Петров А. А.

Компьютерная безопасность

Криптографические методы защиты

Москва, 2000

ББК 32.973.26-018.2
П30

Петров А. А.
П30 Компьютерная безопасность. Криптографические методы защиты. – М.: ДМК, 2000. – 448 с.: ил.

ISBN 5-89818-064-8

В книге рассматриваются актуальные вопросы защиты данных при создании распределенных информационных систем масштаба предприятия, приводятся подробные описания принципов применения современных криптографических средств, имеющихся на рынке («Криптон», «Верба», «Шип», «Игла» и др.). Значительное место уделяется проблемам сохранения тайны при финансовых обменах через Internet, а также электронной коммерции.

Завершают книгу приложения, посвященные практическим рекомендациям по самым острым вопросам обеспечения защиты информации.

ББК 32.973.26-018.2

ISBN 5-89818-064-8

Содержание

О КНИГЕ

В настоящее время издано много руководств, справочников, пособий по проблемам защиты информации. Это и переводные книги, и оригинальные работы отечественных авторов, и вузовские учебники. У подавляющего большинства авторов книги либо чересчур «теоретичны», либо предельно популярны. И то, и другое отпугивает читателей-практиков, которым понимание теории необходимо для ее конкретного применения.

Монография А. Петрова «Компьютерная безопасность. Криптографические методы защиты» является в этом смысле приятным исключением. Наряду с описанием шифрующих преобразований и цифровой подписи в книге содержатся и актуальные сведения о современных криптографических протоколах, и практическая информация, посвященная принципам применения имеющихся на рынке сертифицированных криптографических средств («Криптон», «Верба», «Шип», «Игла» и др.).

Нельзя не отметить и те разделы книги, в которых излагаются теоретические основы компьютерной безопасности. Здесь впервые систематизированы и всесторонне рассмотрены атаки на современные компьютерные системы и различные способы защиты от них.

Монография, несомненно, представляет интерес для широкого круга читателей, в том числе обучающихся по типовым программам специальностей «Введение в криптографию», «Компьютерная безопасность», «Программно-аппаратные средства обеспечения информационной безопасности».

А. Ю. Щербаков,
доктор технических наук, научный консультант подразделения ФАПСИ,
руководитель группы систем защиты от информационного оружия,
почетный член Европейской ассоциации хакеров,
профессор кафедры «Информационная безопасность»
Московского государственного института электроники и математики

Разработчикам информационных систем повышенной сложности, таких как корпоративные информационные системы или информационные системы масштаба крупного предприятия, в последнее время приходится решать задачи, удачное решение которых полностью зависит от точного соблюдения основополагающих и общепринятых правил проектирования и эксплуатации виртуальных конструкций, как-то:

- поддержка стандартов, подразумевающая соответствие общепринятым технологическим стандартам, которыми руководствуются фирмы-производители программного обеспечения;
- масштабируемость, означающая, что программное обеспечение должно работать с приемлемой производительностью без внесения в него существенных изменений при увеличении мощности и количества используемого оборудования;
- многозвенность, где требуется, чтобы каждый уровень системы (клиент, Web-сервер, сервер приложений, сервер баз данных) отвечал и реализовывал только те функции, которые ему наиболее присущи;
- обеспечение (по возможности) аппаратно-платформенной независимости программного обеспечения, используемого при разработке системы;
- коммуникативность и интегрированность, что означает возможность различных уровней системы взаимодействовать между собой, как по данным, так и по приложениям;
- защищенность.

Безусловно, существуют и другие, менее важные задачи, но здесь хотелось бы отметить, что любая из перечисленных проблем внедрения корпоративных информационных систем на базе технологий Internet/intranet может быть успешно реализована только при наличии хорошо организованных средств защиты информации, централизованного управления информационными ресурсами и разграничения доступа к ним. Особенно это важно при обеспечении доступа пользователей из внешних сетей к ресурсам корпоративной ИС по так называемой extranet-технологии.

Как раз одному из этих направлений – защите информации в распределенных сетях – и посвящена книга А. Петрова «Компьютерная безопасность. Криптографические методы защиты», которую мне особенно приятно представить коллегам и читателям.

На сегодняшний день обеспечение безопасности потоков данных является одним из самых перспективных направлений науки и техники, имеющих большое теоретическое и практическое значение.

Автору удалось собрать в книге солидный фактический материал, систематизировать его, изложить свою точку зрения по конкретным вопросам, касающимся практического использования не только отечественных, но и зарубежных криптографических методов и средств защиты информации.

В приложениях, написанных сотрудниками ЦБ России, приведен список известных серверов по проблемам защиты информации. Эти сведения, безусловно, позволят читателям расширить свои знания в данной области.

Я надеюсь, что после знакомства с представляемой книгой моим коллегам станут ближе и понятнее проблемы защиты информации, в том смысле, что работа начинается с проектирования архитектуры информационной системы и заканчивается выбором средств на уровне шифрования.

Занимаясь вопросами создания реальных информационных систем, нам часто не хватает времени познакомиться и тем более сравнить различные варианты реализации средств защиты информации. Надеюсь, что данная книга и приложения к ней позволят уважаемым читателям и коллегам различить новые направления в такой широкой и в то же время такой специальной теме, как проблема построения защищенных информационных систем.

В. С. Лаптев,
кандидат технических наук,
президент Фонда развития программной инженерии

Предисловие

В настоящее время первостепенным фактором, влияющим на политическую и экономическую составляющие национальной безопасности, является степень защищенности информации и информационной среды. Вот почему важное значение приобретают вопросы обеспечения безопасности информационных и телекоммуникационных технологий и гарантированной защиты данных в компьютерных сетях экономически значимых структур. О необходимости надежной защиты свидетельствуют многочисленные компьютерные преступления, совершаемые как в кредитно-финансовой сфере, так и в государственных органах. При этом заметно увеличилось число противоправных деяний, совершенных путем удаленных атак с использованием территориально-распределенных сетей передачи данных. Подобные правонарушения опасны тем, что на сегодняшний день устоявшейся практики борьбы с ними не существует.

Вместе с тем необходимо отметить, что, несмотря на резкое возрастание интереса к проблемам защиты информации, в отечественной научно-технической литературе данная тема освещена слабо, и автор в меру своих сил попытался заполнить этот пробел. В предлагаемой книге рассматриваются вопросы применения криптографии для защиты информации в современных информационно-телекоммуникационных системах, отражающие только одну из областей компьютерной безопасности, однако, на взгляд автора, на сегодняшний день наиболее значимую. В книге освещаются как теоретические аспекты применения классической криптографии (глава 1) и современных криптографических протоколов (глава 2), так и практические вопросы (глава 3), которые возникают при осуществлении защиты информации криптографическими методами и средствами.

В данной книге также затрагиваются проблемы защиты платежных систем в Internet, безопасность современных операционных систем (Windows NT и Novell NetWare) и ряд других актуальных вопросов компьютерной безопасности.

Во введении обсуждается терминология, а также цели и задачи, возникающие при обеспечении информационной безопасности. Здесь определяется роль криптографических протоколов как наиболее перспективного

средства защиты в общей задаче сохранения конфиденциальности, целостности и достоверности информационных потоков.

В главе 1 рассматриваются некоторые теоретические аспекты криптографии и описаны способы построения часто используемых на сегодняшний день криптографических алгоритмов. Раздел 1.1 посвящен теоретическим основам применения и реализации криптографических алгоритмов в современных информационно-телекоммуникационных системах. В разделе 1.2 представлены традиционные блочные алгоритмы шифрования; здесь же изложены принципы блочного шифрования и виды применения подобных алгоритмов, а также конкретные алгоритмы – DES и ГОСТ 28147-89. Раздел 1.3 знакомит с асимметричными алгоритмами шифрования; в нем уделяется внимание математическим идеям, а также получившему широкое распространение алгоритму RSA (с точки зрения уязвимости и возможности проведения на него теоретических атак и с точки зрения эффективности его реализации). В разделе 1.4 рассказывается об электронно-цифровой подписи (ЭЦП); при этом разбираются не только конкретные схемы (ГОСТ Р 34.10-94 и DSA), но и атаки на схемы ЭЦП, а также вопросы, связанные с арбитражем ЭЦП. В разделе 1.5 описываются хэш-функции, используемые совместно с алгоритмами ЭЦП, и затрагиваются вопросы их стойкости. В разделе 1.6 излагаются вопросы, связанные с генерацией, хранением и распределением ключей. Здесь также рассматривается актуальный на сегодняшний день вопрос о минимальной длине ключа, необходимой для обеспечения адекватного уровня безопасности.

Вторая глава посвящена интересным и острым на сегодняшний день проблемам построения, реализации и применения криптографических протоколов, таких как протоколы аутентификации, протоколы распределения и управления ключевой информацией и специфические протоколы. В разделе 2.1 излагаются основные принципы их построения и безопасного использования, а также формальные методы их анализа. Несмотря на то что подобные протоколы до сих пор считаются мощным средством обеспечения безопасности в современных информационно-телекоммуникационных системах, в настоящий момент в хорошо известных протоколах уже найдено немало уязвимых мест. В разделе 2.2 обсуждаются различные схемы аутентификации, начиная с простой аутентификации и заканчивая аутентификацией, обладающей свойством нулевого знания. В разделе 2.3 рассматривается недостаточно освещенная в отечественной литературе проблема распределения и управления ключевой информацией, причем описываются протоколы с использованием симметричных и асимметричных алгоритмов. Одной из главных трудностей, возникающих при построении криптографической системы защиты информации в распределенных

системах, как раз является распределение ключевой информации, поэтому данный раздел в современном контексте развития Internet особенно актуален. Кроме этих проблем в разделе 2.3 затронуты вопросы сертификатов открытых ключей, центров сертификации, междоменные отношения и т.д. Раздел 2.4 посвящен интересным, но малоизученным на практике специфическим криптографическим протоколам, призванным решать вопросы безопасности легитимного голосования, группового разделения знания секрета. Приведенные в этом разделе результаты можно активно использовать как в кредитно-финансовой сфере, так и в повседневной жизни.

В главе 3 рассматриваются практические вопросы применения криптографических средств защиты информации для решения типовых задач информационной безопасности. Обсуждение ведется на основе анализа проблемных вопросов информационной безопасности и средств защиты информации, применяемых на сегодняшний день в России и за рубежом. В качестве законченных решений для конкретных задач приводятся некоторые корпоративные решения. В данной главе также затронута область электронной коммерции в Internet, так как эта сфера в последние годы становится наиболее активным потребителем новых идей и средств криптографической защиты информационных потоков. Раздел 3.1 представляет собой своеобразный путеводитель по этой специфической области переработки и накопления данных; здесь также описывается общая проблематика и подходы к решению задач обеспечения информационной безопасности в современных информационно-телекоммуникационных системах. Приведен обзор стандартов в области криптографической защиты обрабатываемых и передаваемых сведений. Раздел 3.2 посвящен проблемам защиты информации локальных рабочих станций (не подключенных к каналам передачи данных). В качестве конкретных средств криптографической защиты информации рассматриваются семейство «Верба», программно-аппаратные комплексы «Аккорд», «Криптон» и др. Также затрагиваются вопросы и задачи информационной безопасности, возникающие при использовании современных операционных систем (на примере Windows NT). Раздел 3.3 посвящен защите информации в локальных сетях передачи данных на примере сетей Windows NT (в том числе и Windows NT 5.0) и Novell NetWare, широко распространенных на отечественном рынке. В данной части анализируются уязвимости протоколов PPTP и CIFS (от Microsoft). В качестве средства, позволяющего решить большинство задач информационной безопасности в локальных сетях передачи данных, рассмотрена система Secret Net NT. В разделе 3.4 освещаются вопросы, связанные с защитой информации в сетях, имеющих выход в другие сети передачи данных. Здесь приводятся общие сведения об угрозах

в распределенных IP-сетях и протоколах, применяемых для защиты информации в Internet, а также описываются функциональные возможности и применение криптографических средств защиты информации для построения виртуальных частных сетей («Шип», «ФПСУ», «Игла-П» и «Застава»). Раздел 3.5 посвящен не менее актуальным вопросам защиты информации в распределенных сетях – защите технологий «клиент-сервер». Многообразие средств защиты данных и задачи информационной безопасности в клиент-серверных технологиях тоже представлены в этом разделе. Здесь читатель познакомится как с хорошо известными средствами защиты информации SSL и Kerberos, так и с их реализацией в виде законченного продукта – Trusted Web. В разделе 3.6 описывается применение межсетевых экранов, proxy-серверов, а также подробно излагается реализация криптографических средств защиты информации в Check Point Firewall-1. Следующий раздел посвящен принципам защиты электронной почты, в том числе таким известным протоколам защищенной электронной почты, как PEM (и его разновидности) и PGP. Отдельно представлена интересная проблема – защита информации в архитектуре X.400, и в качестве конкретной реализации данного вида электронной почты рассмотрен Messenger 400. Раздел 3.8 рассказывает об опыте обеспечения информационной безопасности в виде отдельных корпоративных решений. Причем на этих страницах рассматриваются системы перевода денежных средств и средства обеспечения финансовых операции – SWIFT и Smart City. Раздел 3.9 рассказывает об электронной коммерции в Internet и обеспечении информационной безопасности глобальной сети. Здесь читатель познакомится с понятием электронных денег и с проблемами информационной безопасности, возникающими при использовании смарт-карт.

В приложениях представлены результаты тестирования отечественных средств построения виртуальных частных сетей[1] и приведен общий обзор проблемы санкционированного доступа к ресурсам корпоративной информационной системы предприятия[2].

Автор надеется, что книга окажется полезной не только пользователям, начинающим осваивать данную область человеческих знаний, но и специалистам в сфере компьютерной безопасности.

[1] Авторы И. Гвоздев, В. Зайчиков, Н. Мошак, М. Пеленицин, С. Селезнев, Д. Шепелявый.

[2] Авторы В. С. Лаптев, С. П. Селезнев, М. Ю. Шувалов.

Введение

Задачи информационной безопасности

Стремительное развитие средств вычислительной техники и открытых сетей передачи данных обусловило их широкое распространение в повседневной жизни и предпринимательской деятельности. Мощные вычислительные возможности и оперативность передачи информации не только оказали большое влияние на принципы ведения бизнеса, сложившиеся в большинстве традиционных отраслей, но и открыли новые направления развития предпринимательской деятельности. В современных условиях автоматизация банковской деятельности и управления предприятиями является «modus vivendi», а такие слова, как «Internet-banking», «e-commerce» и «smart-cards», уже не вызывают всеобщего удивления и жарких дебатов.

Однако последние достижения человеческой мысли в области компьютерных технологий связаны с появлением не только персональных компьютеров, сетей передачи данных и электронных денег, но и таких понятий, как *хакер*, *информационное оружие*, *компьютерные вирусы* и т.п. Оказывается, что под *информационной безопасностью* подразумевается одно из ведущих направлений развития информационных технологий – круг задач, решаемых в этой области, постоянно расширяется как в количественном, так и в качественном отношении.

Современные методы накопления, обработки и передачи информации способствовали появлению угроз, связанных с возможностью потери, раскрытия, модификации данных, принадлежащих конечным пользователям.

На практике угрозы для *информационно-телекоммуникационных систем* (ИТС) могут быть реализованы непосредственным воздействием как на информацию, представляющую интерес для конечных пользователей подобных систем, так и на информационные ресурсы и телекоммуникационные службы, обеспечиваемые в рамках данной ИТС. Например, существует распространенный вид атаки через Internet – шторм ложных запросов на TCP-соединение, приводящий к тому, что система временно прекращает обслуживание удаленных пользователей. Перечень

подобного типа угроз достаточно обширен, и нередко такие проблемы необходимо решать путем защиты пользовательской информации как части ИТС.

Под *информационной безопасностью* в этой книге мы будем понимать состояние защищенности обрабатываемых, хранимых и передаваемых в ИТС данных от незаконного ознакомления, преобразования и уничтожения (как крайний случай модификации), а также состояние защищенности информационных ресурсов от воздействий, направленных на нарушение их работоспособности.

Основными задачами защиты пользовательской информации являются:

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

Конфиденциальность – свойство информации быть доступной только ограниченному кругу пользователей информационной системы, в которой циркулирует данная информация.

Под *целостностью* понимается свойство информации или программного обеспечения сохранять свою структуру и/или содержание в процессе передачи и/или хранения.

Здесь следует сделать пояснение. Рассматривая вопрос передачи информации в виде сообщений через сеть, можно прийти к заключению, что каждое сообщение по своему смысловому содержанию образует некоторый класс. Другими словами, смысл конечного сообщения останется таким же, как и начального, даже если форма представления информации в электронном виде существенно изменится. Таким образом, каждое сообщение на русском языке будет иметь свой класс эквивалентности, и для данного случая свойство сохранения целостности информации можно сформулировать следующим образом: переданное сообщение X считается сохранившим целостность, если полученное в результате передачи сообщение X1 принадлежит классу эквивалентности сообщения X.

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

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

Юридическая значимость означает, что документ обладает юридической силой. С этой целью субъекты, которые нуждаются в подтверждении юридической значимости передаваемого сообщения, договариваются о повсеместном принятии некоторых атрибутов информации, выражающих ее способность быть юридически значимой. Данное свойство информации особенно актуально в системах электронных платежей, где осуществляется операция по переводу денежных средств. Исходя из сказанного, можно сформулировать некоторые требования к атрибутам информации, выражающим ее свойство быть юридически значимой. Прежде всего ее необходимо сформировать таким образом, чтобы с формальной точки зрения было определенно ясно, что только отправитель, которому принадлежит данный платежный документ, мог его создать. О способах обеспечения юридической значимости платежных документов будет сказано ниже.

Неотслеживаемость – способность совершать некоторые действия в информационной системе незаметно для других объектов. Актуальность данного требования стала очевидной благодаря появлению таких понятий, как электронные деньги и Internet-banking. Так, для авторизации доступа к электронной платежной системе пользователь должен предоставить некоторые сведения, однозначно его идентифицирующие. По мере развития данных систем может появиться реальная опасность, что, например, все платежные операции будут контролироваться, тем самым возникнут условия для тотальной слежки за пользователями информационных систем.

Существует несколько путей решения проблемы неотслеживаемости:

- запрещение с помощью законодательных актов всякой тотальной слежки за пользователями информационных систем;
- применение криптографических методов для поддержания неотслеживаемости.

Как уже говорилось, информационная безопасность может рассматриваться не только по отношению к некоторым конфиденциальным сведениям, но и по отношению к способности информационной системы выполнять заданные функции.

Основные задачи, решаемые в рамках информационной безопасности по отношению к работоспособности ИТС, должны обеспечивать защиту от:

- нарушения функционирования телекоммуникационной системы, выражающегося в воздействии на информационные каналы, каналы сигнализации, управления и удаленной загрузки баз данных коммутационного оборудования, системное и прикладное программное обеспечение;

- несанкционированного доступа к информационным ресурсам и от попыток использования ресурсов сети, приводящих к утечке данных, нарушению целостности сети и информации, изменению функционирования подсистем распределения информации, доступности баз данных;
- разрушения встраиваемых и внешних средств защиты;
- неправомочных действий пользователей и обслуживающего персонала сети.

Приоритеты среди перечисленных задач информационной безопасности определяются индивидуально для каждой конкретной ИТС и зависят от требований, предъявляемых непосредственно к информационным системам.

Следует учесть, что с точки зрения государственных структур защитные мероприятия в первую очередь призваны обеспечить конфиденциальность, целостность и доступность информации. (Понятно, для режимных государственных организаций на первом месте всегда стоит конфиденциальность сведений, а целостность понимается исключительно как их неизменность.) Коммерческим структурам, вероятно, важнее всего целостность и доступность данных и услуг по их обработке. По сравнению с государственными, коммерческие организации более открыты и динамичны, поэтому вероятные угрозы для них отличаются не только количеством, но и качеством.

Для решения задачи обеспечения безопасности в информационно-телекоммуникационных сетях необходимо:

- защитить информацию при ее хранении, обработке и передаче по сети;
- подтвердить подлинность объектов данных и пользователей (аутентификация сторон, устанавливающих связь);
- обнаружить и предупредить нарушение целостности объектов данных;
- защитить технические устройства и помещения;
- защитить конфиденциальную информацию от утечки и от внедренных электронных устройств съема информации;
- защитить программные продукты от внедрения программных закладок и вирусов;
- защитить от несанкционированного доступа к информационным ресурсам и техническим средствам сети, в том числе и к средствам управления, чтобы предотвратить снижение уровня защищенности информации и самой сети в целом;
- организовать требующиеся мероприятия, направленные на обеспечение сохранности конфиденциальных данных.

Конкретная реализация общих принципов обеспечения информационной безопасности может выражаться в организационных либо технических мерах защиты информации.

Следует отметить, что объем мероприятий по защите обрабатываемых и передаваемых данных зависит прежде всего от величины предполагаемого ущерба, который может выражаться в прямой (затраты на покупку нового программного обеспечения в случае нарушения целостности программного обеспечения) или в опосредованной (затраты от простоя информационной системы банка) форме. Правда, в некоторых ситуациях рассчитать величину ущерба затруднительно (например, в случае утечки государственного тайны).

Роль криптографических протоколов в общей задаче обеспечения информационной безопасности

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

В основе криптографических методов лежит понятие *криптографического преобразования информации*, производимого по определенным математическим законам, с целью исключить доступ к данной информации посторонних пользователей, а также с целью обеспечения невозможности бесконтрольного изменения информации со стороны тех же самых лиц.

Применение криптографических методов защиты обеспечивает решение основных задач информационной безопасности. Этого можно добиться путем реализации следующих криптографических методов защиты как пользовательской и служебной информации, так и информационных ресурсов в целом:

- шифрование всего информационного трафика, передающегося через открытые сети передачи данных, и отдельных сообщений;
- криптографическая аутентификация устанавливающих связь разноуровневых объектов (имеются в виду уровни модели взаимодействия открытых систем);
- защита несущего данные трафика средствами *имитозащиты* (защиты от навязывания ложных сообщений) и электронно-цифровой подписи с целью обеспечения целостности и достоверности передаваемой информации;
- шифрование данных, представленных в виде файлов либо хранящихся в базе данных;

- контроль целостности программного обеспечения путем применения криптографически стойких контрольных сумм;
- применение электронно-цифровой подписи для обеспечения юридической значимости платежных документов; применение затемняющей цифровой подписи для обеспечения неотслеживаемости действий клиента в платежных системах, основанных на понятии электронных денег.

При реализации большинства из приведенных методов криптографической защиты возникает необходимость обмена некоторой информацией. Например, аутентификация объектов ИТС сопровождается обменом идентифицирующей и аутентифицирующей информации.

В общем случае взаимодействие объектов (субъектов) подобных систем всегда сопровождается соблюдением некоторых договоренностей, называемых *протоколом*. Формально протоколом будем считать последовательность действий объектов (субъектов) для достижения определенной цели. Она в данном случае определяет структуру и специфику применения протокола.

В свою очередь, *криптографическими протоколами* будем называть те, в которых участники для достижения определенной цели используют криптографические преобразования информации.

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

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

На сегодняшний день благодаря повсеместному применению открытых сетей передачи данных, таких как Internet, и построенных на их основе сетей intranet и extranet криптографические протоколы находят все более широкое применение для решения разнообразного круга задач и обеспечения постоянно расширяющихся услуг, предоставляемых пользователям таких сетей.

Кроме вышеперечисленных классических областей применения протоколов существует широкий круг специфических задач, также решаемых с помощью соответствующих криптографических протоколов. Это прежде

всего раскрытие части сведений без обнародования самого секрета в его подлинном объеме, а также частичное раскрытие секрета. Так, например, участники могут для достижения какой-то общей цели сообщить друг другу часть своей информации или объединить усилия для раскрытия секрета, неизвестного каждому из них в отдельности.

Стремительное развитие криптографических протоколов в большей степени стимулируется развитием систем электронных платежей, интеллектуальных карточек, появлением электронных денег и т.д. По зарубежным оценкам, темпы развития электронной коммерции постоянно ускоряются. Например, судя по прогнозам, количество компаний, занимающихся этим видом коммерции во всем мире, вырастет с 111 тысяч в 1996 году до 435 тысяч в 2000 году. При этом суммарный объем продаж через Internet увеличится с 9,5 до 196 млрд долларов.

Что касается розничных продаж через Internet, то они, по тем же оценкам, с 500 млн долларов в 1996 году вырастут до 7 млрд в 2000 году. При этом более половины покупок будет оплачено с помощью новых средств платежей – электронных денег. Таким образом, прогнозируется не только увеличение числа компаний, занимающихся бизнесом в сети, и их общего оборота, но и резкое возрастание среднего дохода, приходящегося на одну такую компанию. Поскольку на сегодняшний день основным криптографическим средством защиты информации в Internet являются протоколы, можно констатировать, что развитие подобных средств защиты коммерческих тайн будет продолжаться и в количественном, и в качественном отношении.

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

Учитывая, что основой любого криптопротокола являются так называемые *криптоалгоритмы*, в рамках данной книги рассматриваются вопросы построения и практического применения основных типов подобных механизмов. Кроме хорошо известных и повсеместно используемых зарубежных криптоалгоритмов, здесь уделяется достаточное внимание отечественным разработкам в области информационной безопасности и стандартов на криптоалгоритмы.

Глава 1

Общие сведения по классической криптографии

1.1. Общие сведения

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

Под системой засекреченной связи будем понимать систему передачи информации, в которой смысл передаваемой информации скрывается с помощью криптографических преобразований. При этом сам факт передачи информации не утаивается. В основе каждой системы засекреченной связи – использование алгоритмов шифрования как основного средства сохранения конфиденциальности.

Зашифрование – процесс криптографического преобразования множества открытых сообщений в множество закрытых сообщений.

Расшифрование – процесс криптографического преобразования закрытых сообщений в открытые.

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

Множество открытых сообщений может быть представлено в виде битового потока, сетевого фрейма, файла и т.д.

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

сообщением (см. рис. 1.1). Ключи зашифрования и расшифрования могут в общем случае быть различными, хотя для простоты рассуждений (применительно к этой главе) предположим, что они идентичны. Множество, из которого выбираются ключи, называется *ключевым пространством*. Совокупность процессов зашифрования, множества открытых сообщений, множества возможных закрытых сообщений и ключевого пространства называется *алгоритмом зашифрования*. Совокупность процессов расшифрования, множества возможных закрытых сообщений, множества открытых сообщений и ключевого пространства называется *алгоритмом расшифрования*.

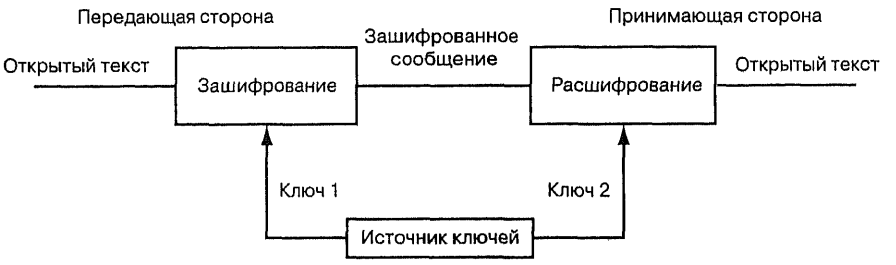

Рис. 1.1. Общая структура системы засекреченной связи

Работу системы засекреченной связи можно описать следующим образом:

1. Из ключевого пространства выбирается ключ зашифрования K и отправляется по надежному каналу передачи.
2. К открытому сообщению C, предназначенному для передачи, применяют конкретное преобразование Fk, определяемое ключом K, для получения зашифрованного сообщения M – M = Fk(C).
3. Полученное зашифрованное сообщение M пересылают по каналу передачи данных.
4. На принимающей стороне к полученному сообщению M применяют конкретное преобразование Dk, определяемое из всех возможных преобразований ключом K, для получения открытого сообщения C: C = Dk(M).

Канал передачи данных, используемый для отправки зашифрованных сообщений, считается ненадежным, то есть любое зашифрованное сообщение может быть перехвачено противником (злоумышленником). Здесь мы предполагаем, что передаваемая информация сохраняет целостность, хотя на практике это не менее важная задача по сравнению с обеспечением конфиденциальности передаваемых сообщений.

Наличие потенциального противника приводит к тому, что система засекреченной связи может быть скомпрометирована. На практике обеспечение надежности функционирования подобной системы сводится к стойкости используемых алгоритмов шифрования, лежащих в основе всех операций. Это обусловлено тем, что стойкость всей системы не может быть выше стойкости алгоритмов шифрования, однако может быть и гораздо ниже, хотя бы в силу технической реализации самой системы. Так, противник, анализируя побочные излучения от аппаратуры, на которой реализуются алгоритмы шифрования, в состоянии получить интересующий его ключ. Использование технических средств для обработки, передачи и хранения конфиденциальной информации порождает сложную научно-техническую задачу обеспечения ее защиты (под защитой мы будем подразумевать сохранение свойств информации, необходимых пользователям). В рамках данной книги технические аспекты защиты информации рассматриваться не будут, мы остановимся только на криптографических методах и средствах.

Каким требованиям должны удовлетворять алгоритмы шифрования, чтобы не быть скомпрометированными в случае, если противник обладает неограниченными возможностями (временем, вычислительными средствами, количеством перехваченных зашифрованных сообщений), что такое стойкость алгоритмов шифрования – эти и подобные вопросы будут рассмотрены в настоящей главе.

1.1.1. Стойкость алгоритмов шифрования

Для каждого открытого сообщения существует априорная вероятность выбора, поскольку механизм выбора открытых сообщений можно представить как некоторый вероятностный процесс. Аналогично выбор каждого ключа тоже имеет априорную вероятность. Противник, перехватывающий зашифрованные сообщения, может вычислить апостериорные вероятности как появления открытого сообщения, так и вероятность появления ключа. Набор апостериорных вероятностей представляет собой систему принадлежащих противнику сведений об используемых ключах и передаваемых открытых сообщениях. Причем перед началом перехвата шифрованных сообщений противник имеет в своем распоряжении некоторый набор априорных вероятностей об открытых сообщениях и ключах. С практической точки зрения это означает, что противник осведомлен об используемой системе засекреченной связи.

Предположим, что противнику известно все криптографические преобразования, используемые в системе засекреченной связи, а также ключевое пространство, причем, как было сказано выше, секретность системы зависит от выбора конкретного ключа. В результате перехвата некоторого объема зашифрованных сообщений и вычисления апостериорных вероятностей противник поймет, что им будет соответствовать единственное решение об использовании ключа или передаче открытого сообщения (точка единственности принятия решения), удовлетворяющего данным вероятностям. Понятно, что подобный вывод вполне может привести к раскрытию системы противником. (Строгие математические доказательства существования точки единственности принятия решения и расчеты в данной книге не приводятся.) Под раскрытием системы засекреченной связи или алгоритма шифрования мы будем понимать одну из следующих планируемых противником операций, направленных на достижение этой цели:

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

Наука, занимающаяся вопросами раскрытия алгоритмов шифрования, называется *криптоанализом.*

Каждой из определенных выше целей соответствует свой объем знаний об используемом ключе или переданном открытом сообщении. Чтобы увеличить его, противнику необходимо иметь достаточное количество перехваченных зашифрованных сообщений. Вот почему он осуществляет ряд атак на используемую систему засекреченной связи. В дальнейшем будем полагать, что целью атаки являются применяемые алгоритмы зашифрования (расшифрования).

Таким образом, под *стойкостью алгоритма шифрования* будем понимать способность противостоять всем возможным атакам против него. В вероятностных терминах это определение можно перефразировать следующим образом: стойким считается алгоритм, в котором перехват зашифрованных сообщений не приводит к появлению точки единственности принятия решения об используемом ключе или переданном открытом сообщении. На

практике же получение противником требуемых для вскрытия алгоритма шифрования апостериорных вероятностей зависит от наличия у него определенных ресурсов, среди которых можно отметить следующие:

- конкретный объем перехваченных зашифрованных сообщений;
- временные ресурсы. Здесь подразумевается время, необходимое для проведения определенных вычислений; в некоторых случаях временные затраты противника могут превышать время жизни секретной информации (табл. 1.1). Время жизни секретной информации можно определить как время, в течение которого информация должна сохранять свое свойство конфиденциальности;
- вычислительные ресурсы. Имеется в виду количество памяти в вычислительных системах, используемых для успешной реализации атаки.

Таблица 1.1. Время жизни некоторых типов информации

Тип информации	Время жизни
Военная тактическая информация	минуты/часы
Заявления о выпуске продукции	дни/недели
Долгосрочные бизнес-проекты	годы
Производственные секреты	десятилетия
Секрет создания водородной бомбы	>40 лет
Информация о разведчиках	>50 лет
Личная информация	>50 лет
Дипломатическая тайна	>65 лет
Информация о переписи населения	100 лет

Стойким считается алгоритм, который для своего вскрытия требует от противника практически недостижимых вычислительных ресурсов или недостижимого объема перехваченных зашифрованных сообщений или времени раскрытия, которое превышает время жизни интересующей противника информации.

Существует достаточно распространенный подход к формальной оценке этого понятия. Стойкость криптографического алгоритма необходимо рассматривать относительно пары «атака-цель», где под целью противника понимается планируемая угроза. В мировой литературе проработана классификация различных типов атак на криптографические алгоритмы:

- *атака с известным шифртекстом* (ciphertext-only attack). Предполагается, что противник знает криптосистему, то есть алгоритмы шифрования,

но не знает секретный ключ. Кроме того, ему известен лишь набор перехваченных криптограмм;

- *атака с известным открытым текстом* (known plaintext attack). То же, что предыдущая, но противник получает в свое распоряжение еще некоторый набор криптограмм и соответствующих им открытых текстов;
- *простая атака с выбором открытого текста* (chosen-plaintext attack). Противник имеет возможность выбрать необходимое количество открытых текстов и получить соответствующие им криптограммы;
- *адаптивная атака с выбором открытого текста* (adaptive-chosen-plaintext attack). В этом случае противник имеет возможность выбирать открытые тексты с учетом того, что криптограммы всех предыдущих открытых текстов ему известны;
- *атака с выбором шифртекста* (chosen-ciphertext attack). Противник имеет возможность выбрать необходимое количество криптограмм и получить соответствующие им открытые тексты;
- *адаптивная атака с выбором шифртекста* (adaptive-chosen-ciphertext attack). Противник, выбирая очередную криптограмму, знает все открытые тексты, соответствующие предыдущим криптограммам;
- *атака с выбором текста* (chosen-text attack). Противник имеет возможность выбирать как криптограммы (и дешифровывать их), так и открытые тексты (и зашифровывать их);
- *атака с выбором ключа* (chosen-key attack). Противник знает не сами ключи, а некоторые различия между ними.

В этом перечне атаки представлены по мере возрастания их силы. Так, с точки зрения криптоанализа последняя атака является самой сильной, хотя существует еще более сильная атака, заключающаяся в получении секретного ключа путем вымогательства, шантажа или подкупа.

Анализируя атаки и их цели, можно прийти к выводу, что наибольшей стойкостью алгоритм обладает в том случае, когда он способен противостоять самой сильной атаке, проводимой противником, при условии, что он преследует самую слабую из возможных целей атаки (угроза).

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

- длина ключа и длина открытого сообщения должны быть одинаковы;
- ключ должен использоваться только один раз;
- выбор ключа из ключевого пространства должен осуществляться равновероятно.

Данные требования приводят к тому, что абсолютно стойкие алгоритмы с практической точки зрения являются труднореализуемыми. Например, осуществление первого и второго условий приводит к тому, что необходимо иметь запас ключей большой длины, что практически невыполнимо. В результате применение современной аппаратно-программной базы приводит к неабсолютной стойкости используемых алгоритмов шифрования.

Рассмотрим самые распространенные на сегодняшний день причины осуществления успешных атак на алгоритмы шифрования:

- наличие статистической структуры исторически сложившихся языков. То есть существуют определенные символы или комбинации символов, наиболее часто встречающиеся в естественной речи. Таким образом, при перехвате зашифрованного сообщения для некоторых типов алгоритмов шифрования можно подсчитать частоту появления определенных символов и сопоставить их с вероятностями появления определенных символов или их комбинаций (биграммы, триграммы и т.д.), что в некоторых случаях может привести к однозначному дешифрованию отдельных участков зашифрованного сообщения;
- наличие вероятных слов. Речь идет о словах или выражениях, появление которых можно ожидать в перехваченном сообщении. Так, в деловой переписке присутствуют шаблонные слова; в английском языке, например, наиболее часто встречаются «and», «the», «are» и т.д.

Следует учесть, что существует ряд методов, позволяющих сделать зашифрованные сообщения практически непригодными для статистического анализа и анализа посредством вероятных слов. К ним относятся:

- *рассеивание*. Влияние одного символа открытого сообщения распространяется на множество символов зашифрованного сообщения. Этот метод хотя и приводит к увеличению количества ошибок, однако с его помощью удается скрыть статистическую структуру открытого сообщения;
- *запутывание*. Развитием принципа рассеивания стал принцип запутывания, в котором влияние одного символа ключа распространяется на множество символов зашифрованного сообщения;
- *перемешивание*. Принцип перемешивания основывается на использовании особых преобразований исходного сообщения, в результате чего вероятные последовательности как бы рассеиваются по всему пространству возможных открытых сообщений. В качестве примера эффективного перемешивания можно привести произведение двух

простых некоммутирующих операций. Развитием метода перемешивания явилось применение составных алгоритмов шифрования, состоящих из последовательности простых операций перестановки и подстановки.

Примерами изложенных выше методов могут служить стандарты шифрования, такие как DES (Data Encryption Standard) и ГОСТ 28147-89, подробнее о которых будет сказано далее.

Хотелось бы отметить, что получение строгих оценок стойкости алгоритмов шифрования является достаточно сложной проблемой, решение которой невозможно без рассмотрения самого алгоритма шифрования.

В последнее время с развитием теоретико-сложностных и теоретико-числовых направлений в математике наметился новый подход к построению и оценкам стойкости криптографических алгоритмов. Суть его сводится к выбору в качестве основы алгоритма сложной математической задачи и оценке стойкости в соответствии со сложностью решения этой задачи. С таким подходом мы познакомимся в разделе «Асимметричные алгоритмы шифрования».

1.1.2. Типы алгоритмов шифрования

Приведенная на рис. 1.1 схема является достаточно общей и не позволяет судить обо всем многообразии применения криптографических алгоритмов. Для представления всего диапазона существующих алгоритмов шифрования в этом разделе приведена их классификация (рис. 1.3), а вопросы практического применения отражены в ходе изложения всей книги в целом.

В основе криптографических алгоритмов лежат математические преобразования, позволяющие добиваться высокой практической стойкости большинства асимметричных алгоритмов шифрования. Было доказано, что в криптографии существуют только два основных типа преобразований – *замены* и *перестановки*, все остальные являются лишь комбинацией этих двух типов. Таким образом, есть криптографические алгоритмы, построенные на основе замены, перестановки и объединения этих двух преобразований.

В перестановочных шифрах символы открытого текста изменяют свое местоположение. Например, в шифрах колонной замены открытый текст выписывается в виде матрицы с нумерованными столбцами (рис. 1.2).

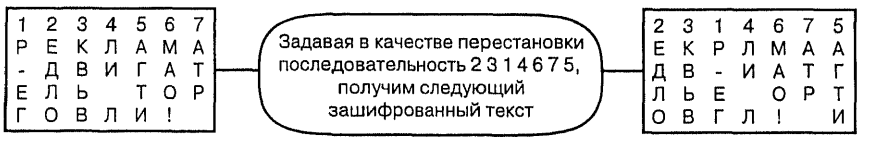

Рис. 1.2. Общая схема перестановки

С другой стороны, в шифрах замены один символ открытого текста замещается символом зашифрованного текста.

В классической криптографии различают четыре типа шифров замены:

- *шифры простой замены.* Один символ открытого текста заменяется одним символом зашифрованного текста;
- *шифры сложной замены.* Один символ открытого текста заменяется одним или несколькими символами зашифрованного текста, например: «А» может быть заменен «С» или «PO4E»;
- *шифры блочной замены.* Один блок символов открытого текста заменяется блоком закрытого текста, например: «ABC» может быть заменен «CPT» или «KAP»;
- *полиалфавитные шифры замены,* в которых к открытому тексту применяются несколько шифров простой замены.

Классическая криптография, в частности теория связи в секретных системах, основанная К. Шенноном, исходила из того, что ключи 1 и 2 (рис. 1.1), используемые соответственно для шифрования и расшифрования, являются секретными и одинаковыми, и передача их должна осуществляться по надежному каналу обмена ключевой информации. Подобные алгоритмы были названы *симметричными*, так как зашифрование и расшифрование происходит на одинаковых ключах. Однако развитие теории построения алгоритмов шифрования с открытыми ключами, родоначальниками которой стали Диффи и Хэлман, положило начало повсеместному использованию *асимметричных* алгоритмов шифрования, в которых ключи зашифрования и расшифрования различны. В зависимости от применения один из ключей будет открытым, то есть общедоступным, а другой необходимо хранить в секрете.

Спустя некоторое время симметричные алгоритмы были разделены на два больших класса – блочные и поточные. В первых открытый текст

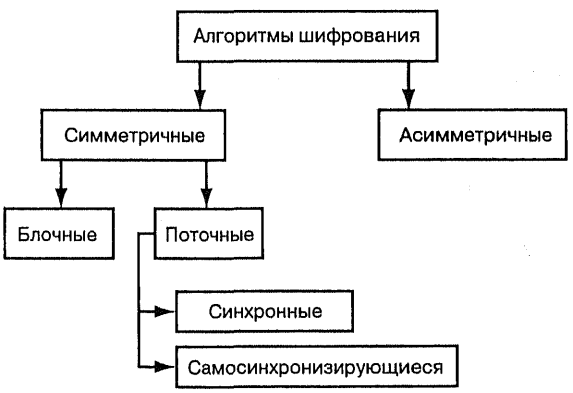

Рис. 1.3
Классификация алгоритмов шифрования

разбивается на блоки подходящей длины (например, размер блоков шифрования в DES равен 64 битам) и фактически каждый блок шифруется, хотя существуют различные варианты применения алгоритмов блочного шифрования, но об этом будет сказано в главе, посвященной блочным алгоритмам. В поточных алгоритмах каждый символ открытого текста зашифровывается независимо от других и расшифровывается таким же образом. Иначе говоря, преобразование каждого символа открытого текста меняется от одного символа к другому, в то время как для блочных алгоритмов в рамках шифрования блока используется одно и то же криптографическое преобразование. Главная идея, воплощенная в алгоритмах поточного шифрования, заключается в выработке на основе секретного ключа последовательности символов из входного алфавита, с которым работает алгоритм шифрования. Это могут быть как, например, символы английского языка, так и цифры десятичной системы исчисления, при этом входной текст преобразуется в соответствии с выбранным алфавитом. Следует учесть, что такая последовательность имеет длину, которая равна открытому тексту. Ее иногда называют гаммой. Зашифрование и расшифрование может, например, осуществляться путем модульного суммирования символа открытого текста с символом гаммы. Стойкость поточных алгоритмов шифрования зависит от того, насколько выработанная гамма будет обладать свойством равновероятности появления очередного символа. Основная проблема в обеспечении безопасности при использовании поточных алгоритмов шифрования заключатся в том, что выработанную гамму недопустимо использовать более одного раза.

Покажем это на примере:

$$c_1 = p_1 \oplus к$$
$$c_2 = p_2 \oplus к$$
$$c_1 \oplus c_2 = к_1 \oplus к_2,$$

где к – символ гаммы;

p_1 и p_2 – символы разных открытых текстов, зашифрованных на одной гамме;

c_1 и c_2 – зашифрованные символы, соответствующие p_1 и p_2.

Далее, применяя к полученной сумме метод вероятностных слов или статистического анализа, есть вероятность найти оба открытых сообщения. На одном ключе может быть выработано конечное число символов гаммы, поскольку через определенное количество символов она начнет повторяться, то есть генератор гаммы всегда имеет некоторый период. Поэтому

при реализации поточного алгоритма шифрования необходимо добиваться как можно большего периода или чаще менять секретный ключ.

Есть еще одна проблема при употреблении алгоритмов данного типа – для правильного расшифрования следует подчиниться требованию синхронности выполнения операций шифраторами на приемной и передающей сторонах. Существует два метода обеспечения синхронизации работы шифраторов:

- *самосинхронизирующиеся шифраторы*, в которых очередной символ гаммы зависит от определенного количества уже образованных символов гаммы. Основной недостаток этого типа шифраторов заключается в разрастании ошибок при расшифровании, если произошла ошибка в ходе передачи.
- *синхронные шифраторы*, то есть шифраторы, осуществляющие синхронизацию своей работы только при вхождении в связь; дальнейшая работа на приемной и передающей сторонах осуществляется синхронно. Основным недостатком этого типа является необходимость заново устанавливать связь между шифраторами при их рассинхронизации, хотя они и не обладают свойством разрастания ошибок.

Поточные алгоритмы обладают высокой скоростью шифрования, однако при программном использовании возникают определенные трудности, что сужает область их практического применения, хотя структура поточных алгоритмов шифрования предполагает эффективную аппаратную реализацию.

1.1.3. Аппаратная и программная реализация алгоритмов шифрования

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

- должны быть адаптированы к новейшей программно-аппаратной базе (например, алгоритмы блочного шифрования в программной реализации должны быть адаптированы к операциям с 64-разрядными числами);

- объем ключа должен соответствовать современным методам и средствам дешифрования зашифрованных сообщений (о минимальной длине ключа будет сказано позже);
- операции зашифрования и расшифрования должны по возможности быть простыми, чтобы удовлетворять современным требованиям по скоростным характеристикам;
- не должны допускать появления постоянно увеличивающегося числа ошибок;
- должны сводить к минимуму объем сообщения в ходе выполнения операций шифрования.

Аппаратная реализация

До недавних пор алгоритмы шифрования реализовывались в виде отдельных устройств, что обусловливалось использованием криптографии для засекречивания различных видов передачи информации (телеграф, телефон, радиосвязь). С развитием средств вычислительной техники и общедоступных сетей передачи данных появились новые возможности применения криптографических алгоритмов. Однако аппаратная реализация до сих пор широко используется не только в военной сфере, но и в коммерческих организациях. Подобная «живучесть» аппаратных средств криптографической защиты информации объясняется рядом факторов.

Во-первых, аппаратная реализация обладает лучшими скоростными характеристиками, нежели программно реализуемые алгоритмы шифрования. Использование специальных чипов, адаптированных к реализации на них процедур зашифрования и расшифрования, приводит к тому, что, в отличие от процессоров общего назначения, они позволяют оптимизировать многие математические операции, применяемые в алгоритмах шифрования.

Во-вторых, аппаратные средства защиты информации обладают несравнимо большей защищенностью как от побочных электромагнитных излучений, возникающих в ходе работы аппаратуры, так и от непосредственного физического воздействия на устройства, где осуществляются операции шифрования и хранение ключевой информации. Что касается побочных электромагнитных излучений, то они вполне могут служить каналом утечки критической информации, связанной с работой алгоритма шифрования и используемых ключей. Физические же воздействия являются удобным средством получения промежуточных сведений о работе алгоритма шифрования, а то и непосредственно ключевой информации, что, в свою очередь, позволит противнику более эффективно и с минимальными затратами

провести атаку на используемые алгоритмы шифрования. Современные микросхемы, на которых реализуются алгоритмы шифрования и осуществляется хранение ключевой информации, способны успешно противостоять любым попыткам физического воздействия – в случае обнаружения несанкционированного доступа к микросхеме она саморазрушается. Реализовать защиту от побочного излучения и утечки по цепям электропитания на обычных персональных компьютерах можно, но решить эту задачу будет гораздо сложнее, нежели использовать устройство, которое соответствует стандартам по защите от побочных электромагнитных излучений.

В-третьих, аппаратные средства более удобны в эксплуатации, так как позволяют осуществлять операции зашифрования и расшифрования для пользователя в прозрачном режиме; кроме того, их легко инсталлировать.

В-четвертых, учитывая многообразие вариантов применения средств криптографической защиты информации, аппаратные средства повсеместно используются для защиты телефонных переговоров, отправки факсимильных сообщений и других видов передачи информации, где невозможно использовать программные средства.

Программная реализация

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

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

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

К этому можно добавить, что программная реализация средств криптографической защиты не в состоянии обеспечить выполнение некоторых

характеристик, требуемых для надежного использования алгоритмов шифрования. Например, генерация ключевой информации не должна производиться программными датчиками случайных чисел; для этой цели необходимо использовать специальные аппаратные устройства.

Программно-аппаратная реализация

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

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

1.2. Алгоритмы блочного шифрования

1.2.1. Общие сведения

Блочные алгоритмы шифрования являются основным средством криптографической защиты информации, хранящейся на компьютере пользователя или передаваемой по общедоступной сети. Такое пристальное внимание к данному типу алгоритмов обусловлено не столько многолетней историей, сколько преимуществами практического применения, среди которых следует отметить:

- возможность эффективной программной реализации на современных аппаратно-программных средствах;
- высокую скорость зашифрования/расшифрования как при аппаратной, так и при программной реализации;
- высокую гарантированную стойкость; причем стойкость алгоритма блочного шифрования может быть доказана при помощи математического аппарата (для большинства асимметричных алгоритмов стойкость

основана на «невозможности» решения какой-либо математической задачи).

Входная последовательность блочных алгоритмов шифрования разбивается на участки определенной длины (обычно 64 бита для удобства реализации на процессорах с внутренними регистрами длиною 32 или 64 бита), и преобразования в алгоритме блочного шифрования совершаются над каждым блоком отдельно. Соответственно выходная последовательность алгоритма блочного шифрования представляет собой блоки, длина которых равна длине входных блоков. В случае, когда длина открытого текста некратна длине входных блоков в алгоритме шифрования, применяется операция *дополнения* (padding) последнего блока открытого текста до необходимой длины. Дополнение осуществляется приписыванием необходимого числа нулей или случайного набора символов. В общем случае содержание того, чем мы дополняем блок открытого текста, не играет роли с точки зрения криптографической стойкости. На приемной стороне необходимо знать, какое количество символов было добавлено, вот почему вместе с данными дополнения приписывается длина этих данных.

Суть алгоритмов блочного шифрования заключается в применении к блоку открытого текста многократного математического преобразования. Многократность подобных операций приводит к тому, что результирующее преобразование оказывается криптографически более сильным, чем преобразование над отдельно взятым блоком. Основная цель подобного трансформирования – создать зависимость каждого бита блока зашифрованного сообщения от каждого бита ключа и каждого бита блока открытого сообщения. Преобразования, базирующиеся на данных алгоритмах, можно разделить на «сложные» (в современных алгоритмах это обычно нелинейные операции) и «простые», в основе которых лежат перемешивающие операции. Аналитическая сложность раскрытия алгоритмов блочного шифрования заключается в конструкции первого типа преобразований.

Специфика организации различных типов секретной связи обусловила появление следующих алгоритмов блочного шифрования:

- *режим простой замены*, или *режим электронной кодовой книги* (Electronic Codebook Mode – ECB);
- *режим гаммирования*;
- *режим гаммирования с самовосстановлением*, или *гаммирование с обратной связью* (Cipher-Feedback mode – CFB);
- *режим гаммирования с обратной связью по выходу* (Output-Feedback mode – OFB);

• *режим шифрования со сцеплением блоков* (Cipher Block Chaining mode – CBC).

Режим простой замены

В данном режиме блоки открытого текста шифруются независимо от других блоков на одном ключе (рис. 1.4). Этот режим назван режимом электронной кодовой книги, поскольку теоретически существует возможность создать книгу, в которой каждому блоку открытого текста будет сопоставлен блок зашифрованного текста. Однако в случае, если длина блока равна 64 битам, то книга будет содержать 2^{64} записи, и каждая книга будет соответствовать одному ключу.

Зашифрование может быть описано следующим уравнением:

$C_i = F(P_i)$ для $i = 1 \div N$,

где C_i и P_i – блоки зашифрованного и открытого текста соответственно, а F – шифрующее преобразование, реализуемое алгоритмом блочного шифрования.

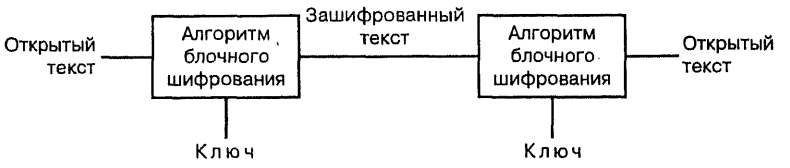

Рис. 1.4. Режим простой замены

Идентичные блоки открытого текста на одном и том же ключе будут зашифрованы одинаковым образом. С точки зрения криптоанализа данный режим является самым «слабым» (поскольку существует большое количество криптографических атак), кроме того, он критичен к пропаданию и появлению новых блоков. Так, злоумышленник, перехватив блок открытого текста, сможет заново передать его, и на принимающей стороне он может быть воспринят как блок, отправленный передающей стороной.

Режим гаммирования

В данном режиме алгоритм блочного шифрования используется для усложнения предварительной гаммы, выработанной одноканальной линией задержки (рис. 1.5). Ошибка во время передачи всего сообщения приводит к искажению при расшифровании только одного блока. Таким образом, при использовании этой методики отсутствует возможность распространения ошибки (хотя данный режим критичен к вставке и пропаданию блоков

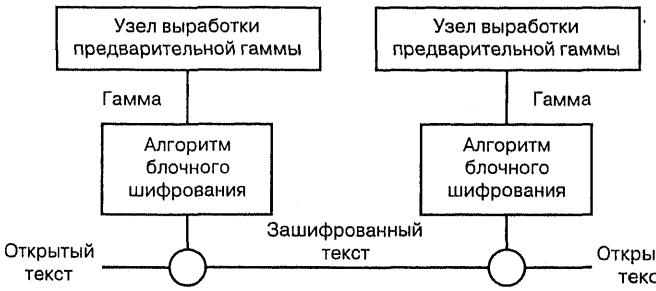

Рис. 1.5
Режим гаммирования

в процессе передачи) за счет рассинхронизации узлов выработки предварительной гаммы на передающей и принимающей сторонах. Для предотвращения этого нежелательного явления на практике применяются устройства синхронизации работы шифраторов, если шифратор реализуется аппаратно. Начальное состояние узла выработки исходной гаммы задается инициализирующим вектором (синхропосылка), который передается по открытым каналам связи в зашифрованном или открытом виде. Гамма, полученная узлом выработки предварительной гаммы, проходит обработку через алгоритм блочного шифрования, после чего результирующая гамма суммируется по модулю с блоком открытого текста.

Зашифрование можно представить в следующем виде:

$$C_i = P_i \oplus F(Y_i)\ i = 1 \div N,$$

где Y_i – выработанная гамма; Y_1 – синхропосылка.

Режим гаммирования с самовосстановлением

Этот режим характеризуется тем, что шифратор в данном случае обладает свойством самосинхронизации и ошибка при передаче приведет к следующему: только два блока открытого текста останутся непрочитанными (рис. 1.6). Режим гаммирования с самовосстановлением, как и предыдущий, критичен к пропаданию и вставлению блоков зашифрованного текста при передаче. Начальное заполнение накопителя, который на практике обычно реализуется в виде сдвигового регистра, является синхропосылкой (Y), которая передается по открытому каналу передачи данных.

Уравнение зашифрования имеет следующий вид:

$$C_1 = P_1 \oplus F(Y)$$
$$C_i = P_i \oplus F(C_{i-1})$$

Соответственно расшифрование имеет вид:

$$P_1 = C_1 \oplus F(Y)$$
$$P_i = C_i \oplus F(C_{i-1})$$

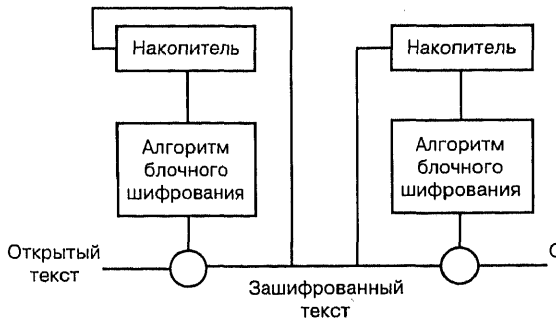

Рис. 1.6
Режим гаммирования с самовосстановлением

Режим гаммирования с обратной связью по выходу

Данный режим во многом похож на предыдущий с той только разницей, что обратная связь не зависит от открытого и зашифрованного текста. Она в этом случае происходит по гамме с выхода алгоритма блочного шифрования (рис. 1.7). В этом режиме алгоритм блочного шифрования используется для организации процесса поточного зашифрования, так же как и в вышеперечисленных режимах гаммирования. Как и ранее, начальное заполнение регистра сдвига является синхропосылкой, передаваемой по открытым каналам связи.

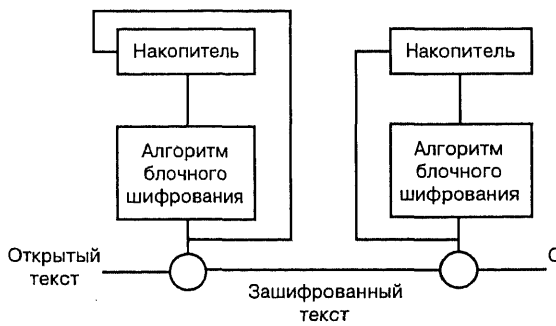

Рис. 1.7
Режим гаммирования с обратной связью по выходу

Процесс зашифрования можно представить следующим образом:

$C_i = P_i \oplus S_i \qquad Si = F(C_{i-1})$

Соответственно расшифрование осуществляется по закону:

$P_i = C_i \oplus S_i \qquad Si = F(C_{i-1})$

Режим шифрования со сцеплением блоков

Характерной особенностью этого способа является наличие обратной связи по зашифрованному тексту. Кроме того, преобразованию в алгоритме блочного шифрования подвергается результат сложения по модулю блока

открытого текста с предыдущим блоком зашифрованного текста (рис. 1.8.). Накопитель, в который поступает предыдущий результат зашифрования, обычно реализуется в виде регистра сдвига и служит для последующего зашифрования. Таким образом, по сравнению с предыдущими режимами, гаммирование в этом случае осуществляется перед тем, как результирующий блок будет преобразован алгоритмом блочного шифрования.

Процесс зашифрования можно представить следующим образом:

$C_i = F(P_i \oplus C_i - 1)$

Соответственно процесс расшифрования будет выглядеть так:

$P_i = F(C_i) \oplus C_i - 1$

Как было сказано выше, начальное заполнение накопителя является синхропосылкой, передаваемой по открытому каналу связи. При передаче блока в данном режиме ошибка в одном бите приведет к неправильному расшифрованию всей оставшейся части блока и ошибке в одном бите при расшифровании следующего блока. Но, как и в предыдущих случаях, этот режим критичен к удалению или добавлению блоков. Основной угрозой безопасности во время его использования может оказаться вставка блоков в начало или конец передачи, поскольку на принимающей стороне данные блоки будут восприняты как истинные, поэтому при использовании режима шифрования со сцеплением блоков желательно знать структуру передаваемого открытого текста.

Как и в предыдущем режиме, при потере синхронности работы регистров сдвига на приемной и передающей сторонах расшифрование открытого текста станет невозможным. Вот почему синхронизация в работе алгоритмов должна обеспечиваться дополнительными средствами. В случае

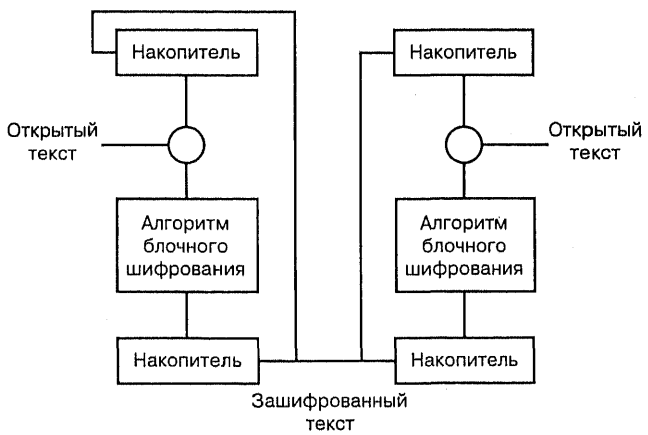

Рис. 1.8
Режим шифрования со сцеплением блоков

применения данного алгоритма шифрования безопасность базируется на длине регистра сдвига, которая должна быть не меньше длины входного блока. Подобное требование обусловлено тем, что выработанная гамма должна иметь гарантированный период, то есть при зашифровании на одном и том же ключе она не должна повторяться.

1.2.2. Алгоритм DES

Одним из самых распространенных алгоритмов блочного шифрования, рекомендованных Национальным бюро стандартов США совместно с АНБ в качестве основного средства криптографической защиты информации как в государственных, так и в коммерческих структурах, является Data Encryption Standard (DES). Он был разработан в 1977 г., однако уже в 1988 г. АНБ рекомендовало использовать DES только в системах электронного перевода. С учетом выявленных недостатков DES в начальный вариант стандарта постоянно вносятся изменения; появляются также и новые алгоритмы, использующие в качестве основы DES – NewDES, TripleDES и др. Необходимость разработки новых алгоритмов обусловлена большим количеством атак, которым подвергался DES за долгие годы своего существования. Кроме того, бурное развитие средств вычислительной и микропроцессорной техники привело к тому, что 56-битного ключа, используемого в оригинальном варианте DES, стало недостаточно, чтобы противостоять атакам, совершаемым методом «грубой силы». Тем не менее в коммерческой сфере и в системах электронных расчетов DES и на сегодняшний день остается одним из самых популярных алгоритмов блочного шифрования.

DES является блочным алгоритмом шифрования с длиной блока 64 бита и симметричными ключами длиной 56 бит. На практике ключ обычно имеет длину 64 бита, где каждый восьмой бит используется для контроля четности остальных битов ключа.

Структура алгоритма приведена на рис. 1.9 и отражает последовательность действий, совершаемых в одном раунде; всего для получения блока зашифрованного сообщения проходит 16 раундов. Эта величина используется в DES по следующим причинам:

- 12 раундов являются минимально необходимыми для обеспечения должного уровня криптографической защиты;
- при аппаратной реализации использование 16 раундов позволяет вернуть преобразованный ключ в исходное состояние для дальнейших преобразований;
- данное количество раундов необходимо, чтобы исключить возможность проведения атаки на блок зашифрованного текста с двух сторон.

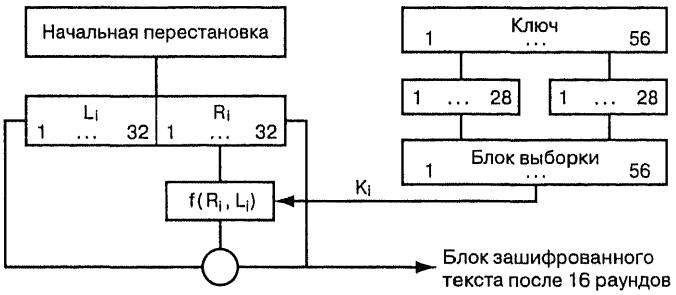

Рис. 1.9
Структура алгоритма DES

В некоторых реализациях DES блоки открытого сообщения перед тем, как они будут загружены в регистр сдвига длиной две ячейки и размером ячейки 32 бита, проходят процедуру начальной перестановки, которая применяется для того, чтобы осуществить начальное рассеивание статистической структуры сообщения. Пример начальной перестановки приведен в табл. 1.2.

Таблица 1.2. Начальная перестановка

58	50	42	34	26	18	10	2	60	52	44	36	28	20	12	4
62	54	46	38	36	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

В случае использования начальной перестановки после завершения 16 раундов к полученному блоку применяется обратная перестановка.

Работа алгоритма заключается в следующем:

1. Входной блок разбивается на две части по 32 бита в каждой (L_i – левая половина, R_i – правая половина).
2. Правая половина преобразуется функцией f с использованием текущей ключевой последовательности длиной 48 бит, снятой с выхода блока выработки ключевой последовательности.
3. Результат преобразования правой части складывается по модулю 2 с левой частью, а результат сложения записывается в исходный регистр, при этом исходная правая часть при помощи операции сдвига записывается на место исходной левой части.

Таким образом, в регистре оказывается следующая последовательность:

$L_i = R_{i-1}$
$R_i = L_{i-1} \oplus f(R_{i-1}, K_i)$

Данная процедура повторяется 16 раз, только в последнем цикле замены местами правой и левой части не происходит. По завершении последнего

цикла полученная последовательность проходит процедуру завершающей перестановки, которая задается подстановкой (табл. 1.3), являющейся обратной к начальной подстановке и учитывающей, что в последнем цикле половины блока открытого текста не меняются местами.

Таблица 1.3. Завершающая перестановка

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

Преобразование f (рис. 1.10) начинается с операции расширения исходной 32-битной последовательности до 48 бит. Эта операция предполагает дописывание в исходную последовательность отдельных битов в соответствии с подстановкой (см. табл. 1.4).

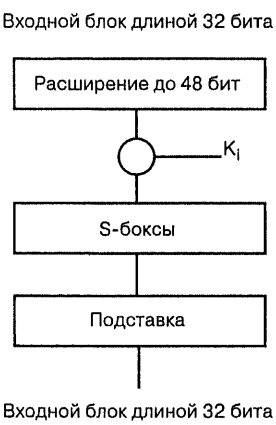

Рис. 1.10
Структура функции f

Таблица 1.4. Подстановка расширения

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

Результат преобразования суммируется по модулю 2 с 48-битной ключевой последовательностью, которая вырабатывается из 56-битного ключа, записанного в два 28-битных циклических регистра сдвига, которые перемещают содержимое в каждом такте на количество битов, зависящее от номера раунда (табл. 1.5).

Таблица 1.5. Таблица зависимости количества сдвигаемых битов от номера раунда

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

Результирующая ключевая последовательность получается путем выборки 48 бит из содержимого регистров в соответствии с подстановкой (табл. 1.6).

Таблица 1.6. Подстановка для выборки ключа

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

Полученный путем сложения 48-битный вектор поступает на вход S-боксов, основная задача которых заключается в замене 48-битного вектора на 32-битный. Всего в DES используются восемь S-боксов с 6-битными входами и 4-битными выходами. Подстановка в S-боксах осуществляется в соответствии с табл. 1.7: здесь номер строки задается первым и последним входом S-бокса, а номер столбца – средними четырьмя битами входа. Битовое представление числа в ячейке задано входной последовательностью и будет являться выходом S-бокса.

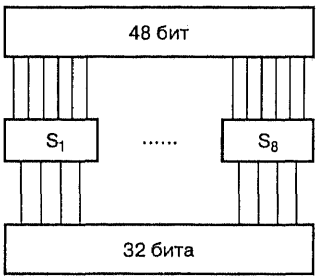

Рис. 1.11. Структура S-боксов

Таблица 1.7. Подстановки в S-боксах

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

Таблица 1.7. Подстановки в S-боксах (окончание)

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

Аналитическая сложность дешифрования DES зависит от математических свойств S-боксов, поскольку именно в них реализуются нелинейные преобразования. Все остальные операции в этом алгоритме носят линейный характер, и аналитическое вычисление подобной зависимости не представляет труда для криптоаналитика противника.

Выход S-боксов поступает на P-блок, где происходит перестановка (табл. 1.8).

Таблица 1.8. Перестановка в P-блоке

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

Расшифрование в алгоритме DES происходит аналогично зашифрованию с той только разницей, что выборка ключевой последовательности в раундах расшифрования будет обратная, то есть K_{16}, K_{15}...K_1, если в процессе зашифрования выборку ключевой последовательности обозначать K_1, K_2...K_{15}, K_{16}.

Стойкость DES

Говоря о DES, невозможно обойти стороной тему безопасности данного алгоритма и возможных атак на него. Многолетний опыт эксплуатации DES и его открытость (исходные тексты алгоритма и документацию на него можно встретить в открытых источниках) привели к тому, что DES стал одним из популярнейших алгоритмов с точки зрения проверки тех или иных методов дешифрования и криптоанализа. За все время существования алгоритма на него было проведено немало атак; при этом внимательно изучались и учитывались его многочисленные слабости, выявленные за столь долгий срок эксплуатации. Следует иметь в виду: некоторые из атак реализуемы только исходя из предположения, что атакующий обладает некоторыми возможностями (вычислительными, временными и т.д.), и в ряде случаев подобные попытки с точки зрения практического осуществления относятся к разряду возможных лишь в теоретическом плане. Хотя не исключено, что со временем они вполне могут стать практически осуществимыми.

Среди основных недостатков DES, существенно снижающих уровень его безопасности, можно выделить следующие:

- наличие слабых ключей, вызванное тем, что при генерации ключевой последовательности используются два регистра сдвига, которые работают независимо друг от друга. Примером слабого ключа может служить 1F1F1F1F 0E0E0E0E (с учетом битов контроля четности), В данном случае результатом генерации будут ключевые последовательности, одинаковые с исходным ключом во всех 16 раундах. Существуют также разновидности слабых ключей, которые дают при генерации всего лишь две (четыре) ключевые последовательности. Для неполнораундовых схем DES характерно наличие связанных ключей (например, ключ, полученный из другого ключа посредством инверсии одного бита);
- небольшая длина ключа 56 бит (или 64 бита с контролем четности). На современном уровне развития микропроцессорных средств данная длина ключа не может обеспечивать должной защиты для некоторых типов информации (табл. 1.1 и 1.10). Применение тройного DES (Triple DES, схема его реализации приведена на рис. 1.12) не дает ощутимого результата, хотя в новой версии используются три разных ключа ($К_1$, $К_2$, $К_3$). Дело в том, что в конечном итоге работа с тремя ключами эквивалентна зашифрованию на другом ключе $К_4$, то есть для любых $К_1$, $К_2$, $К_3$ найдется такой ключ $К_4$, что:

  $EK_3(DK_2(EK_1(P))) = EK_4(P)$;

- избыточность ключа, обусловленная контролем четности для каждого в отдельности байта ключа. Бихам и Шамир предложили достаточно эффективную атаку на реализацию DES в смарт-картах или банковских криптографических модулях, использующих EEPROM-память для хранения ключей. Эта методика наглядно высветила еще одну очевидную слабость DES, заключающуюся в наличии контроля четности каждого байта ключа, в силу которой и создается его избыточность, что позволяет восстанавливать ключи, хранящиеся в памяти устройства, при сбое в данном участке памяти;
- использование статических подстановок в S-боксах, что, несмотря на большое количество раундов, позволяет криптоаналитикам проводить атаки на этот алгоритм.

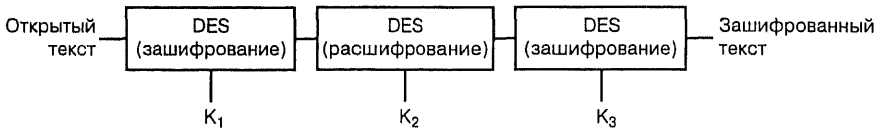

Рис. 1.12. Схема Triple DES

Здесь следует заметить, что на сегодняшний день автору неизвестны успешные атаки на 16-раундовый DES, проведенные на основании последнего в списке факта, однако успешные атаки на неполнораундовые схемы DES имели место. Так, Мартин Хэллман предложил атаку на 8-раундовый DES. Она позволяет восстанавливать на рабочей станции SUN-4 10 бит ключа за десять секунд. В случае выбора 512 открытых текстов вероятность успеха составляет 80%, а при выборе 768 открытых текстов – 95%. Восстановив 10 бит ключа, можно воспользоваться алгоритмами перебора всех оставшихся вариантов, сводя таким образом задачу нахождения 56-битного ключа к поиску 46-битного ключа.

Учитывая вышеизложенное, можно с уверенностью сказать, что на сегодняшний день (с точки зрения криптографической стойкости и обеспечения надежного функционирования систем криптографической защиты информации) использование DES в практических целях является весьма опасным решением.

1.2.3. Алгоритм блочного шифрования

Отечественным аналогом DES является алгоритм блочного шифрования, специфицированный в ГОСТ 28147-89. Разработчики данного алгоритма сумели органично соединить в нем две важные, но трудносочетающиеся друг с другом характеристики:

- высокую криптографическую стойкость алгоритма;
- возможность эффективного программного исполнения (за счет использования узлов, реализуемых на современной аппаратно-программной платформе).

Алгоритм (рис. 1.13) работает с 64-битными входными блоками, 64-битными выходными блоками и ключами длиной 256 бит. Использование 256-битного ключа позволяет не обращать внимания на возможность атаки с применением «грубой силы», поскольку мощность большинства ключей равна 2^{256}. При этом данный алгоритм предполагает как эффективную программную, так и аппаратную реализацию.

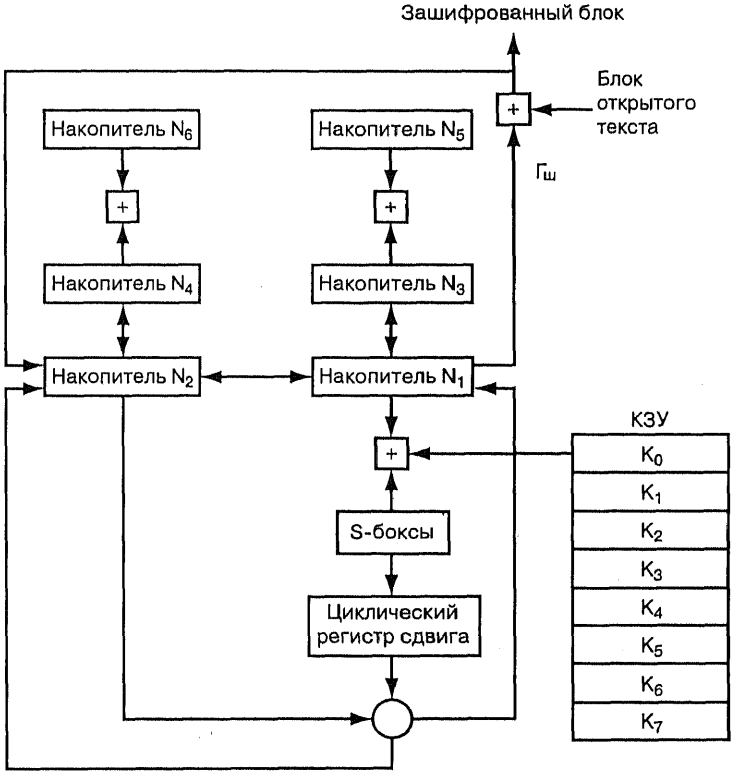

Рис. 1.13
Схема алгоритма ГОСТ 28147-89

Алгоритм может применяться в следующих рабочих режимах:

- простая замена;
- гаммирование;
- гаммирование с обратной связью;
- выработка имитовставки.

Для всех четырех режимов применяется одно общее преобразование, которое можно записать в следующем виде:

$L_i = R_{i-1}$
$R_i = L_{i-1} \oplus f(R_{i-1}, K_i)$ $i = 1 \div 31$,

где L и R – соответственно левая и правая часть 64-битного блока.

Это преобразование характерно для всех раундов (всего их 32), кроме последнего. Для него преобразование будет иметь вид:

$L_i = L_{i-1} \oplus f(R_{i-1}, K_i)$
$R_i = R_{i-1}$ $i = 32$

Для каждого раунда функция f остается неизменной. Перед началом преобразования блок открытого текста разбивается на две 32-битные половины L и R, которые помещаются в два 32-разрядных регистра N_1 и N_2 соответственно. Также перед началом зашифрования 256-битный ключ, разбитый на восемь частей по 32 бита каждая (K_0...K_7), помещается в ключевое запоминающееся устройство (КЗУ). Далее содержимое регистра N_1 суммируется по модулю 2^{32} с очередным заполнением устройства, а ключевые последовательности выбираются из КЗУ в следующем порядке:

- в раундах с 1-го по 24-й – K_0, K_1, K_2, K_3, K_4, K_5, K_6, K_7;
- в раундах с 25-го по 32-й – K_7, K_6, K_5, K_4, K_3, K_2, K_1, K_0.

Полученная битовая последовательность разбивается на восемь блоков по 4 бита в каждом и поступает на вход S-боксов. В них реализуется подстановка, имеющая, например, следующий вид:

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

Числовое представление битового входа однозначно определяет последовательный номер числа в подстановке и битовое представление соответствующего числа; таким же оно будет и на выходе S-бокса. Для каждого S-бокса задается своя подстановка, которая является долговременным секретным ключом. Генерация подстановок в S-боксах является сложной математической задачей, от решения которой как раз и зависит стойкость этого алгоритма. Подобный прием служит также для развязывания различных сетей передачи данных – очевидно, что расшифрование будет успешным, если принимающая и передающая стороны используют одинаковые подстановки. Таким образом, в сети передачи данных абоненты с одинаковыми подстановками имеют возможность обмениваться зашифрованной информацией, что позволяет создавать выделенные группы пользователей с засекреченной связью.

После S-боксов последовательность поступает в циклический регистр сдвига, который осуществляет смещение на 11 шагов влево. Использование

данного регистра вместо перестановок, используемых в DES, обусловлено тем, что данный регистр легко реализуется как программно (за счет использования битовой операции циклического сдвига), так и аппаратно. При этом эффект распространения влияния каждого бита блока открытого текста на каждый бит блока зашифрованного текста достигается за счет прохождения преобразуемого блока через данный регистр 32 раза.

Далее происходит поразрядное суммирование по модулю 2 содержимого регистра циклического сдвига с содержимым регистра N_2. Результат суммирования записывается в регистр N_1, а содержимое регистра N_2 принимает предыдущее значение N_1.

Все эти операции составляют один раунд преобразования (реализация функции f) очередного блока открытого текста. Специфика различных типов применения ГОСТ вносит свои дополнения в раунды зашифрования и соответственно расшифрования.

Режим простой замены

В данном режиме блок открытого текста шифруется описанным выше методом без каких-либо добавлений. Результат зашифрования после прохождения 32 раундов находится в регистрах N_1 и N_2.

Процесс расшифрования аналогичен процессу зашифрования с той лишь разницей, что ключи считываются из КЗУ в следующем порядке:

- с 1-го по 8-й раунд – K_0, K_1, K_2, K_3, K_4, K_5, K_6, K_7;
- с 9-го по 32-й раунд – K_7, K_6, K_5, K_4, K_3, K_2, K_1, K_0.

Этот режим рекомендуется применять только для коротких сообщений, поэтому на практике его в основном используют для генерации сеансовых ключей (или другой ключевой информации) и обеспечения имитозащиты сеансовых ключей, передаваемых в зашифрованном виде по открытым каналам связи.

Режим гаммирования

В этом режиме зашифрование происходит путем побитного сложения по модулю 2 блока открытого текста и блока гаммы длиной 64 бита, содержащейся в регистрах N_1 и N_2. Если блок открытого текста короче 64, то лишние разряды гаммы отбрасываются.

Выработка гаммы происходит следующим образом:

1. В накопители N_1 и N_2 записывается синхропосылка S длиной 64 бита.
2. S шифруется в режиме простой замены, и результат зашифрования из регистров N_1 и N_2 записывается в регистры N_3 и N_4 соответственно.

3. Содержимое регистра N_4 суммируется по модулю $(2^{32} - 1)$ с содержимым регистра N_6, в котором находится константа C_1 $(2^{24} + 2^{16} + 2^8 + 2^2)$, а содержимое регистра N_5 суммируется по модулю 2^{32} с содержимым регистра N_5, в котором находится константа C_2 $(2^{24} + 2^{16} + 2^8 + 1)$.
4. Содержимое регистров N_3 и N_4 записывается в регистры N_1 и N_2 соответственно, и их содержимое образует первый 64-битный блок гаммы.
5. Алгоритм генерации остальных блоков гаммы заключается в суммировании содержимого регистров N_3 и N_4 с содержимым регистров N_5 и N_6 соответственно, с сохранением результата в N_3 и N_4, переписывании содержимого N_3 и N_4 в N_1 и N_2 соответственно и последующем шифровании в режиме простой замены содержимого регистров N_1 и N_2.

Синхропосылка S передается на приемную сторону в открытом или в зашифрованном виде. В некоторых системах связи генерация синхропосылки проходит процедуру согласования между сторонами, участвующими в информационном обмене.

Расшифрование происходит аналогично шифрованию путем сложения по модулю 2 блоков выработанной на приемной стороне гаммы и блоков зашифрованного текста.

Режим гаммирования с обратной связью

Шифрование в этом режиме производится путем суммирования по модулю 2 блоков открытого текста с блоками выработанной гаммы. Синхропосылка S зашифровывается в режиме простой замены, и полученный результат из регистров N_1 и N_2 является первым блоком гаммы, которая суммируется по модулю 2 с первым блоком открытого текста. Блок зашифрованного текста одновременно является заполнением регистров N_1 и N_2 для шифрования следующего блока открытого текста.

Режим выработки имитовставки

Данный режим предназначен для защиты от навязывания ложных сообщений во время передачи информации по открытым каналам. Защита обеспечивается путем генерации *имитовставки* – отрезка информации фиксированной длины, полученного по определенному правилу из открытых данных и ключа и добавленного к зашифрованным данным для обеспечения имитозащиты. Его длина L выбирается в зависимости от требований, выдвигаемых к системам имитозащиты в конкретных сетях передачи данных, при этом необходимо учитывать, что вероятность навязывания ложных сообщений обычно принимается равной 2^{-L}.

Генерация имитовставки производится следующим образом:

1. Первый блок открытого текста записывается в регистры N_1 и N_2 и зашифровывается в режиме простой замены на первых 16 раундах. Заполнение КЗУ соответствует ключам, на которых производится зашифрование блоков открытого текста.
2. Полученный результат из регистров N_1 и N_2 суммируется по модулю 2 со следующим блоком открытого текста.
3. Результат суммирования заносится в регистры N_1 и N_2 и подвергается зашифрованию в режиме простой замены на 16 раундах и т.д. до последнего блока открытого текста.
4. Из содержимого регистров N_1 и N_2 выбирается битовая последовательность длины L, которая и является имитовставкой, передаваемой по открытому каналу связи вместе с блоками зашифрованного текста.

Процедура проверки имитовставки на приемной стороне заключается в расшифровании блоков открытого текста и выработке из них имитовставки способом, описанным ранее. Далее сравнивают полученную по каналу связи имитовставку с сгенерированным значением, и если они не совпадают, то блоки открытого текста считаются переданными неверно.

Стойкость ГОСТ 28147-89

Говоря о безопасности алгоритма ГОСТ 28147-89, необходимо отметить, что на сегодняшний день для него пока не предложено практически реализуемых атак, более эффективных, чем атака методом «грубой силы». Высокие криптографические свойства этого алгоритма достигаются за счет:

- большой длины ключа (256 бит); если учесть, что подстановки в S-боксах могут являться секретными, то общий объем секретной информации окажется равным 610 битам;
- 32 раундов преобразований, используемых в ГОСТе. Например, уже после восьми раундов единичное изменение входной последовательности отразится на изменении всех битов выхода.

Однако при использовании ГОСТа особое внимание следует уделять выбору подстановок S-боксов, поскольку ошибка или недосмотр может существенно снизить криптостойкость алгоритма.

1.2.4. Применение алгоритмов блочного шифрования

Основной на сегодняшний день задачей, решаемой с помощью алгоритмов блочного шифрования, является обеспечение конфиденциальности информации пользователя. Принадлежащие ему сведения, критичные

к потере этого свойства, могут быть представлены в виде файлов или записей в базе данных на локальной машине; их также можно отсылать и получать по общедоступным сетям передачи данных. Подобный вариант использования алгоритмов блочного шифрования считается классическим, и здесь мы не будем подробно его описывать.

Другим вариантом практического применения блочных алгоритмов является использование их для обеспечения имитозащиты передаваемой по каналам связи информации. Имитозащита сообщения, разбитого на блоки, представляет собой дополнительный блок (имитовставку), включаемый при передаче в конец сообщения. Для оформления имитовставки применяются алгоритмы блочного шифрования в режиме гаммирования с самовосстановлением или шифрования со сцеплением блоков. Процедура выработки имитовставки для большинства алгоритмов блочного шифрования аналогична процедуре, описанной в подразделе 1.2.3.

В настоящее время в связи с повсеместным использованием сетевых ОС пользователям предоставляется большое количество услуг и дается возможность работать как локально, так и удаленно; более того, распределенную структуру могут иметь даже компоненты программного обеспечения. Надежное функционирование подобных систем напрямую связано с обеспечением безотказного и качественного взаимодействия удаленных (распределенных) компонентов системы. При таких тесных операционных связях появляется некоторая служебная информация, критичная с точки зрения надежности и бесперебойности работы всей системы в целом. Так, например, в домене Windows NT обмен информацией между контроллером домена и компьютером, на котором пользователь пытается пройти авторизацию, происходит в открытом виде, что дает злоумышленнику, следящему за сетевыми соединениями, потенциальную возможность заменить идентификатор безопасности пользователя идентификатором безопасности администратора. При этом злоумышленник получит возможность несанкционированного доступа к не предназначенной для него информации. Защитой от подобной угрозы может являться шифрование сетевого трафика с помощью алгоритмов блочного шифрования.

Таким образом, с повышением сложности ОС и качественного уровня предоставляемого обслуживания появляется необходимость применения алгоритмов блочного шифрования, особенно для защиты потоков служебной информации.

1.3. Асимметричные алгоритмы шифрования

1.3.1. Общие сведения

Развитие основных типов криптографических протоколов (ключевой обмен, электронно-цифровая подпись, аутентификация, так называемые *эзотерические* протоколы) было бы невозможно без создания открытых ключей и построенных на их основе асимметричных алгоритмов шифрования.

В данном разделе мы ограничимся кратким описанием идеи построения асимметричных алгоритмов (на примере алгоритма RSA), а также затронем вопросы теоретической стойкости подобных алгоритмов (в частности, практические аспекты обеспечения стойкости алгоритма RSA). В силу того, что с точки зрения практической реализации асимметричные алгоритмы по своей природе являются достаточно «медленными», здесь также будет уделено внимание вопросам ускорения основных типов вычислений, используемых в этих алгоритмах.

С одной стороны, идея асимметричных алгоритмов тесно связана с развитием теории односторонних функций, с другой – с теорией сложности. Под односторонней функцией мы будем понимать легковычисляемое отображение $f(x)$: $X \rightarrow Y$, $x \in X$, при этом обратное отображение является сложной задачей. Она называется *трудновычисляемой*, если нет алгоритма для ее решения с полиномиальным временем работы. *Легковычисляемой* будем называть задачу, имеющую алгоритм со временем работы, представленным в виде полинома низкой степени относительно входного размера задачи, а еще лучше алгоритм с линейным временем работы.

Заметим, что на сегодняшний день теоретически не доказано существование односторонних функций. Использование их в качестве основы асимметричных алгоритмов шифрования допустимо только до тех пор, пока не найдены эффективные алгоритмы, выполняющие обращение односторонних функций за полиномиальное время.

Широкое распространение асимметричных алгоритмов шифрования вызвано необходимостью иметь два ключа – открытый для зашифрования (хотя в случае применения асимметричного алгоритма с целью организации, например электронно-цифровой подписи, открытый ключ может использоваться для расшифрования) и закрытый для расшифрования. Соответственно, вводя понятие открытого ключа, то есть ключа, потенциально известного всем, мы избавляемся от необходимости решать сложную задачу обмена секретными ключами. Так, например, в некоторых случаях

необходимость хранить секретные ключи приводит к образованию больших объемов статистической информации, что порой практически неосуществимо. Такое, в частности, может случиться при необходимости использования сети Internet как среды передачи данных и – одновременно – при желании иметь должный уровень безопасности.

Следовательно, имея в своем распоряжении механизм распределения открытых ключей (заметим, что данная задача с практической точки зрения наиболее осуществима, нежели задача обмена секретными ключами), мы можем послать ключ по открытым каналам связи и затем установить защищенный канал передачи данных, пусть даже при этом возникнут проблемы обеспечения безопасности открытых ключей. Злоумышленник в случае такого обмена или при хранении ключей в открытых справочниках старается подменить их, что может привести к установлению ложной связи, и применять их вместо легального пользователя, чей открытый ключ был скомпрометирован. Пути и методы решения данной проблемы будут рассмотрены ниже, в разделе, посвященном обмену ключевой информацией.

Развитием идеи односторонних функций явилось построение односторонних *функций с секретом*. Такой функцией называется $f(x) = y$, значение которой, как и в предыдущем случае, легко вычислить, тогда как обратное значение без знания некоторого секрета трудно вычислить. Знание же секрета позволяет достаточно просто реализовывать операцию обращения односторонних функций с секретом. На практике при применении асимметричного алгоритма шифрования в роли секретного ключа выступает само знание секрета, а в роли открытого ключа – знание процедуры вычисления односторонней функции с секретом.

Вместе с тем необходимо отметить, что стойкость большинства современных асимметричных алгоритмов базируется на двух математических проблемах, которые на данном этапе являются трудновычисляемыми даже для метода «грубой силы»:

- дискретное логарифмирование в конечных полях;
- факторизация больших чисел.

Поскольку на сегодняшний день не существует эффективных алгоритмов решения данных задач либо их решение требует привлечения больших вычислительных ресурсов или временных затрат, эти математические задачи нашли широкое применение в построении асимметричных алгоритмов. Их стойкость рассматривается как возможность свести проблему вскрытия алгоритмов к решению одной из вышеперечисленных математических головоломок.

1.3.2. Стандарт асимметричного шифрования RSA

Самым распространенным алгоритмом асимметричного шифрования является алгоритм RSA, названный по первым буквам имен его создателей (Rivest, Shamir, Adleman). Разработчикам данного алгоритма удалось эффективно воплотить идею односторонних функций с секретом. Стойкость RSA базируется на сложности факторизации больших целых чисел.

В 1993 году метод RSA был обнародован и принят в качестве стандарта (PKCS # 1: RSA Encryption Standard). RSA можно применять как для зашифрования/расшифрования, так и для генерации/проверки электронно-цифровой подписи (ЭЦП).

Генерация ключей

Каждый участник информационного обмена генерирует пару ключей (открытый и секретный) в соответствии со следующими правилами:

1. Выбираются два больших простых целых числа p и q приблизительно одинакового размера. Выбор чисел p и q определяется следующими соображениями:
   - увеличение порядка чисел ведет к замедлению операции зашифрования/расшифрования;
   - увеличение порядка чисел p и q ведет к увеличению стойкости алгоритма, поэтому при выборе чисел следует руководствоваться практической необходимостью. Так, например, при реализации RSA в интеллектуальных карточках с точки зрения скоростных параметров системы не следует выбирать слишком большие p и q в связи с ограниченными вычислительными возможностями, заложенными в данных устройствах. На практике обычно рекомендуется выбирать числа, содержащие порядка 150–200 десятичных знаков.
2. Вычисляется модуль системы $n = pq$ и $\varphi(n) = (p - 1)(q - 1)$ – функция Эйлера.
3. Выбирается достаточно большое число e, удовлетворяющее условию $1 < e < \varphi(n)$, и взаимно простое с $\varphi(n)$.
4. Используя расширенный алгоритм Евклида, вычисляется большое целое число d, отвечающее условию:

   $ed = 1 \pmod{\varphi(n)}$
   $1 < d < \varphi(n)$

Таким образом, секретным ключом является пара чисел (n, d), а открытым – пара чисел (n, e). Открытый ключ помещается в общедоступный

справочник (детально этот процесс будет описан ниже). В соответствии с PKCS # 1 формат представления ключей имеет следующий вид:

```
RSAPublicKey := Sequence                    {
modulus Integer, n                          (модуль n)
public Exponent Integer, e                  (открытая экспонента
                                            шифрования)

                                            }
Формат RSAPublicKey и RSAPrivateKey представлен в соответствии
с ASN.1 (X.208 CCITT).
Секретный ключ имеет вид:
RSAPrivateKey: = Sequence                   {
version Version                             Идентификатор версии
                                            предназначен для
                                            совместимости с последующими
                                            версиями.
modulus Integer, n                          Модуль n.
public Exponent Integer, e                  Открытая экспонента
                                            зашифрования e.
private Exponent Integer, d                 Секретная экспонента
                                            расшифрования d.
prime 1 Integer, p                          Простое целое p.
prime 2 Integer, q                          Простое целое q.
exponent1 Integer, d mod (p-1)              Данные экспоненты
                                            используются для обеспечения
                                            эффективности работы
                                            алгоритма.
exponent 2 Integer, d mod (q-1)
coefficient Integer, (inverse of q) mod p   Инверсия q (q^-1(mod p),
                                            удовлетворяющая условию
                                            qq^-1 = 1(mod p).
                                            }
Version := Integer
```

Зашифрование/расшифрование

После выбора параметров системы (n, e, d) абонент готов к приему зашифрованных сообщений. Их передача состоит из следующих шагов:

1. Входное сообщение разбивается на блоки m_i, их размер определяется целым k, соответствующим неравенству $10^{k-1} < n < 10^k$.
2. Вычисляется значение $c_i = m_i^e \bmod n$.
3. Значение c_i, которое является зашифрованным блоком сообщения, посылается по открытым каналам передачи данных.

Расшифрование заключается в вычислении значения $m_i = c_i^d \bmod n$.

Доказательством того факта, что расшифрование выполняется только абонентом, знающим секретную экспоненту зашифрования d, является $c_i^d \bmod n = m_i^{e\,d} \bmod n$, с учетом, что $ed = 1 \pmod{\varphi(n)}$. При этом получаем $ed = 1 + k\varphi(n)$, где k – целое число, удовлетворяющее этому равенству. Поскольку $m_i^{\varphi(m)} \bmod n$ – единичный элемент группы относительно операции умножения, элемент m_i тоже принадлежит к данной группе. В этом случае:

$c_i^d \bmod n = m_i m_i^{k\varphi(m)} \bmod n = m_i$.

1.3.3. Стойкость алгоритма RSA

Очевидно, что стойкость RSA и асимметричных алгоритмов в целом зависит от сложности обращения односторонних функций, лежащих в основе данного типа алгоритмов. В RSA сложность обращения односторонней функции $F(x) = x^y \pmod n$ зависит от сложности разложения на множители модуля n. При этом следует иметь в виду, что это утверждение является всего лишь предположением, поскольку строгих математических доказательств эквивалентности данных проблем пока не существует.

В настоящем разделе рассматриваются основные атаки на алгоритм RSA, известные на сегодняшний день. Следует отметить, что стойкость RSA может быть снижена за счет некорректного выбора параметров алгоритма.

Среди наиболее известных недостатков алгоритма RSA можно выделить некорректный выбор значений p и q. Числа p и q должны быть простыми и не содержаться ни в одной из известных таблиц простых чисел. Эти числа также не должны быть близкими друг к другу. Если $(p - q)/2$ мало и $(p + q)/2$ немного больше, чем $\sqrt{n}$, то при $(p + q)^2/4 - n$ C $(p - q)^2/4$ левая часть равенства образует полный квадрат. При факторизации n перебираем целые числа на соответствие неравенству $x > \sqrt{n}$ до тех пор, пока не найдем такое значение, что $x^2 - n = y^2$. Тогда $p = x + y$ и $q = x - y$. Учитывая изложенный факт, а также ряд других атак, основанных на неправильном выборе чисел p и q, сформулируем следующие требования к выбору чисел p и q:

1. Данные числа должны быть большими числами одинаковой длины; например: если n = 1024 бита, то длина p и q должна быть равна 512 битам.
2. Различие между p и q должно быть большим.
3. Числа p и q должны быть строго простыми. Число называется строго простым, если выполнены условия:

   а) p – 1 должно иметь большой простой делитель (обозначим его через r);

б) p + 1 должно иметь большой простой делитель;
в) r – 1 должно иметь большой простой делитель.

Требование а) обусловлено противодействием алгоритмам факторизации n (один из таких алгоритмов был предложен Поллардом). Аналогично обосновывается требование б). Выполнение на практике требования в) позволяет противостоять циклическим атакам.

Циклические атаки

Пусть $c = m^e \bmod n$ – зашифрованное сообщение, k – положительное целое, при этом $c^{e^k} \equiv c \pmod n$. Тогда зашифрованием является подстановка на множестве $\{0,1 \ldots n-1\}$, в которой присутствует k, что может привести к ситуации, когда $c^{e^{k-1}} = m \pmod n$. Злоумышленник вычисляет значения $c^e \bmod n$, $c^{e^2} \bmod n$ и т.д. до тех пор, пока не будет получено значение c. Если $c^{e^k} \bmod n = c$, тогда $c^{e^{k-1}} = m \pmod n$. Основой циклической атаки является нахождение такого небольшого положительного целого u, при котором выполняется условие $f(u) = НОД\ (c^{e^u} - c, n) > 1$. Если $c^{e^u} = c \pmod p$ и $c^{e^u} \neq c \pmod q$, то $f(u) = p$. Аналогично, если $c^{e^u} \neq c \pmod p$ и $c^{e^u} = c \pmod q$, то $f(u) = q$. В любом из этих случаев злоумышленник получает разложение n на множители.

При этом существует вероятность возникновения одной из следующих ситуаций:

- если выбрать общий модуль n для каждого участника при различных парах (e_i, d_i), появляется потенциальная возможность разложить на множители модуль n (правда, для этого необходимо знать каждую пару (e_i, d_i)). Более того, если одно и то же сообщение отправлено несколькими участниками, злоумышленник, перехватив эти сообщения, может восстановить открытый текст на основе знания общедоступных параметров системы. Учитывая сказанное, при выборе параметров в асимметричных алгоритмах шифрования необходимо, чтобы n создавалось для каждого участника системы отдельно. Кроме того, развитие эффективных алгоритмов факторизации больших целых чисел предъявляет жесткие требования к размеру n; на сегодняшний день минимально рекомендуемая длина n должна быть равна 768 битам;
- при выборе экспонент e и d тоже могут возникать ситуации, существенно ослабляющие стойкость RSA. При выборе экспоненты e обычно ориентируются на обеспечение приемлемой скорости операций зашифрования/расшифрования, что приводит к выбору малых значений e. Например, рассмотрим случай, когда e = 3 для участников A, B и C (предполагается также, что модули n у каких-либо двух

участников взаимно просты). В этом случае злоумышленник, перехватив одинаковые сообщения, переданные всем участникам (например, многоадресная доставка), $c_i = m^e \bmod n$, где i = A, B, C, может вычислить число $x = m^3 \bmod (n_A n_B n_C)$. Поскольку $m^3 < n_A n_B n_C$, это возможно только в случае, если $x = m^3$. Тогда злоумышленник, вычислив кубический корень, может найти m. С точки зрения достижения приемлемых скоростных параметров и обеспечения должного уровня безопасности имеет смысл выбрать число $e = 2^{16} + 1$, являющееся числом Ферма и имеющее в битовом представлении всего лишь две единицы. Выбор небольшого числа d также опасно тем, что если НОД (p – 1, q – 1) и d приблизительно равны n/4, то существуют алгоритмы факторизации n на основе знания только открытого ключа. Кроме того, в случае небольшого значения d оно может быть найдено простым перебором всех возможных значений.

Для RSA характерно наличие так называемых *нешифруемых* блоков открытого сообщения, то есть таких m, для которых выполняется равенство $m^e = m \pmod n$. Например, m = 0, m = 1 и m = n – 1 всегда являются нешифруемыми блоками сообщения. Необходимым и достаточным условием нешифруемости сообщения является $m^{e-1} = 1 \pmod n$.

В общем случае под единицей в равенстве следует понимать отдельный элемент группы, к которой принадлежит m. Как видно из приведенного условия нешифруемости, таких блоков сообщения совсем немало. Точное количество нешифруемых блоков сообщения равно (1 + НОД(e, p – 1)) × × (1 + НОД (e – 1, q – 1)). Данная особенность алгоритма RSA налагает определенные требования при выборе значения e. Так, например, если e равно 1 + k[φ(q), φ(p)], все элементы кольца сообщений окажутся нешифруемыми.

Необходимо отметить, что RSA критичен также к адаптивной атаке с выбором зашифрованных сообщений. Этот метод основан на гомоморфных свойствах RSA. Другими словами, для любых открытых сообщений m_1 и m_2 и для соответствующих зашифрованных сообщений c_1 и c_2 выполняется следующее равенство: $(m_1 m_2)^e = m_1^e m_2^e = c_1 c_2 \pmod n$.

1.3.4. Методы ускорения вычислений, применяемых в асимметричных алгоритмах

Одной из основных проблем, возникающих при использовании асимметричных алгоритмов, является низкая скорость проведения операций зашифрования/расшифрования. Этот факт особенно характерен для реализаций алгоритмов на устройствах с небольшими вычислительными

возможностями (реализация ЭЦП на интеллектуальных карточках). Один из выходов – уменьшение размерности параметров системы, но, как показано в разделе 1.3.5, это чревато уменьшением стойкости алгоритма.

Другим решением этой же проблемы является применение эффективных процедур проведения основных видов математических вычислений, используемых в асимметричных алгоритмах шифрования. Причем эффективность в зависимости от ситуации может трактоваться не только как высокая скорость работы, но и как минимальный объем памяти, минимальный программный код и их совокупность. Суть большинства методов ускорения вычислений заключается в том, чтобы свести операции модульного умножения и возведения в степень к выполнению последовательности операций модульного сложения/вычитания.

Алгоритм модульного умножения

Выполнение операции $A \times B \bmod n$ обычным путем, то есть умножение A на B и затем применение модульной редукции, не является ни быстрым и экономичным способом умножения. Данное вычисление гораздо лучше выполнить, используя, например, разложение числа B вида $B = b_n\lambda^n + + b_{n-1}\lambda^{n-1} + ... + b_1\lambda + b_0$ и последовательное умножение вида $c = Ab_i$ (при этом получаются числа с размерностью $n + 1$), а затем вычислить mod n. (рис. 1.14).

```
  F=0
  Q=0
  for i from n to 0 do
begin:
  F=2*F+Ab_i
  Q=|F/n|
  F=F-Q*n
end
  return ( F )
```

Рис. 1.14
Алгоритм ускоренного модулярного умножения A × B mod n

По мнению авторов данного алгоритма, он не является одним из самых действенных алгоритмов модулярного умножения. Среди более эффективных следует отметить алгоритм умножения (см. рис. 1.15).

Модульное возведение в степень $A^B \bmod n$

Проведение операции модульного возведения в степень путем последовательного умножения числа A самого на себя, применяющейся для зашифрования/расшифрования в RSA, требует большого объема памяти

```
P = A
  P = P - N
  (A - N) = P
  for i from n to 0 do
begin:
  P = 2 * P   Реализуется сдвигом влево.
  if b_i = 1 then
  begin:
     if B < 0 then P = P + A
     else P = P + (A - N)
  end
  if P < 0 then P = P + N
  else P = P - N
end
  return ( F )
```

Рис. 1.15
Алгоритм модулярного умножения A × B mod n

для хранения полученных чисел и времени. Для решения этой проблемы существует метод последовательного возведения в квадрат, с помощью которого можно добиться, чтобы не возникали числа, большие A^2, и при этом существенно возросла скорость возведения в степень. Суть его состоит в следующем:

1. B представляется в двоичном виде:
   $B = \sum_{i=0}^{k} x_i 2^i$, где $x_i = 0 \div 1$ и $k = [\log_2 B] + 1$.
2. Затем вычисляются числа вида ($A^2 \bmod n$, где $i = 0 \div k$); данные вычисления требуют k возведений в квадрат.
3. Далее числа $A^2 \bmod n$ последовательно перемножаются с одновременным сведением по mod n; этих операций будет не более $k - 1$.

Таким образом, получаем не более $2k - 1$ операций модулярного умножения с числами, меньшими n. Данный метод разрешается модернизировать за счет сокращения числа ненулевых x_i в битовом представлении B: $B = b_n\lambda^n + b_{n-1}\lambda^{n-1} + ... + b_1\lambda + b_0$, где $b_n \neq 0$, $0 < b_i < \lambda$, $i = 0 \div n$. Выбор основания λ производится в соответствии с представлениями чисел в машине. Так, при употреблении двоичного представления числа используется на 20% меньше памяти, нежели при двоично-десятичном представлении.

Модульная редукция X mod n

Суть метода заключается в вычислении значения $r = X \bmod n$ (рис. 1.16), для чего предварительно вычисляется $\mu = [\lambda^{2k}/n]$. Эти числа можно запомнить, если выполняются многократные вычисления с использованием

данного модуля, при этом λ является основанием в разложении числа X; обычно на практике его значение принимается равным разрядности процессора.

Входные значения:
$X = x_{2k-1}\lambda^{2k-1} + \ldots + x_1\lambda + x_0$;
$n = n_{k-1}\lambda^{k-1} + \ldots + n_1\lambda + n_0$ $(m_{k-1} \neq 0)$;
$\mu = [\lambda^{2k}/n]$.

$q_1 = [x/\lambda^{k-1}]$
$q_2 = q_1\mu$
$q_3 = [q_2/\lambda^{k+1}]$
$r_1 = x \bmod \lambda^{k+1}$
$r_2 = q_3 n \bmod \lambda^{k+1}$
$r = r_1 - r_2$
if $r < 0$ then $r = r + \lambda^{k+1}$
while $r >= n$ do: $r = r - n$
return (r)

Рис. 1.16
Модульная редукция методом Барета

Несмотря на то что в представленном алгоритме производится большое количество операций деления, все они легко реализуются с помощью правого сдвига по основанию λ. При этом необходимо учитывать, что q_2 используется для вычисления q_3. В связи с этим в q_2 используется только $k + 1$ младших разрядов.

1.3.5. Практическое применение

Асимметричные алгоритмы используются в основном для решения задачи распределения сеансовых ключей по общедоступным каналам передачи данных. То есть симметричный ключ передается зашифрованным при помощи асимметричного алгоритма шифрования, и дальнейший обмен данными производится с использованием симметричных алгоритмов шифрования. Алгоритмы подобного рода также применяются в процедурах генерации и проверки электронно-цифровой подписи (ЭЦП). Хотя в общем случае эти алгоритмы нужны для зашифрования/расшифрования данных, но для длинных сообщений их использование в связи с низкой скоростью работы оказывается неэффективным. Например, алгоритм RSA в 1000 раз медленнее алгоритма DES.

Асимметричные алгоритмы шифрования также являются основополагающими для большинства криптопротоколов (данный аспект применения асимметричных алгоритмов будет рассмотрен ниже). В этом разделе уделяется внимание практическому применению алгоритма RSA в соответствии с PKCS # 1.

Зашифрование

Входными данными для процедур шифрования является строка (8-бит) с данными D, а также параметры алгоритма RSA (n, e). Строка D имеет длину в октетах не более, чем k – 11, где k – длина в октетах модуля n. Поскольку процесс шифрования не гарантирует контроля целостности данных, она обеспечивается за счет формата представления информации с вероятностью 2^{-16}. Процесс шифрования состоит из следующих шагов (рис. 1.17):

Рис. 1.17. Процесс зашифрования

1. Форматирование строки данных. Строка октетов D преобразуется в строку EB путем дописывания служебной информации и имеет следующий формат: EB = 00||BT||PS||00||D, где || – операция конкатенации строк. Значение BT указывает на тип строки EB. В существующей версии стандарта BT может принимать значение 00, 01 или 02. Для операции с секретным ключом используются значения 00 и 01, а для операции с открытыми ключами – значение 02. PS являются дополнительными данными (набивка) длиной k – 3 – ||D|| (||D|| – длина D в октетах) и используются для того, чтобы длина EB была эквивалентна k. В случае если BT = 00, то PS содержит значения 00; если BT = 01, то PS содержит ff; если BT = 02, то PS содержит псевдослучайные ненулевые значения. Использование значений 00 в EB обеспечивает уверенность в том, что при преобразовании EB в целое число полученное значение не будет больше n. Для значения BT = 00 строка D должна начинаться с ненулевого октета или иметь известную длину, чтобы обратное форматирование было произведено однозначно. Для типов блока 01 и 02 обратное форматирование тоже производится однозначно, поскольку PS не содержит нулевых октетов и к тому же отделен от D значением 00. При использовании BT = 02 псевдослучайное значение PS должно генерироваться для каждого сообщения отдельно. При BT = 02 необходимо, чтобы длина PS в октетах была меньше 8; данное требование обусловлено соображениями безопасности.
2. Перевод строки октетов в целочисленное значение осуществляется в соответствии со следующим равенством: $x = \sum_{i=1}^{k} 2^{8(k-1)} EB_i'$, где EB_i' – целочисленное значение i-го октета.

3. RSA вычисления $y = x^e \bmod n$.
4. Перевод целочисленного значения y в строку октетов. Полученная таким образом строка и будет результатом зашифрования сообщения D.

Расшифрование

Входными значениями для расшифрования является строка октетов ED и параметры алгоритма RSA (n, d). В случае, если длина D не равна k, считается, что произошла ошибка, и на практике данные обычно передаются заново. Процесс расшифрования состоит из следующих шагов (рис. 1.18):

1. Преобразование строки октетов в целочисленное значение. В случае, если y не удовлетворяет неравенству, $0 \leqslant y < n$, считается, что произошла ошибка;
2. RSA вычисления. Вычисляется значение $x = y^c \bmod n$, где c = e или d;
3. Перевод целочисленного значения x в строку октетов EB длины k;
4. Обратное форматирование строки EB. Строка октетов EB разбивается на составные части: BT, PS и D. В данном случае считается, что произошла ошибка, если:
   - строка EB не может быть однозначно «разобрана»;
   - строка PS содержит меньше 8 октетов либо ее содержимое не соответствует значению BT.
   - расшифрование происходит на открытом ключе и BT не равно 00 или 01 или процесс осуществляется на секретном ключе и BT не равно 02.

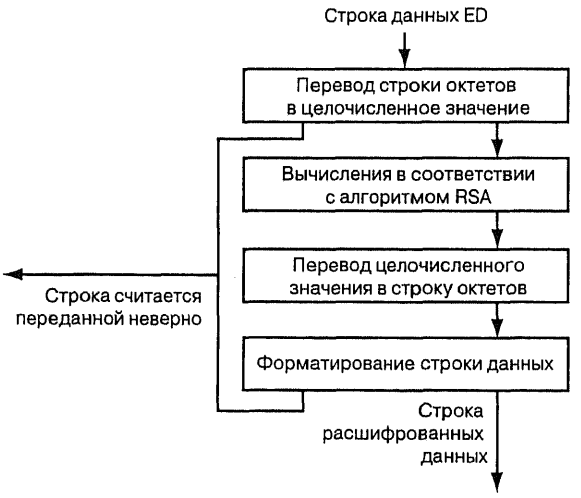

Рис. 1.18
Процесс расшифрования

Если проверки завершились успешно, расшифрованными данными является строка D.

1.4. Электронно-цифровая подпись

1.4.1. Общие положения

На протяжении многих веков при ведении деловой переписки, заключении контрактов и оформлении любых других важных бумаг подпись ответственного лица или исполнителя была непременным условием признания его статуса или неоспоримым свидетельством его важности. Подобный акт преследовал две цели:

- гарантирование истинности письма путем сличения подписи с имеющимся образцом;
- гарантирование авторства документа (с юридической точки зрения).

Выполнение данных требований основывается на следующих свойствах подписи:

- подпись аутентична, то есть с ее помощью получателю документа можно доказать, что она принадлежит подписывающему (на практике это определяется графологической экспертизой);
- подпись неподделываема, то есть служит доказательством, что только тот человек, чей автограф стоит на документе, мог подписать данный документ, и никто другой не смог бы этого сделать;
- подпись непереносима, то есть является частью документа и поэтому перенести ее на другой документ невозможно;
- документ с подписью является неизменяемым, то есть после подписания его невозможно изменить, оставив данный факт незамеченным;
- подпись неоспорима, то есть человек, подписавший документ, в случае признания экспертизой, что именно он засвидетельствовал данный документ, не может оспорить факт подписания;
- любое лицо, имеющее образец подписи, может удостовериться в том, что данный документ подписан владельцем подписи.

С переходом к безбумажным способам передачи и хранения данных, а также с развитием систем электронного перевода денежных средств, в основе которых – электронный аналог бумажного платежного поручения, проблема виртуального подтверждения аутентичности документа приобрела особую остроту. Развитие любых подобных систем теперь немыслимо без существования электронных подписей под электронными

документами. Однако применение и широкое распространение *электронно-цифровых подписей* (ЭЦП) повлекло целый ряд правовых проблем. Так, ЭЦП может применяться на основе договоренностей внутри какой-либо группы пользователей системы передачи данных, и в соответствии с договоренностью внутри данной группы ЭЦП должно иметь юридическую силу. Но будет ли электронная подпись иметь доказательную силу в суде, например при оспаривании факта передачи платежного поручения?

Хотя ЭЦП сохранила практически все основные свойства обычной подписи, все-таки некоторые особенности реализации электронного автографа делают ее отдельным классом подписей. Поэтому юридические, правовые и методологические аспекты применения ЭЦП должны учитывать ее специфику.

Существует несколько методов построения схем ЭЦП, а именно:

- шифрование электронного документа (ЭД) на основе симметричных алгоритмов. Данная схема предусматривает наличие в системе третьего лица (арбитра), пользующегося доверием участников обмена подписанными подобным образом электронными документами. Взаимодействие пользователей данной системой производится по следующей схеме:
  - участник А зашифровывает сообщение на своем секретном ключе $К_А$, знание которого разделено с арбитром, затем шифрованное сообщение передается арбитру с указанием адресата данного сообщения (информация, идентифицирующая адресата, передается также в зашифрованном виде);
  - арбитр расшифровывает полученное сообщение на ключе $К_А$, производит необходимые проверки и затем зашифровывает на секретном ключе участника В ($К_В$). Далее зашифрованное сообщение посылается участнику В вместе с информацией, что оно пришло от участника А;
  - участник В расшифровывает данное сообщение и убеждается в том, что отправителем является участник А.

  (Авторизацией документа в данной схеме будет считаться сам факт зашифрования ЭД секретным ключом и передача зашифрованного ЭД арбитру. Основным преимуществом этой схемы является наличие третьей стороны, исключающей какие-либо спорные вопросы между участниками информационного обмена, то есть в данном случае не требуется дополнительной системы арбитража ЭЦП. Недостатком схемы является наличие третьей стороны и использование симметричных алгоритмов шифрования. На практике эта схема не получила широкого распространения.);

- использование асимметричных алгоритмов шифрования. Фактом подписания документа в данной схеме является зашифрование документа на секретном ключе его отправителя. Эта схема тоже используется довольно редко вследствие того, что длина ЭД может оказаться критичной. Из предыдущего раздела стало ясно, что применение асимметричных алгоритмов для зашифрования сообщений большой длины неэффективно с точки зрения скоростных характеристик, что, например, для системы валовых расчетов является одним из основных показателей. В этом случае не требуется наличия третьей стороны, хотя она может выступать в роли сертификационного органа открытых ключей пользователей;
- развитием предыдущей идеи стала наиболее распространенная схема ЭЦП, а именно: зашифрование окончательного результата обработки ЭД хэш-функцией при помощи асимметричного алгоритма (см. раздел «Хэш-функции»). Структурная схема такого варианта построения ЭЦП представлена на рис. 1.19.

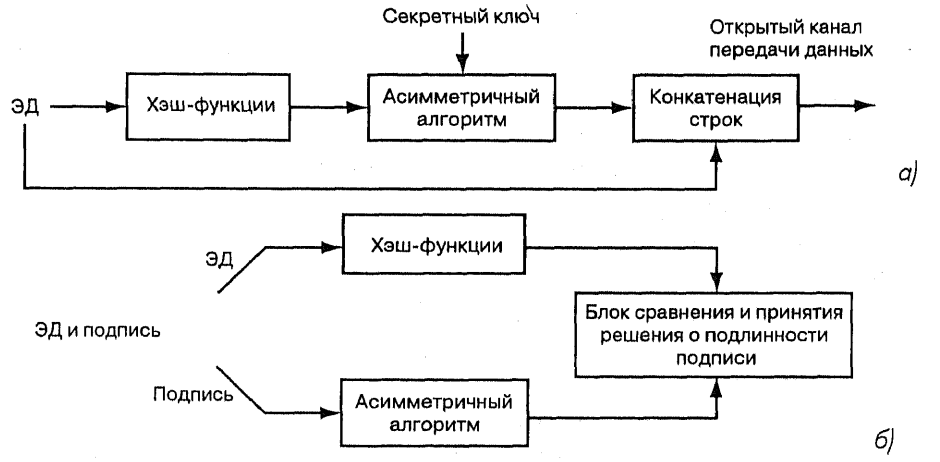

Рис. 1.19. ЭЦП

Генерация подписи происходит следующим образом:

1. Участник А вычисляет хэш-код от ЭД. Полученный хэш-код проходит процедуру преобразования с использованием своего секретного ключа. После чего полученное значение (которое и является ЭЦП) вместе с ЭД отправляется участнику В.
2. Участник В должен получить ЭД с ЭЦП и сертифицированный открытый ключ участника А, а затем произвести расшифрование на нем

ЭЦП, сам ЭД подвергается операции хэширования, после чего результаты сравниваются, и если они совпадают, то ЭЦП признается истинной, в противном случае ложной.

Эффективность реализации данной схемы по сравнению с предыдущей состоит в применении медленных процедур асимметричного шифрования к хэш-коду ЭД, который значительно короче самого ЭД.

Стойкость данного типа ЭЦП основана на стойкости асимметричных алгоритмов шифрования и применяемых хэш-функций.

Кроме вышеперечисленных существуют «экзотические» варианты построения схем ЭЦП (групповая подпись, неоспариваемая подпись, доверенная подпись и т.п.). Появление этих разновидностей обусловлено многообразием задач, решаемых с помощью электронных технологий передачи и обработки ЭД. Подробно такие виды ЭЦП рассмотрены в разделе 2.4.

В общем случае подписанный ЭД выглядит как пара, состоящая из бинарных строк (M, S), где M представляет собой ЭД, а S – решение уравнения $F_K(S) = M$, где F_K является функцией с секретом. В связи с вышеизложенным определением ЭЦП можно выделить следующие ее свойства:

- является неподделываемой, поскольку решить уравнение FK(S) = M может только обладатель секрета K;
- однозначно идентифицирует автора, то есть человека, подписавшего данный документ;
- верификация подписи (проверка) производится на основе знания функции FK;
- является непереносимой на другой ЭД; исключение составляет случай, когда для используемой хэш-функции найдены коллизии (см. раздел «Хэш-функции»);
- ЭД с ЭЦП может передаваться по открытым каналам, поскольку любое изменение ЭД приведет к тому, что процедура проверки ЭЦП обнаружит данный факт.

1.4.2. Атаки на ЭЦП

Широкое применение ЭЦП в различных областях электронных коммуникаций привело к тому, что в криптографии уже сложился обширный теоретический задел, посвященный созданию схем ЭЦП, их использованию, а также вопросам стойкости большинства схем ЭЦП и безопасности их применения.

Как уже отмечалось, стойкость большинства схем ЭЦП зависит от стойкости асимметричных алгоритмов шифрования и хэш-функций. Поэтому данный раздел мы посвятим описанию существующих на сегодняшний день атак и угроз на схемы ЭЦП.

Приведем классификацию атак на схемы ЭЦП:

- атака с известным открытым ключом. Она является, очевидно, самой слабой из всех перечисленных ниже, поскольку злоумышленник всегда может получить открытый ключ пользователя;
- атака с известными подписанными сообщениями. Противник кроме открытого ключа имеет еще и набор подписанных сообщений;
- простая атака с выбором подписанных сообщений. Противник имеет возможность выбирать подписанные сообщения, при этом открытый ключ он получает после выбора сообщений;
- направленная атака с выбором сообщений. Представляет собой вариант предыдущей и отличается тем, что, получая подписанные сообщения, противник знает открытый ключ;
- адаптивная атака с выбором сообщений. В данной атаке противник знает открытый ключ, может выбирать подписанные сообщения и, кроме того, имеет подписи всех ранее подписанных сообщений.

Каждая атака на схему ЭЦП преследует определенные цели (угрозы), которые можно разделить на следующие классы:

- *полное раскрытие*. Противник находит секретный ключ пользователя;
- *универсальная подделка*. Противник находит алгоритм, функционально эквивалентный алгоритму генерации ЭЦП;
- *селективная подделка*. Подделка подписи для сообщения, выбранного противником;
- *экзистенциальная подделка*. Подделка подписи хотя бы для одного случайно выбранного сообщения.

Оценка стойкости схемы ЭЦП производится относительно пары (атака-угроза), то есть стойкость ЭЦП можно определить как способность подписи противостоять достижению противником строго определенной цели при проведении атаки на схему ЭЦП.

На практике применение ЭЦП позволяет предотвратить либо выявить следующие действия нарушителя (он по отношению к системе может быть как *внутренним*, так и *внешним*):

- отказ одного из участников от факта отправления сообщения. Участник информационного обмена А заявляет, что он не посылал сообщение участнику В, хотя на самом деле посылал;

- модификация принятого ЭД. Участник В, приняв сообщение, изменяет его и утверждает, что именно данное сообщение он принял от участника А;
- подделка сообщения. Участник В создает сообщение и утверждает, что данное сообщение он принял от участника А, хотя на самом деле А ничего не передавал;
- навязывание сообщений в процессе передачи. Злоумышленник перехватывает обмен сообщениями между А и В и модифицирует их;
- имитация передачи сообщения. Злоумышленник пытается отправлять сообщения от имени одного из участников информационного обмена.

При этом следует учесть, что существуют нарушения, от которых невозможно оградить систему обмена сообщениями, а именно:

- повтор передачи сообщения. Злоумышленник или один из участников информационного обмена пытается снова передать ранее отправленное сообщение. Подобные нарушения особенно распространены в системах электронного перевода денежных средств;
- фальсификация времени отправления сообщения.

Противодействие данным нарушениям может основываться на использовании временных вставок и строгом учете входящих сообщений.

1.4.3. Алгоритм DSA

В 1991 году Национальный институт стандартизации и технологий (NIST) США опубликовал стандарт на ЭЦП (Digital Signature Standard, DSS), в основе которого – алгоритм DSA. Он является аналогом механизма, предложенного Эль Гамалем, но с некоторыми изменениями, в частности за счет уменьшения числового порядка одного из параметров схемы.

В данном стандарте подпись представляет собой два больших целых числа, полученных в соответствии с процедурами и параметрами, определенными в DSS.

Алгоритм DSA является «классическим» примером схемы ЭЦП на основе использования хэш-функции и асимметричного алгоритма шифрования.

Стойкость системы в целом основана на сложности нахождения дискретных логарифмов в конечных полях.

Генерация ЭЦП

Поскольку алгоритм Эль Гамаля является асимметричным алгоритмом шифрования, то параметры системы делятся на три группы:

- общие параметры;
- секретный ключ;
- открытый ключ.

Общие параметры необходимы для функционирования системы в целом. Секретный ключ используется для формирования ЭЦП, а открытый – для проверки ЭЦП. Общими параметрами системы являются простые целые числа p, q и g, удовлетворяющие следующим условиям:

- p: $2^{511} < p < 2^{512}$;
- q: простой делитель числа (p – 1), который удовлетворяет условию $2^{159} < q < 2^{160}$;
- g: так называемый генератор, удовлетворяющий равенству $g = h^{p-1/q} \bmod p$, где h – любое целое $0 < h < p$, при котором $h^{p-1/q} \bmod p > 1$.

Параметры p, q и g опубликовываются для всех пользователей системы обмена ЭД с ЭЦП.

Секретный ключ x случайно выбирается пользователем из диапазона [1, q] и держится в секрете.

Открытый ключ y вычисляется следующим образом: $y = g^x \bmod p$.

Также при описании данной схемы будут использоваться следующие обозначения и дополнительные параметры: m – входное сообщение пользователя для схемы ЭЦП; k – случайное число, удовлетворяющее условию $0 < k < q$, хранящееся в секрете и меняющееся от одной подписи к другой; H – хэш-функция; h – хэш-код сообщения.

Процесс генерации ЭЦП состоит из следующих этапов:

1. Вычисляется хэш-код сообщения m h = H(m).
2. Из диапазона [1, q] случайным образом выбирается значение k и вычисляется $r = (g^k \bmod p) \bmod q$.
3. Вычисляется $s = (k^{-1}(h + xr)) \bmod q$, где k^{-1} удовлетворяет условию $(k^{-1}k) \bmod q = 1$.

Значения r и s являются ЭЦП сообщения m и передаются вместе с ним по открытым каналам связи.

Проверка ЭЦП

Пусть принято сообщение m_1 и его подпись s_1 и r_1 при передаче (m, s, r). Проверка ЭЦП происходит следующим образом:

- проверяется выполнение условий $0 < r_1 < q$ и $0 < s_1 < q$, и если хотя бы одно из них нарушено, подпись отвергается;

- вычисляются значения:
  $w = s_i^{-1} \bmod q$
  $u1 = (H(m_1)w) \bmod q$
  $u2 = ((r_1/w) \bmod q$
  $v = ((g^{u1}y^{u2}) \bmod p) \bmod q$
- проверяется равенство $v = r_1$.

Если последнее равенство выполняется, то подпись принимается. В данном стандарте также специфицируется процедура генерации основных параметров системы и проводится доказательство того, что если $v = r_1$, то $m_1 = m$, $r_1 = r$ и $s_1 = s$.

Стойкость DSA

Криптографическая стойкость схемы DSA против атак методом «грубой силы» в первую очередь зависит от размера параметров p и q (в данном случае 512 и 160 бит). Соответственно криптостойкость против атаки методом «грубой силы» на параметр p будет равна 2^{160}. А успешная атака на параметр q возможна только в том случае, если злоумышленник может вычислять дискретные логарифмы в полях Галуа $GF(2^{512})$ с количеством предварительных вычислений пропорционально

$$L(p) = e^{\sqrt{\ln p \ln\ln p}}.$$

Одной из теоретически возможных атак на схему DSA является компрометация параметра k. При каждой подписи требуется новое значение k, которое должно быть выбрано случайным образом. Если злоумышленник найдет значение k, употреблявшееся при подписании сообщения (такое возможно, если будут обнаружены некоторые слабости в процедуре генерации k), секретный ключ x может быть воспроизведен. Другой возможный вариант – две подписи были сгенерированы на одном значении k. В этом случае злоумышленник тоже в состоянии восстановить x. Следовательно, одним из факторов, повышающих безопасность использования схем ЭЦП, является наличие «хорошего» генератора случайных чисел.

1.4.4. Стандарт на процедуры выработки и проверки ЭЦП

Отечественным стандартом на процедуры выработки и проверки ЭЦП является ГОСТ Р 34.10-94. Схема ЭЦП, предложенная в данном стандарте, во многом напоминает подпись в DSA, поэтому большая часть рассуждений о предыдущем алгоритме справедлива и для данной схемы ЭЦП.

Цифровая подпись представляет собой два больших целых числа. Общедоступные параметры схемы ЭЦП (p, q и a) должны удовлетворять следующим условиям:

- p: $2^{509} < p < 2^{512}$ или $2^{1020} < p < 2^{1024}$;
- q: простой делитель $(p - 1)$ и удовлетворяет неравенству $2^{254} < q < 2^{256}$;
- r: $1 < a < p - 1$ и $a^q \pmod p = 1$.

Секретный ключ пользователя x выбирается случайным образом и должен удовлетворять неравенству $0 < x < q$. Открытый ключ пользователя вычисляется в соответствии с равенством $y = a^x \pmod p$.

Генерация ЭЦП

Процедура генерации подписи сообщения m состоит из следующих шагов:

1. Вычисляется хэш-код сообщения m: $h = H(m)$ (хэш-функция, используемая в данном стандарте в соответствии с ГОСТ Р 34.10-94), если $h(m) \pmod p = 0$, то $h(m)$ присваивается значение $0...0_{255}1$.
2. Случайным образом выбирается значение k (аналогично DSA), удовлетворяющее условию $0 < k < q$.
3. Вычисляется значение $r = a^k \pmod p$ и $r_1 = r \pmod p$; если $r_1 = 0$, следует вернуться к предыдущему этапу и выработать другое значение k.
4. Вычислить значение $s = (xr_1 + kh(m)) \pmod p$; если $s = 0$, то необходимо выработать другое значение k. В противном случае подписью сообщения являются числа r_1 и s.

Проверка ЭЦП

Процедура проверки ЭЦП состоит из последовательности действий:

1. Проверяется выполнение условий $0 < s < q$ и $0 < r_1 < q$, и если хотя бы одно из них не выполняется, подпись отвергается.
2. Вычисляется хэш-код принятого сообщения $h = H(m)$; если $h(m) \times \times \pmod p = 0$, то битовое представление $h(m)$ равно $0...0_{255}1$.
3. Вычисляется значение $v = (h(m))^{q-2} \pmod p$.
4. Вычисляются значения: $z_1 = sv \pmod p$; $z_2 = (q - r_1)v \pmod p$.
5. Вычисляется значение $u = (a^{z1}y^{z2} \pmod p) \pmod q$.
6. Проверяется равенство $r_1 = u$; если оно выполняется, подпись принимается. То есть в этом случае можно считать, что сообщение подписано данным отправителем и в процессе передачи не была нарушена его целостность.

1.4.5. Практическое применение ЭЦП

Наличие широкого спектра достоинств у схем ЭЦП предопределило их широкое распространение в системах обмена электронными документами, а также во многих других случаях, где требуется обеспечить целостность и достоверность передачи информации.

Практическое использование схем ЭЦП и их разновидностей описано ниже. В этом разделе рассматривается вопрос практической реализации ЭЦП в соответствии со стандартом PKCS # 1. Стандарт криптографии с открытым ключом (PKCS) содержит требования для практического применения процедур зашифрования/расшифрования и генерации/проверки на основе использования асимметричного алгоритма RSA.

Схема ЭЦП, описанная в PKCS # 1, основана на применении хэш-функции (MD 2 или MD 5) к входному сообщению и зашифровании хэш-кода сообщения при помощи алгоритма RSA.

В табл. 1.9 представлены основные обозначения и их показатели, используемые в данном разделе. Следует отметить, что в соответствии с PKCS # 1 все битовые последовательности представляются в виде строки октетов (октет имеет размерность 8 бит), то есть если x – строка октетов, то x_i – i-й октет (отсчет начинается слева направо).

Таблица 1.9. Обозначения, используемые в разделе

k	Длина n в октетах (kV≥11)	EB	Блок данных для зашифрования
n	Модуль в RSA 28(k–1)≤n≤28k	ED	Зашифрованные данные
p и q	Целые числа такие, что n=pq	ab	Шестнадцатеричное значение октета
e и d	Экспоненты в алгоритме RSA	BT	Тип блока
M	Сообщение	PS	Дополнительные данные (набивка)
MD	Хэш-код сообщения	S	Подпись
MD1	Хэш-код, вычисляемый на этапе проверки	‖X‖	Длина X в октетах

Формат данных в соответствии с PKCS # 1

Здесь данные представляют собой строку октетов D, где $\|D\| \leqslant k - 11$. BT – строка октетов, шестнадцатеричное значение которой может принимать значение 00 или 01. PS – строка октетов, $\|PS\| = k - 3 - \|D\|$ – длина строки PS, которая зависит от значения BT. Если BT = 00, то все октеты в PS равны 00, если BT = 01, то все октеты в PS равны FF. Формат EB имеет следующий вид: 00‖BT‖PS‖00‖D. Первые 00 предназначены для того, чтобы значения строки EB были меньше модуля n.

Генерация ЭЦП

В данной процедуре (рис. 1.20) входным является сообщение m, а общедоступными параметрами системы – модуль n и секретный компонент d.

Процесс генерации включает следующие шаги:

1. Получение хэш-кода сообщения; результатом данного шага является строка октетов MD.
2. Из MD и значения, идентифицирующего используемую хэш-функцию в соответствии с рекомендациями ASN. 1 и основными правилами кодирования (BER encoding), получают строку октетов.
3. Строка D преобразуется в строку EB на основе процедуры, описанной выше.
4. Из строки октетов EB получают ее целочисленное значение. Обозначим строку EB как $EB_1 \| EB_2 \| \ldots \| EB_k$, где EB_i – i-й октет строки. Далее через EB_i' обозначим целочисленное значение октета EB_i, тогда целочисленное значение EB (обозначим его как m) будет равно $m = \sum_{i=1}^{k} 2^{8(k-i)} EB_i'$.
5. Зашифровывают m в соответствии с алгоритмом RSA, то есть $s = m^d \bmod n$;
6. Преобразуют целое значение s в строку октетов ED, таким образом, ED и есть цифровая подпись сообщения m.

Рис. 1.20. Генерация ЭЦП

Проверка ЭЦП

В данной процедуре (рис. 1.21) входными параметрами являются сообщение m, подпись сообщения s и параметры алгоритма RSA, а именно: модуль n и открытая компонента e.

Процесс проверки подписи включает следующие шаги:

1. Преобразование подписи s, представленной в виде строки октетов, в целочисленное значение (см. раздел «Генерация ЭЦП»), причем в этом случае происходят следующие проверки:
   - если длина строки в битах s некратна восьми, подпись s отвергается;
   - если s меньше n, подпись отвергается;

2. Расшифрование s в соответствии с алгоритмом RSA, то есть m = $= s^e \bmod n$.
3. Преобразование целочисленного значения m в строку октетов EB × × (‖EB‖ = k); см. раздел «Генерация ЭЦП».
4. Проверку формата строки EB на соответствие типу, заданному в BT, а также проверку строки PS. В ходе проверки возможны следующие случаи непризнания подписи s:
   - если формат EB не может быть однозначно распознан;
   - если значение BT отличается от 00 или 01;
   - если PS состоит из менее восьми октетов или формат PS несовместим с типом, указанным в BT.
5. Из строки D в соответствии с основными правилами кодирования и идентификатором используемой хэш-функции получают строку MD. В случае, если идентификатор не указывает на MD 2 или MD 5, подпись s отвергается.
6. Получение хэш-кода MD_1 из сообщения m с использованием хэш-функции, определенной в соответствии с идентификатором хэш-функции. Проверка равенства MD = MD_1; подпись s сообщения m принимается только в том случае, если равенство верно.

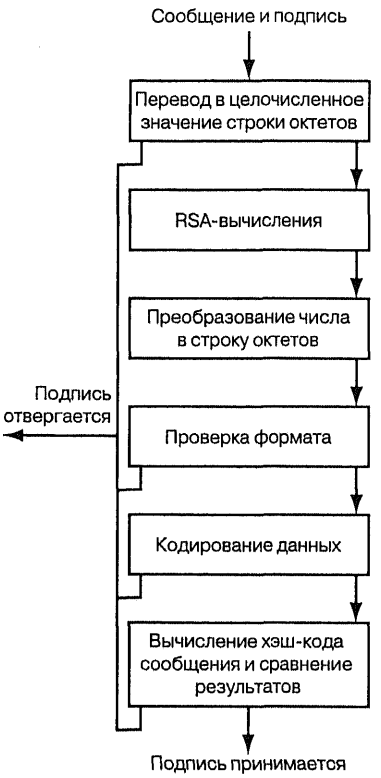

Рис. 1.21. Проверка ЭЦП

1.4.6. Арбитраж ЭЦП

Практическое применение ЭЦП, кроме процедур генерации и проверки электронной подписи, требует присутствия системы разбора конфликтных ситуаций (*арбитраж ЭЦП*). Любой алгоритм ЭЦП подобных процедур в своем описании не содержит. Это связано с тем, что построение схем арбитража связано с контекстом использования ЭЦП, так как его проведение – не столько техническая, сколько организационная задача.

В большинстве существующих систем использование арбитража ЭЦП основано на том, что подписать сообщение может только держатель секретного ключа (в этом случае ответственность за его компрометацию ложится на пользователя). Это утверждение нельзя считать верным в ситуации, когда $h(m_1) = h(m)$, то есть одна и та же подпись появляется под двумя разными сообщениями. Арбитр в подобном случае не сможет разрешить возникший спор, хотя очевидно, что кто-то из участников нашел коллизии для хэш-функции. Поэтому в большинстве случаев процедуры арбитража будут неполными, за исключением варианта, когда схемы ЭЦП разработаны специально с учетом проведения подобного разбирательства. Из существующих на сегодняшний день схем электронной подписи арбитраж эффективней всего может быть проведен для ЭЦП, построенных на основе симметричного шифрования при участии третьей стороны. При использовании схем ЭЦП, построенных без участия арбитра, следует с особой тщательностью выбирать процедуру подписания ЭД, чтобы арбитр в случае возникновения спорной ситуации мог решить, какая из сторон является правой.

Конкретная реализация процедур арбитража зависит прежде всего от потребностей пользователей, технической реализации ЭЦП и от обстоятельств, из-за которых возникла необходимость в проведении арбитража. Для примера рассмотрена процедура, возникшая при отказе пользователя от факта подписания документа m; при этом в качестве используемой ЭЦП берется алгоритм DSA.

Здесь исходными данными являются общедоступные параметры DSA – p, q, g; ЭД – m и подпись – r, s. ЭД и подпись представляются арбитру участником B, который хочет доказать, что данная подпись принадлежит участнику A; в свою очередь, участник A отказывается признать, что данная подпись была сгенерирована им.

Прежде всего арбитр требует от участника A предъявления секретного ключа. Это требование для любого пользователя ЭЦП в конкретной описываемой системе должно выполняться безоговорочно. Арбитр проверяет открытый ключ из общедоступного справочника на соответствие представленному секретному ключу. В случае несовпадения арбитр обращается в центр сертификации открытых ключей и требует предоставления заверенного участником A документа, содержащего открытый ключ. Если выясняется, что открытый ключ в общедоступном справочнике не совпадает с указанным в документе, виновным признается центр сертификации открытых ключей. Когда открытые ключи в справочнике и в документе

совпадают, это означает, что предъявлен некорректный секретный ключ, и участник А признается виновным.

В случае, если открытый и секретный ключи соответствуют ранее созданным образцам, арбитр производит следующие вычисления:

$W = s^{-1} \bmod q$
$u1 = (h(m)w) \bmod q$
$u2 = (rw) \bmod q$
$v = ((g^{u1}y^{u2}) \bmod p) \bmod q$

Завершающей фазой является проверка равенства v = r. Если оно выполняется, то подпись признается истинной, если нет – ложной.

1.5. Хэш-функции

1.5.1. Общие сведения

Во всем многообразии проблем обеспечения информационной безопасности, решаемых при помощи криптографических методов и средств, задача обеспечения целостности и достоверности передаваемой информации представляется на сегодняшний день одной из самых острых. С учетом современных требований к информационно-телекоммуникационным системам эта задача все чаще и чаще превращается в серьезную проблему. Особенно актуальна она в финансовой сфере, поскольку для надежного функционирования платежной системы необходимым условием является сохранение всеми документами целостности и достоверности.

Как уже говорилось ранее, неотъемлемой частью электронно-цифровой подписи является использование так называемых *хэш-функций*. Кроме того, они находят широкое применение и для решения ряда других вопросов, связанных с обеспечением защиты потоков данных, например для хэширования паролей пользователей с целью дальнейшего их шифрования и хранения в базе данных. Данный метод применяется в ОС Windows NT (используется хэш-функция MD 4 совместно с DES).

Одной из самых важных характеристик хэш-функций, обусловивших их широкое внедрение в практику, оказалась способность получать из открытого текста большой длины (например, в хэш-функции SHA максимальная длина открытого текста ограничена 2^{64} битами) хэш-кода гораздо меньшей длины (в отечественном стандарте ГОСТ Р 34.11-94 длина хэш-кода равна 256 битам, западные хэш-функции в основном имеют хэш-код

длиной 160–180 бит), что в некоторых случаях позволяет достаточно эффективно сократить сетевой трафик. Применение хэш-функций позволяет устранить избыточность открытого текста, что при дальнейшем криптографическом преобразовании хэш-кода открытого текста положительно сказывается на криптографических свойствах зашифрованного сообщения. Например, к хэш-коду открытого текста невозможно применить атаку методом протяжки вероятного слова.

Таким образом, хэш-функции являются необходимым элементом при применении ряда криптографических алгоритмов и протоколов. Под хэш-функциями понимаются функции, отображающие последовательность произвольной длины в значение фиксированной длины, называемой *хэш-кодом* или *выжимкой*.

1.5.2. Типы хэш-функций

Существует три типа построения хэш-функций:

- на основе какой-либо трудновычисляемой математической задачи;
- на основе алгоритмов блочного шифрования;
- разработанные с нуля.

Каждый из вышеперечисленных методов имеет свои достоинства и недостатки, однако наиболее распространенными на сегодняшний день оказались последние два. Это связано с тем, что при построении хэш-функций с нуля появляется возможность учитывать такое их свойство, как эффективная программная реализация. Широкое применение хэш-функций, построенных на основе алгоритмов блочного шифрования, является результатом тщательной проработки вопроса стойкости многих из существующих алгоритмов.

В данном разделе приведены два примера практической реализации хэш-функций (SHA, построенная с нуля, и ГОСТ Р 34.11-94 на основе блочного алгоритма шифрования ГОСТ 28147-89). Вопросы криптографической стойкости хэш-функций рассмотрены в разделе 1.5.3.

Стандарт Security Hash Algorithm

Security Hash Algorithm (SHA) разработан в NIST совместно с NSA для использования со стандартом на цифровую подпись DSS. Этот алгоритм предназначен для работы с входными последовательностями длиной $< 2^{64}$ бит и имеет хэш-код длиной 160 бит. Принципиальную основу SHA составляет алгоритм MD 4, созданный Ривестом.

Работа алгоритма начинается с того, что входная последовательность делится на блоки по 512 бит. Перед тем как разбить ее, необходимо, чтобы длины полученных блоков в битовом выражении были равны 512 битам. Для этого к данной последовательности приписываются единица и необходимое количество нулей, чтобы ее длина стала на 64 бита меньше числа, кратного 512. Затем к последовательности приписывается 64-битное представление длины входной последовательности.

Далее инициализируются пять переменных по 32 бита:

A = 0x67452301 B = 0xtfcdab89
C = 0x98badcfe D = 0x10325476
E = 0xc3d2e1f0

Основной цикл, совершаемый над одним 512-битным блоком, состоит из четырех подциклов, в каждом из которых используются по 20 операторов. Перед тем как начать преобразования, создаются копии перечисленных выше пяти переменных a, b, c, d и e.

Каждый оператор представляет собой набор нелинейных функций от трех переменных (B, C и D) и операций циклического сдвига и суммирования. Эти функции имеют следующий вид:

1 раунд $f_1(X, Y, Z) = (X \wedge Y) \vee ((\neg X) \wedge Z)$
2 раунд $f_2(X, Y, Z) = X \oplus Y \oplus Z$
3 раунд $f_3(X, Y, Z) = (X \wedge Y) \vee (X \wedge Z) \vee (Y \wedge Z)$
4 раунд $f_4(X, Y, Z) = X \oplus Y \oplus Z$

Для каждого раунда определяется одна константа:

$K_1 = 0x5a827999 = 2^{1/2}/4$ $K_2 = 0x6ed9eba1 = 3^{1/2}/4$
$K_3 = 0x8f1bbcdc = 5^{1/2}/4$ $K_4 = 0xca62c1d6 = 10^{1/2}/4$

Каждый обрабатываемый блок (512 бит) разбивается на 16 подблоков по 32 бита в каждом ($M_0 \div M_{15}$), затем к нему добавляется до 80 подблоков длиной 32 бита ($W_{16} \div W_{79}$) согласно следующему правилу:

$W_t = M_t$ для $t = 0 \div 15$ (t является номером оператора)
$W_t = (W_{t-3} \oplus W_{t-8} \oplus W_{t-14} \oplus W_{t-16}) << 1$ для $t = 16 \div 79$

Отметим, что первоначально при формировании подблоков W в SHA не было левого циклического сдвига. Результатом его введения явилось устранение одной из уязвимостей оригинального стандарта.

Каждый основной оператор имеет вид:

FOR t = 0 to 70
{ TEMP = (a << 5) + f(b, c, d) + e + W_t + K_i

e = d
d = c
c = b << 30
a = TEMP }, где i – 1 ÷ 4

Завершается цикл суммированием A + a, B + b, C + c, D + d, E + e и конкатенацией A, B, C, D и E.

ГОСТ Р 34.11-94

Этот стандарт разработан для использования совместно со стандартом на цифровую подпись (ГОСТ Р 34.10-94). Основу данной хэш-функции составляет алгоритм блочного шифрования ГОСТ 28147-89, который оперирует с блоками длиной 64 бита и длиной ключа 256 бит, соответственно длина выходной последовательности хэш-функции составляет 256 бит.

Алгоритм вычисления хэш-функции работает с любой двоичной последовательностью, то есть не накладывает ограничения на длину входной последовательности. Описать работу алгоритма можно следующим образом:

$H_i = h(M, H_{i-1})$,

где H_{i-1} – значение хэш-кода предыдущей итерации;
M – входная последовательность, не подвергшаяся обработке.

Процедура вычисления функции h состоит из последовательности шагов:

- **шаг 1**:
  - инициализируются переменные L (текущее значение длины обработанной части входной последовательности) и I (значение контрольной суммы);
  - если длина входной необработанной последовательности больше 256 ($|M| > 256$), алгоритм переходит к **шагу 3**, в противном случае производятся следующие действия:
- **шаг 2**:
  - $L = (L + |M|) \bmod 256$;
  - $I = (I + (0^{256-|M|} \| M) \bmod 256$ $0^{256-|M|}$ – последовательность битовых нулей длиной 256 – |M|, || – конкатенация битовых строк. Таким образом, на этом этапе вычисляется текущее значение контрольной суммы;
  - $H = \chi((0^{256-|M|} \| M), H)$ – вычисляется значение функции хэширования χ от аргументов, представляющих собой хэшируемый блок и начальный вектор хэширования (H);
  - $H = \chi(L, H)$;

– $H = \chi(I, H)$ – результат данного вычисления и является окончательным результатом хэш-функции;
– конец работы алгоритма.

Из входной последовательности выбирается очередной блок длиной 256 бит ($M = M0 \| M1$) и производятся следующие действия:

• **шаг 3**:

– $H = \chi(M0, H)$;
– $L = (L + 256) \bmod 256$;
– $I = (I + M_0) \bmod 256$;
– $M = M_0$;
– переход к **шагу 2**.

Вычисление функции х состоит из следующих этапов:

• генерация четырех секретных ключей длиной 256 бит;
• зашифрование четырех подблоков (составляющие начальный вектор хэширования H) по 64 бита каждый в соответствии с алгоритмом ГОСТ 28147-89 в режиме простой замены;
• применение перемешивающей функции к результату зашифрования.

Генерация секретных ключей

При генерации секретных ключей используются данные H и M (входная последовательность, представленная в двоичном виде) и инициализируются следующие константы: $C_2 = C_4 = 0^{256}$ и $C_3 = 1^8 0^8 1^{16} 0^{24} 1^{16} 0^8 (0^8 1^8)^2 1^8 0^8 \times \times (0^8 1^8)^4 (1^8 0^8)^4$, где a^k – двоичная последовательность из к бинарных знаков а – {0,1}. После чего выполняется алгоритм:

$I = 1$ $U = H$ $V = M$
$W = U \oplus V$
$K_1 = P(W)$
FOR i = 2 to 4
{$U = A(U) \oplus C_i$;
$V = A(A(V))$
$W = U \oplus V$
$K_i = P(W)$}

Функция P представляет собой перестановку $s(i + 1 + 4(k - 1)) = 8i + k$, $i = 0 \div 3$, $k = 1 \div 8$ над подблоками длиной 8 бит исходной двоичной последовательности длиной 256 бит. Таким образом, последовательность $x_{32} \| ... \| x_1$ преобразуется в $x_{s(32)} \| ... \| x_{s(1)}$. Функцией A(X) обозначают следующее

преобразование последовательности, разделенной на 4 подблока по 64 бита каждый: $A(X) = (x_1 \oplus x_2) \| x_4 \| x_3 \| x_2$.

Шифрующее преобразование

Начальный вектор H разбивается на 4 подблока h_i длиной 64 бита, и каждый подблок зашифровывается на своем ключе K_i, то есть $s_i = Ek_i(h_i)$ (i=1,2,3,4). В результате получается последовательность $S = s_1 \| s_2 \| s_3 \| s_4$.

Перемешивающая функция

Сначала исходная последовательность разбивается на шестнадцать 16-битных подблоков n_i $i = 1 \div 16$ и при помощи регистра сдвига с обратной связью преобразуется в последовательность следующего вида:

$$n_1 \oplus n_2 \oplus n_3 \oplus n_4 \oplus n_{13} \oplus n_{16} \| n_{16} \| \ldots \| n_2.$$

Если обозначить данную функцию через φ, то исходная функция хэширования – $\chi(M,H) = \varphi^{61}((H \oplus \varphi(M \oplus \varphi^{12}(S)))$, где степень функции φ обозначает, сколько раз она применяется к битовой последовательности.

Криптографическая стойкость данной хэш-функции основана на стойкости применяемого в ней алгоритма блочного шифрования, используемого в режиме простой замены. На момент подготовки книги к изданию автору были неизвестны алгоритмы «взлома» данной хэш-функции, работающие эффективнее, нежели метод, основанный на «парадоксе дня рождения».

1.5.3. Требования к хэш-функциям

При практическом использовании хэш-функций должны выполняться следующие требования:

- алгоритм должен обладать высокой скоростью обработки информации (это особенно актуально для банковских приложений, где требуется особая оперативность обработки информации);
- хэш-функция должна быть стойкой против атаки методом «грубой силы»;
- программная реализация хэш-функции должна быть оптимизирована под использование на современной аппаратно-программной базе.

Этим требованиям должен удовлетворять как сам алгоритм выработки хэш-значения, так и хэширующая функция.

В современных условиях алгоритмическое повышение скорости выработки хэш-значения может быть достигнуто за счет применения простого

преобразования, которое переводит одно сообщение в другое посредством элементарной операции, например удаления произвольного блока сообщения. Подобными преобразованиями можно также описать зависимость между двумя практически не отличающимися друг от друга сообщениями. Данный тип сообщения очень часто встречается в банковском деле, например с целью заполнения бланков платежных поручений. Отсюда следует, что для увеличения скорости обработки необходимо, чтобы алгоритм выработки хэш-значения включал в себя также алгоритм вычисления хэш-значения одного сообщения из хэш-значения другого сообщения, которое получается из начального с помощью элементарного преобразования.

1.5.4. Стойкость хэш-функций

С точки зрения криптографической стойкости важным свойством хэш-функций является отсутствие коллизий, то есть невозможность найти такие значения $x \neq y$, чтобы $h(x) = h(y)$. В криптографических приложениях важным понятием является *криптографически стойкая хэш-функция*, для которой не существует эффективного алгоритма нахождения значений $x \neq y$, где выполнялось бы условие $h(x) = h(y)$ (функция, стойкая в сильном смысле), или не существует эффективного алгоритма нахождения коллизии при заданном x такого $y \neq x$, что $h(x) = h(y)$ (функция, стойкая в слабом смысле). Росс Андерсон показал, что отсутствие коллизий не позволяет судить о практической стойкости хэш-функций. Другими словами, данное требование носит формальный характер. Практически значимым является отсутствие у хэш-функции корреляции. Свободной от корреляции называется хэш-функция, у которой невозможно найти пары таких значений $x \neq y$, что вес Хэмминга двоичного вектора $h(x)$ xor $h(y)$ будет меньше веса Хэмминга применительно к двоичному вектору $h(M)$ для некоторого малого M. Свобода от корреляции с точки зрения криптографической стойкости является гораздо более сильным свойством хэш-функции, чем свобода от коллизий. Данный факт подтверждается тем, что из любой хэш-функции, являющейся свободной от коллизий и одновременно свободной от корреляций, можно построить другую хэш-функцию, которая тоже будет свободной от коллизий, но при этом может не сохранить свойство свободы от корреляции.

Пример

Возьмем функцию h, обладающую двумя вышеперечисленными свойствами, зафиксируем небольшое по сравнению с длиной входной последовательности значение k. Разобьем входную последовательность S на S_1,

состоящую из первых k бит, и на S_2 – оставшуюся часть входной последовательности ($S = S_1 \| S_2$).

Теперь определим функцию h_1 как $h_1(S) = S_1 \| h(S_2)$. Таким образом, в выходной последовательности первые k бит останутся без изменений. Очевидно, что h_1 свободна от коллизии, поскольку таковой является функция h. Однако h_1 не унаследовала свойство свободы от корреляции, так как можно легко построить X и Y, при которых значение h(X) отличается от h(Y) на k бит.

Необходимо отметить, что основным способом поиска коллизий у хэш-функций является метод, основанный на «парадоксе дня рождения». Суть его заключается в генерации двух наборов по $2^{n/2}$ сообщения в каждом. Согласно парадоксу дня рождения вероятность того, что в этих наборах найдется пара сообщений, имеющих одинаковые хэш-значения, больше 1/2. Из недостатков метода следует отметить значительные временные затраты для генерации данных наборов сообщений, необходимость иметь большой объем памяти для их хранения и наличие быстрых алгоритмов сортировки. Этот метод в случае короткой длины хэш-значения является эффективным как для хэш-функций, построенных с нуля, так и для хэш-функций, построенных на основе известных алгоритмов блочного шифрования. Хотя к хэш-функциям, построенным на основе алгоритмов блочного шифрования, применимы методы криптоанализа алгоритмов блочного шифрования.

1.6. Ключевая информация

1.6.1. Общие сведения

При создании и эксплуатации систем криптографической защиты потока данных приходится сталкиваться с проблемой создания, хранения и распределения ключей или исходной ключевой информации, на основе которой в дальнейшем создаются непосредственно сами ключи. Повышенное внимание к решениям подобных проблем вызвано тем, что злоумышленнику бывает гораздо проще провести атаку на ключевую систему, нежели на криптографические алгоритмы. По своему назначению ключи бывают нескольких типов (в соответствии со стандартом ANSI X9.17):

- для зашифрования ключей;
- для зашифрования данных.

Кроме того, при использовании криптографических средств защиты межмашинного взаимодействия приходится создавать так называемые сеансовые ключи, которые применяются только в рамках одного сеанса связи. В зависимости от того, к какой категории принадлежит ключ, будут меняться и пути решения вышеперечисленных проблем.

1.6.2. Генерация ключевой информации

При использовании криптографических алгоритмов одной из самых сложных задач является генерация ключей. В этом случае основной проблемой оказывается поддержание определенных криптографических свойств создаваемых ключей. Существуют два метода генерации ключей – детерминированный и недетерминированный.

Детерминированные методы

В основе этих методов лежит формирование из случайной последовательности малой длины псевдослучайной последовательности большей длины, которая не отличалась бы по своим статистическим свойствам от первоначальной. Одним из самых распространенных методов формирования псевдослучайных ключевых последовательностей является использование сдвиговых регистров с линейными обратными связями. Их функционирование описывается линейными рекуррентными последовательностями, применение которых в качестве генераторов псевдослучайных ключевых последовательностей не всегда является допустимым. В связи с этим широкое распространение получили процессы, которые носят псевдослучайный характер и имеют физическую природу (движение мыши, время реакции пользователя на работу с устройствами ввода/вывода и т.д.).

Существует ряд математических критериев, оценивающих, насколько распределение полученной последовательности близко к полиномиальному равновероятному распределению. Выбор критериев зависит прежде всего от критических параметров, отклонение которых от заданной величины может повлечь за собой угрозу криптографической стойкости. В любом случае при генерации ключей желательно ориентироваться на следующие критерии:

- проверку частот появления последовательности из символов k (k-грамма) по критерию x-квадрат;
- проверку частот исходов по обобщенному критерию x-квадрат;
- проверку максимального и минимального значения маркировки;
- проверку длины интервалов непопаданий в заданный диапазон;
- проверку на монотонность.

Недетерминированные методы

В основе данных методов – использование случайных физических процессов, исходы которых могут служить для дальнейшего изготовления ключей. Простейшим примером получения случайных исходов является подбрасывание игральных костей или монеты. Теоретически они вполне могли бы послужить генераторами ключевой последовательности, хотя на практике их использование неприемлемо ввиду низкой производительности.

В настоящее время широко применяются физические генераторы шума, выходные последовательности которых являются случайно распределенными (например, шумящие диоды, импульсные генераторы, счетчики Гейгера и т.д.). Снятые с этих приборов сигналы оцифровываются и представляются в виде двоичных последовательностей для дальнейшего побитового сложения в потоковых шифраторах или служат исходной последовательностью для формирования ключей.

Недетерминированные генераторы ключевой последовательности предотвращают наблюдение за работой генератора или вмешательство в нее, обеспечивают стабильность выхода генератора и контроль за смещением криптографически важных характеристик. Проверка физических генераторов шума на случайность выходного сигнала осуществляется с помощью различных статистических критериев, причем это должно происходить постоянно с учетом недетерминированного характера источника.

Приведенная выше классификация относится к генерации ключей для симметричных алгоритмов шифрования. Проблема генерации ключей для асимметричных алгоритмов связана с получением больших простых чисел и проверкой их на простоту.

Необходимо добавить, что некоторые типы алгоритмов имеют так называемые слабые ключи, использование которых значительно уменьшает криптографическую стойкость зашифрования на данных ключах. Для алгоритма DES, например, с длиной ключа 56 бит существует 16 слабых ключей. Проверка слабых ключей может производиться как экспериментальным способом, так и посредством анализа используемого алгоритма шифрования.

Генерация сеансовых ключей

Существует огромное количество методов генерации сеансовых ключей. В этом подразделе мы рассмотрим метод генерации сеансовых ключей на базе стандарта ANSI X9.17. Подобная операция производится на основе

секретного ключа пользователя R и секретного начального заполнения V_0 длиной 64 бита. В качестве алгоритма генерации рекомендуется использовать алгоритм шифрования DES. Для вычисления случайного ключа R_i производятся следующие действия:

R_i = Er (Er (Ti) Vi)
V_{i+1} = Er (Er (Ti) ⊕ Ri), гдеTi – метка времени

В результате получаем 64-битный сеансовый ключ Ri. Если необходимо получить 128-битный ключ, вышеуказанным способом создаются два ключа, и посредством операции конкатенации получается 128-битный ключ.

Генерация ключей на основе пароля пользователя

Одним из самых распространенных способов создания сеансовых ключей является их генерация на основе пароля пользователя. Пароль подвергается криптографическому преобразованию, в результате чего получается ключ, который может быть использован для дальнейшего зашифрования, например сетевого трафика пользователя.

К достоинствам этого метода следует отнести отсутствие необходимости хранить секретные ключи, пользователю нужно лишь помнить свой пароль, на основе которого в дальнейшем будет создан его секретный ключ.

Недостатком данного метода является возможность проведения противником атаки со словарем. Суть заключается в том, что можно угадать пароль пользователя путем перебора наиболее вероятных слов, поскольку при задании пароля для удобства запоминания, скорее всего, выбирается легко запоминающийся набор символов. Подробно данный метод описан в стандарте PKCS № 5.

1.6.3. Хранение ключей

При создании надежной системы криптографической защиты информации одной из первоочередных задач является обеспечение надежного хранения ключей пользователя на его рабочем месте. Злоумышленнику гораздо легче получить несанкционированный доступ к ключам путем подкупа сотрудников или с использованием специальных программных средств, таких как вирусы и программы прямого доступа к памяти, чем осуществлять дорогостоящие, требующие больших вычислительных и временных затрат атаки на криптографический алгоритм. Приведем некоторые основные способы организации надежного хранения ключей на рабочем месте пользователя:

- работа с криптоустройствами, имеющими защищенную от несанкционированного доступа память для хранения ключей;
- хранение ключей в зашифрованном виде непосредственно на персональном компьютере пользователя;
- использование внешних устройств для хранения ключей.

Первый способ, хотя и является самым надежным, но не всегда у пользователя есть возможность работать с такими устройствами. Кроме того, как правило, память криптоустройств имеет ограниченный объем, поэтому на практике в них хранятся ключи для зашифрования или генерации сеансовых ключей или ключей для файлов (директорий).

Второй способ не нуждается в особых комментариях и с точки зрения практической реализации не представляет особого интереса. Необходимо лишь отметить, что в большинстве реализованных систем криптографической защиты таким образом хранятся лишь ключи для зашифрования файлов (директорий).

Третий способ на сегодняшний день считается одним из самых перспективных. Дело в том, что в последнее время появилось большое количество устройств, имеющих возможность долгосрочного хранения ключевой информации. Защита таких устройств легко решается организационными методами, а возможность их совмещения с современными средствами вычислительной техники расширяет диапазон применения, поэтому данный метод является предпочтительным для практической реализации. Среди устройств, в которых хранится ключевая информация, следует выделить магнитные карты, гибкие магнитные диски, микросхемы ППЗУ, электронные пластиковые ключи и электронные карты (smart-cards). Из вышеперечисленных наиболее перспективными являются смарт-карты, поскольку одновременно с ключевым носителем они могут использоваться в качестве технического средства реализации криптосхем с программируемой логикой.

1.6.4. Распределение ключей

Для обеспечения надежного функционирования систем криптографической защиты необходимо наличие надежной и устойчивой системы распределения ключевой информации. До появления криптографии с открытым ключом, а также территориально-распределенных сетей передачи данных, охватывающих миллионы пользователей, распределение ключей осуществлялось в основном через надежные каналы связи, например фельдъегерскую почту, путем их передачи в зашифрованном виде по коммуникационным каналам и т.д.

Увеличение числа пользователей засекреченной связи, повсеместное применение средств вычислительной техники, а также повышение требований к безопасности, оперативности и работоспособности сети связи определило два направления развития систем распределения ключевой информации:

- создание автоматизированных комплексов распределения ключей в зашифрованном виде через открытые каналы передачи данных и автоматизированный ввод ключей в шифротехнику. Это направление предполагает наличие территориальных центров создания и рассылки ключевой информации;
- использование асимметричных алгоритмов и криптографических протоколов для безопасного и оперативного распределения ключевой информации. Такой подход получил широкое распространение благодаря появлению глобальных сетей передачи данных (Internet, intranet и т.д.). Поскольку в этом случае задача распределения ключевой информации в основном решается с помощью использования криптографических протоколов, более подробно о данном направлении будет сказано ниже.

1.6.5. Минимальная длина ключа

Какова же минимальная длина ключа для обеспечения соответствующего уровня безопасности? Найти ответ на этот острый и популярный вопрос без привязки к конкретной ситуации достаточно трудно. Можно дать только некоторые рекомендации по выбору необходимой длины ключа.

Интуитивно понятно, чем большую длину имеет ключ, тем больше времени потребуется злоумышленнику для проведения атаки методом тотального опробования каждого возможного ключа. Так, добавление одного битового разряда приведет к увеличению количества возможных ключей в два раза, но работа с достаточно длинными ключами существенно увеличивает время проведения операций зашифрования/расшифрования, что может отрицательно сказаться на работе приложений, требующих оперативного доступа к защищаемой информации. Поэтому при выборе ключа для улучшения скоростных характеристик функционирования системы и обеспечения должного уровня безопасности приходится искать компромиссные решения. Необходимо также учитывать время жизни защищаемой информации и возможности (временные, стоимостные, вычислительные

и профессиональные) предполагаемого противника, от которого защищается информация.

Современная микропроцессорная и вычислительная техника позволила уже сегодня за достаточно приемлемое время находить 40-битные ключи методом тотального опробования. Кроме того, на рынок поступили FPGA-чипы стоимостью 200 долларов, обладающие возможностью перебирать до 30 миллионов ключей в секунду, и разработаны ASIC-чипы со скоростью 200 миллионов ключей в секунду и стоимостью 10 долларов.

Поэтому, выбирая необходимую длину ключа, вам сначала нужно учесть такие факторы, как стоимость информации (хотя не всегда есть возможность оценить ее в денежном эквиваленте), возможности потенциального противника и время жизни информации (см. табл. 1.10).

Таблица 1.10. Сравнительный анализ времени и средств, затрачиваемых различными группами злоумышленников, на нахождение ключей методом тотального опробования

Тип атакующего	Бюджет	Средства	Время и средства, затрачиваемые на ключ		Необходимая длина ключа
			40 бит	56 бит	
Хакеры	Небольшой	Средства вычислительной техники	1 неделя	Неосуществимо	45
	$400	FPGA	5 часов ($0,08)	38 лет ($5000)	50
Небольшие фирмы	$10000	FPGA	12 минут ($0,08)	18 месяцев ($5000)	55
Корпоративные департаменты	$300000	FPGA	24 секунды ($0,08)	19 дней ($5000)	60
		ASIC	18 секунд ($0,001)	3 дня ($38)	
Большие компании	$10000000	FPGA	7 секунд ($0,08)	13 часов ($5000)	70
		ASIC	0,005 секунды ($0,001)	6 минут ($38)	

Все вышесказанное относится к симметричным алгоритмам. В случае использования асимметричных алгоритмов шифрования необходимо иметь в виду, что при одной и той же длине ключа атака методом тотального опробования на симметричные алгоритмы шифрования потребует гораздо больших временных затрат, нежели разложение модуля той же длины на составляющие в алгоритмах асимметричного шифрования,

построенных на модульном возведении в степень. Поэтому при обеспечении одинакового уровня безопасности асимметричный алгоритм всегда будет иметь большую длину ключа по сравнению с симметричным алгоритмом (табл. 1.11).

Таблица 1.11. Соответствие длины ключей асимметричного и симметричного алгоритмов шифрования при обеспечении одинакового уровня безопасности против атаки методом тотального опробования

Длина ключа для симметричного алгоритма	Длина ключа для асимметричного алгоритма
56 бит	384 бита
64 бита	512 бит
80 бит	768 бит
112 бит	1792 бита
128 бит	2304 бита

Глава II

Теоретические аспекты создания и применения криптографических протоколов

2.1. Общие сведения

2.1.1. Область применения

Одним из важнейших средств решения задач информационной безопасности в сетях передачи данных являются *криптографические протоколы* (далее – *криптопротоколы*). Их применение обусловлено использованием обширных механизмов межсетевого взаимодействия, причем под межсетевым взаимодействием следует понимать обмен информацией как на сетевом уровне модели взаимодействия открытых систем, так и на вышележащих уровнях.

В общем случае под криптопротоколом будем понимать распределенный алгоритм, реализованный при помощи последовательности действий, позволяющий двум или более участникам информационного обмена решать определенные задачи.

При этом под безопасным будем понимать криптопротокол, в котором участники достигают своей цели, а злоумышленник – нет.

Большинство криптопротоколов в своей основе используют криптографические алгоритмы (блочного шифрования, ЭЦП и хэш-функции), хотя это утверждение не является обязательным – вместо криптографических алгоритмов могут применяться необратимые преобразования, такие как модульное возведение в степень.

Использование криптографических алгоритмов в криптопротоколах (причем в некоторых протоколах используются несколько различных

алгоритмов) приводит к необходимости решить задачу согласования используемых алгоритмов и их параметров между сторонами информационного обмена.

Многообразие механизмов межсетевого взаимодействия, в свою очередь, способствует появлению различных криптопротоколов. Причем они могут решать задачи информационной безопасности как в виде отдельных механизмов (SSL, SHTTP и др.), так и входить в состав других продуктов, связанных с этой областью (например, в TrustedWeb используется Kerberos). Типичным примером использования криптопротоколов может являться решение такой распространенной задачи: клиент (например, HTTP-клиент) хочет получить доступ к серверу (например, к Web-серверу) через открытые сети передачи данных и установить с ним защищенный канал передачи данных. Данная проблема эффективно разрешима только с применением криптопротоколов.

В свою очередь, средства обеспечения информационной безопасности в сетях передачи данных тоже требуют решения ряда специфических задач, поддерживающих их надежное функционирование, что также расширяет сферу применения криптопротоколов. Типичным примером их использования является обеспечение ключевого обмена и согласование параметров безопасности (тип алгоритма, режим применения и т.д.).

Основными задачами сегодняшнего дня, которые решаются криптопротоколами в сетях передачи данных, являются:

- аутентификация и идентификация (см. раздел 2.2);
- ключевой обмен (см. раздел 2.3).

Существует также целый ряд криптопротоколов, предназначенных для решения более специфических задач (см. раздел 2.4).

Следует иметь в виду, что один и тот же криптопротокол может применяться в самых разных областях. Например, криптопротокол Kerberos в ходе своей работы позволяет произвести аутентификацию пользователей и осуществить ключевой обмен между участниками.

В качестве примера можно привести достаточно простой криптопротокол ключевого обмена. Его участники, A и B, хотят выработать общий секретный ключ для симметричного алгоритма шифрования, используя открытые каналы передачи данных. Для этого они используют следующую последовательность действий:

1. Участник B выбирает схему асимметричного шифрования, создает пару ключей для данной схемы – открытый и секретный.
2. Участник B посылает свой открытый ключ участнику A.

3. Участник А создает секретный ключ для симметричного алгоритма шифрования; зашифровывает его на открытом ключе участника В и отсылает ему полученный результат.
4. Участник В, получив зашифрованный секретный ключ от участника А, расшифровывает его на своем секретном ключе для асимметричного алгоритма.

Теперь А и В, используя симметричный алгоритм шифрования, могут обмениваться зашифрованной информацией по открытым каналам связи. Приведенный выше криптопротокол не является безопасным, но хорошо отражает саму идею построения подобных средств защиты информации.

2.1.2. Вопросы безопасности криптопротоколов

Как и в случае использования криптографических алгоритмов, главный вопрос, который задают заинтересованные пользователи, заключается в том, насколько стойким является тот или иной криптопротокол. Ответ на него можно найти при сравнении целого ряда факторов, которые мы рассмотрим ниже. Однако, поскольку в основе многих криптопротоколов лежат именно криптографические алгоритмы, очевидно, что окончательная стойкость будет не больше стойкости используемых криптографических алгоритмов. Она может быть существенно снижена в следующих случаях:

- использование слабых криптографических алгоритмов и некорректная реализация некоторых ее составляющих;
- некорректная логика работы криптопротокола;
- некорректное использование криптографических алгоритмов.

Трудности первой категории решаются в рамках классической криптографии. Типичным примером в этом смысле является использование слабых генераторов случайных чисел.

Проблемы, относящиеся ко второй категории, наиболее распространены. На практике именно по этим «болевым точкам» обычно проводятся атаки на криптопротоколы. Активный нарушитель может оказывать влияние на функционирование криптопротокола путем следующих атак:

- *атака с известным ключом.* Злоумышленник, получив ключ предыдущей сессии, пытается узнать ключ новой сессии;
- *повторная передача.* Перехватив в предыдущих сессиях определенную порцию информации, злоумышленник передает ее в последующих сессиях;

- *подмена стороны информационного обмена.* В процессе установления сеанса связи между легальными пользователями злоумышленник в случае подобной атаки пытается выдать себя за одного из них либо инициировать от имени легального пользователя установление связи;
- *атака со словарем.* Заключается в подборе пароля, содержащего наиболее часто встречающиеся слова или комбинацию букв/цифр. Обычно преобразованные при помощи неключевой хэш-функции пароли хранятся в файлах на компьютере. Задача злоумышленника состоит в том, чтобы получить искомый файл, преобразовать свой словарь посредством данной хэш-функции и произвести сравнения с целью найти совпадающие значения. Подобный тип атак может применяться и для сообщений;
- *подмена сообщений.* Производится путем замены во время работы криптопротокола сообщений или данных.

В качестве примера успешной атаки такого типа рассмотрим криптопротокол распределения секретных сеансовых ключей между двумя участниками информационного обмена (рис. 2.1). Авторами этого метода являются Нидхэм и Шредер. Уязвимость рассматриваемого криптопротокола впервые была обнаружена Денингом и Сакко. Каждый участник данного протокола должен разделить знание своего секретного ключа с сервером аутентификации. Протокол состоит из следующих шагов:

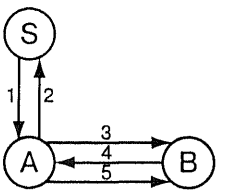

Рис. 2.1. Протокол распределения секретных сеансовых ключей

1. A → S: A, B, N_a. Участник A посылает запрос серверу S, в котором он указывает, что необходимо установить сеанс связи с участником B. В данном запросе присутствуют следующие значения:
   – A и B – имена или идентификаторы участников;
   – N_a – уникальное для данного сеанса число; используется для предотвращения повторных передач путем включения его в последующие сообщения.
2. S → A: $\{N_a, B, K_{ab}, \{K_{ab}, A\}_{Kbs}\}_{Kas}$. Сервер отвечает сообщением, зашифрованным на секретном ключе сервер-участник A (K_{as}), в котором находится сеансовый ключ (K_{ab}), а также еще одна копия этого ключа, зашифрованного на секретном ключе сервер-участник B (K_{bs}).
3. A → B: $\{K_{ab}, A\}_{Kbs}$. Участник B расшифровывает данное сообщение, поскольку ему известен ключ K_{bs}; в результате он получает ключ K_{ab}.

4. B → A: $\{N_b\}_{Kab}$. Данное сообщение участник B посылает для того, чтобы убедиться, что A владеет ключом K_{ab}, и показать участнику A свое знание K_{ab}.
5. A → B: $\{N_b - 1\}_{Kab}$. Участник A доказывает свое владение ключом K_{ab}, и на этом протокол заканчивает свою работу, в результате участники A и B получают общий секретный сеансовый ключ K_{ab}.

Уязвимость данного криптопротокола состоит в том, что если сеансовый ключ K^1_{ab} из предыдущей сессии был скомпрометирован, то злоумышленник (обозначим его через C), получивший возможность контроля сетевого трафика на шаге 3 новой (нескомпрометированной) сессии, перехватывает сообщение (A → B: $\{K_{ab}, A\}_{Kbs}$) и вместо него посылает сообщение (C → B: $\{K^1_{ab}, A\}_{Kbs}$). Участник B отвечает сообщением (B → A: $\{N_b\}K^1_{ab}$), полагая, что A хочет установить с ним новую сессию с ключом $K^1_{a\,b}$. C, получая данное сообщение, расшифровывает его и отправляет B сообщение (C → B: $\{N_b - 1\}\ K^1_{a\ b}$). Таким образом, C устанавливает сессию с B от имени A.

С точки зрения понимания возможных последствий, возникающих при обнаружении недоработок в логике работы криптопротокола, этот пример является очень показательным. Для устранения подобных проблем были специально разработаны средства и методы анализа корректности построения логики работы криптопротоколов. Применение формальных методов позволяет также обнаружить коммуникационную избыточность в криптопротоколах, что является немаловажным плюсом в условиях постоянно возрастающих требований к оперативности обработки информации в современных сетях передачи данных.

Третья группа атак, существенно понижающих уровень безопасности при использовании криптопротоколов, является наименее распространенной. Об этом, по крайней мере, можно судить по частоте их использования злоумышленниками. Но с другой стороны, в силу их трудного обнаружения, такие попытки являются наиболее опасными, поскольку они во многом зависят от математических свойств используемых криптографических алгоритмов. Подобные уязвимые места не поддаются формальному анализу и поэтому не могут обнаруживаться даже при использовании хорошо изученных криптопротоколов. Особенно характерно появление слабых точек при использовании асимметричных алгоритмов, так как они в основном построены на невозможности эффективного решения некоторых математических задач и не поддаются строгим математическим доказательствам и исследованиям с помощью формальных методов.

Нам остается дать несколько полезных советов, с помощью которых в какой-то мере можно избежать отдельных атак на криптопротоколы.

Полезные советы

Первый совет заключается в том, что перед зашифровыванием данных их необходимо подписывать. Если добавить подпись к уже зашифрованным данным, то может возникнуть ситуация, когда нельзя убедиться в том, что подписывающий имел представление о содержании подписываемой информации, а это обстоятельство приводит к невозможности доказать арбитру аутентичность подписи.

Приведем следующий пример.

Предположим, что участник A сначала зашифровывает сообщение M на открытом ключе e_B участника B, а затем подписывает его на своем секретном ключе d_A. Сообщение, передаваемое по открытым каналам связи, будет иметь вид:

$$((M^{eb} (\text{mod } n_B))^{da} (\text{mod } n_A)$$

Участник B имеет разложение n_B, и его множители могут оказаться длиной 200–300 бит. Тогда, используя теорему об остатках, участник B может найти такое x, что:

$$(M^1)^X = M(\text{mod } n_B),$$

где M^1 – сообщение, на которое необходимо перенести подпись участника A.

Для успешного проведения атак участнику B нужно только зарегистрировать следующий открытый ключ (xe_B, n_B), после чего он может доказать, что участник A подписал сообщение M^1, а не M. Атака подобного рода может быть реализована на протоколе CCITT X.509, где подписываются сообщения вида $\{T_A, N_A, B, X, \{Y\}^{Eb}(\text{mod } n_B)\}$, в которых X и Y – данные пользователя. Она также работает и в случае использования алгоритма Эль-Гамаля.

Следующий совет сводится к необходимости иметь механизмы, дающие пользователю возможность однозначно различать (идентифицировать) участников информационного обмена, а также позволяющие отличать разные сессии при использовании криптопротокола. Это требование особенно важно в том случае, когда подписание и расшифрование производится с применением асимметричного алгоритма (например, RSA), в котором используются одинаковые секретные ключи. Это вполне реальный вариант, поскольку в RSA, например, процессы генерации подписи и расшифрования являются одинаковыми математическими операциями.

Поэтому рекомендуется избегать ситуаций, когда для генерации подписи и для расшифрования сообщения используются одни и те же ключи или ключи, связанные друг с другом некоторым элементарным преобразованием.

Для обеспечения идентификации участника информационного обмена существует большое количество методов и средств. Необходимо отметить, что идентификация является синонимом термина «аутентификация пользователя» (см. раздел 2.2). Различие в сессиях протокола при этом реализуется, как правило, двумя методами:

- с помощью временных вставок;
- посредством случайного числа.

При этом, однако, сразу возникают сложности, связанные с доказательством того, что число, характерное для какой-либо сессии, ранее не использовалось. С временными вставками дело обстоит несколько иначе, поскольку проблемой в данном случае является синхронизация времени у всех участников информационного обмена.

Еще один совет (или принцип) заключается в том, чтобы каждый заинтересованный пользователь разумно применял избыточность в передаваемых сообщениях (например, в сообщениях установления сеансового ключа или передачи сертификата). Будьте уверены, что в данном контексте приложения вам необходима определенная избыточность и при этом наличие дополнительного количества битов не приводит к уязвимости системы, но и в этом вопросе следует соблюдать меру. Распространенным примером отыскивания злоумышленником слабых мест подобного вида является использование избыточной информации в протоколе «игра в покер по телефону», что позволяет установить, является ли переданное сообщение квадратичным вычетом, а далее, принимая во внимание свойства квадратичных вычетов, с большой долей вероятности «взломать» данный протокол.

Теперь необходимо упомянуть об обязательной проверке данных, пересылаемых в процессе выполнения протокола, на предмет установления истинного источника сообщения, и не принимать информацию без проведения соответствующих проверок. Это пожелание тоже очень важно для обеспечения безопасности передаваемой информации.

И наконец, используйте в работе только хорошо известные криптографические механизмы защиты информации.

2.1.3. Формальные методы анализа криптопротоколов

Выявление уязвимостей в известных криптопротоколах, которые до определенного момента считались надежными, предполагает разработку

формальных методов их анализа. Из существующих на сегодняшний день подходов к этому вопросу можно выделить четыре основных:

- моделирование и проверка работы протокола. Для этой цели полезно использовать специализированные языки и инструментарии, которые не создавались для анализа криптопротоколов;
- создание экспертных систем, которые разработчики криптопротоколов могут применить для апробирования различных сценариев функционирования криптопротокола;
- моделирование требований к семейству криптопротоколов. При этом можно употребить логику, разработанную специально для анализа таких свойств криптопротокола, как «знание» и «доверие»;
- разработка формальных моделей, основанных на алгебраических свойствах криптографических систем.

Каждый из перечисленных подходов не привязан к лежащим в основе криптопротоколов механизмам, а направлен только на анализ логики работы протокола.

Использование специализированных языков и инструментариев

Данный подход является наименее распространенным, поскольку его основная идея заключается в представлении криптопротокола как программы с последующей попыткой доказательства его корректности. Однако тут же необходимо отметить, что из доказательства корректности протокола не следует доказательство его безопасности. Наиболее успешными в данном направлении считаются формальные методы, автором которых является Кеммерер, и формальный язык LOTOS (Language of Temporal/Ordering Specification), созданный для анализа протоколов аутентификации. Кеммерер в своих работах выделил две цели, для достижения которых можно использовать формальные методы анализа криптопротоколов:

- формальный анализ того, что криптопротокол удовлетворяет требованиям безопасности;
- обнаружение уязвимостей в криптопротоколах.

Формальная модель построена на использовании конечных автоматов, по отношению к которым криптопротокол рассматривается как набор различных состояний, отличающихся друг от друга значениями переменных состояния. При этом значения переменных могут быть изменены только с помощью строго определенных процедур.

Примером использования конечных автоматов является анализ протоколов аутентификации. Для каждой процедуры (итерации криптопротокола)

определяется состояние криптопротокола (системы), выражающееся в описании состояний участников и коммуникационных каналов между ними. После чего каждый набор состояний анализируется на предмет корректности и отсутствия тупиковых ситуаций.

Применение экспертных систем

Использование экспертных систем для исследований криптопротоколов заключается в апробировании различных сценариев работы криптопротокола. В основе применения данных систем – задание некорректных состояний и изучение результатов их обработки криптопротоколом. Этот подход позволяет более эффективно, чем в первом случае, определять наличие в криптопротоколе уязвимостей, но не доказывает безопасность его использования, а также не дает возможности работать с ним в автоматических инструментариях для изучения различных атак на криптопротоколы. Другими словами, он позволяет определить, содержит ли криптопротокол известную уязвимость, но найти с его помощью новые уязвимости достаточно сложно.

На практике экспертные системы применяются совместно с BAN-логикой или формальными моделями. Так, например, метод, зафиксированный в данном подходе, реализован как часть протокольного анализатора NRL.

BAN-логика

Это направление на сегодняшний день наиболее развивающееся. В его основе – логика, разработанная для анализа свойств «знания» и «доверия» работы криптопротокола в целом или его отдельных частей. Ярким представителем подобного метода является BAN-логика, которая и послужила началом развития этого же направления.

Использование BAN-логики позволяет найти ответы на следующие вопросы:

1. Каких результатов в конечном счете можно достичь с помощью криптопротокола?
2. Содержатся ли в данном криптопротоколе избыточные шаги, которых можно было бы избежать, сохранив безопасность работы криптопротокола на прежнем уровне?
3. Необходимо ли зашифровывать данное сообщение или его можно передать в открытом виде?
4. Нужно ли включить в данный криптопротокол дополнительные шаги?

Постулаты, применяемые в BAN-логике

P believes X – P верит в то, что X истинно.

P sees X – кто-либо послал P сообщение, содержащее X, и P может прочитать и повторить X (возможно после проведения процедуры расшифрования).

P said X – P когда-либо посылал сообщение, содержащее X, и при этом P доверял X в момент его передачи.

P control X – P имеет права на X.

#(X) – данная конструкция означает, что X не было использовано в предыдущих итерациях протокола.

$P \overset{K}{\longleftrightarrow} Q$ – P и Q разделяют между собой ключ K, соответственно P и Q доверяют друг другу.

$\overset{K}{\longrightarrow} P$ – P имеет открытый ключ K, соответствующий секретному ключу K^{-1}, который никогда не будет раскрыт другими участниками криптопротокола.

$P \overset{X}{\longleftrightarrow} Q$ – X – секретная формула, известная только P и Q, и эту формулу P и Q могут использовать для идентификации друг друга.

$\{X\}_K$ *from P* – данная конструкция означает, что формула X была зашифрована на ключе K, принадлежащем P.

Некоторые правила BAN-логики

Приведенные правила при анализе криптопротоколов могут дополняться, однако в некотором роде их можно считать основополагающими, поскольку они описывают наиболее распространенные ситуации.

$$\frac{P believes Q \overset{K}{\longleftrightarrow}, P sees \{X\}_K}{P believes Q said X}$$

Если P верит, что Q и P разделяют между собой секретный ключ K, и видит сообщение X, зашифрованное на ключе K, и к тому же P не зашифровывал данное X на ключе K, тогда P имеет основания верить, что X было послано Q.

$$\frac{P believes \#(X), P believes Q believes X}{P believes X}$$

Если P верит в то, что X до этого не использовалось и что Q посылал X, тогда P может полагать, что Q доверяет X.

$$\frac{P believes Q controls X, P believes Q believes X}{P believes X}$$

Если P верит, что Q имеет права на X, и P верит, что Q доверяет X, тогда P доверяет X.

Согласно BAN-логике, прежде чем приступить к анализу криптопротокола, его необходимо представить в идеализированной форме. Например, процесс передачи сообщения $A \rightarrow B$: $\{A, K_{ab}\}_{Kbs}$ в идеализированной форме может быть зафиксирован в следующем виде: $A \rightarrow B$: $\{A \xleftrightarrow{Kab} B\}_{Kbs}$. Когда же B получит это сообщение, его можно в соответствии с правилами BAN-логики записать так: B sees $\{A \xleftrightarrow{Kab} B\}_{Kbs}$. В идеализированной форме часть сообщения, которая не участвует в доказательстве, опускается. Значит, открытая часть сообщения не включается в идеализированную форму, поскольку она может быть изменена злоумышленником. В общем случае идеализированная форма сообщения выглядит так: $\{X_1\}_{K1} \ldots \{X_n\}_{Kn}$, где каждое зашифрованное сообщение представлено независимо от других.

Отсюда можно сделать вывод, что анализ предполагает следующие шаги:

1. Представление протокола в идеализированной форме.
2. Присвоение начальных значений.
3. Применение логических формул к состояниям протокола для получения утверждений о состоянии системы после каждого шага протокола.
4. Применение логических постулатов к начальным значениям и последовательности утверждений для изучения доверия частям протокола.

Итак, криптопротокол в BAN-логике – это последовательность состояний $S_1 \ldots S_n$, каждое из которых представляется в виде $P \rightarrow Q$: X, где $P \neq Q$. Между состояниями вставляется последовательность утверждений. Они заключаются в комбинировании постулатов вида P believes X. Структурно это может быть представлено в следующем виде:

Начальные значения
S_1[утверждение 1]$S_2 \ldots$ [утверждение n –1]S_n

Выводы

На практике BAN-логика зарекомендовала себя с положительной стороны, особенно после того, как с ее помощью были найдены уязвимости в хорошо известных криптопротоколах, таких как Needham-Schroeder, CCITT X.509 и др. Точно так же была доказана избыточность в Kerberos, Yahalom, Andrew RPC handshake и CCITT X.509. Однако и в BAN-логике существует ряд проблемных вопросов. Нессет продемонстрировал простой пример, показывающий, что BAN-логика способна доказать свойства безопасности протокола, являющиеся заведомо ложными.

Формальные модели

Суть данного подхода – в разработке формальных моделей, позволяющих анализировать криптопротоколы на основе изучения алгебраических свойств криптографической системы. Среди достоинств данного подхода следует отметить возможность создания инструментариев, с помощью которых допускается автоматизировать процесс исследования криптопротоколов. В результате появился протокольный анализатор (NRL Protocol Analyzer), посредством которого были обнаружены как известные, так и неизвестные уязвимости. На сегодняшний день данный подход к анализу криптопротоколов практически не используется.

2.2. Протоколы аутентификации

2.2.1. Общие сведения

Расширение круга пользователей распределенных систем и усложнение задач, решаемых с помощью подобных сетей, привели к осознанию первостепенной важности проблем, связанных с обеспечением информационной безопасности при межсетевом взаимодействии, прежде всего с использованием открытых каналов передачи данных, создающих потенциальную возможность для действий как пассивного нарушителя, имеющего возможность только просматривать доступные ему сообщения, так и активного нарушителя, который наряду с прослушиванием может модифицировать доступные ему сообщения.

Вот почему одной из важнейших задач обеспечения безопасности является разработка методов и средств, позволяющих одной стороне (*проверяющему*) убедиться в идентичности другой стороны (*доказывающего*). Широко распространенными механизмами решения данной проблемы являются специальные приемы, дающие возможность проверить корректность сообщений, которые демонстрируют, что доказывающий владеет определенным секретом. Обычно такие методы и средства называются *идентификацией* (*аутентификацией*) пользователей, или *проверкой идентичности*. Существует также особая разновидность подобных протоколов, называемая *аутентификацией сообщений* (или *аутентификацией источника данных*), которая реализуется при помощи симметричных алгоритмов или/и цифровой подписи.

Основное отличие между аутентификацией пользователя и аутентификацией сообщения состоит в том, что последняя не обеспечивает временных гарантий того, когда данное сообщение было создано, а аутентификация

пользователей дает возможность доказывающему идентифицировать себя в ходе сессии.

Существуют также инструментарии, реализующие одновременно аутентификацию и распределение ключей; осуществление некоторых из них заключается в проведении аутентификации сообщений, которые представляют собой ключ (этот вопрос подробно рассматривается в следующем разделе).

Аутентификация пользователей – процесс, с помощью которого одна сторона убеждается в идентичности другой стороны, при этом другая сторона тоже активно участвует в процессе обмена информацией.

В рамках этой книги мы будем считать, что термины *идентификация*, *аутентификация пользователей* и *аутентификация* являются синонимами.

С точки зрения проверяющего к протоколам аутентификации предъявляются следующие критичные требования:

- при взаимном доверии сторон A и B стороне A необходимо убедится в идентичности стороны B; это так называемая *корректная завершенность протокола*;
- сторона B не должна иметь возможности повторного обмена информацией, переданной ранее в ходе взаимодействия со стороной A, с целью выдать себя за сторону A при аутентификации со стороной C; то есть протокол аутентификации должен быть *непереносимым*;
- различия в аутентификационном обмене между сторонами должны быть настолько существенными, чтобы ни одна из сторон не смогла реализовать обмен данными от имени другой стороны;
- информация, пересылаемая в рамках аутентификационного обмена, должна препятствовать получению статистических сведений, на основе которых пассивный нарушитель смог бы имитировать обмен от имени какой-либо стороны; то есть в результате информационного обмена знания участников друг о друге не должны увеличиться. О реализации данного требования будет сказано ниже.

Все протоколы аутентификации могут быть разделены на следующие категории:

- на основе знания чего-либо. Примером могут служить стандартные пароли, *персональные идентификационные номера* (PIN), а также секретные и открытые ключи, знание которых демонстрируется в протоколах типа «запрос-ответ».
- на основе обладания чем-либо. Обычно это магнитные карты, смарт-карты, touch memory и персональные генераторы, которые используются для создания одноразовых паролей.

- на основе каких-либо неотъемлемых характеристик. Эта категория включает методы, базирующиеся на проверке пользовательских биометрических характеристик (голос, сетчатка глаза, отпечатки пальцев). В данной категории не используются криптографические методы и средства. Аутентификация на основе биометрических характеристик применяется для контроля доступа в помещения либо к какой-либо технике.

Также можно классифицировать протоколы аутентификации по уровню обеспечиваемой безопасности или по возможности противостоять определенному классу атак. В соответствии с данным подходом протоколы аутентификации разделяются на следующие типы:

- *простая аутентификация* (на основе использования паролей);
- *строгая аутентификация* (на основе использования криптографических методов и средств);
- протоколы, обладающие свойством доказательства с нулевым знанием.

С точки зрения безопасности каждый из перечисленных типов способствует решению своих специфических задач, поэтому все протоколы активно применяются на практике. Единственным исключением являются протоколы аутентификации, обладающие свойством доказательства с нулевым знанием. Пока интерес к ним носит скорее теоретический, нежели практический характер, но времена меняются, и, возможно, уже в недалеком будущем их начнут использовать при обмене данными (подробно о данном типе протоколов аутентификации рассказано в конце этого раздела).

Говоря о классификации по уровню обеспечиваемой безопасности, необходимо упомянуть о типовых атаках на протоколы аутентификации и общих методах, помогающих их избежать (см. табл. 2.1). При этом следует учесть, что оценка уровня безопасности, обеспечиваемого протоколом аутентификации, может быть определена только по отношению к конкретным типам атак. Необходимо также отметить, что в данном случае подразумевается атака, направленная на нарушение логики работы протокола, или атака, учитывающая уязвимости в логике его работы.

Основными атаками на протоколы аутентификации являются:

- *самозванство* (impersonation). Заключается в том, что один пользователь пытается выдать себя за другого;
- *повторная передача* (replay attack). Заключается в повторной передаче аутентификационных данных каким-либо пользователем;
- *подмена стороны* аутентификационного обмена (interleaving attack). Злоумышленник в ходе данной атаки участвует в процессе

аутентификационного обмена между двумя сторонами и имеет возможность модификации проходящего через него трафика;

- *отражение передачи* (reflection attack). Один из вариантов предыдущей атаки, в ходе которой злоумышленник в рамках данной сессии протокола пересылает обратно перехваченную информацию;
- *вынужденная задержка* (forced delay). Злоумышленник перехватывает некоторую информацию и передает ее спустя некоторое время;
- *атака с выборкой текста* (chosen-text attack). Злоумышленник перехватывает аутентификационный трафик и пытается получить информацию о долговременных ключах.

Таблица 2.1. Атаки на протоколы аутентификации и пути их обхождения

Тип атаки	Пути обхождения
Повторная передача	Использование протоколов типа «запрос-ответ»; использование меток времени или случайных чисел; вставка идентифицирующий информации в ответы
Подмена стороны	Связывание всех сообщений в рамках данной сессии протокола (например, используя одноразовый и уникальный идентификатор во всех сообщениях)
Отражение	Использование идентификаторов сторон в передаваемых сообщениях; построение протокола таким образом, чтобы каждое сообщение имело отличную от других форму
Выборка текста	Использование протоколов, обладающих свойством доказательства с нулевым знанием; включение в каждое сообщение случайных чисел
Задержка передачи	Комбинирование использования меток времени вместе со случайными числами

Самая опасная угроза протоколам аутентификации возникает в том случае, когда нарушитель выдает себя за какую-либо другую сторону, обычно обладающую существенными привилегиями в системе. Именно получение привилегий и возможность работы от лица другого пользователя (в случае, если в системе используется система аудита) являются основными целями злоумышленника.

Кроме перечисленных атак на протоколы аутентификации существует атака следующего вида: после успешного прохождения аутентификации между двумя пользователями и установления соединения нарушитель «выкидывает» какого-либо пользователя из соединения и продолжает работу от его имени. Для устранения подобного рода атак существуют следующие приемы:

- периодическое выполнение процедур аутентификации в рамках уже установленного сеанса связи;
- привязка результата аутентификации к последующим действиям пользователей в рамках системы. Примером подобного подхода может

служить аутентификационный обмен секретными сеансовыми ключами, с использованием которых осуществляется дальнейшее взаимодействие пользователей.

Учитывая описанные выше атаки на протоколы аутентификации, а также вопросы практического применения протоколов аутентификации, можно выделить следующие свойства протоколов данного типа:

- *взаимная аутентификация.* Свойство, отражающее необходимость обоюдной аутентификации между сторонами аутентификационного обмена;
- *вычислительная эффективность.* Количество операций, необходимых для выполнения протокола;
- *коммуникационная эффективность.* Данное свойство отражает количество сообщений и их длину, необходимую для осуществления аутентификации;
- *наличие третьей стороны.* Примером третьей стороны может служить доверенный сервер распределения симметричных ключей или сервер, реализующий дерево сертификатов для распределения открытых ключей;
- *основа гарантий безопасности.* Примером могут служить протоколы, обладающие свойством доказательства с нулевым знанием;
- *хранение секрета.* Здесь имеется в виду, каким способом реализовано хранение критичной ключевой информации.

В заключение этого раздела приведем некоторые примеры использования протоколов аутентификации (подробно практическая часть будет изложена позже). Одной из основных задач, решаемых во время работы подобных протоколов, является разграничение доступа к ресурсам, когда при привилегированном доступе необходимо убедиться в идентичности пользователей (например, локальный или удаленный доступ к ресурсам компьютера или доступ к приложениям со стороны пользователей и т.д.). Схемы, основанные на паролях и применяющиеся для того, чтобы обеспечить доступ к *пользовательской учетной записи* (user account), могут быть рассмотрены в качестве примера матрицы контроля доступа: каждый ресурс содержит список идентификаторов пользователей, имеющих к нему доступ, при этом доступ к этому ресурсу разрешен только тем пользователям, которые указаны в списке контроля и успешно прошли процедуру аутентификации. Во многих приложениях (например, сотовая связь) идентификация пользователей осуществляется для выставления счетов за работу.

2.2.2. Простая аутентификация

Общие сведения

Одной из разновидностей схем аутентификации является простая аутентификация, в основе которой – применение двусторонне согласованного пароля с одновременным согласованием средств его использования и обработки. Идея простой аутентификации заключается в следующем: пароль (последовательность букв и цифр какого-либо алфавита) служит для обеспечения доступа пользователя к определенному системному ресурсу или к нескольким ресурсам (например, сервер, принтер и учетная запись пользователя) вместе с идентификатором пользователя. Пользователь передает свой пароль и идентификатор системе, которая проверяет их и в случае совпадения идентифицирует пользователя.

При этом передача пароля и идентификатора может производиться следующими способами:

- в открытом виде (например, при доступе к IIS (Internet information server) по протоколу HTTP в качестве одного из типов поддерживаемой IIS аутентификации (basic autentification));
- передача выделенного идентификатора пользователя, пароля пользователя, случайного числа и/или метки времени, причем все передаваемые данные защищены посредством однонаправленной функции.
- в защищенном виде совместно со случайным числом и/или меткой времени, причем все данные защищены с помощью однонаправленной функции.

Очевидно, что первый тип простой аутентификации не может служить основой средств обеспечения информационной безопасности, поскольку гарантирует минимальный уровень таковой и подвержен многочисленным атакам. Процедура простой аутентификации показана на рис. 2.2.

Справочник может быть реализован как локально, например на рабочей станции участника B, и тогда шаги 2 и 3 осуществляются в рамках рабочей станции, так и удаленно, то есть шаги 2 и 3 представляют собой сетевой обмен данными и создают дополнительную возможность для проведения атак.

Схемы организации простой аутентификации отличаются не только методами передачи и проверки паролей, но и видами хранения паролей.

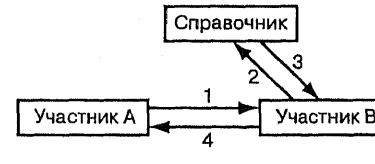

Рис. 2.2. Простая аутентификация

Схемы фиксированных паролей

Наиболее распространенный подход к организации хранения паролей пользователей – это их хранение в открытом виде в системных файлах; на них при этом устанавливаются атрибуты защиты от чтения и записи (например, при помощи описания соответствующих привилегий в списках контроля доступа операционной системы). Система сопоставляет введенный пользователем пароль с хранящейся в файле паролей записью. При этом не используются криптографические механизмы, такие как однонаправленные функции или шифрование. Недостатком данного метода является то, что злоумышленник может получить в системе привилегии администратора (в соответствии с которыми обладающий данным типом привилегий имеет все права доступа к системным файлам и ресурсам). Для усиления защиты этого способа хранения паролей может служить запись парольных файлов на внешние носители информации, такие как touch memory, смарт-карты и гибкие магнитные диски.

Более предпочтительным с точки зрения безопасности является метод хранения паролей с использованием односторонних функций (рис. 2.3) или шифрования. При проверке введенного пользователем пароля система вычисляет одностороннюю функцию и сравнивает результат с хранящимся значением в таблице паролей для данного идентификатора пользователя. В подобном случае файл, в котором хранится таблица, должен быть защищен только от записи. Применение односторонних функций или шифрование позволяет также защищать пароли в случае передачи их по общедоступным каналам. При использовании односторонней функции или шифрования следует помнить, что:

- не требуется ключевой информации, и эти методы не подпадают под действие экспортных ограничений;
- шифрование с точки зрения безопасности более предпочтительно.

Отдельно хотелось бы сказать о правилах формирования паролей. Некоторые системы, чтобы предотвратить выбор пользователем «слабых» паролей и не допустить или затруднить проведение атак по словарю на использующиеся пароли, предлагают собственные наборы знаков. Обычно правила формирования паролей ограничивают его длину снизу (например, 8 или 12 символами) и требуют для каждой такой записи наличия хотя бы одного знака из каждого набора символов (например, алфавит верхнего регистра, цифры и т.д.) или проверки, что предлагаемая запись не представляет собой слово, содержащееся в словаре, или имени пользователя (идентификатора пользователя), или специфической для данного пользователя

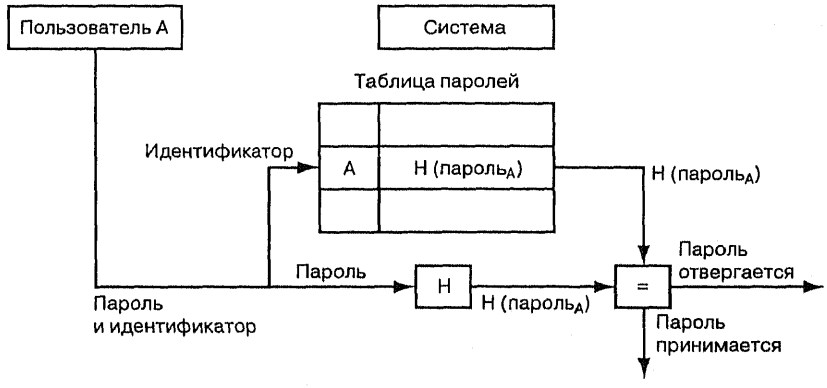

Рис. 2.3. Использование односторонних функций для проверки паролей

информации. Нарушитель, владея информацией об используемых в системе правилах формирования паролей, может применить модифицированную словарную атаку. Целью подобных правил является уменьшение энтропии пароля пользователя, что позволяет сделать его недосягаемым для словарной атаки или для полного перебора.

Чтобы ослабить эффект от словарной атаки, каждый пароль перед применением односторонней функции может быть расширен с помощью случайной строки. Оба пароля – хэшированный и дополненный – в произвольной последовательности записываются в файл паролей. Когда пользователь вводит пароль, система генерирует случайную последовательность и применяет к введенному паролю и случайной последовательности одностороннюю функцию. Проведение атаки методом полного перебора затруднится вследствие увеличения объема вычислений (например, если случайная последовательность состоит из t символов, то сложность перебора увеличится в 2^t), а также вследствие необходимости привлечения дополнительных ресурсов для проведения атак данного типа. Заметим, что при помощи случайной последовательности два пользователя, выбрав одинаковые пароли, будут иметь различные записи в файлах паролей. В таких системах в качестве случайной последовательности может использоваться идентификатор пользователя.

Другим вариантом организации парольных схем является использование целых фраз. Хотя обладающая смыслом фраза по своей сути не может иметь случайный характер, тем не менее энтропия такого пароля остается на достаточно высоком уровне вследствие того, что фразу легче запомнить, чем пароль, имеющий случайный набор символов, вот почему в подобной записи можно использовать гораздо большее количество символов.

Пример

В качестве примера простой аутентификации рассмотрим организацию парольной защиты в операционной системе (ОС) Windows NT. Прежде чем пользователь получит доступ к некоторым ресурсам системы, он должен пройти через процедуру входа в систему, при этом подсистема безопасности должна распознать его по имени и проверить подлинность запроса по паролю. Пароль пользователя хранится в базе данных в двух вариантах – в виде, необходимом для проведения аутентификации между компьютерами, работающими под управлением ОС Windows NT (NT hash), и в виде, необходимом для аутентификации между компьютерами, работающими под управлением ОС Windows 95 или Windows for WorkGroups (Lan manager hash). Для формирования NT hash используется алгоритм MD 4. Для формирования Lan manager hash все алфавитные символы исходной строки пароля приводятся к верхнему регистру. Каждая из двух половин 14-байтного символьного пароля обрабатывается независимо от другой. Если длина пароля меньше 14 символов, то вторая половина заполняется нулями. На основании этих двух 7-байтных половин формируется ключ для шифрования по DES некоторого 64-битного числа. В результате получаются две половины 16-байтного хэшированного пароля.

Схемы одноразовых паролей

Применение схем одноразовых паролей явилось заметным шагом вперед в использовании простой аутентификации. Дело в том, что при фиксированном (постоянном) наборе символов нарушитель, его перехвативший, может произвести повторную передачу с целью выдачи себя за легального пользователя. Частным решением этой проблемы как раз и является применение одноразовых паролей: каждый пароль в данном случае используется только один раз. На практике требуются следующие схемы одноразовых паролей:

- разделяемый список одноразовых паролей. Пользователь и проверяющий используют последовательность или набор секретных паролей, где каждый подбор употребляется только один раз. Данный список должен быть заранее распределен между сторонами аутентификационного обмена. Вариантом данного метода является использование таблицы запросов/ответов, в которой содержатся запросы и ответы, использующиеся сторонами для проведения аутентификации, причем каждая пара должна применяться только один раз;
- последовательность преобразуемых одноразовых паролей. Здесь изначально распределенным является только один пароль. В ходе очередной

сессии аутентификации пользователь создает и передает пароль именно для данной сессии, зашифрованный на секретном ключе, полученном из пароля предыдущей сессии;

- последовательности паролей, основанные на односторонней функции. Суть данного метода составляет последовательное использование односторонней функции (схема Лампорта, представленная ниже). Этот метод с точки зрения безопасности предпочтительней, чем метод последовательно преобразуемых паролей.

В схеме, предложенной Лампортом, пользователи начинают с разделения секрета w. Односторонняя функция H применяется для того, чтобы определить последовательность паролей: w, H(w), H(H(w)), ..., $H^t(w)$. Пароль, который будет использоваться в i-й сессии аутентификации, равен $w_i = H^{t-i}(w)$, а количество сессий, для которых назначается начальный секрет w, обозначим через t (на практике обычно t = 100 или 1000). Пользователь в ходе i-й сессии аутентификации (значение предыдущей сессии w_{i-1} сохраняется проверяющим) передает w_i, а проверяющий производит сравнение $H(w_i) = w_{i-1}$, и при совпадении идентичность пользователя принимается.

2.2.3. Строгая аутентификация

Общие сведения

Идея строгой аутентификации, реализованная в криптографических протоколах, заключается в том, что сторона доказывает свою идентичность, демонстрируя знание некоторого секрета (например, этот секрет должен быть предварительно распределен безопасным способом между сторонами аутентификационного обмена) при помощи последовательности запросов и ответов с использованием криптографических методов и средств. Существенным здесь является тот факт, что демонстрируется только знание секрета, но сам секрет в ходе аутентификационного обмена не раскрывается. Это обеспечивается посредством ответов на различные запросы, причем результирующий запрос зависит только от пользовательского секрета и начального запроса, который обычно представляет произвольно выбранное в начале протокола большое число. Во многих случаях строгая аутентификация заключается в том, что каждый пользователь идентифицируется по признаку владения своим секретным ключом. Другой пользователь имеет возможность определить, владеет ли его партнер по связи секретным ключом и может ли использовать его в качестве подтверждения того, что его партнер по связи действительно является предполагаемым пользователем.

В соответствии с рекомендациями X.509 процедуры строгой аутентификации могут быть следующих типов:

- аутентификация в одном направлении, в ходе которой обмен информацией идет только в одном направлении. Данный тип аутентификации позволяет:
  - подтвердить подлинность только одной стороны информационного обмена;
  - обнаружить нарушение целостности передаваемой информации;
  - обнаружить проведение атаки типа «повтор передачи»;
  - гарантировать, что передаваемыми аутентификационными данными может воспользоваться только проверяющая сторона;
- двусторонняя аутентификация по сравнению с первым типом содержит дополнительный ответ проверяющей стороны доказывающей стороне, который должен убедить ее, что связь устанавливается именно с той стороной, которой были предназначены аутентификационные данные;
- трехсторонняя аутентификация, которая содержит дополнительную передачу данных от доказывающей стороны проверяющей. Этот подход позволяет избежать использования меток времени при проведении аутентификации.

Необходимо отметить, что данная классификация условна. Эти приемы носят только теоретический характер, а набор используемых средств зависит непосредственно от контекста применения того или иного типа строгой аутентификации. Следует учитывать, что проведение строгой аутентификации требует дополнительного согласования сторонами используемых криптографических алгоритмов и ряда дополнительных параметров.

Рассмотрение практических вопросов построения и применения протоколов строгой аутентификации начнем с одноразовых параметров, используемых в протоколах аутентификации. Одноразовые параметры позволяют избежать повтора передачи, подмены стороны аутентификационного обмена и атаки с выбором открытого текста. При помощи одноразовых параметров можно обеспечить уникальность, однозначность и временные гарантии передаваемых сообщений. Среди используемых на сегодняшний день одноразовых параметров следует выделить: метки времени, случайные числа и номера последовательностей. Основная идея подобных параметров заключается в том, что они используются не более одного раза и с одной и той же целью. Различные их типы могут употребляться как отдельно, так и дополнять друг друга. Следует отметить, что данные параметры широко используются и в других вариантах криптографических протоколов (например, в протоколах распределения ключевой информации).

Существует несколько точек зрения на применение одноразовых параметров:

- проверка своевременности в протоколах, построенных по принципу запрос/ответ. В ней могут использоваться случайные числа, метки времени с синхронизацией часов или номера последовательностей для конкретной пары (проверяющий, доказывающий);
- обеспечение своевременности или гарантий уникальности заключается в контроле протокольных одноразовых параметров напрямую (посредством выбора случайного числа) или косвенно (анализируя информацию, содержащуюся в разделяемом секрете);
- однозначная идентификация сообщения или последовательности сообщений возможна при помощи выработки одноразового значения из монотонно возрастающей последовательности (например, последовательность серийных номеров, метки времени и т.д.) или при помощи случайного числа соответствующего размера;
- применение комбинации различных типов одноразовых параметров, например, когда случайные числа соединяются с метками времени. Подобный прием позволяет гарантировать, что псевдослучайные числа не используются повторно.

Строгая аутентификация, основанная на симметричных алгоритмах

Протоколы аутентификации, построенные на основе симметричных алгоритмов, требуют, чтобы проверяющий и доказывающий с самого начала разделили секретный ключ. Для закрытых систем с небольшим количеством пользователей каждая пара пользователей может априорно разделить его между собой. В больших распределенных системах применение симметричных ключей часто приводит к необходимости использования в протоколах аутентификации доверенного сервера, с которым каждая сторона разделяет знание ключа. Такой сервер распределяет сеансовые ключи для каждой пары пользователей всякий раз, когда один из них запрашивает аутентификацию другого. Кажущаяся простота данного подхода является обманчивой, на самом деле разработка протоколов аутентификации этого типа является сложной и с точки зрения безопасности не очевидной.

Протоколы с симметричными алгоритмами шифрования

Яркими представителями протоколов, обеспечивающих аутентификацию пользователей с привлечением в процессе аутентификации третьей доверенной стороны, являются протокол Kerberos и протокол распределения секретных ключей Нидхэма и Шредера.

Ниже приводятся примеры отдельных протоколов аутентификации, специфицированных в ISO/IEC 9798-2. В ходе работы подобные протоколы будут требовать предварительного распределения секретных ключей (что в дальнейшем позволит отказаться от третьей – доверенной – стороны). Мы рассмотрим следующие варианты аутентификации:

- односторонняя с использованием меток времени, состоящая из двух шагов;
- односторонняя с использованием случайных чисел, состоящая из трех шагов;
- двусторонняя, требующая от двух до трех шагов.

В каждом из этих случаев пользователь доказывает свою идентичность, демонстрируя знание секретного ключа, так как эта операция производится на основе зашифрования запросов с помощью этого секретного набора символов.

При использовании в процессе аутентификации симметричного шифрования необходимо также реализовывать механизмы обеспечения целостности передаваемых данных. Так, например, применение блочного алгоритма в режиме ECB не позволяет обнаружить перестановку или удаление блоков шифрованного сообщения; использование же шифрования в режиме CBC является частным решением. Поэтому протоколы аутентификации, выполненные под конкретные задачи, должны также реализовывать механизмы обеспечения целостности на основе общепринятых способов.

В ходе рассмотрения протоколов аутентификации в данном разделе мы будем применять следующие обозначения:

- r_A – случайное число, сгенерированное участником A.
- t_A – метка времени, сгенерированная участником A.
- E_k – симметричное шифрование на ключе k (ключ k должен быть предварительно распределен между A и B).

1. Односторонняя аутентификация, основанная на метках времени:

   $A \rightarrow B: E_k(t_A, B)$

   После получения и расшифрования данного сообщения участник B убеждается в том, что метка времени действительна, и имя, указанное в сообщении, совпадает с его собственным. Предотвращение повторной передачи данного сообщения основывается на том, что без знания ключа невозможно изменить метку времени.
2. Односторонняя аутентификация, основанная на использовании случайных чисел:

A←B: r_B
A→B: $E_k(r_B, B)$

Участник B расшифровывает полученное сообщение и сравнивает случайное число, содержащееся в сообщении, с тем, которое он послал участнику A. Дополнительно он проверяет имя, указанное в сообщении. Этот протокол может быть модифицирован с целью предотвращения атак с выборкой открытого текста. Для этого во второе сообщение может быть добавлено дополнительное случайное значение или изменена форма первоначального сообщения, с тем чтобы нарушитель, перехвативший его, не смог сделать вывод о случайном значении, которое будет добавлено во второе сообщение.

3. Двусторонняя аутентификация, использующая случайные значения:

A←B: r_B
A→B: $E_k(r_A, r_B, B)$
A←B: $E_k(r_B, r_A)$

Применение третьего сообщения позволяет A выяснить, что он имеет дело именно с участником B, на основе проверки значений r_A и r_B. Кроме представленного варианта двусторонняя аутентификация может быть реализована при помощи односторонних типов аутентификации в двух направлениях.

Протоколы, основанные на использовании однонаправленных ключевых функций

Протоколы, представленные выше, могут быть модифицированы путем замены алгоритмов симметричного шифрования на ключевые хэш-функции. Это бывает необходимо, если алгоритмы блочного шифрования недоступны (например, в случае экспортных ограничений). Тогда в представленных выше протоколах должны быть внесены следующие изменения:

- функция шифрования E_k заменяется функцией h_k;
- поверяющий вместо установления факта совпадения полей в расшифрованных сообщениях с предполагаемыми значениями на основе предполагаемых значений вычисляет результат однонаправленной функции и сравнивает его с полученным от другого участника обмена информацией;
- в протоколе 1 t_A может передаваться дополнительно в открытом виде, а в протоколе 2 r_A может дополнительно передаваться в открытом виде.

Приведем в качестве примера модификацию протокола 3. Результирующий протокол известен как SKID 3 и имеет следующую структуру:

$A \leftarrow B: r_B$
$A \rightarrow B: r_A, h_k(r_A, r_B, B)$
$A \leftarrow B: h_k(r_B, r_A, A)$

Реализация аутентификации с использованием дополнительных устройств

В этом примере рассмотрен механизм аутентификации с использованием генератора аутентификационной информации. На практике данная реализация используется в банковской сфере и предполагает реализацию генератора, например на смарт-картах.

В ходе протоколов, построенных по принципу запрос/ответ, в некоторых случаях возникает необходимость использования вычислительных устройств и устройств хранения долговременной ключевой информации. Кроме того, такие устройства, как смарт-карты, могут реализовывать в себе устройства хранения ключевой информации и вычислитель. Здесь мы рассмотрим устройство, которое используется для генерации кодов или одноразовых паролей (рис. 2.4). Оно называется *генератором кодов* (passcode generator) и содержит специфичный для данного устройства секретный ключ.

Полученный пользователем запрос от системы поступает на вход генератора, и на его основе вырабатывается ответ. При этом обязательно применяется секретный ключ и PIN-код пользователя, а также функция преобразования, реализованная в генераторе (это может быть алгоритм блочного шифрования, асимметричного шифрования и т.д.). Система, задействовав

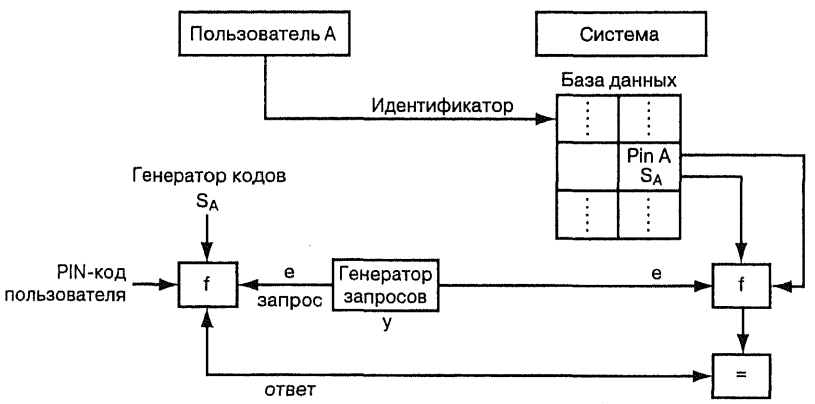

Рис. 2.4. Алгоритм работы генератора кодов

информацию, хранящуюся в базе данных, формирует предполагаемый ответ и сравнивает его с данными, полученными от пользователя.

Строгая аутентификация, основанная на асимметричных алгоритмах

Алгоритмы с открытыми ключами также могут использоваться в протоколах строгой аутентификации. В этом случае доказывающий может продемонстрировать знание секретного ключа одним из следующих способов:

- расшифровать запрос, зашифрованный на открытом ключе;
- подписать запрос проверяющего.

Пара ключей, необходимая для аутентификации, по соображениям безопасности не должна использоваться для других целей (например, для шифрования). Здесь следует предостеречь читателей – выбранная система с открытым ключом должна быть устойчивой к атакам с выборкой зашифрованного текста даже в том случае, если нарушитель попытается получить критичную информацию, выдав себя за проверяющего и действуя от его имени.

Аутентификация с использованием асимметричных алгоритмов шифрования

В качестве примера протокола, построенного на использовании асимметричного шифрования и аутентификаторов, можно привести следующий протокол аутентификации:

$A \leftarrow B$: $h(r)$, B, $P_A(r, B)$
$A \rightarrow B$: r

Участник B выбирает случайным образом r и вычисляет значение $x = h(r)$ (значение демонстрирует знание r без раскрытия самого r), далее вычисляется значение $e = P_A(r, B)$. Под P_A подразумевается алгоритм асимметричного шифрования (например, RSA), а под h – хэш-функция. Участник A расшифровывает e и получает значения r^1 и B^1, а также вычисляет $x^1 = h(r^1)$. После этого производится ряд сравнений, доказывающих, что $x = x^1$ и что полученный идентификатор B^1 действительно указывает на участника B. В случае успешного проведения сравнения участник A посылает r. Получив его, участник B проверяет, то ли это значение, которое он отправил в первом сообщении.

В качестве следующего примера приведем модифицированный протокол Нидхэма и Шредера, основанный на асимметричном шифровании. Достаточно подробно данный протокол описывается в разделе, посвященном распределению ключевой информации, поскольку основной вариант

протокола используется для аутентификационного обмена ключевой информацией. Рассматривая протокол Нидхэма и Шредера, использующийся только для аутентификации, будем подразумевать под P_B асимметричное шифрование участником B. Протокол имеет следующую структуру:

$A \rightarrow B: P_B(r_1, A)$
$A \leftarrow B: P_A(r_2, r_1)$
$A \leftarrow B: r_2$

Аутентификация, основанная на использовании цифровой подписи

Аутентификация, специфицированная в рекомендациях X.509, основана на использовании цифровой подписи, меток времени и случайных чисел. Поскольку она относится к разряду аутентификационного ключевого обмена, то о ней рассказывается в разделе, посвященном распределению ключевой информации.

Для описания примеров введем следующие обозначения:

- r_A и t_A – случайное число и временная метка соответственно;
- S_A – подпись, сгенерированная участником A;
- $cert_A$ – сертификат открытого ключа участника A.

В случае, если участники изначально имеют аутентичные открытые полученные друг от друга ключи, сертификатом можно и не пользоваться, в противном случае они служат для подтверждения подлинности открытых ключей.

В качестве примеров приведем следующие протоколы аутентификации:

1. Односторонняя аутентификация с применением меток времени:

   $A \rightarrow B: cert_A, t_A, B, S_A(t_A, B)$

   После принятия данного сообщения B проверяет действительность метки времени, указанный идентификатор и (используя открытый ключ из сертификата) корректность подписи.
2. Односторонняя аутентификация с использованием случайных чисел:

   $A \leftarrow B: r_B$
   $A \rightarrow B: cert_A, r_A, B, S_A(r_A, r_B, B)$

   Участник B проверяет корректность подписи и адресата сообщения. Случайное число r_A используется для предотвращения атак с выборкой открытого текста.
3. Двусторонняя аутентификация с использованием случайных чисел:

   $A \leftarrow B: r_B$
   $A \rightarrow B: cert_A, r_A, B, S_A(r_A, r_B, B)$
   $A \leftarrow B: cert_B, A, S_B(r_B, r_A, A)$

2.2.4. Протоколы аутентификации, обладающие свойством доказательства с нулевым знанием

Одна из уязвимостей протоколов простой аутентификации заключается в том, что после того, как доказывающий передаст проверяющему свой пароль, проверяющий может, используя данный пароль, выдать себя за проверяемого. Немногим лучше обстоит дело с протоколами строгой аутентификации. Дело в том, что A, отвечая на запросы B, обязан продемонстрировать знание секретного ключа, пусть даже и одноразово; при этом передаваемая информация не может быть напрямую использована B. Тем не менее некоторая ее часть поможет B получить дополнительную информацию о секрете A. Например, B имеет возможность так сформировать запросы, чтобы передаваемые ответы анализировались на предмет содержания дополнительной информации.

Протоколы доказательства с нулевым знанием были разработаны специально для решения данной проблемы. Этой цели можно добиться при помощи демонстрации знания секрета, однако проверяющий должен быть лишен возможности получать дополнительную информацию о секрете доказывающего. Если сформулировать эту мысль в более строгой форме, то протоколы *доказательства с нулевым знанием* (далее – ZK-протоколы) позволяют установить истинность утверждения и при этом не передавать какой-либо дополнительной информации о самом утверждении.

ZK-протоколы являются примером систем интерактивного доказательства, в которых проверяющий и доказывающий обмениваются многочисленными запросами и ответами, обычно зависящими от случайных чисел (в идеальном случае зависимость может реализовываться в виде подбрасывания монеты), которые позволяют сохранить секрет в тайне. Цель доказывающего – убедить проверяющего в истинности утверждения. Проверяющий же принимает или отклоняет доказательство. Необходимо отметить, что доказательство носит скорее вероятностный, нежели абсолютный характер.

Интерактивное доказательство, использующееся для идентификации, может быть сформулировано как доказательство знания. A владеет некоторым секретом s и с помощью последовательных ответов на вопросы (при наличии согласованных входных данных и функций) пытается убедить B в знании s. Заметим, что доказательство знания s отличается от доказательства того факта, что s существует. Например, доказательство знания простых множителей n (основания в RSA) отличается от доказательства того, что n является составным. Интерактивное доказательство может быть названо доказательством знания в том случае, если выполнены свойства стойкости и целостности.

Интерактивный протокол обладает свойством целостности, если только доверенные стороны смогут успешно закончить протокол (когда проверяющий принимает доказательство).

Интерактивный протокол является стойким при условии, что не существует алгоритма M, работающего за полиномиальное время и обладающего следующим свойством: если злоумышленник, пытающийся выдать себя за A, может с малой вероятностью успешно закончить протокол с участником B, тогда M в ходе работы протокола может использоваться для получения дополнительных знаний о секрете A, которые с высокой вероятностью позволяют успешно выполнить некоторые шаги протокола (в худшем случае весь протокол).

Несмотря на то что в основе построения ZK-протоколов лежат те же математические идеи, что и в протоколах, построенных на открытой криптографии, между ними есть некоторые отличия. Прежде всего они касаются:

- уменьшения эффективности в ходе использования. Протоколы доказательства с нулевым знанием не уменьшают свою эффективность даже при повторных передачах и соответственно являются стойкими к атаке с выборкой открытого текста. Данное свойство в некоторых случаях позволяет предпочесть использование ZK-протоколов;
- использования шифрования. Многие ZK-протоколы поволяют избегать употребления шифрования явным образом, что дает несомненные преимущества, например в случае экспортных ограничений;
- эффективности. Коммуникационная и вычислительная эффективность в ZK-протоколах основана прежде всего на том факте, что они по своей природе являются интерактивными протоколами доказательства.

Среди сходств данных протоколов можно отметить следующие:

- наличие недоказумых предположений. ZK-протоколы, так же как и протоколы, основанные на открытой криптографии, напрямую используют недоказуемые предположения (например, доказуемая стойкость к факторизации);
- доказуемость обладания свойством ZK. На практике достаточно сложно доказать наличие данного свойства у протокола.

Для практического рассмотрения концепции построения протоколов, обладающих свойством ZK, приведем оригинальную версию протокола Фиата и Шамира (Fiat-Shamir). Целью данного протокола является демонстрация знания секрета s доказывающей стороной, при этом информация о самом секрете не раскрывается проверяющему и он не имеет предварительно

распределенных сведений. Безопасность протокола основана на сложности извлечения квадратного корня по модулю n (имеет то же значение, что и в RSA), не зная разложения n.

Опишем общую структуру протоколов данного класса:

A доказывает знание секрета s B при помощи t-шагов по три сообщения в каждом.

Параметры протокола:

- доверенный центр T выбирает и публикует модуль $n = pq$ (p и q сохраняются в секрете);
- каждый доказывающий выбирает секрет s взаимно простой с n, $1 \leqslant s \leqslant n - 1$, вычисляет $v = s^2 \bmod n$ и регистрирует v у T в качестве своего открытого ключа.

Сообщения, передаваемые в рамках каждого шага:

A→B: $x = r^2 \bmod n$
A←B: $e \in \{0,1\}$
A→B: $y = rs^e \bmod n$

B принимает доказательство за t шагов, и последовательность действий в рамах протокола имеет следующий вид:

- A выбирает случайное число r, $1 \leqslant r \leqslant n - 1$ и посылает B $x = r^2 \bmod n$.
- B выбирает случайным образом e и посылает его A.
- A вычиляет y и посылает его B, где $y = r$ $(e = 0)$ или $y = rs$ $(e = 1)$.
- B отвергает доказательство, если $y = 0$, иначе производится проверка $y^2 \equiv xv^e \bmod n$. В зависимости от e $y^2 = x \bmod n$ или $y^2 = xv \bmod n$, иначе $v = s^2 \bmod n$.

Значение параметра t выбирается равным 20 или 40.

A→B: начальные значения
A←B: запрос
A→B: ответ

Доказывающий выбирает произвольное начальное значение, являющееся секретом, и на его основе вычисляет некоторый открытый параметр. Выбор начального значения зависит прежде всего от дальнейших шагов протокола, при помощи которых производится доказательство. Конструкция протокола построена таким образом, что только сторона, владеющая секретом, сможет ответить на запросы в рамках протокола, и при этом ответы не несут какой-либо информации о данном секрете.

2.3. Протоколы распределения и управления ключевой информацией

В этом разделе будут рассмотрены протоколы распределения ключевой информации между несколькими сторонами, используемые для этой цели криптографические методы и средства, а также механизмы управления ключевой информацией, содержащие процедуры хранения, использования и ее смены. При этом речь пойдет как о протоколах распределения ключевой информации между двумя сторонами с использованием третьей стороны и без использования таковой, так и о протоколах распределения ключей для конференц-связей. Далее рассказывается о проблемах управления ключами, рассматриваются вопросы обновления ключей в режиме online, процедуры хранения/восстановления ключей, их смена, а также вопросы управления сертификатами.

2.3.1. Протоколы распределения ключевой информации

Общие сведения

Под *распределением ключевой информации* будем понимать процесс или протокол, с помощью которого знание секретного ключа разделяется между двумя или более сторонами для последующего использования его в криптографических механизмах.

В свою очередь распределение ключей можно разграничить на *ключевой транспорт* и *соглашения о ключах*.

Ключевой транспорт (может быть реализован в виде протокола или механизма) – способ распределения ключей, позволяющий стороне, создавшей или получившей ключ, передать его другой стороне безопасным способом.

Соглашение о ключах (может быть реализовано в виде протокола или механизма) – способ распределения ключей, позволяющий разделить знание секрета между двумя (или более) сторонами в виде значения функции от требуемой к распределению информации, и при этом ни одна из сторон не сможет предварительно вычислить результирующее значение.

Кроме непосредственно механизмов распределения ключей существуют ключевые транспорты и соглашения о ключах, реализующие различные формы их обновления.

Протоколы распределения ключей в ходе распределения ключевого материала также требуют аутентификации, которая должна гарантировать, что только проверенная сторона получит доступ к распределяемой ключевой

информации. Существуют и другие схемы распределения ключей, требующие предварительного распределения некоторого ключевого материала. Такие системы называются схемами с предварительным распределением ключей.

Названная группа протоколов противоположна схемам динамического распределения ключей, которые в ходе выполнения протокола вырабатывают общие ключи для пары или группы пользователей. Другими словами, динамические схемы не требуют предварительной передачи какого-либо ключевого материала. Их также часто называют системами распределения сеансовых ключей.

По уровню обеспечиваемой в ходе работы безопасности различие между предварительным и динамическим распределением ключей заключается в том, что динамическое распределение является стойким к атакам с выборкой известных ключей. Общая классификация схем распределения ключей представлена в табл. 2.2.

Таблица 2.2. Обобщенная классификация схем распределения ключей

	Распределение ключей		
	Ключевой транспорт	Соглашения о ключах	
Симметричные механизмы	Динамические схемы распределения	Динамические схемы распределения	Схемы с предварительным распределением
Асимметричные механизмы	Динамические схемы распределения	Динамические схемы распределения	Схемы с предварительным распределением

Многие протоколы распределения ключей в ходе своей работы требуют наличия третьей (доверенной) стороны, которая обычно называется *доверенная третья сторона, доверенный сервер, центр распределения ключей, аутентификационный сервер, центр трансляции ключей* и т.д. Особенности различных схем организаций доверенной стороны будут рассмотрены ниже.

Среди характеристик, по которым можно классифицировать схемы распределения ключей, можно выделить следующие:

- основа аутентификации, которой может быть аутентификация пользователей, аутентификация ключей и подтверждение приема ключа;
- взаимность аутентификации. Каждый из приведенных выше типов аутентификации, используемых в схемах распределения ключей, в свою очередь может относиться к односторонней и взаимной;

- наличие механизмов, гарантирующих уникальность и своевременность ключа (то есть ключ ранее не использовался, и доставка его была произведена в соответствии с принятыми в данной схеме распределения ключей договоренностями);
- контроль ключей. В некоторых схемах (ключевой транспорт) одна из сторон сама выбирает ключ. В других же (соглашения о ключах) ключ получается из распределенной информации, и одна из сторон обмена ключами может желать, чтобы выработанный ключ не мог быть предсказан другой стороной;
- эффективность. Различается по:
  - количеству передаваемых по каналам связи сообщений;
  - количеству битов в передаваемых сообщениях;
  - сложности производимых каждой стороной вычислений;
  - возможности производить предварительные вычисления;
- наличие третьей стороны. Этот параметр можно разделить на:
  - требуется в режиме online;
  - требуется в режиме offline;
  - не требуется;
- наличие сертификатов открытых ключей;
- невоспроизводимость. То есть протокол, например, должен обеспечивать подтверждение приема ключа и фиксацию использованных ключей.

Ознакомившись с вышеперечисленными характеристиками, можно сделать вывод о том, насколько оправдано использование той или иной схемы распределения ключей. Однако оценка этих схем не заканчивается учетом представленных пунктов, необходимо также принять во внимание, насколько данная система распределения ключей справляется с функциями, возлагаемыми на нее со стороны механизмов управления ключами (но об этом позже).

При выборе некоторых предпочтительных характеристик схемы распределения ключей не следует также забывать о вопросах безопасности конкретной схемы. Как уже говорилось, все уязвимости криптопротоколов разделяются на три вида, поэтому особое внимание следует уделять используемым криптографическим алгоритмам. Даже в случае выбора надежных алгоритмов и правильного их применения существует возможность незаконного воздействия на схему распределения ключей, поскольку обычно сообщения передаются по незащищенным каналам передачи данных, где злоумышленник может действовать как пассивно, так и активно.

По отношению к схемам распределения ключей атаки, проводимые нарушителем, могут быть следующего рода:

- определение сеансового ключа; для чего используется информация, полученная с помощью перехвата информации в сети;
- тайное или скрытое участие в процессе инициирования обмена ключами от имени одной из сторон и совершение действий, направленных на изменение передаваемой информации;
- инициирование обмена ключевой информации с одной или более сторонами, в результате чего появляется возможность модифицировать передаваемый через нарушителя трафик;
- попытка выдать себя за доверенную сторону. Не имея возможности вычислить ключ, злоумышленник может попытаться получить таким образом доступ к информации, разделяемой в ходе обмена информацией.

В зависимости от характера доступной ему информации злоумышленник может быть классифицирован следующим образом:

- *внешний нарушитель* – имеющий доступ только к информации, передаваемой по каналам передачи данных (в рамках этой классификации не рассматривается нарушитель, пытающийся получить доступ к информации, используя свойства физической среды передачи и обработки ключевой информации);
- *внутренний нарушитель* – имеющий доступ к дополнительной информации (например, сеансовый ключ или часть некоторой секретной информации) на основе владения некоторыми привилегиями в системе обработки и хранения ключевой информации (например, физический доступ к закрытым ресурсам компьютера и т.д.);
- *одноразовый внутренний нарушитель* – получивший информацию случайным способом;
- *постоянный внутренний нарушитель*, то есть злоумышленник, имеющий постоянный доступ к критической информации.

По отношению к схемам распределения ключей нарушитель может ставить перед собой следующие цели:

- получить долговременный секретный ключ или текущий сеансовый ключ;
- получить ранее использовавшийся сеансовый ключ.

Анализируя цели нарушителя, можно заметить, что, хотя второй пункт и кажется бессмысленным, особенно по отношению к сеансовым ключам,

и практически безопасным (опасность заключается только в получении доступа к ранее передаваемой информации, при этом сами сведения могут не представлять практического интереса), в зависимости от конкретных схем распределения ключей достижение противником поставленной цели может привести к негативным последствиям и для будущего ключевого обмена. Так, в протоколе Диффи-Хэлмана, одно время признававшимся безопасным, была обнаружена уязвимость, возникающая в случае компрометации ранее использовавшегося сеансового ключа. Подобные атаки на схемы распределения ключей называются *атаками с известным ключом* (known-key attack). Возможность проведения атаки с известным ключом может быть существенно снижена по следующим причинам:

- вероятность компрометации ранее использовавшегося сеансового ключа в системах с криптографией средней стойкости гораздо выше, нежели компрометация долговременного ключа;
- в некоторых системах в силу различных причин не обеспечивается хранение использованных сеансовых ключей или их надежное удаление.

Ключевой транспорт, построенный на основе симметричного шифрования

В этом разделе рассматриваются ключевые транспорты, построенные на основе симметричных алгоритмов, которые во время работы не используют третью сторону и требуют наличия таковой (табл. 2.3). Соответствующие протоколы представлены в табл. 2.2.

Таблица 2.3. Ключевой транспорт, основанный на симметричном шифровании

Наименование протокола	Тип используемой третьей стороны	Использование меток времени	Количество пересылаемых сообщений
Протокол обмена ключами типа «точка-точка»	Не используется	Дополнительно	1–3
Протокол Шамира	Не используется	Нет	3
Kerberos	Центр распределения ключей	Используется	4
Протокол распределения ключей Нидхэма и Шредера	Центр распределения ключей	Нет	5
Протокол Otway-Rees	Центр распределения ключей	Нет	4

Ключевой транспорт, основанный на симметричном шифровании и не использующий третью сторону

Этот тип схем распределения ключей разбивается на два подвида: требующие предварительного распределения определенной ключевой информации, с использованием которой будут в последующем распределятся сеансовые ключи; и не требующие предварительного обмена.

Протокол обмена сеансовыми ключами типа «точка-точка» построен на симметричном шифровании с использованием долговременного секретного ключа K. Этот ключ должен быть предварительно распределен между двумя сторонами с помощью безопасного канала передачи данных (например, фельдъегерская почта) или с использованием схем предварительного распределения ключей. Таким образом, секретный ключ K должен быть распределен между каждой парой абонентов в сети. В ходе описания протокола мы будем использовать следующие обозначения:

- r_A, t_A и n_A означают случайное число, метку времени и номер последовательности сообщений;
- E_K – симметричное шифрование на секретном ключе K.

Существует несколько разновидностей схем распределения ключей этого типа:

1. A→B: $E_K(r_A)$, где r_A – сеансовый ключ для участников A и B. Очевидно, что представленный вариант имеет только теоретическое значение, поскольку с точки зрения безопасности данный протокол является нестойким по отношению ко многим атакам (например, к повторной передаче). Модифицировать протокол можно при помощи добавления t_A и B в передаваемое сообщение, что позволит избежать повторных передач и идентифицировать получателя данного сообщения. Для обеспечения двустороннего согласования сеансового ключа B, в свою очередь, может ответить сообщением, содержащим $f(r_A, r_B)$, где f является односторонней функцией, результат ее вычисления и будет результирующим сеансовым ключом.
2. Этот вариант построен на механизме запросов/ответов:

   A←B: n_B
   A→B: $E_K(r_A, n_B, B)$

   Так же как и в предыдущем варианте, r_A является секретным ключом. Отказ от использования меток времени позволяет избежать согласования времени на рабочих станциях участников обмена. Вместо них используются уникальные номера последовательности сообщений, которые позволяют гарантировать неповторимость сообщений,

содержащих ключи. Если требуется, чтобы сеансовый ключ вырабатывался двумя пользователями, можно представить следующий вариант данного сообщения:

$A \leftarrow B: n_B$
$A \rightarrow B: E_K(r_A, n_A, n_B, B)$
$A \leftarrow B: E_K(r_B, n_B, n_A, A)$

Недостатком этого типа протоколов является то, что при компрометации долговременного секретного ключа будут скомпрометированы все остальные сеансовые ключи для данной пары абонентов, и для восстановления защищенной связи между данными абонентами придется заново производить распределение долговременного секрета. Необходимо отметить, что подобная процедура может занять достаточно длительное время или потребует применения дополнительных мер распределения ключей. Вот почему этот тип протоколов на практике не применяется.

Следующий протокол позволяет избежать недостатков, свойственных предыдущему протоколу. Схема распределения ключей Шамира (в основе ее – протокол Диффи-Хэлмана) построена на симметричных алгоритмах и позволяет распределить сеансовые ключи без предварительного распределения пользователями некоторой ключевой информации. Все, что нужно пользователям, – это владение собственными секретными симметричными ключами. Протокол обеспечивает защиту от пассивного нарушителя, но не обеспечивает аутентификации.

Протокол начинается с того, что выбирается общий параметр схемы распределения ключей – модуль n. Далее каждый пользователь выбирает свое секретное число (a для участника A и b для участника B), при котором $1 \leqslant a \leqslant n - 2$ и является взаимно простым с $n - 1$. Сам протокол состоит из следующих шагов:

1. Участник A выбирает случайный ключ K, такой, что $1 \leqslant K \leqslant n - 1$, вычисляет значение $K^a \bmod n$ и передает его участнику B.
2. Участник B возводит в степень b полученное значение и результат пересылает участнику A.
3. Участник A возводит полученное значение в степень $a^{-1} \bmod n - 1$ и пересылает результат участнику B.
4. Участник B возводит полученное значение в степень $b^{-1} \bmod n - 1$, получая при этом сеансовый ключ K.

В общем виде данный протокол будет иметь следующую структуру:

$A \rightarrow B: K^a \bmod n$
$A \leftarrow B: (K^a)^b \bmod n$
$A \leftarrow B: (K^{ab})^{a^{-1}} \bmod n$

К сожалению, этот протокол унаследовал все уязвимые точки оригинального протокола Диффи-Хэлмана, которые, в свою очередь, могут быть устранены с помощью доработок этого же протокола.

Ключевой транспорт, основанный на симметричном шифровании и использующий третью сторону

Ключевые транспорты, построенные по данному принципу, при обмене ключами используют третью сторону (доверенный сервер), с помощью которой обе стороны, желающие выработать общий секретный ключ, должны предварительно получить некоторую ключевую информацию (например, это может быть симметричный секретный ключ, использующийся между данной стороной и доверенным сервером). В таких схемах доверенный сервер может быть следующих типов (в зависимости от выполняемых им функций):

- центр распределения ключей (KDC), непосредственно занимающийся доставкой сеансового ключа;
- центр трансляции ключей (KTC), где с помощью перешифрования сеансовый ключ одной стороны становится доступным для другой стороны.

Ярким представителем подобного семейства схем распределения ключей является протокол Kerberos, который наиболее часто применяется на практике (речь об этом протоколе пойдет позже).

В данной главе мы остановимся на рассмотрении двух протоколов: протокола распределения ключей Нидхэма и Шредера и протокола Otway-Rees.

Протокол распределения ключей Нидхэма и Шредера имеет долгую историю и за все время существования для устранения появляющихся уязвимостей не раз подвергался многочисленным преобразованиям, тем не менее его различные модификации широко используются на практике. Для работы этот механизм требует наличия KDC (обозначим его как S), с которым взаимодействуют участники A и B, желающие выработать общий сеансовый ключ.

Протокол имеет следующую структуру:

$A \rightarrow S$: A, B, N_a

$A \leftarrow S$: $E_{Kas}(N_a, B, k, E_{Kbs}(k, A))$

$A \rightarrow B$: $E_{Kbs}(k, A)$

$A \leftarrow B$: $E_k(N_b)$

$A \rightarrow B$: $E_k(N_b - 1)$,

где N_a и N_b – случайные числа, выбранные A и B соответственно;

K_{as} и K_{bs} – секретные симметричные ключи, предварительно распределенные участниками A и B соответственно с KDC;
E – симметричное шифрование;
K – сеансовый ключ.

Данный протокол наряду с распределением ключей реализует аутентификацию пользователей A и B при помощи четвертого и пятого сообщения. Основная уязвимость данного протокола заключается в следующем, что B не может узнать, что k до этого не использовался, а это критично при компрометации k. В данном случае злоумышленник имеет возможность распределить с B скомпрометированный секретный ключ.

Протокол Otway-Rees построен с использованием KDC и путем передачи четырех сообщений для распределения сеансового ключа и обеспечения аутентификации передаваемых сообщений, содержащих ключи. Однако наряду с предоставлением гарантий уникальности ключа описываемый протокол не обеспечивает аутентификации участников и подтверждения приема ключа. Протокол имеет следующую структуру:

$A \rightarrow S$: M, A, B, E_{Kas} (N_a, M, A, B)
$B \rightarrow S$: M, A, B, E_{Kbs}(N_a, M, A, B), E_{Kbs}(N_b, M, A, B)
$B \leftarrow S$: E_{Kas}(N_a, k), E_{Kbs}(N_b, k)
$A \leftarrow B$: E_{Kas}(N_a – 1, k)

Все обозначения здесь совпадают с обозначениями из предыдущего протокола, а M – случайное число, служащее идентификатором обмена. Если проверка всех передаваемых сообщений прошла успешно, то гарантируется, что k является уникальным. Данный протокол может быть расширен с помощью добавления пятого сообщения, при этом четвертое сообщение может быть дополнено значениями, демонстрирующими знание B относительно случайных чисел N_a и N_b (например, $E_K(N_a, N_b)$), тогда пятое сообщение будет иметь следующий вид: $A \rightarrow B$: $E_k(N_b)$.

Соглашения о ключах, основанные на симметричных алгоритмах

В данном разделе речь пойдет о ключевых соглашениях, построенных на основе использования симметричных алгоритмов шифрования и схем предварительного распределения ключей, которые отчасти похожи на протоколы Диффи-Хэлмана с фиксированной экспонентой.

Системой распределения ключей будем называть метод, с помощью которого в процессе инициализации системы доверенный сервер либо служба создает и распределяет секретные данные, которые в дальнейшем могут быть использованы для генерации ключей каждой пары абонентов.

Подобная система распределения ключей для фиксированного ключа парной связи будет выступать в роли схемы предварительного распределения. Примером построения системы с распределением ключей может служить следующая простая схема: доверенная служба выбирает для каждой пары пользователей (будем считать, что всего пользователей n) отдельный ключ и для каждого пользователя высылает безопасным способом n – 1 ключей, пронумерованных соответствующим способом. Это может быть строка матрицы размером n×n (столбцы и строки матрицы обозначены номерами абонентов сети секретной связи и на пересечении строки i и столбца j находится ключ парной связи i-го абонента с j-м), в которой количество строк совпадает с количеством абонентов в сети. Такой способ обеспечивает совершенную безопасность (в теоретико-информационном смысле), но практическое применение данного метода связано с необходимостью организационно-технического обеспечения хранения и использования носителей информации, на которых записаны данные строки. В качестве таких носителей могут использоваться гибкие магнитные диски, смарт-карты, touch memory и т.д.

В качестве примера рассмотрим схему Блома (Blom), с помощью которой может быть распределено j ключей, где $j < n-2$ (n – количество абонентов в сети). Данная схема не является интерактивной, пользователю нужно только получить свой номер в сети i ($1 \leqslant i \leqslant n$), который будет идентифицировать используемый данным абонентом ключ. В распределенных системах номера абонентов могут состоять, например, из номера сети, номера абонента в сети и серии ключей. Далее каждый пользователь получает безопасным способом нужную информацию, на основе которой он в дальнейшем вырабатывает ключи парной связи.

Результатом работы схемы для каждой пары абонентов A_i и A_j будет секретный ключ парной связи $K_{i,j}$ длиной m бит. Протокол начинается с того, что опубликовывается генератор матрицы G размером n×k, которая содержит элементы конечного поля F_q в виде MDS-кодов. Далее доверенная служба S создает секретную симметричную матрицу размером k×k над полем F_q. S передает каждому пользователя A_i секретный ключ S_i, являющийся i-ой строкой матрицы $S = (DG)^S$. Пользователи A_i и A_j вычисляют общий ключ $K_{i,j} = K_{j,i}$ длиной m бит, где $m = \lg(q)$. Используя S_i и столбец j из G, A_i вычисляет для данной пары пользователей симметричную матрицу $K = (DG)^S G$; применяя S_j и столбец i из G, A_j вычисляет точно такую же матрицу K.

Подобные схемы используются в небольших корпоративных сетях, где есть возможность оперативного предварительного обмена секретной

ключевой информацией. Соответственно в глобальных сетях передачи данных эти схемы неприменимы.

Ключевой транспорт, основанный на асимметричном шифровании

Ключевой транспорт, основанный на асимметричном шифровании, позволяет одной стороне, выбравшей симметричный ключ, передать его второй стороне, используя зашифрование на открытом ключе. В рамках данного подхода реализуется также аутентификация инициатора (на основе расшифровывания на секретном ключе), но для него не обеспечивается аутентификация и подтверждение приема ключа. Дополнительные гарантии безопасности могут быть получены при помощи дополнительных сообщений, ЭЦП и симметричного шифрования в дополнение к ЭЦП. Аутентификация в этих схемах может быть реализована как с использованием ЭЦП, так и без таковой и обеспечивает:

- аутентификацию участников при помощи шифрования на открытых ключах;
- аутентификацию источника сообщений при помощи цифровой подписи.

Ключевой транспорт, основанный на асимметричных алгоритмах и не использующий ЭЦП

Рассматривая схемы распределения ключей данного типа, начнем с протокола Нидхама и Шродера, основанного на использовании открытых ключей. В данном протоколе кроме взаимного распределения симметричных ключей (результирующий ключ получается путем вычисления односторонней функции от тех секретных данных, которые были распределены в ходе работы протокола) обеспечивается взаимная аутентификация участников и аутентификация ключей. До начала работы протокола должна быть реализована структура открытых ключей, которые должны быть доступны надлежащим образом участникам распределения ключевой информации (построение архитектур с открытыми ключами будет рассмотрено позже). Таким образом, предварительно требуется распределить только открытые ключи участников. В ходе протокола выполняются следующие шаги:

1. Участник A выбирает свою часть секретного ключа k_1 и посылает участнику B следующее сообщение: AfB: $P_B(k_1, A)$, где P_B обозначает зашифрование на открытом ключе участника B.
2. Участник B расшифровывает полученное сообщение, вырабатывает свою часть сеансового ключа k_2 и отправляет участнику A следующее сообщение: AaB: $P_A(k_1, k_2)$, где k_1 используется для аутентификации участника B.

3. После расшифрования полученного сообщения участник А проверяет значение k_1 на предмет совпадения с отправленным в первом сообщении и посылает следующее сообщение: A→B: $P_B(k_2)$, которое служит для аутентификации участника А.
4. После расшифрования полученного сообщения В проверяет значение k_2. По окончании протокола А и В могут выработать общий сеансовый ключ при помощи $f(k_1, k_2)$.

Данный протокол может быть модифицирован добавлением случайных чисел r_1 и r_2, созданных участниками А и В соответственно. В этом случае протокол будет иметь вид:

A←B: $P_B(k_1, A, r_1)$
A→B: $P_A(k_2, r_1, r_2)$
A→B: r_2

Ключевой транспорт, совмещающий использование асимметричного шифрования с ЭЦП

На практике, кроме секретности передаваемой ключевой информации, обычно требуется обеспечить аутентификацию источника, в таких случаях совместно используется асимметричное шифрования и ЭЦП. Точно так же, как и для шифрования на асимметричных алгоритмах, использование ЭЦП требует реализации процедур предварительного распределения открытых ключей проверки подписи. Существует несколько разновидностей совместного использования зашифрования на открытом ключе и генерации ЭЦП, а именно:

- сначала ключ подписывается, затем он зашифровывается на открытом ключе;
- сначала ключ подписывается, затем зашифровывается неподписанный ключ на открытом ключе (данный подход является безопасным только при определенных обстоятельствах, о которых будет сказано позже);
- сначала ключ зашифровывается на открытом ключе, затем подписывается зашифрованный ключ (о уязвимостях данного подхода уже говорилось);
- использование симметричных алгоритмов в добавление к асимметричному шифрованию и ЭЦП (данный подход будет обсуждаться в рамках так называемых *гибридных схем* распределения ключей).

Далее в добавление к уже применяемым символам будем использовать следующие обозначения:

- $S_A(y)$ – генерация подписи на секретном ключе участника А;
- $P_B(y)$ – зашифрование на открытом ключе участника В;
- K – сеансовый ключ.

Зашифрование подписанного ключа

$A \rightarrow B$: $P_B(k, t_A, S_A(B, k, t_A))$

Использовать метку времени (t_A) необязательно, ее присутствие необходимо только для обеспечения своевременности передаваемой информации. Негативные моменты в применении данного подхода заключаются в следующем: если данные имеют большую длину, зашифрование на открытом ключе будет неэффективно с вычислительной точки зрения.

Независимое зашифрование и генерация подписи

Этот подход применяется в случае, если ЭЦП не позволяет злоумышленнику получить информацию об открытых данных, например, если генерация подписи применяется к хэш-коду открытых данных. Тем не менее криптоаналитик, вероятно, сможет получить открытую информацию (или часть открытой информации) на основе подписи (даже в случае использования хэш-функции). В общем случае данный подход может быть реализован передачей следующего сообщения:

$A \rightarrow B$: $P_B(k, t_A)$, $S_A(B, k, t_A)$

Подписание зашифрованного ключа

$A \rightarrow B$: t_A, $P_B(A, k)$, $S_A(B, t_A, P_B(A, k))$

Зашифрование параметра A обеспечивает аутентификацию отправителя данного сообщения. В случае его отсутствия злоумышленник сможет от чужого имени послать данное сообщение B. Подобный подход, несмотря на некоторые уязвимости, получил широкое практическое распространение в связи с его использованием в рекомендациях X.509 (раздел строгой аутентификации).

Эти рекомендации принадлежат к классу аутентификационного распределения ключей. Как уже говорилось, строгая аутентификация подразделяется на несколько типов: с передачей одного сообщения, с передачей двух сообщений и с передачей трех сообщений. Наиболее распространенной является аутентификация с обменом двумя сообщениями. В ходе работы данного варианта X.509 протокола обеспечивается:

- идентификация A и гарантия того, что принятые B данные были действительно созданы;
- гарантия того, что полученные B данные были действительно предназначены для него;
- гарантия того, что принятые данные переданы в течение текущего сеанса обмена информацией и ранее не передавались;
- взаимное согласование и распределение выбранного ключа.

Перед началом работы протокола каждая сторона должна предварительно выполнить процедуры выработки и распределения открытого и секретного ключа для шифрования и ЭЦП, а также процедуры сертификации открытых ключей. Протокол состоит из следующих шагов:

1. Участник А вырабатывает метку времени t_A, случайное число r_A, зашифрованную на открытом ключе участника В часть сеансового ключа k_1, а также дополнительно может выбрать данные для проведения аутентификации источника сообщения $data_1$. После чего оправляется следующее сообщение: $A \rightarrow B$: $cert_A$, D_A, $S_A(D_A)$, где $cert_A$ – сертификат открытых ключей А для зашифрования и генерации подписи, $D_A = (t_A, r_A, B, data_1, P_B(k_1))$.
2. Участник В производит проверку $cert_A$ (подпись центра сертификации, содержащиеся данные, дата и т.д.), из которого, в случае успешного прохождения проверок, выбирается открытый ключ для проверки подписи. В случае ее успешной проверки идентифицируются остальные параметры в сообщении (метки времени, случайное число и идентификатор получателя). Если все проверки закончились успешно, участник В отправляет следующее сообщение: $A \leftarrow B$: $cert_B$, D_B, $S_B(D_B)$, где $D_B = (t_B, r_B, A, r_A, data_A, P_A(k_2)$.
3. Участник А производит аналогичные проверки, и в случае их успешного завершения считается, что аутентификация участника В прошла успешно.

Общий сеансовый ключ вырабатывается на основе секретных частей k_1 и k_2 с использованием однонаправленных функций. Представленный оригинальный вариант протокола аутентификационного распределения ключей использует для проверки подписи и зашифрования один и тот же открытый ключ. Как уже отмечалось, с точки зрения безопасности такой подход содержит ряд уязвимостей, однако данный протокол может быть модифицирован для использования раздельных открытых ключей. В этом случае для зашифрования и проверки подписи в пересылаемые сообщения добавляются два сертификата вместо одного.

Соглашения о ключах, основанные на асимметричных алгоритмах

Наиболее распространенным механизмом распределения ключей, относящимся к соглашению о ключах, является протокол неаутентификационного распределения ключей Диффи-Хэлмана. Базовый вариант этого протокола обеспечивал защиту от пассивного нарушителя, но имел проблемы с защитой от активного нарушителя. Протокол Диффи-Хэлмана служит для распределения ключей между двумя сторонами и не

требует в ходе своей работы наличия третьей стороны. Существенный недостаток данной системы – не обеспечивается аутентификация пользователей.

Чтобы протокол функционировал, необходимо выбрать и опубликовать следующие параметры: простое p и генератор a группы Z_p ($2 \leqslant a \leqslant p - 2$). Работа протокола состоит из следующих этапов:

1. Участник A выбирает случайное число x ($1 \leqslant x \leqslant p - 2$), хранящееся в секрете, и посылает участнику B следующее сообщение: A→B: $a^x \bmod p$.
2. Участник B выбирает случайное число y ($1 \leqslant y \leqslant p - 2$), хранящееся в секрете, и посылает участнику A следующее сообщение: A←B: $a^y \bmod p$.
3. Участник A получает a^x и вычисляет сеансовый ключ: $K = (a^x)^y \bmod p$.
4. Участник B получает a^y и вычисляет сеансовый ключ: $K = (a^y)^x \bmod p$.

Распределение ключей для конференц-связи

Протоколы распределения ключей для конференц-связи являются обобщением протоколов распределения ключей для двух сторон. При этом необходимо учитывать, что, несмотря на внешнюю схожесть, протоколы распределения ключей фундаментально отличаются от протоколов динамического распределения ключей между двумя сторонами. Основные требования к протоколам распределения ключей для конференц-связи сводятся к следующим пунктам:

- различные группы участников конференц-связи должны вырабатывать различные сеансовые ключи;
- сеансовые ключи должны распределятся динамически (за исключением схем с предварительным распределением);
- каждая сторона должна индивидуально вычислять сеансовый ключ.

Типичным примером применения протоколов конференц-связи является необходимость обеспечить зашифрование трафика при разговоре по телефону сразу нескольких участников. Очевидно, что первым шагом к распределению ключей при конференц-связи является разделение каждым из t ($t \geqslant 3$) участников конференц-связи знания своего симметричного ключа с доверенной третьей стороной. Далее третья сторона может создать сеансовый ключ и передать его каждому участнику конференц-связи, зашифровывая его на секретных ключах. Недостатком данного метода является то, что, во-первых, не для всех случаев практического применения конференц-связи имеется возможность использовать третью сторону, во-вторых, требования, предъявляемые к производительности третьей стороны, достаточно серьезны.

Наиболее интересной работой в этом направлении считается протокол распределения ключей для конференц-связи, предложенный Бурместером (Burmester) и Десмедом (Desmedt). В ходе работы протокола t участников ($U_0 \dots U_{t-1}$) вычисляют собственную экспоненту $z_i = \alpha^{ri} \bmod p$. Конференц-ключ в данном случае будет иметь вид: $K = \alpha^{r0r1+r1r2,\dots+rt-1r0} \bmod p$. Определяя $A_j = \alpha^{rjrj+1} \bmod p$ и $X_j = \alpha^{rj+1rj-rjrj-1} \bmod p$, получаем, что $A_j = A_{j-1}X_j$, и тогда K может быть представлен следующим образом: $K = A_0A_1 \dots A_{t-1} = A_{j-1}A_jA_{j+1} \dots$ $\dots A_{j+(t-2)} = A_{j-1}(A_{j-1}X_j)(A_{j-1}X_jX_{j+1}) \dots (A_{j-1}X_jX_{j+1} \dots X_{j+(t-2)})$.

Работа протокола начинается с того, что выбираются общие для системы параметры p и α и рассылаются всем участникам протокола. Каждый пользователь U_i выбирает случайное число r_i ($1 \leqslant r_i \leqslant p - 2$), вычисляет $z_i = \alpha^{ri} \bmod p$ и посылает z_i каждому из $t - 1$ участников протокола. Каждый участник протокола U_i, получая значения z_{i-1} и z_{i+1}, вычисляет $X_i = (z_{i+1}/z_{i-1})^{ri} \times$ $\times \bmod p$ и посылает значение X_i остальным участникам. После получения X_j ($1 \leqslant j \leqslant t$, исключая $j = i$) U_i вычисляет ключ $K = K_i = (z_{i-1})^{tri}X_i^{t-1}X_{i+1}^{t-2} \dots$ $\dots X_{i+(t-3)}^{2}X_{i+(t-2)} \bmod p$.

Очевидно, что при $t = 2$ данный протокол представляет собой оригинальный вариант протокола Диффи-Хэлмана, унаследовав при этом его уязвимость к атакам активного нарушителя. Безопасность протокола основана на вычислительной сложности нахождения дискретных логарифмов в конечных полях. В то же время доказательство абсолютной безопасности для протоколов этого типа представляет теоретический интерес, и на данном этапе строгое доказательство достаточно затруднено.

2.3.2. Управление ключевой информацией

Общие сведения

Практическое использование криптографических средств и методов защиты данных порождает помимо проблемы распределения ключевой информации широкий круг вопросов, связанных с хранением, использованием, уничтожением, депонированием и т.д. ключевой информации. Различные схемы организации защиты информационных ресурсов с использованием криптографических методов и средств требуют для своего надежного функционирования различных схем организации ключевой структуры. Поэтому в данной главе рассматриваются не только вышеперечисленные, но и другие вопросы, связанные с функционированием ключевых структур (например, виды сертификатов открытых ключей, доверенных сторон и т.д.).

Основу управления ключами составляет такое понятие, как ключевые взаимоотношения участников. В общем случае под управлением ключами будем понимать набор методов и/или средств, поддерживающих распределение и обработку ключевого материала (в качестве ключевого материала могут служить не только секретные или открытые ключи, а также инициализирующие векторы и дополнительные несекретные параметры) между авторизованными сторонами.

Управление ключами состоит из следующих процедур:

- инициализация ключевой структуры для пользователей в домене;
- создание, распределение и инсталляция ключевого материала;
- контроль над использованием ключевого материала;
- обновление, восстановление, уничтожение ключевого материала;
- хранение ключевого материала.

Основной целью управления ключами является поддержка ключевых взаимоотношений и ключевого материала таким образом, чтобы не произошло:

- компрометации секретных ключей;
- компрометации системы аутентификации секретной ключевой информации или открытых ключей;
- неавторизованного использования секретных или открытых ключей (например, необходимо обеспечивать невозможность использования ключей, срок действия которых истек).

Реализация системы управления ключами должна проводиться в соответствии с политикой безопасности, принятой в организации. В политику безопасности обычно включены следующие стороны, связанные с управлением ключами:

- использование наряду с техническими средствами управления ключами административных методов;
- ответственность и подотчетность каждой стороны, вовлеченной в управление ключами;
- аудит системы управления ключами.

Схемы организации распределения ключей

Существуют две основные модели построения схем распределения ключей, на основании которых в дальнейшем строятся более сложные схемы. Основными являются схемы типа «точка-точка» и централизованные схемы распределения ключей, которые в свою очередь делятся на схемы с использованием KDC и KTC (рис. 2.5).

Схемы типа «точка-точка» обеспечивают прямой обмен ключами между двумя сторонами.

Централизованные схемы с использованием KDC (KDC используется для распределения сеансовых ключей между двумя сторонами, которые предварительно должны разделить знание своих секретных ключей с KDC).

Централизованные схемы с использованием КТС. Основное отличие от предыдущего варианта заключается в том, что КТС обеспечивает перешифровывание полученной ключевой информации.

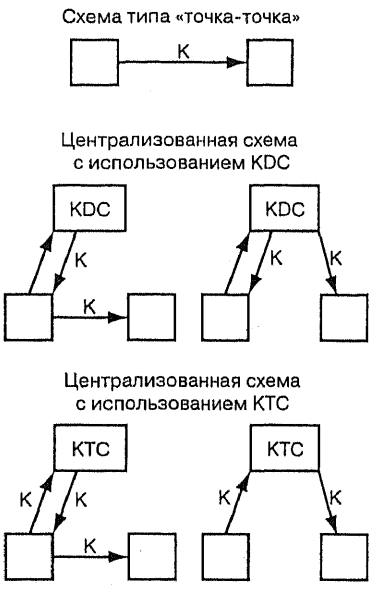

Рис. 2.5. Модели распределения ключей

Функции третьей стороны

В зависимости от того, каким образом третья сторона участвует в обмене ключевой информации между двумя сторонами, ее можно классифицировать следующим образом:

- inline: третья сторона (Т) находится непосредственно между сторонами ключевого обмена, то есть выполняет функции посредника в режиме реального времени;
- online: Т может участвовать в режиме реального времени в ходе ключевого обмена между сторонами, тем не менее стороны могут обмениваться друг с другом напрямую;
- offline: Т используется только для предварительного распределения ключевой информации, а обмен между сторонами в режиме реального времени происходит без участия Т.

Отдельно хотелось бы сказать о роли третьей стороны при использовании сертификатов открытых ключей (рис. 2.6):

- служба сертификации (CA) – обеспечивает распределение и аутентичность открытых ключей. В случае использования сертификатов открытых ключей вышеперечисленное обеспечивается за счет связывания имени владельца открытого ключа, самого открытого ключа и номера сертификата и при помощи подписания данной службой этой информации (на практике сертификаты несут в себе гораздо больше информации);

- служба имен – необходима для выделения пользователю системы уникального имени;
- служба регистрации – служит для авторизации доступа пользователей к службе сертификации и для связывания имени пользователя с его комплектом ключевой информации;
- генерация ключей – на данном этапе создается открытый и секретный ключ пользователя (дополнительно может создаваться симметричный ключ или пароль);
- директория сертификатов – содержит сертификаты всех зарегистрировавшихся пользователей, например в виде базы данных, которая доступна только для чтения.

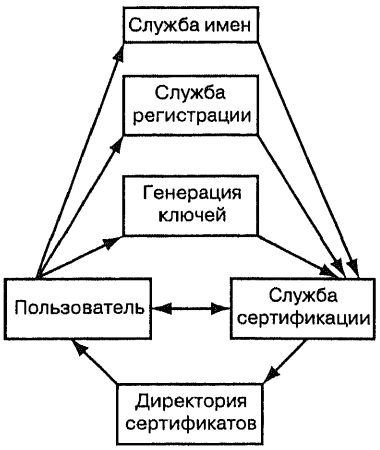

Рис. 2.6. Службы, обеспечиваемые третьей стороной при использовании сертификатов

Среди дополнительных функций третьей стороны можно выделить следующие:

- служба аутентификации пользователей;
- реализация механизмов управления ключами. В рамках CA может быть реализованы процедуры, обеспечивающие безопасное хранение ключевой информации, контроль времени жизни ключевой информации, а также процедуры аудита, связанные с функционированием CA.

К третьей стороне пользователями могут быть выдвинуты требования, определяющие уровень доверия к ней. В соответствии с данными требованиями можно выделить следующие уровни доверия к третьей стороне:

- третья сторона знает секретные ключи каждого пользователя;
- третья сторона не знает секретных ключей пользователей, но может создать аутентификационную информацию, используемую пользователем без обнаружения данного факта;
- третья сторона не знает секретных ключей пользователей и не может создать аутентификационную информацию, используемую пользователем без обнаружения данного факта.

Расширенными функциями третьей стороны являются:

- служба меток времени;
- служба нотариата. Используется для обеспечения невоспроизводимости передаваемой информации; например, может использоваться для проверки ЭЦП;

- служба депонирования ключей. В ее задачу входит обеспечение доступности секретных ключей пользователей третьей стороне. На практике данная служба может использоваться правоохранительными органами.

Иерархия и время жизни ключей

В зависимости от практического использования ключи могут быть следующих типов:

- *главные ключи* – находятся на самом высоком уровне иерархии; их безопасность обеспечивается только надежным хранением на ключевых носителях. Обычно эти ключи используются для зашифрования ключей, находящихся на следующем уровне иерархии;
- *ключи шифрования ключей* – секретные ключи, служащие для зашифрования других ключей, на которых производится непосредственно шифрование данных. Используя эти ключи, можно организовать хранение ключей шифрования данных, например на жестком диске компьютера, или распределить сеансовый ключ;
- *ключи данных* – служат непосредственно для защиты данных. Это могут быть как ключи генерации ЭЦП, так и ключи шифрования данных или выработки хэш-кода.

Используя на практике иерархическую структуру ключевой информации, можно оптимальным образом организовать хранение и применение ключей. Например, при использовании в качестве носителей ключевой информации touch memory, учитывая ограниченный объем внутренней памяти данного типа устройств, на нем можно хранить только главный ключ и/или ключи шифрования ключей, а ключи работы с данными могут быть расположены в зашифрованном виде на жестком диске.
Кроме того, ключи могут быть классифицированы по их *криптопериоду*. Криптопериод ключа – это время, в течение которого данный ключ используется легальным образом.

Понятие криптопериода было введено на практике для того, чтобы:

- ограничить объем информации, доступной криптоаналитику противника;
- ограничить объем незащищенных данных в случае компрометации ключа, на котором обеспечивается защита этих данных;
- ограничить использование существующих технологий по истечении времени использования;

- ограничить время, доступное противнику для проведения атак на основе долговременных вычислений (в приложениях, где не требуется долговременная защита).

В зависимости от криптопериода ключи подразделяются на:

- долговременные;
- краткосрочные (сеансовые, применительно к межсетевому обмену данными).

Краткосрочные ключи в основном используются для защиты информации при межсетевом обмене, в котором они могут применяться как для защиты целой сессии, так и для защиты отдельных пакетов с данными. Долгосрочные ключи необходимы для защиты данных при хранении либо для закрытия сеансовых ключей.

С точки зрения безопасности долговременные ключи должны использоваться для защиты небольшого количества данных (например, других ключей), в то время как краткосрочные ключи в рамках одной сессии могут использоваться для защиты большого количества данных.

Управление открытыми ключами

Протоколы, использующие криптографию с открытым ключом, обычно требуют, чтобы участники владели открытыми ключами друг друга, причем обмен должен происходить с обеспечением аутентификации данных ключей. Хотя, с одной стороны, открытые ключи и не являются конфиденциальными, злоумышленник может произвести подмену открытого ключа пользователя на свой открытый ключ, который будет опубликован от имени легального пользователя, и получится, что нарушитель сможет, например, иметь доступ к конфиденциальной информации, зашифрованной на открытом ключе данного пользователя. Поэтому при построении схем распределения и управления открытыми ключами следует запретить неавторизованное проведение процедур, связанных с генерацией, хранением и опубликованием открытых ключей. Семейство схем распределения ключей с обеспечением аутентификации, кроме использования экспоненциального распределения сеансовых ключей Диффи-Хэлмана, включает также следующие виды распределения открытых ключей:

- доставка типа «точка-точка» по доверенному каналу. Аутентичные открытые ключи пользователей приходят напрямую от пользователей с помощью прямых закрытых каналов (например, фельдъегерская

почта и т.д.). Подобный подход применим только для небольших закрытых организаций и при достаточно редком использования этого способа. Родственным является способ распределения ключей, использующий общедоступные каналы передачи данных с применением дополнительных средств защиты информации, причем для распределения открытых ключей критичным является требование целостности и достоверности передаваемого открытого ключа;

- прямой доступ к файлу открытых ключей. Открытые ключи каждого пользователя хранятся в общедоступном виде в базе данных, при этом обеспечивается гарантия аутентичности открытых ключей. Файл открытых ключей может храниться непосредственно у пользователя, который с определенной периодичностью будет инициировать его обновление;
- использование доверенного сервера в режиме online. Данный сервер обеспечивает передачу необходимых пользователям открытых ключей, подписанных с применением секретного ключа сервера, причем соответствующий открытый ключ сервера должен быть у пользователя для проверки полученного открытого ключа другого пользователя;
- применение доверенного сервера в режиме offline и сертификатов. В этом режиме каждый пользователь предварительно должен пройти процедуру сертификации открытых ключей на доверенном сервере. Пользователь, желающий получить открытый ключ другого пользователя, получает его вместе с сертификатом на открытый ключ;
- использование систем с неявной гарантией аутентичности общедоступных параметров. Такие системы обычно строятся с применением неявным образом сертифицированных открытых ключей, о которых будет сказано позже.

Одним из основополагающих понятий в управлении открытыми ключами является *дерево аутентификации*. Оно обеспечивает доступность открытых данных с одновременной проверкой аутентичности таковых. Эта проверка осуществляется при помощи связанной хэш-функции и аутентификации значения, находящегося в узле. Практическая реализация дерева аутентификации включает в себя:

- аутентификацию открытых ключей (в качестве альтернативы сертификатам открытых ключей). Дерево аутентификации создается доверенной стороной и содержит открытые ключи пользователей, позволяя аутентифицировать большое количество подобных ключей;

- доверенную службу меток времени. Создается в рамках служб, реализуемых деревом аутентификации;
- службу проверки параметров, ассоциированных с пользователем. Обычно сертификат открытого ключа включает в себя достаточно большое количество дополнительных параметров, которые при использовании сертификата должны проходить проверку.

В качестве примера приведем бинарное дерево, состоящее из узлов и ребер (рис. 2.7). Узлы бывают трех типов:

- корневой – имеет два ребра: правое и левое;
- внутренний – имеет три ребра: одно соединяет с вышележащим узлом, а два других – правое и левое;
- листья – соединяются только с вышележащим узлом.

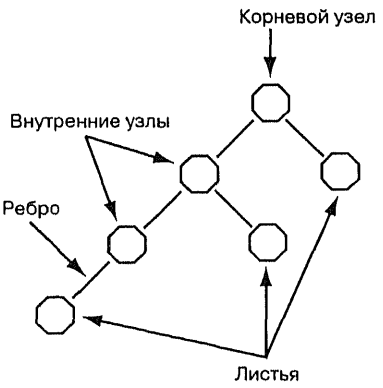

Рис. 2.7. Бинарное дерево

Корневой узел порождает либо внутренние узлы, либо листья, называемые детьми. Они, в свою очередь, порождают другие узлы, и тогда сами называются родителями.

Приведем пример построения дерева аутентификации на основе бинарного дерева. Пусть в системе имеется свободная от коллизий хэш-функция h и t общедоступных значений (например, открытые ключи) $Y_1 \ldots Y_t$. Тогда построение дерева аутентификации будет происходить следующим образом:

1. Обозначим каждое из t общедоступных значений как листья.
2. Ребро, выходящее из листа, обозначим как $h(Y_i)$.
3. Если левое и правое ребро внутреннего узла обозначены как h_1 и h_2, то ребро, соединяющее с вышележащим узлом, обозначим как $h(h_1 \| h_2)$.
4. Если ребра, подходящие к корневому узлу, обозначены как u_1 и u_2, то корневой узел обозначим как $h(u_1 \| u_2)$.

Общедоступное значение Y_1 (рис. 2.8) может быть аутентифицировано, если известны значения $h(Y_3)$, $h(Y_3)$, $h(Y_4)$ и производится последовательность вычислений, состоящая из:

- значения $h(Y_1)$;
- значения $h_1 = h(h(Y_1) \| h(Y_2))$;
- значения $h_2 = h(h_1 \| h(Y_3))$.

Завершающей фазой является сравнение $h(h_2 \| h(Y_4)) = R$ (R – общеизвестное значение, являющееся аутентичным). Если равенство выполняется, то считается, что аутентификация значения Y_1 произведена успешно. На практике процедура аутентификации открытых ключей производится следующим образом:

1. Участник B хочет проверить аутентичность открытого ключа Y_i, принадлежащего участнику A и расположенного в листе дерева аутентификации, при этом зная и доверяя только значению R.
2. Участник A предоставляет участнику B открытый ключ и все хэш-значения, лежащие на пути от данного листа до корневого узла.
3. Участник B проводит вышеперечисленные вычисления и приходит к заключению об аутентичности данного открытого ключа.

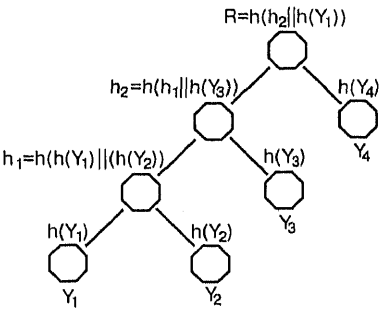

Рис. 2.8. Пример построения дерева аутентификации

Заметим, что если в бинарном дереве аутентификации содержится t листьев, то при достаточно большом значении t для проведения аутентификации необходимо пройти достаточно большое количество ребер в дереве. Параметр, обозначающий количество ребер на пути от листа до корневого узла, назовем *длиной пути*. Длина пути аутентификации может быть минимизирована при помощи соответствующего построения дерева аутентификации. Так, если дерево является *сбалансированным* (любые пути по которому от листа до корневого узла отличаются не больше, чем на единицу), то длина пути аутентификации будет приблизительно равна $\lg(t)$.

Недостатком данного метода является то, что в случае изменения или добавления значений листьев в дереве придется заново вычислять значение корневого узла, и при большом их количестве это может привести к большим вычислительным затратам. Решение этой проблемы приводит к различным способам построения деревьев аутентификации; в некоторых случаях производится добавление новых узлов в уже существующее дерево либо построение новых деревьев. Подробное описание моделей построения деревьев аутентификации выходит за рамки данной книги.

Сертификаты открытых ключей

Сертификаты открытых ключей играют основополагающую роль в современной криптографии с открытыми ключами. Они могут найти практическое применение в многочисленных системах электронной коммерции и разграничения доступа в Internet, без их использования не обходится ни одно средство защиты информации, рассчитанное на массовое употребление в глобальных сетях передачи данных. Основное назначение сертификата заключается в том, чтобы сделать доступным открытый ключ пользователя. Наиболее применяемым стандартом на сертификаты открытых ключей является CCITT X.509.

Сертификатом открытого ключа называется структура данных, состоящая из раздела данных и раздела подписи. Часть, в которой находятся данные, содержит открытую информацию, включающую, как минимум, открытый ключ и данные, идентифицирующие владельца ключа. Раздел подписи содержит ЭЦП, сгенерированную Центром сертификации на раздел данных с ключом, обеспечивая тем самым аутентификацию владельца ключа.

Центр сертификации (CA) является доверенной третьей стороной, обеспечивающей аутентификацию открытых ключей, содержащихся в сертификатах. CA имеет собственную пару ключей (открытый/секретный), где секретный используется для подписания сертификатов, а открытый публикуется и используется пользователями для проверки подлинности открытого ключа, содержащегося в сертификате. При этом обеспечение безопасности передачи открытого ключа CA может осуществляться не только криптографическими средствами защиты информации, например при помощи процедуры персонального обращения в CA, но и посредством сертификации открытого ключа другим CA.

Обычно сертификат содержит:

- период действия открытого ключа;
- номер и серию ключа;
- информацию о владельце ключа (например, Ф.И.О., адрес и т.д.);
- информацию о содержащемся в сертификате ключе (например, тип алгоритма, для которого предназначен данный ключ);
- информацию, использующуюся при проведении процедуры проверки ЭЦП (например, идентификатор алгоритма генерации ЭЦП);
- статус открытого ключа.

Создание сертификата открытого ключа начинается с создания пары ключей, при этом учитывается практическое применение данной пары

и некоторые другие факторы. Процедура генерации ключей может производиться двумя способами:

- CA создает пару ключей. Открытый ключ помещается в сертификат, а соответствующий ему секретный ключ отправляется пользователю с обеспечением аутентификации и конфиденциальности;
- пользователь сам создает пару ключей. При этом открытый ключ передается по защищенному каналу в CA.

При использовании второго случая CA может потребовать от пользователя демонстрации знания соответствующего секретного ключа. Например, в случае использования открытого ключа для ЭЦП пользователь должен будет подписать случайный запрос от CA, а центр, в свою очередь, используя открытый ключ пользователя, может проверить данную подпись.

Аннулирование сертификатов

В ходе функционирования схем с открытыми ключами может произойти компрометация секретного ключа, что приведет к необходимости аннулировать сертификаты с соответствующим открытым ключом. Если открытый ключ может быть получен в режиме реального времени при помощи доверенного сервера, находящегося в режиме online, то открытый ключ должен быть немедленно уничтожен. Ситуация осложняется в случае использования сертификатов открытых ключей, поскольку трудно удалить или аннулировать копии сертификатов, хранящиеся у большого числа пользователей. На практике существует много причин, по которым следует осуществлять процедуры аннулирования сертификатов открытых ключей, например статуса работника в организации, что приводит к необходимости изменения процедуры аутентификации для данного пользователя. Чтобы решить проблему аннулирования сертификатов открытых ключей, необходимо:

- ограничение срока действия сертификатов;
- ручное оповещение об аннулировании сертификата. Все пользователи системы информируются об аннулирование сертификата; данный способ действенен в небольших сетях;
- иметь общедоступный файл аннулированных ключей. Посредством помещения аннулированных сертификатов в общедоступный файл, при помощи обращения к которому пользователь может проверить действительность сертификата;

- иметь список аннулированных сертификатов (CRL). Использование CRL является одним из методов управления общедоступными файлами с аннулированными открытыми ключами;
- иметь сертификаты с реализованными функциями, позволяющими производить их аннулирование, например: используя в сертификате флаги аннулирования или указывая время аннулирования.

Механизмы контроля использования ключей

В данном разделе рассматриваются вопросы контроля применения ключей легальными пользователями. Их можно разделить на два разряда: разделение и ограничение в использовании ключей и контроль использования симметричных ключей.

Информация, которая может быть ассоциирована с криптографическим ключом, разделяется на *ограничивающую* использование ключа и на *используемую* для оперативных целей и обычно содержит следующие разделы:

- идентификатор владельца ключа;
- срок действия ключа;
- идентификатор ключа (номер ключа, серия ключа);
- информацию, связанную с характером использования ключа;
- спецификации алгоритма;
- система или окружение, в котором данный ключ может использоваться;
- имена участников процесса генерации, регистрации и сертификации данного ключа;
- контрольную сумму.

В простых системах управления ключами информация, ассоциированная с ключом, содержит только сведения, использующиеся для авторизации и зависящие от контекста применения. При этом следует учитывать, что эта информация, в свою очередь, может служить объектом для посягательств со стороны злоумышленника и в данном случае необходимо обеспечение гарантий неизменности информации для конкретного ключа. Достигается это за счет выработки контрольных сумм или генерации ЭЦП не только на ключ, но и на ассоциированную с ним информацию.

Смысл принципа разделения в использовании ключей в том, что ключ должен употребляться только для тех целей, которые были заложены в процессе авторизованного создания данного ключа. Ограничение использования ключей состоит в недоступности ключа для чтения неавторизованным пользователям либо в недоступности для неавторизованного использования. Реализация данных принципов решается как организационно, так

и технически (например, с помощью физически защищенных хранилищ ключей).

Существует множество причин, по которым необходимо избегать нецелевого использования ключей. Например, не следует применять ключ генерации ЭЦП для асимметричного расшифрования, поскольку ключи, применяемые для этого, будут иметь различные жизненные циклы, и создание их сопровождается употреблением специализированных для конкретных целей процедур проверки.

Механизмы контроля над использованием ключей можно разделить на три категории:

- с использованием масок ключей и изменяющихся ключей;
- на основе подтверждения подлинности ключей;
- с использованием контрольных векторов.

Маски ключей и изменяющиеся ключи

Маска ключа представляет собой битовый вектор или структуру, которая добавляется к ключу и зашифровывается вместе с ним, появляясь в открытом виде только в случае расшифрования ключа. Такая маска, кроме контроля, решает также проблему, связанную с криптоанализом ключа.

В случае изменяющихся ключей из основного ключа создаются производные от данного, например при помощи добавления некоторой несекретной информации или посредством некоторой функции. В идеальном случае эта функция должна обладать свойством необратимости.

Подтверждение подлинности ключей

Чтобы избежать возможной подмены ключей, некоторые системы требуют проведения строгих процедур, обеспечивающих гарантии подлинности ключей. Данный подход реализуется с помощью криптографических методов и средств, и при этом ключи, задающие криптографические преобразования, находятся у третьей доверенной стороны или у каждого из участников для обеспечения проверки ключей.

В качестве простого примера работы механизмов подтверждения подлинности ключей рассмотрим случай, когда одна из сторон или доверенный сервер должны обеспечить процедуру подтверждения того факта, что данный сеансовый ключ необходим для обмена между участником A и B. В этом случае, обладая предварительно распределенным ключом шифрования ключей K, сеансовый ключ S может быть зашифрован следующим образом: $E_{K \oplus (A \| B)}(S)$. Идентификаторы участников A и B используются при отождествлении того, что данный ключ должен использоваться только между A и B.

Контрольный вектор

Данный подход совмещает в себе два предыдущих. С каждым ключом S ассоциируется *контрольный вектор* C, который представляет себой поля данных, введенных пользователем, и использующиеся, например, совместно с ключом шифрования ключей для закрытия S ($E_{K \oplus C}(S)$). Использование C (например, введенная пользователем случайная строка) приводит к тому, что без знания данного вектора невозможно получить доступ к ключу S.

Междоменные отношения

Область, обслуживаемая одним доверенным сервером, обычно называется *доменом безопасности.* Причем доверенный сервер может выполнять как функции центра сертификации, так и дополнительные функции, связанные с обеспечением безопасности (о доменах безопасности будет сказано позже) и поддержкой выбранной политики безопасности.

В рамках одного домена безопасности пользователи могут устанавливать между собой защищенный канал передачи данных и производить аутентификацию друг друга при помощи разделения знания ключей или паролей с доверенным сервером домена. Таким образом, в рамках одного домена пользователи вполне способны поддерживать заданную политику безопасности.

Ситуация осложняется, когда два пользователя (A и B) хотят, например, установить защищенный канал передачи данных и при этом находятся в разных доменах (D_A и D_B), то есть обслуживаются разными доверенными серверами (T_A и T_B). Для этого необходимо, чтобы:

- пользователи разделяли знание симметричного ключа друг с другом;
- пользователи доверяли открытым ключам друг друга.

В случае, если пользователи уже выполняют одно из вышеперечисленных условий, данный вариант сводится к ситуации, когда пользователи находятся в одном домене. Но когда между пользователями нет прямых отношений, могут возникать такие ситуации:

- серверы T_A и T_B имеют доверенные отношения, выражающиеся в разделении знания ключей или в доверенных взаимоотношениях (A, T_A), (T_A, T_B) и (T_B, B);
- между серверами T_A и T_B не существует доверенных взаимоотношений. В этом случае серверы могут воспользоваться третьей стороной T_C, с которой у них существуют доверенные отношения. Если такой стороны не существует, то ситуация может быть представлена цепочкой доверенных взаимоотношений.

На рис. 2.9 показана общая схема организации междоменных отношений, при этом содержимое передаваемых сообщений может иметь различное значение в зависимости от того, на каких ключах (асимметричных или симметричных) построены взаимоотношения.

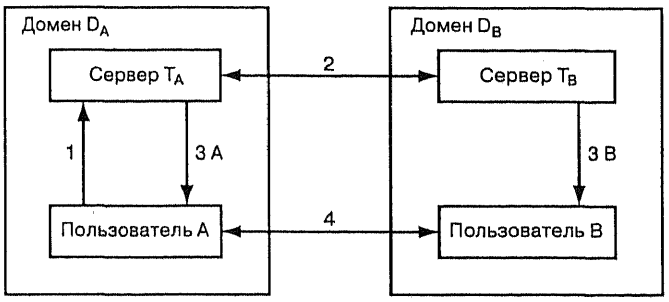

Рис. 2.9 Схема доверенных отношений между пользователями разных доменов

С использованием симметричных ключей

Если взаимоотношения серверов построены на основе схем распределения симметричных ключей, то последовательность передачи сообщений, представленная на рис. 2.10, будет иметь следующие значения:

1. Пользователь А передает запрос серверу T_A на выработку ключа с пользователем В.
2. T_A и T_B распределяют между собой симметричный сеансовый ключ K_{AB}.
3. T_A и T_B передают ключ K_{AB} пользователям А и В с обеспечением аутентификации и конфиденциальности.
4. Пользователи А и В устанавливают защищенный канал с одновременной аутентификацией друг друга.

С точки зрения пользователя А междоменное отношение для него носит прозрачный характер, то есть он взаимодействует с T_A таким образом, как будто этот сервер выполнял функции KDC или КТС.

С использованием асимметричных ключей

Выработка доверительных ключевых отношений с использованием открытых ключей может быть осуществлена на основе применения стандартных средств аутентификации источника сообщений, таких как ЭЦП или коды аутентификации сообщений. В данном случае последовательность сообщений, представленная на рис. 2.10, будет иметь следующий смысл:

1. А запрашивает от T_A открытый ключ пользователя В.
2. T_A запрашивает данный ключ от T_B, при этом должна быть обеспечена аутентификация открытого ключа.

3. T_A передает пользователю А открытый ключ пользователя В с обеспечением аутентификации.
4. Пользователь А, применяя данный ключ, устанавливает с В защищенный канал передачи данных.

Модели междоменных отношений с использованием нескольких центров сертификации

Существует несколько подходов (рис. 2.10) к организации доверенных отношений между центрами сертификации (CA). Данные приемы обычно носят название *модели доверия* или *топологии сертификации*. Доверенные отношения между CA определяют, каким образом сертификат одного CA может быть использован и проверен пользователем, находящимся в области действия другого центра.

Цепочки сертификатов и сертификационные пути

Цепочки сертификатов создаются в случае, если пользователь А, доверяющий одному CA (открытому ключу данного CA), желает проверить подлинность сертификата пользователя В, подписанного CA, которому пользователь А не доверяет. При этом между данными CA нет доверенных отношений. В этом случае находятся другие CA, которые имеют доверенные отношения с CA пользователей А и В. Далее устанавливается так называемый *путь сертификации*. Его суть состоит в том, что на открытые ключи одного центра создается сертификат доверяющего данному ключу CA, и так происходит по всему пути сертификации, то есть создается цепочка сертификатов.

Пусть пользователь А доверяет CA_5 (владеет открытым ключом P_5 данного CA) и желает проверить подлинность открытого ключа пользователя В, который подписан при помощи CA_3. Предположим, что существует путь аутентификации (CA_5, CA_4, CA_3), тогда обозначая $CA_5\{CA_4\}$ как сертификат открытого ключа CA_4, выданный CA_5, выведем следующую цепочку сертификатов ($CA_5\{CA_4\}$, $CA_4\{CA_3\}$). Пользователь А сначала проверяет сертификат $CA_5\{CA_4\}$, получая аутентичный открытый ключ P_4, принадлежащий CA_4, затем, используя P_4, проверяет сертификат $CA_4\{CA_3\}$ и получает аутентичный ключ P_3, принадлежащий CA. В итоге пользователь А может проверить сертификат CA_3, выданный на ключ P_B.

Изолированные друг от друга домены

В небольших сетях передачи данных может существовать один CA, но потребности глобальных сетей передачи данных в сертификации открытых ключей не могут быть удовлетворены одним центром, тогда используются

несколько СА. Если же СА являются изолированными друг от друга, то есть не имеют доверенных отношений, тогда в данной структуре организации сертификации в сети пользователь не сможет проверить аутентичность открытого ключа пользователя, находящегося в другом домене.

Модель доверительных отношений, построенная по принципу строгой иерархии

Решением проблемы криптографического взаимодействия между различными доменами является построение междоменных отношений по принципу строгой иерархии, в которой каждый пользователь изначально

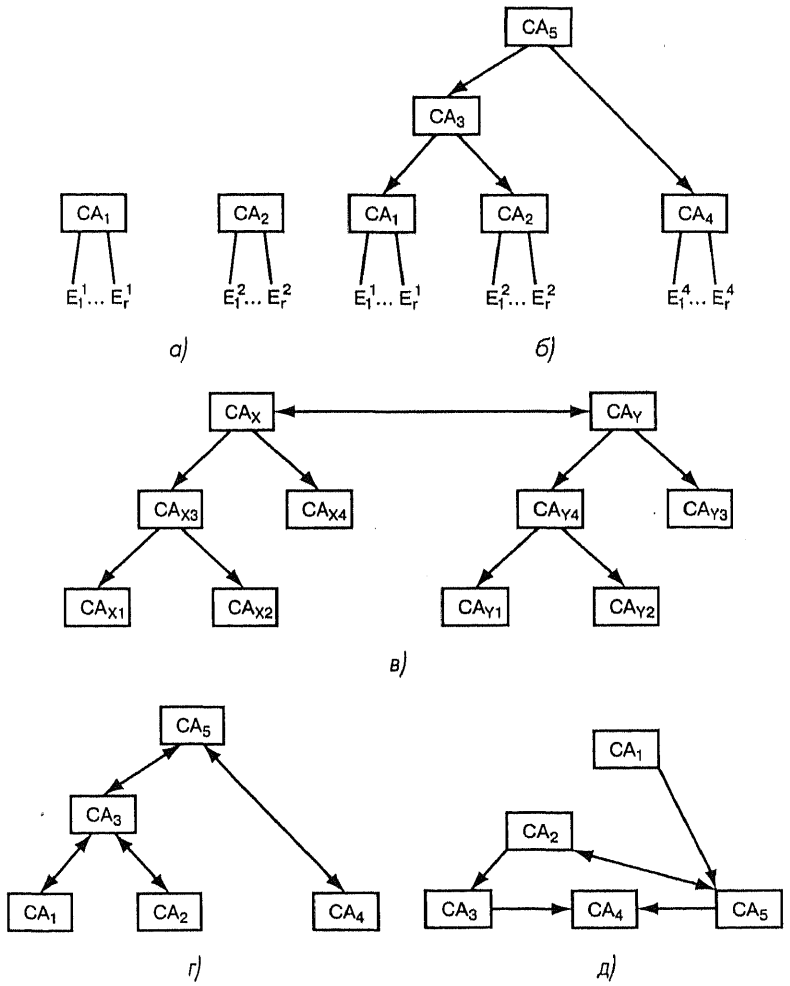

Рис. 2.10. Схемы организации междоменных отношений

имеет и доверяет только открытому ключу CA, находящемуся в корне дерева сертификации. Несмотря на тот факт, что сертификат открытого ключа пользователя E_1^1 подписывается CA_1, данный пользователь доверяет только CA_5, находящемуся в корне. Доверительные отношения между CA_1 и пользователем E_1^1 устанавливаются только через CA_5 и через лежащие между ними CA (в нашем случае – CA_3) на основе использования цепочки сертификатов. Недостатки данного метода заключаются в следующем:

- любые доверительные отношения в системе строятся через корневой узел;
- цепочка сертификатов создается даже в том случае, когда два пользователя зарегистрированы в одном CA;
- в случае глубокого дерева цепочка сертификатов будет иметь большую длину.

Модификацией данной схемы является объединение нескольких таких деревьев в структуру, поддерживающую сквозную сертификацию в обоих направлениях. Стрелка, соединяющая узел CA_X с CA_Y, обозначает сертификат открытого ключа CA_Y, созданный CA_X, что позволяет пользователям дерева с CA_X осуществлять процедуру доверенного получения сертификатов открытых ключей пользователей дерева CA_Y.

Реверсивные сертификаты и модель доверия на основе ориентированного графа

Модель с реверсивными сертификатами похожа по построению на модель со строгой иерархией, с той только разницей, что нижележащий CA может создавать сертификаты для вышележащего CA. В данной модели каждый пользователь изначально доверяет только открытому ключу того CA, в котором он зарегистрирован (то есть данный CA сертифицировал ключ данного пользователя). Любая цепочка сертификатов в этом случае будет начинаться с сертификата локального CA и заканчиваться сертификатом CA конечного пользователя. Таким образом, пользователи, находящиеся в одном домене, смогут напрямую взаимодействовать друг с другом через сертификат открытого ключа их общего CA. Недостатком этой схемы является то, что если пользователи разных доменов (например, CA_1 и CA_4) часто взаимодействуют друг с другом, то обработка достаточно длинной цепочки сертификатов будет привносить дополнительные трудности. Решить данную проблему можно, только используя прямые доверительные отношения между CA_1 и CA_4.

Модель доверительных отношений на основе ориентированного графа является примером распределенной модели доверительных отношений, а приведенные схемы являются моделями децентрализованных доверительных

отношений. В данной схеме не существует центрального узла, и сертификация открытого ключа пользователя одного домена в другом домене может носить сквозной характер.

2.4. Специфические криптографические протоколы

Рассмотренные выше криптографические протоколы составляют основу средств обеспечения безопасности в распределенных системах и решают большой круг задач, связанных с обеспечением информационной безопасности. Но существуют проблемы, решить которые можно с использованием специфических криптографических протоколов. Некоторые задачи носят сугубо теоретический характер, однако со временем они могут найти свое практическое применение. Так, например, задача генерации затемняющей подписи была применена в реализации электронных денег (e-cash).

В качестве примера можно привести задачу, в которой двум или более участникам, обладающим некоторым секретом, для достижения общей цели необходимо разделить друг с другом часть своей информации. Криптографические протоколы, решающие поставленную задачу, называются *протоколами с частичным разделением секрета.*

Здесь мы остановимся на рассмотрении следующих видов специфических криптографических протоколов:

- безопасные выборы;
- совместная подпись контракта;
- групповая подпись;
- доверенная подпись;
- неоспариваемая подпись;
- слепая подпись;
- забывающая передача;
- подбрасывание монеты по телефону;
- разделение знания секрета.

2.4.1. Безопасные выборы

Электронные платежные системы, получившие широкое распространение в последнее время, ставят перед их создателями не только проблемы обеспечения достоверности и целостности передаваемой платежной информации, но и проблему неотслеживаемости действий клиентов. Причем

неотслеживаемость нужно решать скорее техническими мерами, чем организационно-административными, поскольку клиент должен иметь возможность действовать анонимно: банкам или его партнерам нельзя идентифицировать действия клиента. Проблема неотслеживаемости актуальна в связи с применением Internet в повседневной жизни. Так, злоумышленник может оценить сферу интересов того или иного пользователя, производя отслеживание посещаемых данным пользователем серверов, конференций и т.д.

Теоретически данная проблема может быть сведена к проблеме реализации безопасных выборов, в ходе которых кроме конфеденциальности и достоверности передаваемых данных следует обеспечить:

- гарантию того, что только законные избиратели могут подать голос;
- невозможность в ходе голосования узнать выбор, сделанный тем или иным пользователем;
- условие, чтобы ни один пользователь не мог иметь больше одного голоса;
- право каждого избирателя убедиться в том, что его голос был правильно учтен при подведении итогов голосования.

Протокол безопасного голосования основывается на использовании двух доверенных сторон – агентства по проверке голосующего (i) на предмет разрешения участия в голосовании (T_1) и агентства для подведения итогов голосования (T_2). Перед проведением голосования T_1 должно послать T_2 список всех разрешенных идентификаторов голосующих. Протокол начинается с того, что каждый голосующий посылает T_1 некоторую идентифицирующую его информацию, после чего, если голосующему разрешено голосовать, то T_1 отправляет голосующему значение $E_1(i)$, являющееся идентификатором голосующего, и фиксирует факт обращения (участия в выборах). Далее голосующий вычисляет значения $E_2(i)$ (секретный идентификатор) и $E_3(i)$ (результат голосования) и посылает T_2 ($E_1(i)$, $E_2(i)$, $E_3(i)$). T_2 проверяет, существует ли $E_1(i)$ в списке разрешенных идентификаторов голосующих; если существует, то добавляет $E_2(i)$ к списку голосующих за $E_3(i)$. Преобразования E_1, E_2 и E_3 могут носить необратимый характер, либо здесь могут использоваться асимметричные алгоритмы. Приведенный протокол будет уязвим в случае, если T_1 и T_2 состоят в сговоре. Усиление данного протокола возможно в случае использования его вместе с протоколом забывающей передачи данных.

2.4.2. Совместная подпись контракта

Пусть два участника составления контракта хотят подписать его одновременно, вот почему должна быть обеспечена гарантия того, что одна сторона подпишет контракт, если это сделает и другая. Очевидным способом добиться этого является подписание контракта лицом к лицу, но в случае, если участники находятся на значительном расстоянии друг от друга, воспользоваться этим методом не представляется возможным. Простым протоколом совместной подписи контракта является обращение к арбитру, тогда протокол будет иметь следующий вид:

1. Участник A подписывает копию контракта и направляет ее арбитру.
2. Участник B подписывает свою копию контракта и направляет ее арбитру.
3. Арбитр направляет сообщения участникам A и B, информирующие их о том, что оба подписали контракт.
4. Участник A подписывает две копии контракта и посылает их участнику B.
5. Участник B подписывает обе копии контракта, одну из которых оставляет себе, а другую отправляет участнику A.
6. Участники A и B информируют арбитра о том, что они оба подписали контракт.
7. Арбитр уничтожает обе копии контракта, имеющие по одной подписи.

Данный протокол предотвращает мошенничество участников, связанное с отказом одного из них от факта подписания контракта. Если участник B отказывается от факта подписания контракта на шаге 5, то участник A может попросить апелляции у арбитра для получения копии контракта, подписанной участником B.

Следующий протокол не требует присутствия третьей стороны и использует в качестве своей основы DES, хотя здесь можно применить и какой-либо другой алгоритм блочного шифрования. Протокол имеет следующую структуру:

1. Оба участника (A и B) случайным образом выбирают 2n ключа для алгоритма DES, разбитые на пары.
2. Оба участника генерируют n пар сообщений (где L_i – левая часть пары сообщений, а R_i – правая часть пары сообщений) и зашифровывают их на парах ключей DES (левая часть пары сообщений зашифровывается на левой части пары ключей и т.д.).

3. Оба участника направляют друг другу свои зашифрованные сообщения, указывая при этом, какие сообщения половинами каких пар являются.
4. Участники посылают друг другу по одному ключу из каждой пары.
5. Участники расшифровывают те половины сообщений, которые зашифрованы на полученных половинах пар ключей, и убеждаются, что расшифрование прошло успешно.
6. Оба участника посылают друг другу первые биты всех 2n ключей DES, далее – вторые биты всех 2n ключей DES и т.д.
7. Участники расшифровывают оставшиеся части сообщений, и в случае успешного расшифрования считается, что контракт подписан.
8. Участники обмениваются секретными ключами и проверяют, что в ходе выполнения протокола не было мошенничества.

Таким образом, в конце протокола у обеих сторон имеется n пар подписанных сообщений, любое из которых может служить для подписи контракта. Хотя данный протокол имеет ряд уязвимостей, однако с его помощью можно наглядно показать общие подходы к реализации проблемы совместной подписи контракта.

2.4.3. Групповая подпись

Типичной сферой применения групповой подписи является следующий пример: «В офисе компании есть несколько компьютеров, подсоединенных к локальной сети. В каждом отделе есть свой сетевой принтер, и только один человек имеет право печатать на нем или давать разрешение на печать. Следовательно, перед печатью нужно проверить, работает ли данный сотрудник в этом отделе. В то же время компания держит в секрете имя этого пользователя. Если, однако, кто-то в конце дня обнаружит, что принтер печатает слишком много, у директора должна быть возможность найти того, кто использует принтер не по назначению, и послать ему чек».

Групповые подписи обладают следующими свойствами:

- только члены группы могут подписывать сообщения;
- получатель подписи может убедиться, что это правильная подпись группы;
- получатель подписи не может определить, кто именно из членов группы подписал документ;
- при споре подпись будет раскрыта для определения личности подписавшего.

Следующий протокол использует доверенного арбитра:

1. Арбитр создает большую группу пар *открытый ключ – закрытый ключ* и выдает каждому члену группы индивидуальный список уникальных закрытых ключей. Одинаковых ключей в списках нет. (Если в группе n членов, и каждый из них получает m пар ключей, то общее число пар ключей составит $n \times m$.).
2. Арбитр публикует список всех открытых ключей для группы в случайном порядке, сохраняя в секрете, какой ключ кому принадлежит.
3. Когда член группы хочет подписать документ, он случайным образом выбирает ключ из выданного ему списка.
4. Когда кто-то хочет убедиться, что подпись принадлежит члену данной группы, он перебирает главный список в поисках подходящего открытого ключа и проверяет подпись.
5. В случае возникновения споров необходимо обращаться к арбитру, который знает, какие ключи использует каждый член группы.

Проблема безопасности протокола состоит в том, что для него необходим надежный посредник. Арбитр знает закрытые ключи каждого из участников и имеет возможность подделывать подписи.

2.4.4. Доверенная подпись

На практике часто встречаются ситуации, когда кто-то хочет передать полномочия подписания документа другому человеку. При этом необходимо обеспечить:

- различимость. Любой желающий проверить данную подпись должен иметь возможность сразу определить, что данная подпись сделана по доверенности;
- неподделываемость. Сгенерировать действительную доверенную подпись сможет только человек, делегировавший полномочия и его доверенное лицо;
- отличительные характеристики доверенной подписи. Доверенное лицо, создающее подобную подпись, не должно иметь возможности выдать ее за настоящую;
- проверяемость. По доверенной подписи проверяющий может убедиться в том, что подписывающий согласен с содержанием документа;
- идентифицируемость. Доверенное лицо может по доверенной подписи определить подписывающего;
- неотрицаемость. Доверенное лицо не может опровергнуть сгенерированную им подпись.

Протоколы, реализующие доверенную подпись, существуют только в теоретическом плане.

2.4.5. Неоспариваемая подпись

Обычные цифровые подписи можно точно копировать. Иногда это необходимо, например при распространении публичных заявлений, но бывают ситуации, когда требуется, чтобы получатель не смог показать третьей стороне полученное сообщение без согласия лица, его подписавшего.

Неоспариваемость подписи удобна для решения подобных задач. Как и обычная цифровая подпись, неоспариваемая цифровая подпись зависит от подписанного документа и закрытого ключа человека, поставившего ее. Но, в отличие от обычных цифровых подписей, подпись такого вида не может быть проверена без разрешения подписавшего. Существующее название данного вида цифровых подписей объясняется тем, что, если одному из участников придется либо подтверждать, либо отрицать подпись (может быть, даже в суде), он не сможет отказаться от своей настоящей подписи. Несмотря на сложность математических приемов в основе неоспариваемой подписи, основная ее идея и порядок выполнения просты:

1. Участник А предъявляет участнику В подпись.
2. Участник В создает случайное число и посылает его участнику А.
3. Участник А выполняет вычисления, используя случайное число и свой закрытый ключ подписи, и посылает участнику В результат. Участник А может выполнить эти вычисления, только если подпись правильна.
4. Участник В проверяет данный факт.

Также существует дополнительный протокол, позволяющий участнику А доказать, что он не подписывал документ, и не допускающий возможности ложно отказаться от подписи.

У неоспариваемых подписей широкий круг применения, ведь во многих случаях участник того или иного обмена информацией может не согласиться на то, чтобы кто угодно имел возможность проверить его подпись. Например, он не хочет, чтобы подпись под его личной корреспонденцией могла быть проверена журналистами, чтобы его письма были опубликованы и подтверждены независимо от контекста или чтобы нельзя было обнаружить изменения в письмах, сделанные им позже. Когда участник подписывает информацию, которую продает, он вряд ли будет рад, если кто-то,

не заплатив за сведения, захочет подтвердить их достоверность. Защитить свои права участник может, контролируя тех, кто проверяет его подпись.

В ряде вариантов неотрицаемых подписей четко прослеживается связь между подписавшим и сообщением и между подписавшим и подписью. Есть схемы, в которых кто угодно может проверить, что автограф действительно принадлежит его автору, но для этого требуется согласие подписавшего.

Близким к понятию *неоспариваемая подпись* является *доверительная неотрицаемая подпись*. Доверительные неотрицаемые подписи похожи на обычные неотрицаемые подписи, за исключением протокола снятия подписи, который может быть запущен только арбитром. Лишь арбитр, а не участник может потребовать от подписывающего использовать протокол снятия подписи. И если арбитр представляет судебную систему, то он использует этот протокол только для разрешения формального спора.

2.4.6. Слепая подпись

Важным свойством протоколов цифровой подписи является знание подписывающим содержания подписываемого документа. Но на практике бывают ситуации, когда требуется, чтобы, например, нотариус подписал документ, не имея ни малейшего представления о его содержании. Нотариус не отвечает за содержание документа, он только заверяет, что нотариально засвидетельствовал его в определенное время. Происходит это следующим образом:

1. Участник А берет документ и умножает его на случайное число, которое называется *маскирующим множителем*.
2. Участник А посылает замаскированный документ участнику В, выполняющему функции нотариуса.
3. Участник В подписывает замаскированный документ.
4. Участник А удаляет маскирующий множитель, получая оригинальный документ, подписанный участником В.

Этот протокол работает только тогда, когда функция подписи и умножение коммутативны. Если это условие не выполнено, то можно использовать другие способы изменения документа.

Здесь возникает вопрос, может ли нотариус получить какую-нибудь информацию о том, что именно он подписывает, и тем самым нарушить непроницаемость документа? Если маскирующий множитель действительно случаен и делает случайным замаскированный документ, ответ будет отрицательным. Сгенерированная в ходе протокола подпись называется *полностью слепой подписью*.

Полностью слепые подписи обладают следующими свойствами:

- подпись участника B под документом правильна и служит доказательством того, что он подписал этот документ. Она убедит участника B в том, что он подписал этот документ, когда документ будет впоследствии предъявлен ему; при этом она также обладает всеми свойствами цифровых подписей;
- участник B не может связать подписанный документ с процессом подписания документа. Даже если у него хранятся записи обо всех сделанных им слепых подписях, он не сможет определить, когда он подписал конкретный документ.

На практике, конечно, высшей степенью безрассудства можно считать решение подписать документ, не имея понятия о его содержании. Однако и в этом случае существует способ, с помощью которого участник B может узнать, что именно написано в документе, и сохранить при этом полезные свойства полностью слепых подписей. Подобный протокол называется протоколом *генерации слепых подписей*.

Протокол слепой подписи работает аналогично описанному выше протоколу. Участник B получает большую пачку различных замаскированных документов. При этом он откроет, например, все, кроме одного, и затем подпишет последний. Вообразим, что оставшийся замаскированный документ как бы лежит в конверте. Ведь процесс маскировки документа вполне можно рассматривать как запечатывание его в конверт, а процесс удаления множителя маскировки – как вскрытие конверта. Когда документ спрятан, никто его не прочитает. Он подписывается с помощью кусочка копировальной бумаги, помещенной в конверт. В этом случае, когда подписывающий посредством копировальной бумаги ставит свою подпись на конверте, она отпечатывается и под документом.

Следующий протокол описывает идею слепых подписей:

1. Участник A готовит n документов, каждый из которых использует различный идентификатор.
2. Участник A маскирует каждый из документов различным маскирующим множителем.
3. Участник A отправляет n документов участнику B.
4. Участник B случайным образом выбирает n – 1 документ и просит участника A прислать маскирующий множитель для каждого из выбранных документов.
5. Участник A посылает участнику B соответствующие маскирующие множители.

6. Участник B открывает (то есть удаляет маскирующий множитель) n–1 документ и убеждается в том, что он корректен.
7. Участник B подписывает оставшийся документ и посылает его участнику A.
8. Теперь участник A удаляет маскирующий множитель и получает подписанный документ.

Этот протокол надежно защищен от мошенничества со стороны участника A. Теперь он должен точно угадать, какой документ участник B не будет проверять.

Существует прием, который уменьшает вероятность противоправного деяния со стороны участника A. На четвертом этапе участник B случайным образом выбирает n/2 документов для проверки. На пятом – участник A присылает ему соответствующие маскирующие множители. На седьмом этапе участник B перемножает все непроверенные документы и подписывает получившийся мегадокумент. На восьмом – участник A удаляет все маскировочные множители. Подпись участника B будет правильной, только если подписано произведение n/2 идентичных документов. Чтобы смошенничать, участнику A нужно точно угадать, какое подмножество документов будет проверять участник B. Вероятность этого гораздо ниже, чем вероятность угадать единственный документ, который участник B не проверял.

Протоколы данного типа имеют большое практическое значение в связи с развитием идеи электронных денег.

2.4.7. Забывающая передача

В этом случае основная идея состоит в том, чтобы участник A после передачи некоторого секрета участнику B не знал, передавал ли он секрет или нет, но при этом участник B точно знал, получил он секрет или не получил. Данная задача может быть решена с помощью частичного раскрытия секретов и доверенного арбитра. Однако в ряде случаев возникает ситуация, в которой воспользоваться услугами доверенной стороны не представляется возможным, поэтому особый интерес вызывают протоколы забывающей передачи, не применяющие доверенную сторону. Приведем в качестве примера протокол, основанный на том факте, что знание двух различных квадратных корней по модулю n из одного числа позволяет разложить n:

1. Участник B выбирает случайное число, вычисляет значение $x^2 \bmod n$ и посылает его участнику A.

2. Участник A, зная разложение n, вычисляет четыре квадратных корня из x^2 и сообщает один из них участнику B.
3. Участник B проверяет, будет ли полученное значение сравнимо с $\pm x$ по модулю n. Если да, то участник B не получает новой информации. В противном случае участник B имеет два различных квадратных корня из одного числа по mod n и, следовательно, может разложить n.

При этом у участника A нет способа узнать, какой из этих случаев имеет место, и вероятность для участника A выбрать правильное значение квадратного корня с точки зрения участника B равна 1/2.

На практике никто не использует протокол рассеянной передачи, однако он является важным звеном при построении других протоколов.

2.4.8. Подбрасывание монеты по телефону

В некоторых случаях необходимо, чтобы участники, расположенные далеко друг от друга, создали случайную последовательность и при этом не обращались за помощью к третьей стороне. Если участников двое, то действие протокола, решающего поставленную задачу, будет равносильно подбрасыванию монеты по телефону. На практике задача решается посредством однонаправленных функций:

1. Участник A выбирает случайное число x. Он вычисляет y = f(x), где f(x) – однонаправленная функция.
2. Участник A посылает y участнику B.
3. Участник B предполагает, что x четно или нечетно, и отправляет свое предположение участнику A.
4. Если предположение участника B правильно, результатом броска является «орел», если неправильно – «решка». Участник A объявляет результат и посылает x участнику B.
5. Участник B проверяет, что y = f(x).

Безопасность этого протокола обеспечивается однонаправленной функцией. Если участник A сможет найти такие x и x', когда x – четно, а x' – нечетно и y = f(x) = f(x'), он каждый раз сможет обманывать участника B. Кроме того, наименьший значащий бит f(x) должен быть некоррелирован с x. В противном случае участник B сможет обманывать участника A, по крайней мере изредка. Например, если f(x) в 75% случаев четна, у участника B будет преимущество. (Иногда наименьший значащий бит – не лучший выбор для использования в приложении, потому что его вычисление может оказаться слишком простым.)

Интересно отметить, что в протоколе участники A и B узнают о результате подбрасывания не одновременно. В протоколе есть момент, когда

участник A уже знает, какой стороной упала монета, но не может изменить ситуацию. Он, однако, в состоянии скрыть на какое-то время результат. Этот прием называется *броском монет в колодец*. Представьте себе высохший колодец. Участник A стоит рядом с колодцем, а участник B немного подальше. Участник B бросает монету, и она падает на дно колодца. Участник A может теперь заглянуть в колодец и увидеть результат, но не имеет возможности спуститься и изменить его. Участник B не сможет увидеть монету, пока участник A не позволит ему подойти и заглянуть в колодец.

2.4.9. Разделение знания секрета

Идея разделения секретов заключается в том, чтобы расчленить некоторый секрет на несколько частей, которые распределяются между участниками и затем коллективно используются для восстановления исходного секрета.

Следуют сказать, что существуют способы разделения сообщения на части. Каждая часть сама по себе не имеет смысла, но если их сложить, смысл сообщения полностью восстанавливается. Если это рецепт соуса и каждый работник знаком только с частью технологии его приготовления, то лишь собравшись вместе, сотрудники смогут приготовить соус. Если кто-нибудь из работников уволился и унес с собой свою часть рецепта, оставшаяся информация будет бесполезной.

Сообщение делится между двумя людьми по простейшей схеме. Вот протокол, используя который, арбитр имеет возможность разделить его между участником A и участником B:

1. Арбитр генерирует строку случайных битов R такой же длины, что и сообщение M.
2. Арбитр выполняет «исключающее ИЛИ» (XOR) над M и R, создавая S.
3. R A M = S.
4. Арбитр передает участнику A – R, а участнику B – S.
5. Чтобы получить сообщение, участникам A и B нужно выполнить единственное действие – операцию над имеющимися у них частями, восстанавливая сообщение R A S = M.

Этот метод при правильном исполнении абсолютно безопасен. Каждая часть в отдельности совершенно бессмысленна, что существенно: арбитр шифрует сообщение одноразовым блокнотом и дает шифротекст одному человеку, а блокнот – другому. Одноразовые блокноты, обладающие абсолютной безопасностью, обсуждаются в разделе 1.5. Никакие вычислительные средства не смогут восстановить сообщение только по одной его части.

Эту схему легко расширить до большего числа людей. Чтобы разделить сообщение между N-м количеством людей, выполните операцию XOR с большим числом строк случайных битов. В следующем примере арбитр делит сообщение на четыре части:

1. Генерирует три строки случайных битов R, S и T такой же длины, что и сообщение M.
2. Выполняет «исключающее ИЛИ» (XOR) над M и полученными тремя строками, создавая U:

   M A R A S A T = U.

3. Передает участнику A – R, участнику B – S, участнику C – T, а участнику D – U.
4. Участники A, B, C, и D могут восстановить сообщение.
   Они собираются вместе и вычисляют:

   R A S A T A U = M.

Арбитр обладает абсолютной властью и может делать все, что хочет. Он может раздать чепуху и утверждать, что это настоящие части секретной информации. Никто не в состоянии его проверить, пока, собравшись вместе, участники протокола не попробуют прочитать письмо. Он может выдать части секрета участникам A, B, C и D и позже заявить всем, что только участники A, C и D нужны для восстановления секрета, устранив при этом участника B. Но все это не является проблемой, так как делимый секрет принадлежит арбитру.

Тем не менее одна трудность в этом протоколе все-таки присутствует. Если любая из частей будет потеряна, а арбитра не будет поблизости, пропадет и все сообщение. Если участник C, обладая частью рецепта соуса, перейдет работать к конкуренту, оставив свою часть секрета у себя, значит, остальным не повезло. Конкурент не сможет восстановить рецепт, но этого не смогут сделать и участники A, B и D, собравшись вместе. Часть C также критична для восстановления сообщения, как и любая другая. Все, что известно участникам A, B и D, – длина сообщения, и ничего больше. Это истинно, так как у R, S, T, U и M одинаковая длина, следовательно, каждому из участников известна длина M. Помните, сообщение M делится не в обычном смысле этого слова, а подвергается операции XOR со случайными величинами.

Глава III

Компьютерная безопасность и практическое применение криптографии

3.1. Общие сведения

Описанные выше методы и средства криптографической защиты информации нашли широкое применение в практической деятельности человека. В постоянно расширяющейся области использования средств вычислительной техники и передачи данных появляются все новые и новые проблемы сохранения конфиденциальности и целостности информации, ограждения ее от посягательств злоумышленников, порой вооруженных самыми современными приемами и методами, которые способствуют нарушению тайны обмена сведениями.

Особую актуальность вопросы использования криптографических средств и методов защиты информации приобрели в связи с распространением общедоступных сетей передачи данных и повсеместного их применения в хозяйственной деятельности человека, например: Internet-banking (предоставление ряда банковских услуг через Internet), построение корпоративных виртуальных сетей передачи данных или IP-телефония. Понятно, что без сохранения информационной безопасности в этих сферах человеческой деятельности, без решения ряда практических вопросов использования криптографических алгоритмов дальнейшее развитие современных информационно-телекоммуникационных систем вряд ли возможно.

В свою очередь активное развитие криптографического инструментария тоже оказывает стимулирующее влияние на становление специфических областей человеческих знаний, например электронной коммерции, использующей последние достижения теоретической криптографии.

При построении отечественных информационно-телекоммуникационных систем применение современных технологий в совокупности с импортными

аппаратными и программными компонентами заставляет задумываться над некоторыми проблемными вопросами, связанными с обеспечением информационной безопасности. К ним относятся:

- импортные компоненты не могут считаться доверенными, что выдвигает специфические требования к применению и встраиванию средств защиты информации;
- использование разнородных и часто мало совместимых программно-аппаратных составляющих;
- повсеместное использование открытых сетей передачи данных, что с точки зрения угроз выводит на первое место методики проведения удаленных атак.

Построение информационных систем обычно идет по двум направлениям:

- использование стандартизованных и общепринятых идеологий, методов и средств в случае, если цели и требования, закладываемые в реализацию проекта, могут быть достигнуты путем унифицированных решений;
- применение при построении информационных систем частных решений, обеспечивающих специфические потребности (например, SWIFT, UEPS и др.).

В свою очередь средства защиты информации могут быть также унифицированными в зависимости от идеологии и архитектуры построения информационно-телекоммуникационной системы, создаваемой для решения конкретных задач. Это касается и архитектуры самой подсистемы защиты информации, учитывающей особенности построения защищаемой системы. Она может быть организована как с использованием стандартных решений, так и с применением средств защиты информации, созданных специально для решения частных задач.

С точки зрения практического применения средств защиты информации особый интерес представляют именно унифицированные решения. В этой главе им уделяется основное внимание, при этом в некоторых подразделах рассматриваются некоторые наиболее удачные, по мнению автора, реализации корпоративных решений.

Проблемы, связанные с обеспечением информационной безопасности в современных информационно-телекоммуникационных системах (имеющих как распределенный, так и локальный характер), обычно включают:

- защиту информационных ресурсов локальной рабочей станции от *несанкционированного доступа* (НСД). Реализуется применением

устройств типа *электронный замок*, средств абонентского шифрования, штатных средств разграничения доступа, входящих в состав программного обеспечения, установленного на рабочей станции. При этом вполне могут использоваться и дополнительные средства разграничения доступа;

- защиту в локальных сетях передачи данных;
- защиту межсетевого взаимодействия. Этот пункт, в свою очередь, может быть разбит на несколько частей:
  - создание *защищенных виртуальных сетей* (VPN) на базе общедоступных сетей передачи данных. В ходе решения подобной проблемы необходимо, кроме установления защищенного канала передачи данных между территориально разнесенными локальными сетями, обеспечить также защиту от несанкционированного доступа к информационным ресурсам, предоставляемым в данных сетях;
  - обеспечение защиты технологии клиент/сервер. В зависимости от практических потребностей методы создания защиты в таких технологиях могут подразделяться на два типа: а) обеспечение защищенного взаимодействия между клиентом и сервером; б) обеспечение авторизованного доступа клиента к ресурсам, предоставляемым целевым сервером. В качестве примера, описанного в этой главе, выбраны технологии защиты при доступе к Web-серверам;
  - обеспечение защиты информационных ресурсов корпоративной сети, имеющей выход в общедоступные сети передачи данных, от атак извне;
  - средства защиты пользовательского взаимодействия. В качестве примера рассматриваются средства, обеспечивающие шифрование данных при обмене информацией типа «точка-точка»;
- защиту электронной почты и документооборота;
- защиту электронных платежных систем (в том числе и систем, осуществляющих платежные операции через Internet).

В зависимости от практических потребностей решения проблем по обеспечению информационной безопасности, включенных в приведенную классификацию, могут осуществляться по отдельности, а также совместно с вопросами, изложенными в других пунктах. Их реальное воплощение в рамках выбранной в организации политики безопасности может выражаться как в виде отдельных программно-аппаратных решений, так и в объединении организационно-административных мер и программно-технических средств. Состав средств и методов защиты информации для конкретной информационно-телекоммуникационной системы определяется в соответствии с заданной политикой безопасности.

Необходимость обеспечения надежности и оперативности функционирования средств защиты, использующих криптографические алгоритмы, выдвигает особые требования и к выбору ключевых систем, которые должны удовлетворять следующим требованиям:

- устойчивость ключевой системы к компрометации ключей. Это выражается в том, что компрометация ключа у одного пользователя не должна сказаться на работе всей системы в целом;
- у пользователей должно находиться минимальное число ключей, и защищать их необходимо с помощью особых организационных мер, предпочтение при этом отдается техническим мерам защиты ключей;
- предусматривать защиту от копирования;
- иметь механизмы плановой смены ключей и сертификатов открытых ключей.

Здесь следует добавить, что для современных информационно-телекоммуникационных систем, использующих межсетевое взаимодействие, на практике необходимо обеспечивать не только конфиденциальность, целостность и достоверность информации, но и ее доступность, а также доступность информационных ресурсов. Другими словами, необходимо предусмотреть защиту от атак типа *отказ в обслуживании*.

Эффективность применения систем защиты информации зависит от того, учитываются ли при их построении следующие требования:

- масштабируемость. Другими словами, при построении системы защиты должна быть обеспечена возможность ее дальнейшего расширения с учетом роста защищаемой инфраструктуры;
- интегрируемость. Средства защиты информации должны оптимальным образом встраиваться в существующую инфраструктуру;
- контролируемость. События, связанные с функционированием системы защиты информации, должны отражаться системами мониторинга и аудита;
- структурированность. В рамках построения системы защиты в ней должны быть выделены функциональные подсистемы, выполняющие отдельные задачи по обеспечению информационной безопасности, например подсистема разграничения доступа или подсистема аудита;
- сертифицируемость. Применяемые при построении системы защиты средства защиты информации должны удовлетворять национальным стандартам и требованиям;
- эшелонированность. Надежная система защиты должна состоят из нескольких рубежей, на каждый из которых возлагаются определенные задачи и выдвигаются определенные требования. Суть эшелонированной

защиты состоит в том, что нарушение штатного функционирования одного из рубежей не должно повлечь катастрофических последствий для всей системы безопасности в целом.

Особо следует отметить требования, выдвигаемые к быстродействию средств защиты, которые должны быть обеспечены современными высокоскоростными методами передачи и обработки информации:

- механизмы обеспечения безопасности должны эффективно обрабатывать мультиплексированные данные на уровне пакетов;
- средства безопасности не должны вносить существенные задержки в процесс обработки и передачи информации:
- ключевая инфраструктура в силу высоких скоростей обмена информацией и, следовательно, большого объема передаваемой информации в единицу времени должна содержать эффективные средства обновления ключей (либо стоит позаботиться об увеличении срока действия ключей);
- криптографические алгоритмы должны обрабатывать большие объемы информации в единицу времени. Например, для некоторых систем скорость обработки информации должна составлять несколько гигабитов в секунду;
- реализация криптографических алгоритмов должна позволять работу в системах с различной пропускной способностью.

На практике уровень безопасности и надежности системы защиты информации зависит не только от стойкости выбранных средств или политики безопасности, но и от эффективности интеграции средств защиты информации в целевую систему. Очевидно, что схемы интеграции будут значительно отличаться в зависимости от реализации средств защиты, которые обычно подразделяются на программные, аппаратные и аппаратно-программные.

Интеграция аппаратных средств заключается в разработке и реализации физических процедур сопряжения подобных средств защиты информации в целевую систему. На практике обычно используются стандартные процедуры и интерфейсы подключения устройств к рабочей станции, например RS-232, PCI-слот и т.д. Применение аппаратных устройств с точки зрения эффективности реализации оказывается предпочтительней, поскольку минимизируется негативное влияние среды, в которую интегрируется данное устройство. (Целевая система может носить недоверенный характер, то есть допускать наличие недокументированных возможностей, оказывающих негативное влияние на работу средств защиты информации.)

В случае применения программных средств критичным становится вопрос о проверке среды (обычно под средой подразумевается операционная система) на наличие недокументированных возможностей системы и вариантах построения доверенной программной среды (отдельные аспекты этой проблемы будут рассмотрены далее). Программные средства защиты информации могут быть реализованы или в виде законченного программного продукта, интеграция которого заключается в обычной его инсталляции, или в виде дополнительных процедур модулей, встраивающихся в программное обеспечение целевой системы.

Второй случай наиболее распространен, в частности, если необходимо обеспечить гибкость работы системы или прозрачное выполнение функций защиты информации при работе с каким-либо программным продуктом. В качестве примера можно привести случай, когда при передаче данных по электронной почте необходимо обеспечить их шифрование на прикладном уровне. При этом можно, конечно, применить предварительное зашифрование подклеиваемого к письму файла, но тогда тело письма останется в открытом виде; можно также воспользоваться услугами защищенной почтовой системы с уже реализованными функциями шифрования данных.

Встраивание программных средств защиты информации может осуществляться одним из следующих способов:

- с использованием программных интерфейсов;
- с использованием криптосервера.

Существуют также некоторые специфические способы интеграции программных средств защиты информации в целевые системы, например интеграция в операционную систему (ОС) Windows NT криптопротокола Kerberos через архитектуру DCE.

Очевидно, что основная проблема встраивания заключается в корректном использовании вызываемых функций. Субъекты, связанные с выполнением подобных операций, могут использовать некоторое общее подмножество криптографических функций логического преобразования объектов (в частности, алгоритмы контроля целостности объектов). При проектировании конечной системы защиты исторически сложившийся подход к использованию общего ресурса связан с применением разделяемых субъектов, выполняющих общие для других субъектов функции.

Программным интерфейсом (далее – API) называется детальное описание функций и используемых ими параметров. Формат обращения к функциям, описанный в программном интерфейсе, поддерживает прикладное ПО.

Для того чтобы интегрировать защитный механизм в данное ПО, необходимо согласовать форматы функций (вызываемых и реализованных). Однако при таком подходе возникает следующая трудность: не всегда API предоставляет все необходимые форматы обращения для обеспечения надежной безопасности. Например, API может предоставлять возможность обращения к функциям работы с шифрованием и ЭЦП, но при этом возможность обращения к функциям распределения сеансовых ключей предоставлена не будет.

Наиболее известными на сегодняшний день API являются CryptoAPI, GSS API и SSPI. В ОС Windows NT 4.0 используется программный интерфейс Crypto API, дающий возможность шифрования (RC2, RC4, DES, RSA), выработки и проверки электронной подписи, однако в этой системе нет действующих программных средств с механизмами криптографической защиты. Но необходимо отметить, что алгоритмы шифрования, применяемые в Crypto API, используют укороченные ключи.

В рамках данного подхода необходимо реализовывать процедуры проверки прикладным ПО программных модулей (например, библиотек dll), в которых реализован механизм защиты, на предмет того, что это именно те программные модули, которые предполагалось использовать. Реализовать этот процесс можно, используя механизм запросов и ответов между прикладным ПО и программными модулями, выполняющими защитные функции, или проводя контроль целостности ПО средствами защиты перед тем, как начать с ним работать. Необходимость выполнения этого требования основывается на следующем: злоумышленник может подменить программные модули средства защиты и пользователь не сможет обнаружить, что его открытая информация, которая должна была быть зашифрована, уйдет в канал связи незашифрованной.

Следующий подход с использованием криптографического сервера заключается в локализации средства защиты информации, работающего с критической к компрометации информацией (секретные ключи, пароли и т.д.) на особо выделенной рабочей станции. В прикладное ПО, работающее на другой (находящейся на связи) рабочей станции, встраиваются только вызовы функций, реализованных в ПО криптосервера. Использование криптосервера позволяет выполняться средству защиты информации в доверенной программной среде.

Данный подход используется в случае, когда сложно гарантировать отсутствие негативного влияния прикладного ПО на функционирование средства защиты. Например, такой подход применяется к *мейнфреймам*. Взаимодействие же криптосервера с целевой станцией осуществляется

с использованием выделенного сегмента локальной сети; несанкционированный доступ к этому сегменту предотвращается организационными мерами.

В большинстве случаев непосредственное встраивание осуществляется при работе с исходными текстами прикладного ПО и программного обеспечения средств защиты и последующем совместном компилировании и линковании.

Использование аппаратно-программных средств защиты информации позволяет вынести некоторые, особенно критичные к негативному воздействию процедуры, осуществляемые в данном средстве защиты, для реализации в аппаратной части, что заметно повышает уровень защиты информации. Например, функции подсчета контрольных сумм защищаемых файлов реализуются в аппаратной части, где и хранятся выработанные значения этих сумм.

Средства защиты информации подразделяются в зависимости от уровня взаимодействия открытых систем (в качестве телекоммуникационной основы рассмотрим стек протоколов TCP/IP). Причем для каждого уровня существует свой набор механизмов и служб обеспечения безопасности и соответственно свои наборы угроз. При этом каждый уровень с точки зрения реализации средств безопасности имеет свои преимущества и недостатки.

Каждая подобная конфигурация также может быть охарактеризована набором специфических действий, которые способна предпринять атакующая систему сторона. Если рассматривать проблему в теоретическом плане, такие атаки проводятся с целью компрометации, изменения или уничтожения критичных ресурсов данного уровня. В соответствии с возможными угрозами каждый уровень должен быть защищен специальными средствами защиты, которые взаимно дополняют друг друга и в совокупности обеспечивают информационную безопасность системы в целом.

3.1.1. Физический и канальный уровни

На этих уровнях в качестве механизмов безопасности обычно осуществляется шифрование соединения или трафика (зашифровываться может как весь трафик, так и его выборочная часть), то есть обеспечивается конфиденциальность передаваемой информации. В качестве вероятных угроз здесь можно выделить следующие: а) несанкционированное подключение, б) ошибочная коммутация, в) прослушивание, г) перехват, д) фальсификация

информации, е) имитоатаки, ж) физическое уничтожение канала связи. Для защиты информации на данном уровне обычно применяют *скремблирование, шифрующие модемы, специализированные канальные адаптеры.* Из достоинств реализации средств защиты на уровнях этого типа следует отметить:

- простоту применения;
- аппаратную реализацию;
- полную защиту трафика;
- прозрачность выполнения средствами защиты информации своих функций.

К недостаткам применения средств защиты на канальном или физическом уровне следует отнести:

- негибкость решения (фиксированный тип канала, сложность адаптации к сетевой топологии, фиксированная производительность);
- низкая совместимость,
- высокая стоимость.

3.1.2. Сетевой уровень

При защите информации на сетевом уровне необходимо решить следующие задачи:

- защита информации непосредственно в пакетах, передаваемых по сети;
- защита трафика сети, то есть шифрование всей информации в канале;
- контроль доступа к ресурсам сети, то есть защита от нелегальных пользователей и от доступа легальных пользователей к информации, им не предназначенной.

Реализация средств защиты на сетевом уровне представляет гораздо больше возможностей для обеспечения безопасности передаваемых данных. Так, например, на данном уровне может быть реализована аутентификация рабочей станции, являющейся источником сообщений. Из угроз, специфичных для сетевого уровня, следует выделить:

- анализ служебной информации сетевого уровня, то есть адресной информации и топологии сети;
- атаки на систему маршрутизации. Другими словами, возможна модификация маршрутизационных таблиц через протоколы динамической маршрутизации и ICMP;
- фальсификацию IP-адресов; атаки на систему управления;

- прослушивание, перехват и фальсификацию информации;
- имитоатаки.

В качестве традиционно применяемых решений, способных устранить подобные угрозы, следует отметить:

- пакетную фильтрацию;
- административную защиту на маршрутизаторах (в том числе шифрующих);
- протоколы защиты информации сетевого уровня (защита и аутентификация трафика);
- туннелирование;
- векторизацию;
- динамическое распределение сетевых адресов;
- защиту топологии.

Здесь следует также перечислить достоинства средств защиты на сетевом уровне. К ним относятся:

- полнота контроля трафика;
- универсальность;
- прозрачность;
- совместимость;
- адаптивность к сетевой топологии.

К недостаткам средств защиты на сетевом уровне можно отнести только неполноту контролируемых событий (неподконтрольность транспортных и прикладных протоколов). Защита информации на этом уровне имеет ряд преимуществ с архитектурной точки зрения. Сетевой уровень – это та ступень организации, на которой сеть становится полносвязной системой. На более низких уровнях защита может быть реализована только как набор двухточечных защищенных звеньев.

Только на сетевом уровне появляется возможность установления защищенного соединения между двумя компьютерами, расположенными в произвольных точках сети; на этой ступени появляется и понятие топологии, можно различить внешние и внутренние каналы. При сетевом взаимодействии реализуются такие возможности защиты информации, как фильтрация трафика между внутренней (корпоративной) сетью и внешней коммуникационной средой, защита от несанкционированного доступа из внешней сети во внутреннюю, маскировка топологий внутренних сетей. С другой стороны, сетевой уровень все-таки является достаточно низким в архитектуре взаимодействия открытых систем, вот почему не оказывает существенного влияния на прикладные системы.

3.1.3. Транспортный уровень

Самыми опасными угрозами на этом уровне следует считать:

- несанкционированные соединения, разведку приложений;
- атаки на систему управления;
- прослушивание, перехват и фальсификацию информации;
- имитоатаки.

Традиционными решениями подобных проблем являются:

- защита в составе межсетевых экранов (контроль доступа к приложениям, управление доступом к серверам);
- proxy-системы;
- протоколы защиты транспортного уровня (защита и аутентификация данных).

Среди достоинств средств защиты на транспортном уровне можно выделить следующие:

- развитая функциональность;
- высокая гибкость защиты.

Недостатками средств защиты на транспортном уровне являются:

- неполнота защиты
- неподконтрольность событий в рамках прикладных и сетевых протоколов.

3.1.4. Прикладной уровень

Часто при создании и пересылке сложного электронного документа, состоящего из нескольких полей, необходимо обеспечить защиту какого-то отдельно взятого поля данных. Решение этой задачи может заключаться в применении средств криптографической защиты на прикладном уровне. Информация, зашифрованная на прикладном уровне, затем разбивается на пакеты и сетевые кадры, надежность доставки которых обеспечивается транспортной средой, следовательно, криптографическая защита объекта прикладного уровня должна быть действительной и для всех нижестоящих уровней.

Необходимо отметить, что при защите информации на прикладном уровне процедуры передачи, разборки на пакеты, маршрутизации и обратной сборки не могут нанести ущерба конфиденциальности информации. Кроме механизмов обеспечения целостности и конфиденциальности на данном уровне, существует возможность обеспечения аутентификации

пользователя, породившего информацию, – таким образом осуществляется жесткая связь информации с породившим ее пользователем.

Из основных угроз, возникающих на прикладном уровне, следует отметить:

- НСД к данным;
- разведку имен и паролей пользователей;
- атаки на систему разграничения прав доступа пользователей;
- маскировку под легитимного пользователя;
- атаки на систему управления, атаки через стандартные прикладные протоколы;
- фальсификацию информации;
- имитоатаки.

Традиционно применяемыми решениями по защите информации и информационных ресурсов прикладного уровня являются встроенная защита приложений, межсетевые экраны с фильтрацией прикладных протоколов, proxy-системы и т.д. Достоинством размещения средств защиты на прикладном уровне является полнота и высокая функциональность в рамках конкретного приложения и контроль на уровне действий конкретного пользователя (процесса). Недостатками размещения средств защиты на прикладном уровне считаются: отсутствие универсальности, ограниченность рамками заданного набора приложений, неподконтрольность событий в нижележащих уровнях управления.

Криптографическая защита информации на прикладном уровне является наиболее предпочтительным вариантом защиты информации с точки зрения ее гибкости, однако эти меры могут оказаться наиболее сложными в части программно-технической реализации.

Особое место в современных информационно-телекоммуникационных системах занимают вопросы стандартизации методов и средств защиты информации и информационных ресурсов.

На сегодняшний день существует большое количество международных и зарубежных стандартов и рекомендаций, которые, кроме конкретных программно-технических решений, представляют общие подходы к обеспечению информационной безопасности.

3.1.5. Обзор стандартов в области защиты информации

Международные стандарты

Эти документы представлены серией ISO и ISO/IEC и выпущены в свет Международной организацией по стандартизации (ISO) и Международной электротехнической комиссией (IEC).

ISO 7498-2 (X.800). Стандарт посвящен описанию архитектуры безопасности по отношению к модели взаимодействия открытых систем (OSI), включая описание расположения механизмов и служб безопасности на уровнях OSI.

ISO 8372. Этот стандарт описывает режимы блочного шифрования: а) режим электронной книги (ECB); б) режим сцепления блоков (CBC); в) режим с обратной связью по шифртексту (CFB) и г) режим с обратной связью по выходу (OFB). Данные режимы изначально были представлены в стандарте на DES FIPS 81 (1980) и ANSI X.3 106 (1983), в то время как ISO 8372 впервые был опубликован в 1987 г.

ISO 8730. Данный стандарт совместно со стандартом ISO 8731 описывает процедуру использования алгоритмов генерации MAC в банковских системах и является международным аналогом стандарта ANSI X9.9. В нем вводятся независимые от алгоритмов методы и требования использования MAC, включая форматы представления данных, а также способ, при помощи которого данный стандарт может быть применен к выбранному алгоритму.

ISO 8731. В этом стандарте описываются непосредственно алгоритмы генерации MAC. В части 8731-1 описано использование DES в режиме CBC, а часть 8731-1 посвящена алгоритму MAA.

ISO 8732. Стандарт посвящен управлению ключами в банковских системах и является аналогом стандарта ANSI X9.17.

ISO 9564. Стандарт посвящен методам управления и обеспечения безопасности персонального идентификационного номера (PIN). В части 9564-1 раскрываются принципы и средства защиты PIN в течение его жизненного цикла, позволяющие сохранить его в тайне от злоумышленников, а часть 9564-2 посвящена использованию алгоритмов шифрования для защиты PIN.

ISO/IEC 9594-8 (X.509). Данный стандарт посвящен двум типам аутентификации – простой и строгой. Представленная строгая аутентификация состоит из двух и трех передаваемых сообщений и основана на использование ЭЦП и параметров, зависящих от времени. В стандарте также описывается процедура аутентичного распределения открытых ключей на основе сертификатов в формате X.509.

ISO 9807. Стандарт описывает процедуры аутентификации сообщений применительно к банковским системам и является аналогом стандарта ANSI X9.19.

ISO/IEC 9796. Стандарт определяет унифицированный механизм, реализующий ЭЦП с восстановлением сообщения. Основная часть стандарта посвящена избыточным схемам, предназначенным в общем случае для использования совместно с большим классом схем ЭЦП.

ISO/IEC 9797. Этот стандарт описывает алгоритм генерации кодов аутентификации сообщений (MAC), основанный на использовании алгоритма блочного шифрования.

ISO/IEC 9798. Данный стандарт состоит из пяти частей. В первой части (9798-1) описывается механизм аутентификации пользователей, основанный на использовании симметричного алгоритма шифрования (9798-2), алгоритма генерации ЭЦП с асимметричными ключами (9798-3), криптографической функции проверки целостности или MAC (9798-4), некоторых дополнительных механизмов (9798-5).

ISO/IEC 9979. Стандарт описывает процедуры, позволяющие некоторым участникам зарегистрировать алгоритм шифрования в регистрирующем органе в ISO. Результатом является назначение уникального идентификатора-алгоритма, используемого в процедурах взаимодействия и выработки контекста безопасности.

ISO/IEC 10116. Данный стандарт, аналогично ISO 8372, описывает четыре типа использования алгоритмов блочного шифрования, однако ISO/IEC 10116 специфицирует операции для общего случая построения алгоритмов блочного шифрования, где длина блока имеет n бит.

ISO/IEC 10118. Стандарт состоит из нескольких частей и описывает алгоритмы генерации хэш-кода. В 10118-1 представлены основные определения и требования, часть 10118-2 описывает две конструкции хэш-функции, основанные на алгоритмах блочного шифрования. Часть 10118-3 содержит алгоритмы SHA-1, RIPEMD-128 и RIPEMD-160, а часть 10118-4 – описание алгоритмов MASH-1 и MASH-2.

ISO 10126. Стандарт описывает методы обеспечения конфиденциальности финансовых сообщений и является аналогом стандарта ANSI X9.23. Часть 10126-1 посвящена основным методам и принципам, а часть 10126-2 касается вопросов использования DES для этих целей.

ISO 10202. Данный стандарт описывает аспекты безопасности при применении смарт-карт в финансовых транзакциях.

ISO/IEC 10181 (X.810-X.816). Стандарт состоит из описания серии архитектур безопасности, а именно: архитектура аутентификации, архитектура контроля доступа, архитектура обеспечения неотрицаемости и разбора конфликтных ситуаций, архитектура обеспечения целостности, архитектура обеспечения конфиденциальности и архитектура аудита систем безопасности.

ISO 11131. Стандарт описывает аутентификацию, основанную на применении ЭЦП, и является международным аналогом стандарта ANSI X9.26.

ISO 11166. Стандарт состоит из нескольких частей и описывает асимметричные механизмы распределения симметричных ключей. Часть 11166-1

посвящена основным принципам и процедурам распределения ключей и форматам представления ключей, а также сертификации ключевой информации. Часть 11166-2 описывает применение RSA для шифрования и генерации ЭЦП.

ISO 11568. Стандарт описывает средства и процедуры управления ключами при финансовых операциях для симметричных и асимметричных алгоритмов.

ISO/IEC 11770. Стандарт состоит из нескольких частей, в которых рассматриваются вопросы управления ключами и механизмы распределения ключей. Часть 11770-1 описывает основы управления ключами и понятие цикла жизни ключей; требования по защите ключей и роль третьей стороны в распределении ключей. Часть 11770-2 описывает механизмы распределения ключей, основанные на симметричных алгоритмах и протоколах, подобных Kerberos и Otway-Rees. В части 11770-3 рассматриваются механизмы распределения ключей, основанные на асимметричных алгоритмах, которые, в свою очередь, подразделяются на соглашения о ключах и протоколы передачи ключей.

ISO/IEC 13888. Данный стандарт посвящен проблеме неотрицаемости ряда фактов, связанных с процессом передачи сообщений. В нем также приводятся механизмы, разрешающие случаи, связанные с отказом от факта отправки сообщения, отказа от факта приема сообщения, а также механизмы неотрицаемости, связанные с использованием доверенной третьей стороны.

ISO/IEC 14888. Стандарт состоит из нескольких частей и посвящен построению схем ЭЦП. Часть 14888-1 знакомит с общепринятыми определениями и моделями построения схем ЭЦП. В части 14888-2 рассматриваются схемы ЭЦП, в которых ключ проверки подписи является результатом общедоступной функции от информации, идентифицирующей подписывающего. Часть 14888-3 посвящена распределению открытых ключей при помощи сертификатов открытых ключей.

Стандарты серии ANSI

В этом разделе перечислены стандарты, разработанные Американским национальным институтом стандартов (ANSI) и посвященные криптографическим алгоритмам и их применению в банковских системах.

ANSI X3.92. Стандарт специфицирует DES-алгоритм, который в рамках этого стандарта представляется как алгоритм шифрования данных (Data Encryption Algorithm-DEA).

ANSI X3.106. Стандарт описывает виды применения алгоритма DES.

ANSI X9.8. Стандарт посвящен вопросам безопасности и управления PIN.

ANSI X9.9. Стандарт описывает применение MAC на основе использования DES (в режимах CBC и CFB с блоками длиной 64 бита) в широком круге банковских систем.

ANSI X9.17. Данный стандарт основан на стандарте ISO 8732 и посвящен процедурам управления ключами, а также средствам распределения и защиты ключей применительно к банковским системам. В нем нашли отражения специфические аспекты использования ключей, такие как счетчики ключей, преобразование ключей и подтверждение подлинности ключей. Средства распределения ключей, представленные в данном стандарте, построены по принципу «точка-точка» или используют третью сторону (KDC или KTC).

ANSI X9.19. Стандарт посвящен вопросам использования алгоритма DES для генерации MAC в конкретных типах банковских систем.

ANSI X9.23. Стандарт описывает форматы представления данных при использовании алгоритма DES в банковских системах, включая поля флагов, поля добавленных данных (набивки) и методы обработки получаемых из каналов связи данных.

ANSI X9.24. Данный стандарт описывает методы управления ключами, аутентификацию (основанную на DES) и шифрование PIN, ключей и других данных, а также руководящие принципы защиты ключей на всех этапах жизненного цикла.

ANSI X9.26. Стандарт описывает два основных класса механизмов пользовательской аутентификации, которая требуется для контроля доступа. Первый класс построен на основе использования паролей. Второй – на основе использования симметричных алгоритмов шифрования и предварительно распределенных секретных ключах.

ANSI X9.28. Этот стандарт является продолжением стандарта ANSI X9.17 и описывает процедуры распределения ключей между пользователями, которые не имеют между собой и третьей стороной предварительно распределенной ключевой информации.

ANSI X9.30. Стандарт состоит из нескольких частей и описывает асимметричные алгоритмы: в X9.30-1 – DSA, а в X9.30-2 – SHA.

ANSI X9.31. Стандарт описывает алгоритм генерации и проверки ЭЦП на основе использования алгоритма RSA.

ANSI X9.42. Стандарт описывает несколько вариантов применения алгоритма неаутентификационного обмена ключами типа Диффи-Хэлмана, обеспечивающих распределение симметричных ключей.

ANSI X9.45. Стандарт посвящен применению аутентификационных сертификатов открытых ключей.

ANSI X9.52. В этом стандарте, аналогично стандарту ISO 8372, описывается применение Triple DES и четырех режимов его использования.

ANSI X9.55. В этом стандарте рассматриваются формат и применение сертификата типа X.509 вер. 3.

ANSI X9.57. Стандарт описывает управление сертификатами в сфере электронной коммерции.

Государственные стандарты США

FIPS 46. Стандарт описывает DES-алгоритм.

FIPS 74. Стандарт описывает руководящие принципы применения и использования DES.

FIPS 81. В этом стандарте рассматриваются четыре основных типа использования DES.

FIPS 112. Данный стандарт описывает руководящие принципы использования и управления паролей.

FIPS 113. Стандарт описывает алгоритм генерации MAC на основе использования DES.

FIPS 140-1. Стандарт описывает требования безопасности при разработке и применении криптографических модулей, выполненных как аппаратно, так и программно.

FIPS 171. Стандарт описывает средства распределения ключей при использовании в государственных структурах.

FIPS 180 и 180-1. Стандарты посвящены алгоритмам SHA и SHA-1 соответственно.

FIPS 185. Стандарт описывает процедуру депонирования ключей и описывает алгоритм симметричного шифрования SKIPJACK.

FIPS 186. Стандарт описывает DSA в применении совместно с SHA.

FIPS JJJ. В этом стандарте рассматриваются механизмы аутентификации с применением асимметричных алгоритмов, состоящих из двух-трех шагов.

Существует также большое количество стандартов в области безопасности Internet, например RFC, разработанный IETF (Internet Engineering Steering Group), – табл. 3.1, и стандарты PKCS (Public-key Cryptography Standards).

3.1.6. Подсистема информационной безопасности

Важным понятием в обеспечении информационной безопасности современных информационно-телекоммуникационных систем является *подсистема информационной безопасности* (ПИБ). Понятие ПИБ включает

Таблица 3.1. Некоторые стандарты из серии RFC

Номер стандарта	Назначение
1319	Хэш-функция MD2
1320	Хэш-функция MD4
1321	Хэш-функция MD5
1421	PEM-шифрование и аутентификация
1422	PEM-сертификаты и управление ключами
1423	PEM-алгоритмы, режимы, идентификаторы
1424	PEM-сертификация ключей и службы
1508	Общее описание API к службам безопасности
1510	Kerberos v5
1828	Ключевой вариант MD5
1847	MIME
1938	Системы с одноразовыми паролями

в себя весь комплекс средств и мер по защите информации в ИТС. Основные задачи создания ПИБ:

- обеспечение информационной безопасности в ИТС за счет комплексного использования технологических, организационных, технических, программно-аппаратных, криптографических средств и мер защиты;
- обеспечение требуемой готовности и надежности средств защиты информации в соответствии с регламентом функционирования ИТС;
- использование различных средств защиты информации, находящихся в эксплуатации в региональных отделениях ИТС в рамках единой системы, за счет использования типовых интерфейсов взаимодействия;
- обеспечение открытости системно-технических решений и архитектуры средств защиты, преемственности и развития;
- выполнение непрерывного контроля и регистрации действий пользователей и приложений, ведение аудита и протоколирования.

Типовая ПИБ строится с учетом следующих принципов:

- применяемые в ПИБ средства и технологии защиты должны обладать свойствами модульности, масштабируемости, открытости и возможности адаптации системы к различным организационным и техническим условиям в соответствии с используемыми аппаратными средствами;
- ПИБ ИТС должна предполагать полную независимость функционирования каждого из компонентов защиты. Нарушение функционирования любого компонента не должно приводить к изменению других характеристик защиты, тем более к отказу всей системы;

- применяемые средства и технологии защиты должны обеспечивать открытость архитектуры, наращиваемость, а также предоставлять интерфейс для подключения внешних систем защиты информации;
- ПИБ ИТС должна предусматривать возможность осуществления защитных мер на всех уровнях ISO/OSI. Кроме того, должна быть обеспечена информационная безопасность от уровня структурированной кабельной системы до уровня локальной вычислительной сети и прикладных задач ИТС;
- должны быть обеспечены меры защиты автоматизированных рабочих мест различного вида и назначения, криптографическая защита информации, защита прикладных серверов от несанкционированного доступа (НСД), развитые системы контроля (аудита) безопасности, управления и мониторинга состояния информационной безопасности ИТС;
- в ПИБ ИТС должен быть предусмотрен комплекс организационно-технических мер и мер технологической защиты информации.

В ПИБ каждому классу средств и мер защиты информации выделяется своя задача, в связи с чем ряд средств и мер объединяются в подсистемы, которые (каждая по отдельности) решают строго определенные задачи, но при этом соблюдаются цели создания ПИБ и принципы ее построения.

Типовая ПИБ может иметь следующую структуру (хотя здесь необходимо отметить, что в реальной ситуации конфигурация может меняться; принципы разбиения ПИБ на подсистемы могут также отличаться от изложенных):

- подсистема защиты локальных рабочих мест;
- подсистема защиты локальных вычислительных сетей (ЛВС);
- подсистема защиты межсетевого взаимодействия;
- подсистема аудита и мониторинга (в зависимости от масштабов и сложности построения ИТС данная подсистема может разбиваться на отдельные составляющие);
- подсистема технологической защиты.

Основные задачи подсистемы защиты локальных рабочих мест

К основным задачам этой подсистемы относятся:

- разграничение доступа к ресурсам пользователей *автоматизированных рабочих мест* (АРМ) ИТС;
- криптографическая защита хранящейся на АРМ информации (конфиденциальность и целостность);

- аутентификация пользователей и авторизация их действий;
- обеспечение невозможности влияния программно-технического обеспечения АРМ на регламент функционирования прикладных процессов ИТС и контроль целостности прикладного и системного ПО.

Основные задачи подсистемы защиты локальных вычислительных сетей

Для данной подсистемы характерна не только защита от несанкционированного доступа (НСД) к информации и информационным ресурсам в рамках локальных вычислительных систем (ЛВС), но также и криптографическая защита информации, передаваемой в ЛВС, на прикладном, системном уровне и на уровне структурированной кабельной системы.

Основные задачи подсистемы защиты межсетевого взаимодействия

Для этой подсистемы основным является защита ИТС от негативных воздействий со стороны внешних сетей передачи данных. При этом должна обеспечиваться не только защита информации (конфиденциальность и целостность), передаваемой во внешнюю среду, но также и защита от внешнего разрушающего воздействия информации и информационных ресурсов, находящихся внутри ИТС. Особенно характерным для данной подсистемы является наличие развитых средств мониторинга и аудита событий, связанных с нарушением безопасности.

Основные задачи подсистемы аудита и мониторинга

Основными задачами данной подсистемы являются:

- осуществление централизованного, удаленного, автоматизированного контроля работы подконтрольных серверов, АРМ и других подсистем ПИБ;
- централизованное ведение протокола всех событий, происходящих на подконтрольных серверах, АРМах и подсистемах;
- выполнение автоматизированного анализа механизма принятия решений в нештатных ситуациях.

Эта подсистема предназначена для первичного контроля независимости выполнения основных технологических операций и правомерности событий, связанных с информационной безопасностью.

Основные задачи подсистемы технологической защиты

Задачами подсистемы технологической защиты являются:

- выработка принципов построения технологического процесса обработки информации в ИТС;
- определение контрольных точек технологического процесса;
- определение мест использования средств и методов защиты информации для регистрации, подтверждения и проверки прохождения информации через контрольные точки технологического процесса;
- исключение сосредоточения полномочий у отдельных должностных лиц.

3.2. Защита локальной рабочей станции

Под *локальной рабочей станцией* (ЛРС) будем подразумевать персональный компьютер при условии, что он не подсоединен к каналам передачи данных. Любая защита в современных информационных системах начинается прежде всего с конкретного рабочего места. Круг вопросов, связанных с обеспечением его безопасности, занимает одно из первых мест по остроте стоящих задач и по разнообразию средств и методов защиты ЛРС. Ее защита особенно актуальна еще и потому, что по статистике большинство нарушений в современных информационных системах приходится именно на внутренних нарушителей. Хотя существуют тенденции возрастания и числа удаленных атак, чему способствует широкое развитие глобальных сетей передачи данных.

Вопросы, связанные с функционированием рабочих станций в сети, и межсетевое взаимодействие будут рассмотрены ниже.

Задачи обеспечения информационной безопасности на локальных рабочих станциях осложняются тем, что функционирование средств защиты информации происходит в так называемой *недоверенной программной среде*, то есть в среде, где невозможно однозначно доказать отсутствие влияния программного окружения на функционирование средств защиты информации. Недоверенность программной среды вытекает из того факта, что на сегодняшний день ПО, установленное на ЛРС, чаще всего зарубежного производства, функциональный состав которого в большинстве случаев является неопределенным. При этом под ПО подразумевается как прикладное программное обеспечение, так и обеспечение безопасности ОС.

Следует отметить, что наибольшую опасность представляют недокументированные возможности и закладки, находящиеся в ядре ОС. Так, например,

в ОС Windows NT в интерфейсе Win32 имеется ряд недокументированных возможностей, хотя ни одна из них не является враждебной по отношению к средствам защиты информации и секретной пользовательской информации, но тем не менее наличие других не столь безобидных недокументированных функций и возможностей не исключается. Вопросы создания доверенной программной среды будут рассмотрены ниже.

При этом необходимо учитывать, что большинство ЛРС включает разнородное ПО со сложной структурой, а также аппаратную часть, реализующую большое количество функций и являющуюся разветвленной системой. Дело в том, что сложность построения современных информационных систем – одна из причин успешного проведения атак, так как угрозы часто возникают из-за того, что конечные пользователи требуют от средств вычислительной техники выполнения заданных функций, а проблемам защиты информации уделяется мало внимания. В связи с тем, что реальные системы становятся изо дня в день все сложнее и реализуют все большее количество сервисных функций, то очевидно, что и проблем с безопасностью будет все больше и больше.

3.2.1. Угрозы и задачи информационной безопасности для локальных рабочих станций

Защита локальной рабочей станции – задача комплексная, состоящая из широкого круга проблем, решение которых возможно только после разработки общей для той или иной системы политики безопасности.

Перед рассмотрением специфических угроз, возникающих при работе пользователей на локальных рабочих станциях (ЛРС), полезно остановиться на описании модели нарушителя.

Потенциальных нарушителей можно разделить на следующие категории:

- зарегистрированные пользователи ЛРС;
- имеющие доступ к штатным средствам управления рабочей станцией, таким как клавиатура, устройства считывания информации (дисководы, считыватели (touch memory) и др.);
- не имеющие физического доступа к ЛРС;

При этом нарушители обычно ставят перед собой следующие цели:

- компрометация секретной информации, в том числе и информации, относящейся к работе средств защиты информации ЛРС (ключевая информация, пароли пользователей и др.);

- изменение или уничтожение секретной информации, в том числе и относящейся к функционированию средств защиты информации ЛРС (ключевая информация, пароли пользователей и др.);
- нарушение работоспособности всей системы в целом или ее отдельных компонентов;

Здесь необходимо уточнить, что нарушениями мы будем считать не только умышленные действия, направленные на нарушение политики безопасности, но и неумышленные нарушения и ошибки обслуживающего персонала, приводящие или позволяющие противнику опосредствованно добиваться перечисленных целей.

Список угроз будет прежде всего зависеть от того, как используется ЛРС: в однопользовательском или многопользовательском режиме. Хотя очевидно, что угрозы, возникающие при однопользовательском режиме, являются частным случаем угроз, появляющихся при многопользовательском режиме.

Обобщенный перечень угроз можно классифицировать по следующим признакам:

- по характеру доступа:
  - с доступом к ПО;
  - с доступом только к аппаратным ресурсам;
  - без доступа к ЛРС.

Доступ к ПО подразумевает, что нарушитель имеет возможность взаимодействовать с установленным ПО, которое доступно для модификации и анализа. Доступ к аппаратному обеспечению подразумевает воздействие нарушителем на аппаратную часть с целью введения ее в режим, нарушающий надежное функционирование средств защиты информации. Отсутствие доступа к ЛРС подразумевает, что нарушитель имеет возможность воздействовать на систему только через каналы передачи информации, возникающие в ходе работы ЛРС. В качестве таких каналов следует отметить: а) электромагнитный канал, б) цепи заземления и питания, в) виброакустический канал. Например, электромагнитный канал возникает вследствие появления электромагнитного излучения от ЛРС, причем данное излучение модулируется в соответствии с процессами обработки информации, среди которых могут быть и процессы, обрабатывающие секретную информацию;

- по характеру проявления:
  - активные;
  - пассивные.

Активные характеризуются тем, что нарушитель является инициатором негативных воздействий на систему (воздействия могут быть как умышленными, так и неумышленными), пассивные же угрозы заключаются в анализе информации, возникающей в ходе функционирования ЛРС;

- по используемым в ходе реализации угрозы средствам:
  - с использованием штатных средств, входящих в состав ЛРС;
  - с использованием дополнительных средств.

Далее мы сформулируем перечни конкретных угроз по пунктам классификации (1–7), предложенной для однопользовательской ЛРС.

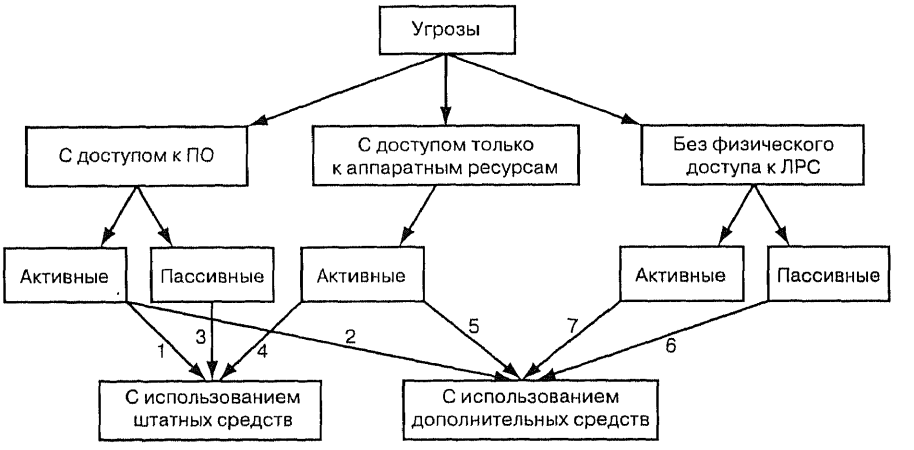

Рис. 3.1. Классификация угроз

Ветвь классификации 1

Угрозы, относящиеся к этой ветви, характеризуются тем, что нарушитель, получивший доступ к ПО, установленному на ЛРС, может воспользоваться штатными средствами данного ПО для проведения атак. К штатным средствам, позволяющим нарушителю проводить атаки, будем относить: средства разработки и отладки ПО, входящие в состав некоторых типов прикладного ПО (например, Microsoft Word имеет в своем составе встроенный Visual Basic); средства управления и конфигурирования, входящие в состав программного обеспечения; ПО, имеющее ошибки в реализации. Недоступность для пользователя данных средств переводит потенциальные угрозы в ветвь классификации 3. При перечислении угроз мы также будем исходить из того, что на ЛРС используются штатные средства защиты ОС, причем их конфигурирование выполнено корректно, так как в противном случае, если таковые средства не используются, нарушитель

получает неограниченный доступ к ЛРС. Предположим, что на ЛРС установлена одна ОС, поскольку, если установлены еще и другие ОС (например, Windows NT и MS DOS), пользователь может получить доступ к служебной информации, которая до этого находилась в монопольном владении другой ОС. Для однопользовательского режима характерно, что нарушителем в данном случае является не зарегистрированный в системе пользователь.

Итак, в эту ветвь входят следующие угрозы:

- создание программ-закладок, вирусов и программ, обеспечивающих прямой доступ к памяти или адресному пространству любого процесса, позволяющих обходить штатные средства защиты ОС и получать привилегии администратора ОС. С помощью подобных программ можно получать доступ к конфиденциальной информации за счет чтения файлов и областей памяти, в которых находятся секретные данные (пользовательский пароль, учетная запись пользователя, настройки системы), или нарушать целостность данной информации;
- нарушение целостности установленного прикладного или системного ПО с целью перевода штатных средств защиты в нештатный режим работы;
- удаление критичной информации или части ПО, позволяющее нарушить работоспособность системы;
- внесение программ-закладок и вирусов в установленное ПО;
- оказание негативного влияния на работу аппаратных средств ЛРС;
- переконфигурирование системы с целью нарушения ее работоспособности;
- ошибочные действия персонала;
- анализ особенности функционирования системы защиты с целью выявления некорректных состояний; то же самое применительно к прикладному и системному ПО;
- доступ к остаточной информации, находящейся во временных файлах или в кластерах жесткого диска после удаления программ штатными средствами (физически данные могут оставаться в кластере, если именно в него не будет произведена перезапись);
- удаление журналов аудита с целью скрыть факт воздействия на систему.

Ветвь классификации 2

В этом случае нарушитель использует свои программы проведения атак, и перечень угроз в данной ветви будет иметь тот же вид, что и ветвь классификации 1. Чтобы запустить подобные программы, нарушитель должен

иметь возможность загрузить ПО с внешних носителей информации (гибкие диски, CD-ROM и т.д.). Также нарушитель может загрузить с внешних носителей свою версию ОС, и тогда последствия будут аналогичны тем, что описаны выше.

Ветвь классификации 3

Это пассивные угрозы, которые возможны только в случае отсутствия штатных механизмов защиты ОС или их некорректной настройки. К пассивным угрозам относятся:

- доступ к незащищенным файлам, содержащим критичную пользовательскую или системную информацию, с помощью штатных средств ПО;
- анализ функционирования ПО, установленного на ЛРС;
- анализ настроек ОС;
- добывание информации о содержащихся ошибках в ПО;
- анализ остаточной информации.

Ветвь классификации 4

Нарушитель имеет доступ к органам управления и средствам ввода/вывода информации в ЛРС, а также к контактным разъемам считывающих устройств, не будучи зарегистрированным в качестве штатного пользователя системы. Нарушитель может воздействовать через органы управления на работу ПО ЛРС, при этом считается, что он не имеет доступа внутрь корпуса ЛРС и получает возможность:

- загрузить свою версию ОС с внешних носителей информации;
- выдать себя за зарегистрированного пользователя системы с целью получить доступ к системному и/или прикладному ПО. После осуществления подобной операции возникает перечень угроз, относящихся к первой, второй или третьей ветви классификации;
- воздействовать на аппаратные средства с целью перевода их в некорректный режим функционирования, например заставить ПО работать на ЛРС, на которой не прошли тесты памяти в процессе загрузки;
- войти в режим конфигурирования BIOS;
- получить доступ к информации, относящейся к функционированию аппаратной части ЛРС;
- воздействовать на аппаратную часть с целью выведения из строя отдельных частей аппаратных компонентов ЛРС или всей ЛРС в целом.

Ветвь классификации 5

Данный перечень конкретных угроз состоит из тех же пунктов, что и предыдущий, с одним добавлением: получение физического доступа к аппаратным компонентам, осуществляющим хранение и обработку информации, нарушение целостности информации или изменение процессов обработки информации с целью доступа к конфиденциальной или критичной системной информации.

Ветвь классификации 6

Угрозы этого типа основаны на том, что аппаратные средства в ходе обработки информации создают побочные физические поля, законы изменения которых отражают процесс обработки и хранения информации. К таким полям относятся электромагнитное и виброакустическое. Также в ходе работы средств вычислительной техники могут возникать побочные каналы утечки информации, выражающиеся в наличии сопутствующих сигналов в линии электропитания или заземления. Поскольку считается, что ЛРС отключена от каналов передачи данных, мы не будем учитывать побочные каналы утечки информации.

Очевидно, что любой сигнал в процессе удаления от источника имеет некоторую величину затухания, вследствие чего на определенном расстоянии побочный канал утечки информации не будет представлять опасности. Зона, в которой побочные сигналы могут представлять опасность, обычно контролируется, и доступ туда посторонним лицам (тем более нахождение аппаратуры, позволяющей фиксировать опасные сигналы) запрещен. Кроме того, само по себе пребывание в опасной зоне мало что значит. Чтобы воспользоваться подобного рода возможностями, нарушитель должен обладать определенными средствами и навыками:

- получения доступа к конфиденциальной пользовательской и системной информации;
- получения информации о процессах функционирования ПО и аппаратной части ЛРС.

Ветвь классификации 7

Угрозы этого типа отличаются от перечисленных следующим: нарушитель в ходе их реализации производит воздействие на ЛРС с помощью электромагнитных каналов и цепей питания и/или заземления. Факт наличия подобных угроз необходим при практической реализации дифференциального криптоанализа, заключающегося в вызове ошибочных ситуаций

в процессах обработки и хранения информации, с целью последующего анализа полученного результата. Подобная атака, например, может привести к компрометации секретных ключей пользователя. Таким образом, основная цель нарушителя в данном случае – вызвать сбой в аппаратной части ЛРС и вывести из строя ее отдельные компоненты.

При использовании локальной рабочей станции в многопользовательском режиме, то есть при наличии нескольких зарегистрированных пользователей на ЛРС, возникают угрозы и со стороны легальных пользователей. Поскольку на одной ЛРС могут работать несколько пользователей, то на ней будет храниться и обрабатываться информация разных пользователей, в том числе и конфиденциальная. Рассматривать угрозы для данного случая будем с учетом предположения, что на ЛРС установлена ОС, имеющая штатные средства защиты информации и разграничения доступа к ресурсам, поскольку при отсутствии таковых любой пользователь сможет беспрепятственно достигнуть одной из поставленных целей, а именно:

- получить доступ к конфиденциальной информации других пользователей, хранящейся в незащищенных файлах;
- применить программы-закладки и вирусы, активизирующиеся в ходе работы на данной ЛРС другого пользователя;
- нарушить целостность пользовательской информации;
- получить доступ к системной информации, относящейся к работе других пользователей;
- выдать себя при регистрации в системе за другого пользователя и провести несанкционированные операции от его имени.

3.2.2. Методы и средства обеспечения информационной безопасности локальных рабочих станций

Создание защищенной системы – задача комплексная, и решается она путем применения программно-технических методов и средств, а также с помощью организационных мероприятий. Конкретные средства защиты информации реализуют только типовые функции по ее защите, реальная же защитная система строится исходя из возможных угроз и выбранной политики безопасности.

Учитывая обобщенный перечень угроз, можно предложить следующий перечень мероприятий по защите данных:

- обеспечение конфиденциальности и достоверности пользовательской информации. Реализуется путем применения средств шифрования и ЭЦП, выполненных как виде абонентского шифрования

(отдельная программа-вызов, запуск которой предоставляет пользователю возможность применять функции шифрования и ЭЦП), так и в виде средств прозрачного шифрования. Например, вызовы функций шифрования встраиваются в используемое пользовательское ПО, в ходе работы которого незаметно для пользователя происходит вызов этих функций из прикладного ПО;

- разграничение доступа пользователей к информации, хранящейся и обрабатываемой на ЛРС при применении многопользовательского режима. Обычно используются штатные средства современных ОС, но, учитывая недостатки в их реализации, часто возникает желание иметь более надежную защиту от НСД, что приводит к необходимости применения дополнительных средств разграничения доступа;
- аутентификация пользователей и авторизация доступа к защищаемым объектам с использованием криптографических методов или индивидуальных носителей секретной информации (touch memory, смарт-карты и гибкие магнитные диски), включая аутентификацию пользователей перед загрузкой ОС или при входе в ОС;
- контроль целостности секретной пользовательской или системной информации. Реализуется путем применения программ, выполняющих функции по генерации контрольных сумм и их проверки; обычно используются бесключевые хэш-функции или имитовставки. Контроль целостности защищаемой информации предполагает систематичность ее проведения. Например: либо администратор безопасности должен инициировать ее с помощью организационных мер, либо в прикладное или системное ПО должны быть встроены функции, реализующие контроль целостности, которые автоматически вызываются ПО непосредственно перед тем, как приступить к работе с защищаемой информацией. В этом случае запуск драйвера, работающего с ключами пользователя, должен происходить с предварительным контролем его целостности, или контроль целостности должен осуществляться на критичные системные файлы (файлы настроек, системные драйверы и т.д.) перед их загрузкой в память или передачей им управления;
- контроль процесса загрузки ОС. Должна быть обеспечена загрузка строго определенной версии ОС и блокирована несанкционированная загрузка с внешних носителей или из сети;
- контроль целостности аппаратных ресурсов, который необходимо осуществлять непосредственно перед тем, как пользователь начнет работать с ЛРС. Если зафиксировано несанкционированное изменение аппаратной части или конфигураций аппаратной части ЛРС, работа с данной ЛРС должна блокироваться;

- исключение негативного влияния на процесс функционирования пользовательского приложения со стороны программного окружения, установленного на ЛРС. Под программным окружением в данном случае понимается другое прикладное ПО, которое работает на данной ЛРС. При этом поставленная задача может быть решена с помощью контроля целостности ядра ОС и гарантии отсутствия программ-закладок в системном ПО. Ее решение особенно актуально в многопользовательских системах, где один зарегистрированный пользователь, являющийся нарушителем, может реализовать программу, которая будет стартовать во время работы другого пользователя, при этом злоумышленник будет иметь возможность получать доступ к секретной информации;
- контроль запуска задач пользователем, то есть пользователю для запуска должны быть доступны только те задачи, которые являются разрешенными в рамках выбранной политики безопасности;
- защита от побочного электромагнитного излучения и утечки информации по цепям питания и заземления. Подобного рода задачи решаются путем анализа спектра излучения аппаратной части ЛРС, проведения специальных работ по экранированию, фильтрации опасных побочных сигналов или создания контролируемой территории, за пределами которой побочные сигналы не представляют опасности в силу того, что их уровень ничтожно мал, чтобы уловить информацию даже с помощью современных средств. Представляется возможным использовать также генераторы шумов, позволяющие создавать высокий уровень помех, уловить за которыми несущий ценную информацию сигнал затруднительно. Здесь, правда, следует учесть, что эта задача напрямую связана с качеством и возможностями аппаратуры, применяемой нарушителем для перехвата побочных каналов утечки информации;
- физическое удаление остаточной информации, возникающей в ходе функционирования ПО. Подобные проблемы возникают в силу того, что прикладное и системное ПО создают временные файлы; в некоторых современных ОС существуют файлы подкачки, или страничные файлы, и не реализовано физическое затирание удаляемой информации;
- аудит и протоколирование работы средств защиты информации, системного и прикладного ПО.

Рассмотрим теперь средства защиты информации, которые широко применяются на сегодняшний день и позволяют решать одну или несколько

из перечисленных выше типовых задач обеспечения информационной безопасности для ЛРС.

Средства криптографической защиты информации семейства «Верба»

Семейство СКЗИ «Верба» (МО ПНИЭИ) представлено целым рядом программных продуктов, выполненных как в виде законченного программного продукта, так и в виде библиотек, встраиваемых в прикладное ПО. СКЗИ «Верба» подходит практически для всех вариантов ОС – Windows 95/NT, MS DOS, а также для различных модификаций UNIX. При этом для ОС Windows существуют 16-разрядные и 32-разрядные варианты данного СКЗИ.

СКЗИ «Верба» подразделяется на два класса: СКЗИ «Верба» и СКЗИ «Верба-О». Их отличия сводятся к устройству ключевой системы: если в первом классе используются симметричные ключи с предварительным их распределением, то во втором классе применяется открытое распределение ключей.

Далее мы подробно опишем двух представителей СКЗИ семейства «Верба», а именно: «Верба-W» и «Верба-OW».

СКЗИ «Верба-W» представляет собой набор библиотек с реализованными в них функциями шифрования, работы ЭЦП и др. Данное средство может служить криптографическим ядром других средств защиты информации.

Основными функциями СКЗИ «Верба-W» являются:

- зашифрование/расшифрование информации, хранящейся в виде файлов или в виде областей памяти;
- генерация ЭЦП как на отдельные файлы, так и на области памяти. При этом ЭЦП может присоединяться к исходному файлу (области памяти) либо сохраняться в отдельном файле (области памяти);
- проверка ЭЦП как на файлах, так и на области памяти;
- получение идентификатора ключа абонента, зашифровавшего файл (область памяти);
- получение списка получателей зашифрованного файла (области памяти);
- получение имитовставки для файла, причем имитовставка может считаться как на ключе, так и на пароле пользователя;
- вычисление хэш-кода на файл или область памяти;
- функции работы со справочниками открытых ключей (удаление ключа из справочника, добавление ключа в справочник, получение списка открытых ключей в справочнике и т.д.);

- сервисные функции: загрузка ключа в оперативную память, удаление ключа из оперативной памяти, получение списка открытых ключей в справочнике, генерация случайного числа.

Алгоритм шифрования выполнен в соответствии с требованиями ГОСТ 28147-89. Системы обработки информации. Защита криптографическая.

Цифровая подпись выполнена в соответствии с требованиями ГОСТ Р 34.10-94. Информационная технология. Криптографическая защита информации. Система электронной цифровой подписи на базе асимметричного криптографического алгоритма.

Функция хэширования выполнена в соответствии с требованиями ГОСТ Р 34.11-94. Информационная технология. Криптографическая защита информации. Функция хэширования.

Скорость шифрования информации (PC/AT 486/100) – 500 Кб/с (не включая время, затрачиваемое на контроль работоспособности функций шифрования и обращения к диску).

Время вычисления хэш-функции (PC/AT 486/100) – 400 Кб/с.

Время формирования ЭЦП (PC/AT 486/100) – 0,04 с.

Время проверки ЭЦП (PC/AT 486/100) – 0,2 с.

В комплект поставки данного СКЗИ входят библиотеки с реализованными криптографическими функциями: а) wcryp.dll (содержит функции шифрования, выработки имитовставки, формирования случайного числа); б) wsign.dll (содержит функции формирования электронной цифровой подписи, ее проверки и выработки значения хэш-функции, а также функции работы со справочниками открытых ключей подписи); в) wboth.dll (содержит функции шифрования, выработки имитовставки и случайного числа, формирования электронной цифровой подписи, проверки подписи и выработки значения хэш-функции, функции работы со справочниками открытых ключей подписи). А также ПО, обеспечивающее:

- операции чтения и загрузки ключей с основных носителей в драйвер;
- драйвер хранения ключей для Windows 95 (Windows NT), реализующий функции хранения ключевой информации и инициализации программного датчика случайных чисел;
- интерфейс к драйверу хранения ключей;
- работу с ключевыми носителями типа touch memory и смарт-карта.

Ключевая система построена по принципу полной матрицы (рис. 3.2), обеспечивающей связь каждого с каждым. Совокупность всех исходных секретных ключей парной шифрованной связи в данном случае представляет

$$K = \begin{bmatrix} k_{11}, & k_{12}, & ..., & k_{1N} \\ k_{12}, & k_{22}, & ..., & k_{2N} \\ ... & ... & ... & ... \\ ... & ... & ... & ... \\ k_{1N}, & k_{2N}, & ..., & k_{NN} \end{bmatrix}$$

Рис. 3.2. Ключевая система типа «полная матрица»

собой симметричную матрицу К размером N×N с элементами k_{ij}, $k_{ij} = k_{ji}$, где N – число абонентов сети, k_{ij} – ключ шифрования межу i-м и j-м абонентами (256-мерный двоичный вектор).

К важной информации, хранящейся на ключевых носителях (гибкие магнитные диски, touch memory и смарт-карты) пользователей, относятся: ключи шифрования, основу которых составляет строка матрицы К (для обеспечения шифрованной связи i-му абоненту выдается i-я строка матрицы К), секретные и открытые ключи ЭЦП и служебные ключи, на которых зашифровываются секретные ключи подписи и шифрования для хранения на жестком диске.

При построении систем защиты с использованием СКЗИ этого типа необходимо создать организационную структуру – центр управления ключевой системой (ЦУКС), в задачи которого входят:

- регистрация новых абонентов сети;
- подача заявок на требуемое количество ключевых упаковок для плановой смены ключей в сети;
- получение комплекта ключевых носителей;
- организация и хранение ключевых упаковок (в том числе и резервных);
- ведение журнала учета полученных, хранящихся в ЦУКС и выданных абонентам ключевых носителей;
- выдача абонентам ключевых носителей;
- регистрация открытых ключей ЭЦП абонентов:
- выдача абонентам регистрационных файлов открытых ключей ЭЦП из справочника участников расчетной системы;
- организация схемы оперативного оповещения абонентов обо всех изменениях, происходящих в сети (компрометация ключей, восстановление конфиденциальной связи после компрометации ключей, включение новых абонентов, плановая смена ключей и т.п.);
- разработка парольной системы оповещения в сети;

- управление ключевой системой;
- разбор конфликтных ситуаций и доказательство авторства электронного документа, снабженного ЭЦП.

СКЗИ «Верба-W» отличается от СКЗИ «Верба-OW» своей ключевой системой. СКЗИ «Верба-OW» является системой с открытым распределением ключей. Каждый пользователь вырабатывает свой секретный ключ, из которого затем с помощью некоторой процедуры формируется открытый ключ. Открытые ключи публикуются в общедоступном справочнике.

В СКЗИ «Верба-OW» ключ зашифрования совпадает с ключом расшифрования. При зашифровании сообщения i-м абонентом для j-го абонента общий секретный ключ связи вырабатывается на основе секретного ключа шифрования i-го абонента и открытого ключа шифрования j-го абонента. Соответственно для расшифрования этого сообщения j-м абонентом формируется секретный ключ связи на основе секретного ключа шифрования j-го абонента и открытого ключа шифрования i-го абонента. Таким образом, для обеспечения связи с другими абонентами каждому пользователю необходимо иметь:

- собственный секретный ключ шифрования;
- открытые ключи шифрования пользователей сети конфиденциальной связи, объединенные в справочники.

Таким образом, СКЗИ обеспечивают конфиденциальность, целостность и достоверность информации, хранящейся и передаваемой по общедоступным каналам связи. Но для надежной работы данного вида СКЗИ требуются дополнительные защитные функции, поскольку само средство может стать объектом атаки нарушителя.

Необходимо отметить, что на сегодняшний день данное СКЗИ сертифицировано ФАПСИ совместно с программно-аппаратным комплексом «Аккорд», обеспечивающим защиту от несанкционированного доступа.

В общем случае безопасность работы с СКЗИ «Верба», например реализованного под ОС Windows NT, определяется совокупностью нескольких составляющих:

- защитными мерами, реализованными в самом СКЗИ;
- безопасностью ОС и корректностью настроек ее штатных средств защиты.
- функциями по защите информации, реализованными в «Аккорде»;
- выполнением организационных требований, выдвигаемых производителем данного СКЗИ в соответствующих документах (вместе с СКЗИ поставляется большое количество инструкций, затрагивающих все стороны его работы).

Поскольку СКЗИ «Верба» поставляется в виде динамически подключаемых библиотек, что обеспечивает гибкость его применения, то для практического использования библиотек необходимо осуществить встраивание СКЗИ в ПО, которое наряду со своими сервисными функциями будет осуществлять защиту информации. Грамотное выполнение инструкций по встраиванию данного СКЗИ является еще одним компонентом, обеспечивающим безопасность и надежность.

«Лексикон-Верба»

В качестве законченного программного продукта со встроенным криптографическим ядром «Верба-OW» рассмотрим ПО «Лексикон-Верба» (Арсеналъ).

«Лексикон» выпускается в двух версиях общего назначения: базовой и XL, а также с криптозащитой документов «Лексикон-Верба». «Лексикон» 4.0 и «Лексикон-Верба» позволяют:

- с сохранением оформления считывать и записывать файлы, созданные в предыдущих версиях «Лексикона» для DOS;
- работать с документами, подготовленными в MS Word 6.0–8.0;
- обмениваться документами с другими текстовыми редакторами в формате RTF;
- работать с файлами в формате HTML;
- правильно переносить русские и английские слова;
- находить и исправлять орфографические ошибки в русских и английских словах;
- легко и быстро форматировать текст;
- использовать шаблоны типовых документов;
- создавать и сохранять стили оформления;
- включать в документ иллюстрации, таблицы, врезки;
- размещать таблицы на нескольких страницах;
- создавать многоуровневые списки с различным оформлением;
- создавать разделы с различным оформлением страниц;
- делать сноски на странице, в конце раздела или всего документа;
- набирать текст в несколько колонок;
- печатать брошюры, размещая на листе бумаги по две страницы так, чтобы из листов можно было сшить книжку;
- нумеровать различные объекты в документе (рисунки, таблицы и др.) и следить за правильностью номеров в ссылках;
- использовать гиперссылки как в одном документе, так и в нескольких, включая находящиеся в Internet.

«Лексикон-Верба» по функциональным возможностям близок к «Лексикону» 4.0 и отличается от него наличием средств криптографической защиты и дополнительными возможностями настройки инструментальных панелей.

СКЗИ «Верба-OW» построена по принципу открытого распределения ключей, в связи с чем в «Лексиконе-Верба» применяются открытые ключи пользователей, которые должны быть предварительно занесены в справочник открытых ключей, хранящийся на диске компьютера или на сервере локальной сети. Поскольку открытые ключи необходимо защищать от искажений и подделок, то для обмена ими по незащищенным каналам связи, например по электронной почте, в «Лексиконе-Верба» предусмотрено шифрование с использованием паролей.

В «Лексиконе-Верба» реализована достаточно развитая система управления ключами, что позволяет пользователю сосредоточиться на выполнении сервисных функций, представляемых программой.

Работа с ключами

Ключи могут поддерживать только шифрование или только электронную подпись, поэтому занесение и удаление соответствующих ключей производится по отдельности. Окно справочника открытых ключей вызывается кнопкой или командой из меню **Криптозащита ⇒ Справочник ключей**. Чтобы, например, добавить ключ в раздел подписей (ЭЦП), нажмите в этом разделе справочника кнопку +. Появится панель открытия файлов, с помощью которой следует выбрать файл с типом **Файл открытого ключа ЭЦП**. Таким же образом добавляются ключи шифрования. Их файлы имеют тип **Файл открытого ключа шифрования**. По умолчанию для открытого ключа цифровой подписи принято расширение имени файла LFX, для открытого ключа шифрования – PUB.

Для импорта открытых ключей можно и не открывать окно справочника, а нажать кнопку **Im** или дать команду **Криптозащита ⇒ Импорт ключей**. Появится панель импорта, на которой следует отметить, какие ключи надо импортировать (шифрования, подписи), и ввести соответствующие пути загрузки и имена файлов.

Чтобы другие пользователи системы могли читать ваши сообщения и отправлять свои, им необходимо передать ваш открытый ключ. Для экспорта своего открытого ключа нажмите кнопку **Ex** или дайте команду **Криптозащита ⇒ Экспорт ключей**. Появится панель экспорта файлов, на которой следует отметить сохраняемые открытые ключи (шифрования и подписи), ввести имена файлов и пути записи. Здесь же можно задать и пароль для защиты открытого ключа.

Для группы пользователей, работающих в локальной сети, можно вести общий справочник открытых ключей. При этом он располагается на компьютере администратора, который и обеспечивает его использование. Остальные пользователи получают доступ к справочнику (подкаталог HD в каталоге, где установлен «Лексикон-Верба») в режиме «только чтение». Для работы с сетевым справочником необходимо выбрать в меню **Криптозащита ⇒ Параметры** пункт **Использовать сетевой справочник**, в открывшемся диалоговом окне включить отметку и указать путь к справочнику на компьютере администратора.

Для использования функций криптографической защиты пользователь должен зарегистрироваться – загрузить свой ключ в память компьютера. «Лексикон-Верба» предлагает сделать это сразу же при запуске. Для загрузки ключа вставьте в устройство считывания (дисковод, карт-ридер, считыватель touch memory) соответствующий носитель информации с ключом. После загрузки ключ находится в памяти компьютера и для работы с функциями криптозащиты наличие дискеты в дисководе или смарт-карты в карт-ридере необязательно. Если вы не загрузили ключ при запуске «Лексикона», позже загрузку можно вызвать командой из меню **Криптозащита ⇒ Загрузка ключей** или кнопкой  .

Работа с ЭЦП

Для обеспечения целостности и достоверности обрабатываемых электронных документов данный программный продукт предоставляет пользователю возможность вызывать по мере необходимости функции генерации ЭЦП на электронный документ. Чтобы подписать документ:

1. Щелкните на панели инструментов по кнопке  или дайте команду **Криптозащита ⇒ Подписи**.
2. Нажмите в появившемся диалоговом окне **Цифровые подписи** кнопку **Подписать**. В строке состояния появится надпись **Будет подписано**.

Проверка проставленных в документе подписей выполняется автоматически при каждом открытии документа. До тех пор, пока под документом стоит электронная подпись, любые его изменения приведут к ее разрушению. Поэтому в самом «Лексиконе» редактировать подписанный документ просто невозможно: все попытки правки игнорируются. Если необходимо изменить текст, нужно удалить все проставленные в нем подписи, а после внесения исправлений заново подписать новую версию документа.

Кроме того, каждый подписывающий электронный документ пользователь системы может при этом внести свои комментарии. Добавка комментариев

не нарушает целостности документа и предыдущих подписей. Текст комментария также будет защищен электронной подписью, и в случае его искажения вы увидите соответствующее сообщение.

Использование функций шифрования

Кроме возможности использовать ЭЦП, «Лексикон-Верба» предоставляет пользователю процедуры зашифрования/расшифрования электронного документа. Вызов функций шифрования происходит следующим образом:

1. Нажмите на панели инструментов кнопку  или дайте команду **Криптозащита ⇒ Шифрование**.
2. Включите на появившейся панели сохранения файлов отметку **Шифровать**. Появится список пользователей, чьи открытые ключи есть в вашем справочнике. (Та же самая панель диалога вызывается и по команде **Документ ⇒ Сохранить как**; там тоже можно включать и выключать шифрование).
3. Отметьте в списке пользователей, которым будет открыт доступ к зашифрованному файлу, поставив отметки слева от имен. Кнопка **Для всех** помечает сразу всех имеющихся в списке пользователей, кнопка **Для себя** – только вас. Зашифрованный в режиме **Для себя** документ сможет прочесть только тот, кто его зашифровал.
4. Введите имя файла. Если до этого документ уже сохранялся без шифрования, «Лексикон» предложит сохранить зашифрованный документ в том же файле.
5. Нажмите кнопку **Сохранить**. Документ будет зашифрован и сохранен, после чего в строке состояния, в правом нижнем углу экрана, его статус будет изменен с **Общедоступно** на **Конфиденциально**. Далее документ будет сохраняться в зашифрованном виде до тех пор, пока вы не снимете шифрование.

Для отмены шифрования необходимо перед сохранением документа снять метку **Конфиденциально**. Шифрование всегда снимается при сохранении документа в файле другого формата.

Программно-аппаратный комплекс «Аккорд»

Программно-аппаратный комплекс (ПАК) «Аккорд» (ОКБ САПР и Инфокрипт), устанавливаемый на ЛРС, обеспечивает доступ к работе на нем только зарегистрированных пользователей, контроль целостности

аппаратной части ЛРС и ОС, а также управление доступом пользователей к аппаратным и программным ресурсам ЛРС.

ПАК «Аккорд» прошел испытания на соответствие требованиям Гостехкомиссии «Автоматизированные системы. Защита от несанкционированного доступа к информации. Классификация автоматизированных систем и требования по защите информации».

ПАК поставляется в составе: контроллер, съемник, идентификатор, комплект ПО и комплект эксплуатационной документации.

Дальнейший обзор посвящен аппаратному модулю доверенной загрузки «Аккорд-АМДЗ», созданному на основе ПАК защиты информации от несанкционированного доступа (НСД) и предназначенному для применения на ЛРС типа IBM PC AT. Задача этого комплекса – обеспечивать защиту ЛРС и информационных ресурсов от НСД, а также проводить контроль целостности файлов и областей жесткого диска (в том числе и системных) при многопользовательском режиме их эксплуатации.

Выбор данной реализации ПАК «Аккорд» был обусловлен тем, что программы, реализующие механизм контроля целостности комплекса, администрирования и аудит работы пользователей, защищены от подделки и несанкционированной модификации за счет их хранения в энергонезависимой памяти контроллера комплекса.

При этом обеспечивается режим доверенной загрузки в различных операционных средах: MS DOS, Windows 3.x, Windows 95, Windows NT, OS/2, UNIX.

Комплекс «Аккорд-АМДЗ» реализуется на основе контроллера «Аккорд-4+» (базовый) и его модификаций и по требованию заказчика дополнительно может поставляться со следующими блоками:

- аппаратный датчик случайных чисел – опция **R**;
- аппаратный датчик случайных чисел для криптографического применения – опция **К**;
- блокировка до трех физических каналов (FDD, YDD, CD-ROM и других отчуждаемых носителей) – опция **С**;
- внутренняя энергонезависимая память прямого доступа – опция **М1** (один модуль по 32 К × 8), опция М4 (четыре модуля по 32 К × 8).

Все модификации комплекса «Аккорд-АМДЗ»:

- могут использоваться на ЛРС с процессором 80386 и выше, объемом RAM 640 Кб при наличии свободного слота (ISA) на материнской плате ЛРС;

- содержат для идентификации пользователей уникальные персональные ТМ-идентификаторы DS 199x с объемом памяти до 64 Кбит и предусматривают регистрацию до 16 пользователей на ЛРС;
- используют для аутентификации пользователей пароль до 12 символов, вводимый с клавиатуры;
- блокируют загрузку ОС с гибких магнитных дисков;
- обеспечивают контроль целостности технических средств ЛРС до загрузки ОС;
- обеспечивают контроль целостности программ и данных до загрузки ОС, защиту от внедрения разрушающих программных воздействий (РПВ);
- поддерживают файловые системы следующих типов: FAT12, FAT16, FAT32, NTFS, HPFS, FreeBSD;
- осуществляют регистрацию действий пользователей в системном журнале, размещенном в энергонезависимой памяти контроллера (при наличии опции **М**);
- обеспечивают администрирование системы (регистрацию пользователей и персональных идентификаторов, назначение файлов для контроля целостности, контроль аппаратной части ЛРС, просмотр системного журнала).

Преимуществом комплексов средств защиты информации (СЗИ) от НСД семейства «Аккорд-АМДЗ» является проведение процедур идентификации, аутентификации и контроля целостности до загрузки операционной системы. Это обеспечивается перехватом управления контроллером комплекса во время так называемой процедуры ROM-SCAN, суть которой заключается в следующем.

В процессе начального старта после проверки основного оборудования BIOS компьютера начинает поиск внешних ПЗУ в диапазоне С 800:0000 – Е000:0000 с шагом в 8 К. Доказательством, что ПЗУ существует, является наличие AA55H в первом слове проверяемого интервала. Если данный признак обнаружен, то в следующем байте содержится длина ПЗУ в страницах по 512 байт. Затем вычисляется контрольная сумма всего ПЗУ, и если она корректна, производится вызов процедуры, расположенной в ПЗУ со смещением 3. Такая процедура обычно используется для инициализации.

В комплексе «Аккорд-АМДЗ» в этой процедуре проводится идентификация и аутентификация пользователя. При ошибке возврата из процедуры не происходит, то есть дальнейшая загрузка выполняться не будет.

Во время осуществления контрольных процедур (идентификация, аутентификация пользователя, проверка целостности) блокируется загрузка ОС с диска А.

При касании съемника информации происходит поиск предъявленного ТМ-идентификатора в списке зарегистрированных идентификаторов. Этот список хранится в энергонезависимой памяти контроллера комплекса «Аккорд-АМДЗ». Если предъявленный ТМ-идентификатор в списке обнаружен, то производится аутентификация пользователя и контроль целостности установленных в ПЭВМ (РС) технических и программных средств по перечню, создаваемому для каждого пользователя.

Для проведения процедуры аутентификации предусмотрен режим ввода пароля в скрытом виде – в виде символов <*>. Этим предотвращается возможность раскрытия личного пароля и использования утраченного (похищенного) ТМ-идентификатора.

Чтобы выполнить вышеуказанные условия, необходимо проконтролировать целостность технических и программных средств ЛРС перед каждым сеансом работы пользователя. Этим обеспечивается защита от несанкционированных модификаций и внедрения разрушающих программных воздействий (закладок, вирусов и т.д.).

Контроль целостности в комплексе «Аккорд-АМДЗ» реализован на аппаратном уровне (средствами контроллера) с использованием алгоритма пошагового (ступенчатого) контроля целостности. При этом обеспечивается корректная работа комплекса с загрузчиками различных файловых систем (Boot-менеджерами), что позволяет осуществить доверенную загрузку всех ОС и прикладного ПО при одновременной их установке на дисках или разделах дисков ЛРС.

Надежность функционирования системы защиты ЛРС от НСД при применении ПАК «Аккорд» обеспечивается выполнением следующих условий:

- на ЛРС с проверенным BIOS установлена проверенная операционная среда;
- достоверно установлена неизменность ОС, BIOS и программ для данного сеанса работы;
- в данной программно-аппаратной среде ЛРС не запускалось и не запускается никаких иных программ, кроме проверенных;
- исключен запуск проверенных программ в какой-либо иной ситуации, то есть вне проверенной среды, при установленном специальном ПО СЗИ от НСД;
- вышеперечисленные пункты выполняются в любой момент для всех пользователей, аутентифицированных защитным механизмом комплекса.

Программно-аппаратные комплексы «Криптон», «Криптон-Вето» и «Crypton Lite»

В качестве образца средств защиты, относящихся к семейству «Криптон» (Анкад), выберем ПАК «Криптон-4К/16», работающий в составе IBM PC и предназначенный для криптографической защиты данных.

ПАК «Криптон-4К/16» может применяться для организации защиты файлов электронных документов, содержащих государственную тайну.

Его базовое ПО позволяет:

- тестировать компьютер для настройки устройства (при установке);
- создавать и обслуживать ключевые системы пользователя;
- осуществлять электронную подпись документов;
- шифровать файлы, группы файлов и диски.

ПАК реализует криптографические алгоритмы, являющиеся государственными стандартами Российской Федерации: ГОСТ 28147-89, ГОСТ Р 34.11-94 и ГОСТ Р 34.10-94.

Алгоритм шифрования реализован аппаратно на основе новой специализированной большой интегральной схемы (СБИС) «Блюминг-1К», что позволяет гарантировать целостность алгоритма криптографического преобразования. Использование СБИС также позволяет загружать ключи шифрования до запуска операционной системы в регистры шифропроцессора и хранить их там в процессе обработки файлов без выгрузки в оперативное запоминающее устройство, что гарантирует сохранность ключей от несанкционированного доступа.

ПАК «Криптон-4К/16» имеет встроенный аппаратный датчик случайных чисел – это значительно повышает надежность ключей шифрования и электронной подписи.

ПАК «Криптон-4К/16» конструктивно выполнен как плата расширения персонального компьютера типа IBM PC. Для работы СКЗД необходимо:

- свободный разъем шины ISA;
- пространство в области адресов BIOS компьютера объемом 16 Кб;
- программное прерывание 4Ch;
- порты ввода/вывода с адресами 338, 33A, 33C, 33E, 738, 73A (hex) или 350, 352, 354, 356, 750, 752 (hex).

ПАК «Криптон-4К/16» имеет интерфейс к адаптеру смарт-карт, что позволяет хранить ключи шифрования и электронной подписи на смарт-карте.

Кроме того, у ПАК есть интерфейс к коннектору touch memory, что позволяет хранить ключи шифрования на таблетках touch memory.

В стандартный комплект поставки ПАК «Криптон-4К/16» входят:

- устройство криптографической защиты данных «Криптон-4К/16»;
- базовое ПО шифрования для MS DOS;
- ПО шифрования для Win32 с;
- программа электронной подписи CryptonSign для MS DOS;
- драйверы для работы в ОС Windows 95/NT 4.0.

Базовое ПО функционирует под MS DOS 3.1 и выше, Windows 3.1 – Windows NT. Чтобы работать под Windows 95/NT, необходимо предварительно установить драйверы для Win32 и библиотеку CryptonAPI (входит в комплект поставки драйверов). В Win32 базовое ПО функционирует как DOS-задача.

На основе ПАК «Криптон» в дальнейшем была построена система защиты информации «Криптон-Вето», функционирующая в среде MS DOS. *Программно-аппаратная система защиты информации* (СЗИ) от НСД «Криптон-Вето» предназначена для:

- ограничения доступа к персональному компьютеру;
- разграничения доступа пользователей к данным, расположенным на жестком диске компьютера;
- проверки целостности программ, разрешенных к использованию;
- предотвращения запуска неразрешенных программ;
- прозрачного шифрования логических дисков;
- создания и обслуживания ключевых систем шифрования, электронной подписи, СЗИ от НСД;
- шифрования файлов, групп файлов;
- электронной цифровой подписи файлов, групп файлов.

Криптографическое ядро системы составляет ПАК «Криптон-4» (4К/8, 4К/16). Использование ПАК «Криптон» позволяет при включении питания ЛРС и до загрузки ОС передать управление программным средствам ПАК, размещенным в ПЗУ. Они проверяют целостность ПО ЛРС и только при ее соблюдении передают управление ОС. Такой метод запуска обеспечивает защиту ПО ЛРС от любого искажения (несанкционированной модификации, воздействия вирусов, внесения программных закладок и т.д.).

Функциональным ядром системы являются:

- программа CryptonAccess в части оболочки защиты от НСД и прозрачного шифрования;
- базовое ПО СКЗД в части создания и обслуживания ключевой системы, шифрования и ЭЦП.

СЗИ основана на технологиях «прозрачного» шифрования логических дисков, на которые разбивается жесткий диск (последний из логических дисков не шифруется и отводится под СЗИ). Прозрачное шифрование заключается в том, что хранящиеся на диске данные автоматически расшифровываются при обращении к ним пользователя, а при записи на диск зашифровываются. Также предусмотрена мандатная организация доступа к дискам по уровням конфиденциальности. Каждому логическому диску присваивается уровень конфиденциальности, а каждому пользователю – уровень доступа – от 0 до 7. Пользователю доступ к диску разрешается, если уровень конфиденциальности не превышает уровня доступа.

СЗИ предполагает наличие администратора безопасности, который определяет взаимодействие между управляемыми ресурсами: пользователями, программами, логическими дисками, дисководами и последовательными портами.

Для каждого пользователя администратор определяет идентификатор и пароль, которые необходимо ввести при запуске компьютера, а также право доступа к открытым и шифруемым логическим дискам: а) недоступен (недоступный диск не виден из MS DOS, поэтому системный диск не должен быть заблокирован); б) доступен только для чтения (используется в целях обеспечения целостности и достоверности хранящейся информации) и в) доступен для чтения и записи (не отличается от обычного логического диска.

Администратор определяет перечень разрешенных к запуску на компьютере программ. Они подписываются электронной цифровой подписью администратора и проверяются на целостность при запуске.

СЗИ ведет журнал работы (доступный только администратору), в котором регистрируются следующие события:

- установка системы на компьютере;
- вход пользователя в систему;
- попытка доступа к запрещенному логическому диску;
- зашифрование/расшифрование/перешифровывание диска;
- добавление нового пользователя;
- смена полномочий пользователя;
- удаление пользователя;
- причины останова системы;
- попытка запуска неразрешенной программы;
- нарушение целостности разрешенной программы.

При установке системы на компьютере и при каждом его последующем включении проверяется целостность следующих элементов:

- идентификатора СЗИ;
- ядра комплекса программ CryptonAccess;
- системных областей системного диска;
- таблицы полномочий пользователей.

Система не препятствует шифрованию файлов пользователями, что позволяет защитить их от администратора.

Подобный подход к организации защиты имеет много положительных сторон, однако при использовании СЗИ «Криптон-Вето» под управлением ОС Windows 3.1 или 3.11 возникают некоторые трудности. Например, попытка записи утилиты Smartdrv (для кэширования по записи и чтению) на логический и шифруемый диск, доступный только для чтения, вызовет «зависание» системы. Также невозможно использовать 32-разрядный режим доступа к логическому диску. Во время работы ОС Windows использует файл подкачки. Если разместить его на нешифруемом диске, складывается ситуация, когда секретные данные в этом конкретном файле записываются в открытом виде, что может привести к несанкционируемому доступу к защищаемой информации. Расположение же файла подкачки на шифруемом диске вызовет замедление работы системы примерно в 10–15 раз.

Crypton Lite – это пакет программ шифрования и электронной цифровой подписи, совместимый с устройствами серии «Криптон», обеспечивающий гарантированную защиту файлов электронных документов.

Crypton Lite реализует отечественные стандарты:

- ГОСТ 28147-89 – в части шифрования;
- ГОСТ Р 34.10-94 – в части функции хэширования;
- ГОСТ Р 34.11.-94 – в части цифровой подписи.

Ключевая система шифрования – симметричная. Ключи шифрования (256 бит) и электронной подписи (512 бит – открытый и 256 бит – секретный) создаются пользователем с помощью встроенного датчика случайных чисел.

Пакет предполагает и операторскую работу из оболочки (интерфейс Windows-программы наиболее удобен для пользователя, так как встроен в Windows Explorer), и шифрование/подпись вызывается при нажатии правой клавиши мыши. Аналогичный интерфейс имеет широко распространенная в Internet программа PGP. Для автоматической обработки файлов она вызывает командную строку. В случае, когда необходимо встроить функции шифрования/подписи непосредственно в клиентскую программу, рекомендуем использовать библиотеки соответствующих функций.

Программное обеспечение позволяет осуществлять:

- шифрование файлов, групп файлов и разделов дисков;
- электронную подпись файлов юридических и финансовых документов и ее проверку.

Пакет обрабатывает электронные документы со скоростью более 1,5 Мб/с в зависимости от производительности компьютера.

В пакет заложена концепция расширения системы без ограничения на число ключей, подписей, пользователей. Он используется в системе электронного документооборота ЦБ РФ, а также для защиты информации, циркулирующей между ЦБ и коммерческими банками. Целесообразно использовать пакет в системах электронного документооборота и в системах клиент-банк.

Создание замкнутой программной среды

Большая часть угроз, связанных с безопасностью потоков информации, исходит от программной среды, в которой средству защиты или криптографическому приложению приходится работать. При этом обеспечение безопасной окружающей среды разбивается на две области:

- создание замкнутой программной среды. Она обеспечивает защиту от негативного влияния прикладного и системного ПО, возникающего в связи с потенциальной опасностью внедрения в программную среду программ-закладок, вирусов, программ-шпионов и т.д;
- создание доверенной программной среды. В рамках данного пункта обеспечивается анализ программной среды на наличие недокументированных возможностей и ошибок, способных оказать негативное влияние на безопасность функционирования средств защиты.

Очевидно, что для различных операционных систем (ОС) создание замкнутой программной среды будет заключаться в различных подходах и применении различных средств.

Операционная система MS DOS

В этой однозадачной системе любая программа располагает всей свободной памятью и всеми ресурсами. Опасность для криптографического приложения может исходить со стороны несанкционированных драйверов и резидентных программ, перехватывающих прерывания. В связи с этим для MS DOS необходимо выполнение следующих требований к окружающей среде:

- доверенная загрузка ОС, означающая гарантированную загрузку с легального носителя с идентификацией пользователя и контролем его прав на доступ к ресурсам;
- контроль целостности ОС и библиотек на этапе доверенной загрузки;
- контроль запуска задач – запуск задач только из фиксированного списка.

При формулировке дальнейших требований к окружающей среде будем исходить из того, что средства защиты не оставляют за собой «опасных следов» в памяти и дисках. В связи с этим единственной ситуацией, при которой «опасные следы» могут сохраниться на диске, является аварийная ситуация, связанная с обесточиванием. Именно при отсутствии питания на диске в потерянных кластерах может остаться открытая информация из временных файлов. Отсюда вытекают следующие требования: а) при восстановлении питания сборка потерянных кластеров после запуска системы на правах санкционированного доступа может осуществляться только с применением санкционированных программных средств; б) последующее шифрование (если информация в потерянных кластерах представляет ценность) или физическое затирание потерянных кластеров; в) удаление затертого файла с потерянными кластерами.

Заметим, что затирание может быть сведено к шифрованию на случайном ключе с последующим его обнулением.

Операционные системы Windows 3.x и Windows 95

Поскольку прикладные задачи в Windows 3.x и Windows 95 запускаются в едином адресном пространстве, то при использовании этих ОС должны выполняться все перечисленные выше требования к окружающей среде. Вместе с тем многозадачность данных систем накладывает следующие дополнительные требования:

- исключение возможности запуска программ с внешнего носителя, а также программ, переписанных с внешнего носителя на внутренний. Внешние носители могут содержать только данные, необходимые для работы криптографического приложения, и не могут содержать никаких исполняемых модулей;
- исключение возможности запуска программ из сети;
- анализ программ, запускаемых одновременно с приложением, с целью исключения средств разработки (например, компиляторы), средств просмотра и редактирования памяти (различные отладчики).

Операционная система Windows NT

В данной ОС, по сравнению с Windows 95, каждая задача выполняется в собственном адресном пространстве, и к тому же область памяти, отведенная под ядро ОС, недоступна пользовательским приложениям, то есть находится в монопольном владении этой операционной системы. Однако при этом не исключается возможность вторжения посторонней программы в криптографический процесс (раздел 3.4.2, пункт 4). Отсюда следует, что для Windows NT должны выполняться все требования к изолированной среде, перечисленные для MS DOS и Windows 95.

Выполнение всех приведенных выше требований для соответствующих ОС позволяет создать замкнутую в полном объеме программную среду. Другими словами, подобный комплекс мер защиты обладает достаточной полнотой в том отношении, что его выполнение блокирует возможные стратегии потенциального нарушителя при нападении на СКЗИ.

3.2.3. Организационно-технические меры защиты локальной рабочей станции

Как уже говорилось, надежность и эффективность применения криптографических и других средств защиты информации может быть достигнута только при неукоснительном выполнении определенных организационно-технических требований. При этом следует помнить, что в некоторых случаях меры организационно-технического характера способны выступать надежной заменой любых других средств защиты информации. Они являются дополнительным уровнем обеспечения выбранной политики безопасности, и конкретный набор подобных мер всегда зависит от ситуации. Тем не менее существует типовой перечень подобных мероприятий, который применим в любых условиях и составляет как бы основание для выстраивания всей политики, направленной на сохранение в неприкосновенности потоков информации. Этот перечень основан на выполнении специальных требований, которые в зависимости от их функционального назначения можно разбить на следующие группы:

- требования по размещению технических средств;
- рекомендации по установке СЗИ;
- меры по обеспечению надежности функционирования СЗИ, установленных на ЛРС.

Реализация организационно-технических мероприятий по защите информации должна начинаться с разработки соответствующих инструкций

и рекомендаций, а также создания структурных подразделений, ответственных за реализацию политики безопасности и контроль над их неукоснительным соблюдением.

Требования по размещению технических средств

При размещении технических средств, поддерживающих системы криптографической защиты информации (СКЗИ), следует руководствоваться следующими рекомендациями:

- расположение режимных помещений и размещенного в них оборудования должно исключать возможность бесконтрольного проникновения в эти зоны посторонних лиц и гарантировать сохранность находящихся в них конфиденциальных документов;
- входные двери должны быть оборудованы замками, гарантирующими санкционированный доступ в режимные помещения в нерабочее время. Для контроля над входом должны устанавливаться шифрзамки;
- окна и двери необходимо оборудовать охранной сигнализацией, связанной с пультом централизованного наблюдения за всеми сигнальными устройствами;
- размещение оборудования и технических средств, предназначенных для обработки конфиденциальной информации, должно соответствовать требованиям техники безопасности, санитарным нормам, а также требованиям пожарной безопасности;
- в режимные помещения по утвержденному списку допускаются руководство учреждения, сотрудники отдела безопасности и исполнители, имеющие прямое отношение к обработке, передаче и приему конфиденциальной информации;
- допуск в помещения вспомогательного и обслуживающего персонала (уборщицы, электромонтеры, сантехники и т.д.) производится только при служебной необходимости в сопровождении ответственного за режим, причем нужно позаботиться о мерах, исключающих визуальный просмотр конфиденциальных документов;
- каждый исполнитель работ в качестве пользователя сети конфиденциальной связи обязан зарегистрироваться у администратора службы безопасности;
- внутренняя планировка и расположение рабочих мест в режимных помещениях должны обеспечивать исполнителям сохранность доверенных им конфиденциальных документов и сведений;
- по окончании рабочего дня режимные помещения необходимо закрывать и опечатывать. Затем их (с опечатанными входными дверями)

сдают под охрану отделу безопасности или дежурному по предприятию (по установленному порядку) с указанием времени приема-сдачи и отметкой о включении и выключении охранной сигнализации в журнале учета;

- сдачу ключей и режимных помещений под охрану, а также получение ключей и вскрытие режимных помещений производят сотрудники, работающие в этих помещениях и входящие в утвержденный руководством учреждения список с образцами подписей этих сотрудников. Список хранится у начальника охраны или у дежурного по учреждению;
- перед вскрытием режимных помещений должна быть проверена целостность оттисков печатей и исправность замков. При обнаружении нарушения целостности оттисков печатей, повреждения замков или других признаков, указывающих на возможное проникновение в эти помещения посторонних лиц, помещение не вскрывается, а о случившемся немедленно информируется руководство и отдел безопасности;
- в случае утраты ключа от входной двери режимного помещения немедленно ставится в известность отдел безопасности учреждения;
- на случай пожара, аварии или стихийного бедствия должны быть разработаны специальные инструкции, утвержденные руководством учреждения, в которых предусматривается вызов администрации, должностных лиц, вскрытие режимных помещений, очередность и порядок спасения конфиденциальных документов и дальнейшего их хранения;
- в помещениях, где находятся СЗИ, запрещается приносить и использовать радиотелефоны и другую радиоаппаратуру.

Рекомендации по установке СЗИ

При установке ПО, входящего в состав СЗИ, нужно руководствоваться следующими рекомендациями:

- установка СЗИ производится только лицами, имеющими соответствующую лицензию;
- аппаратную часть ЛРС, на которую устанавливается СЗИ, необходимо проверить на отсутствие аппаратных закладок;
- все программное обеспечение ЛРС, на которой будет устанавливаться СЗИ, должно быть лицензионно чистым, при этом не допускается наличия средств разработки и отладки программ;
- перед установкой СЗИ необходимо проверить ПО ЛРС на отсутствие вирусов и программных закладок;

- должны быть предприняты меры, препятствующие извлечению аппаратной части СЗИ из ЛРС; системные блоки ЛРС должны быть опечатаны специально выделенной для этих целей печатью. Наряду с этим допускается применение других средств контроля за доступом к ЛРС;
- к эксплуатации СЗИ допускаются лица, прошедшие соответствующую подготовку и изучившие эксплуатационную документацию данного СЗИ;
- перед установкой ПО СЗИ необходимо осуществить контроль целостности дистрибутива;
- после завершения установки должны быть приняты меры, необходимые для осуществления ежедневного контроля за установленным СЗИ, а также его программным и аппаратным окружением.

Меры по обеспечению надежности функционирования СЗИ, установленных на ЛРС

В этом разделе представлены основные рекомендации по организационно-техническим мерам защиты, обеспечивающим безопасность функционирования рабочих мест со встроенными СЗИ:

- правом доступа к рабочим местам с установленными СЗИ могут обладать только лица, прошедшие соответствующую подготовку. Администратор безопасности должен ознакомить каждого абонента автоматизированной системы, использующего СЗИ, с правилами пользования или с другими нормативными документами, созданными на их основе;
- должностные инструкции администратора безопасности (его заместителя) и ответственного исполнителя не должны противоречить правилам пользования спецаппаратурой и другим нормативным документам, созданным на их основе;
- администратор безопасности обязан периодически проводить контроль целостности и легальности установленных копий ПО на всех ЛРС со встроенной СЗИ с помощью программ контроля целостности;
- при обнаружении «посторонних» (незарегистрированных) программ, нарушенной целостности программного обеспечения или выявлении факта повреждения печатей на системных блоках работа на ЛРС прекращается. По данному факту должно быть проведено служебное расследование комиссией в составе представителей служб информационной безопасности предприятия-владельца сети и предприятия-абонента сети, где произошло нарушение, а также организованы работы по анализу и ликвидации негативных последствий данного нарушения;

- пользователь должен запускать только те приложения, которые разрешены администратором безопасности;
- установленное программное обеспечение не должно содержать средств разработки и отладки приложений, а также средств, позволяющих осуществлять несанкционированный доступ к системным ресурсам;
- в инструкцию по использованию рабочей станции должен быть включен пункт, запрещающий оставлять без контроля вычислительные средства, входящие в состав СЗИ, при включенном питании и загруженном специальном программном обеспечении СЗИ;
- запретить допуск пользователей в режим конфигурирования BIOS (например, с использованием парольной защиты);
- исключить возможность работы на ЛРС, если встроенные тесты выдают отрицательный результат во время ее начальной загрузки;
- пароли, назначаемые пользователям, должны отвечать требованиям соответствующих инструкций и нормативных документов;
- в случае использования ЛРС несколькими операторами с различными ключами нельзя производить выгрузку ключевой информации (перезагрузку ЛРС).

При этом запрещается:

- осуществлять несанкционированное копирование ключевых носителей;
- разглашать содержимое носителей ключевой информации или передавать сами носители лицам, не имеющим к ним допуска; выводить ключевую информацию на дисплей и принтер (за исключением случаев, предусмотренных данными правилами);
- вставлять ключевой гибкий диск (или другой ключевой носитель) в дисковод ЛРС (или в другое устройство считывания) в режимах, не предусмотренных штатным расписанием, а также в дисководы других ЛРС;
- записывать на ключевые носители постороннюю информацию;
- подключать к ЛРС дополнительные устройства и соединители, не предусмотренные в комплектации;
- работать на компьютере, если во время его начальной загрузки не проходит встроенный тест ОЗУ, предусмотренный в ЛРС;
- вносить какие-либо изменения в программное обеспечение СЗИ;
- несанкционированно устанавливать, создавать и выполнять на ЛРС посторонние программы;
- использовать бывшие в работе ключевые носители для записи новой информации без предварительного уничтожения на них ключевой информации;
- осуществлять несанкционированное вскрытие системных блоков ЛРС.

3.2.4. Штатные средства защиты современных операционных систем на примере Windows NT

Некоторые вопросы защиты ЛРС от возможных атак потенциального нарушителя могут быть решены с использованием штатных средств, входящих в состав современных операционных систем (в качестве конкретной ОС рассматривается Windows NT 4.0).

На сегодняшний день данная система является одной из самых распространенных ОС, кроме того, ей присущи следующие положительные качества:

- защищенность. Национальное агентство безопасности правительства США в августе 1995 года сертифицировало соответствие автономного варианта ОС Windows NT Workstation 3.51 SP3 спецификациям уровня защиты С2. В то же время стоит отметить, что текущая версия 4.0 SP3 не имеет подобного сертификата, кроме того, сертификации подвергался автономный вариант ОС Windows NT Workstation;
- развитая сетевая поддержка. ОС Windows NT имеет в своем составе развитые сетевые средства и позволяет осуществлять информационный обмен по протоколам TCP/IP, IPX/SPX, Netbeui и AppleTalk, а также поддерживает маршрутизацию протоколов TCP/IP и IPX/SPX. Компанией Microsoft разработаны средства, позволяющие осуществлять клиентам сети прозрачный доступ к ресурсам сетевых ОС NetWare и SNA, а также эмулировать сервер NetWare для клиентов одноименной сети; при этом централизованное административное управление и контроль за всей сетью сохраняются. Клиентам сети представляется также возможность удаленного доступа к ресурсам сервера Windows NT через глобальные сети по линиям связи ISDN, frame relay, X.25 и асинхронным коммутируемым каналам. Обеспечивается также одновременное функционирование в сети архитектуры «клиент-сервер» Windows NT нескольких одноранговых подсетей Windows for Workgroups. Таким образом, Windows NT позволяет организовывать гетерогенные сетевые конфигурации, объединяющие различные аппаратные и программные платформы единой политикой безопасности, равными возможностями клиентов и централизованным администрированием;
- поддержка нескольких аппаратных платформ и симметричная мультипроцессорная обработка. Windows NT Server может работать как на Intel-совместимых компьютерах с процессорами 386, 486 и Pentium, так и на компьютерах со следующими типами RISC-процессоров: PowerPC, MIPS R4000 и DEC Alpha. Возможность использования аппаратных

платформ с несколькими процессорными элементами поднимает в ряде задач производительность Windows NT на уровень компьютеров классов мини-ЭВМ и mainframe. Совместно с поддержкой POSIX-приложений указанная характеристика позволяет утверждать о переносимости Windows NT;

- приоритетная вытесняющая многозадачность и многопоточность. Используемая в Windows NT система приоритетов позволяет наиболее оптимально распределить между задачами процессорное время, гарантируя при этом своевременное выполнение критических участков кода и корректную обработку исключений. Разработанные особым образом приложения могут запускать во время своего выполнения несколько собственных потоков, распределяя выделенные приложению вычислительные ресурсы оптимальным образом;
- виртуальная память. Windows NT, являясь многозадачной и многопользовательской оперативной системой, предоставляет надежные механизмы защиты ядра ОС от прикладных процессов и задач, решаемых пользователями друг с другом. *Своппинг* (swapping) на жесткий диск позволяет выделить активной задаче практически все доступные вычислительные ресурсы компьютера;
- поддержка нескольких файловых систем. Помимо возможности выполнения приложений, написанных для других операционных систем (DOS, 16-разрядные приложения Windows, 16-разрядные текстовые приложения OS/2, POSIX-приложения), Windows NT поддерживает несколько файловых систем: стандартную FAT и собственную NTFS;
- многоцелевые функциональные возможности. Сервер Windows NT поддерживает следующие типы услуг:
  - файл-сервер;
  - сервер печати;
  - сервер приложений;
  - сервер удаленного доступа;
  - intranet-сервер;
  - сервер архивирования.

Основная задача подсистемы безопасности Windows NT состоит в отслеживании и управлении порядком доступа к тому или иному объекту, а также фиксации и контроле попыток несанкционированной эксплуатации пользователями системных объектов (файлов, каталогов, разделяемых принтеров, баз данных реестра и т.д.). Подсистема безопасности хранит определенную информацию о каждом пользователе, группе пользователей и каждом защищенном объекте, а также идентифицирующую информацию

о каждом пользователе и каждой группе пользователей. На ее основе проверяются права пользователя на доступ к объекту, и, если таковые отсутствуют, подсистема защиты отказывает ему в разрешении.

В целом подсистема безопасности ОС Windows NT состоит из следующих компонентов (рис. 3.3):

- процессы входа в систему (logon-процессы), принимающие запросы на вход от пользователей. Они включают интерактивный (локальный) logon, показывающий окно для ввода пароля, а также удаленный logon, который позволяет получать доступ удаленным пользователям к процессам сервера Windows NT;
- компонент локальной авторизации (Local Security Authority), который определяет, имеет ли пользователь право доступа к системе. Этот компонент является центральным в подсистеме безопасности Windows NT. Он генерирует маркер доступа, управляет локальной политикой безопасности и предоставляет интерактивный сервис администратору. Локальная авторизация также управляет регистрацией событий и записывает в журнал регистрации события аудита, генерируемые диспетчером доступа;
- менеджер управления счетами пользователей (Security Account Manager, SAM), который строит базу данных счетов пользователей. Эта база данных содержит информацию обо всех счетах пользователей и группах пользователей. Менеджер SAM предоставляет сервис по их полномочиям, используемый компонентом локальной авторизации;
- диспетчер доступа. В его функции входит проверка права пользователя на доступ к объекту для выполнения затребованных им операций. Этот компонент реализует политику управления доступом и следит за выполнением требований по регистрации, определенных подсистемой локальной авторизации. Он предоставляет сервисное обслуживание как в пользовательском режиме, так и в режиме ядра при попытках пользователей и процессов получить доступ к объекту. Этот компонент также генерирует соответствующие сообщения регистрации.

Организация управления доступом

В основе этой подсистемы лежит избирательная политика управления доступом, в соответствии с которой регламентируется доступ субъектов к объектам. Защищенные объекты Windows NT подразделяются на:

- файлы и каталоги на NTFS-разделах;
- объекты пользователя (окна, ресурсы приложений и т.д.);

- объекты ядра ОС;
- объекты реестра;
- частные (private) объекты. Частные объекты создаются и обрабатываются прикладными программами (СУБД, электронными таблицами и т.д.) и служат для реализации в прикладных программах собственного механизма разграничения доступа. Частными объектами являются, например, защищенные поля таблиц Microsoft Access.

С каждым объектом системы связаны возможные выполняемые операции и типы доступа к экземплярам объекта, а также атрибуты безопасности объекта.

Атрибуты безопасности каждого защищаемого объекта задаются в специальной структуре – так называемом *дескрипторе безопасности* (security descriptor). Он связан с объектом и содержит информацию о владельце и группе владельцев объекта, а также списки управления доступом к данному объекту (Access Control List, ACL).

Дескриптор безопасности включает в себя всю информацию о владельце объекта и правах доступа пользователей к объекту. Он может содержать *идентификатор безопасности* (SID, см. ниже) владельца объекта, идентификатор безопасности его первичной группы и *список управления доступом* (ACL), разъясняющий права различных пользователей и групп пользователей по доступу к данному защищаемому объекту.

Список управления доступом к объекту состоит из элементов управления доступом (Access Control Entry, ACE). Каждый защищенный объект может иметь два списка управления доступом: *пользовательский* и *системный*.

Пользовательский список управляется владельцем объекта и определяет права доступа конкретных пользователей и групп к объекту. Когда этот список имеется у объекта, но не содержит ни одного элемента управления доступом, считается, что доступ к объекту запрещен. Однако если избирательный список у объекта отсутствует, это означает отсутствие защиты у объекта (доступ к объекту разрешен в любой форме).

С объектом может быть также связан и системный список управления доступом, контролируемый исключительно администратором. Системный список управления доступом предназначен для указания на необходимость регистрации любых попыток получения пользователями доступа к защищаемому объекту.

Администратор или владелец объекта может назначать полномочия, разрешающие или запрещающие определенным пользователям и группам получать требуемый тип доступа к конкретным объектам.

В Windows NT пользователи и группы идентифицируются так называемыми идентификаторами безопасности (security identifiers, SID). Идентификатор безопасности – это структура переменной длины, уникально идентифицирующая пользователя или группу. Эти идентификаторы хранятся в базе данных системы защиты, и любое приложение может их использовать посредством функций Win32 API. Идентификаторы безопасности используются для указания:

- владельца объекта и группы в дескрипторе защиты объекта;
- получателя разрешений на доступ в элементах (ACE) списка управления доступом (ACL) к объекту;
- пользователя и всех групп, в которые он входит, в так называемом *маркере доступа* (access token) процесса (пользователя). (Описание маркера доступа см. ниже.)

Для выполнения некоторых операций пользователю могут быть присвоены специальные привилегии. Содержащаяся в маркере доступа дополнительная информация о привилегиях пользователя (процесса) используется для усиления избирательного (реализованного на основе ACL) контроля доступа. Диспетчер доступа использует значения привилегий процессов для управления их доступом к системным ресурсам, таким как системное время, признак активизации отладочного режима, признак принадлежности к ядру системы защиты (trusted computer base) и т.п.

При входе в систему каждый пользователь проходит процедуры идентификации и аутентификации. В случае успешного прохождения процедуры аутентификации каждому пользователю присваивается специальная метка – маркер доступа, содержащий идентификатор этого пользователя, идентификаторы всех групп, в которые входит данный пользователь, а также иную информацию.

Поскольку ОС Windows NT поддерживает работу в режиме «клиент-сервер», то в архитектуре ее подсистемы защиты имеется два класса субъектов:

- простой субъект. Это процесс, которому присвоен контекст безопасности при входе соответствующего пользователя в систему по logon. Такой процесс не реализует функции защищенного сервера, который может иметь других субъектов в качестве своих клиентов;
- серверный субъект. Это процесс, запущенный как защищенный сервер (как подсистема Win32) и обслуживающий других субъектов в качестве клиентов. В связи с этим серверный субъект обычно имеет контекст безопасности того клиента, от имени которого он действует.

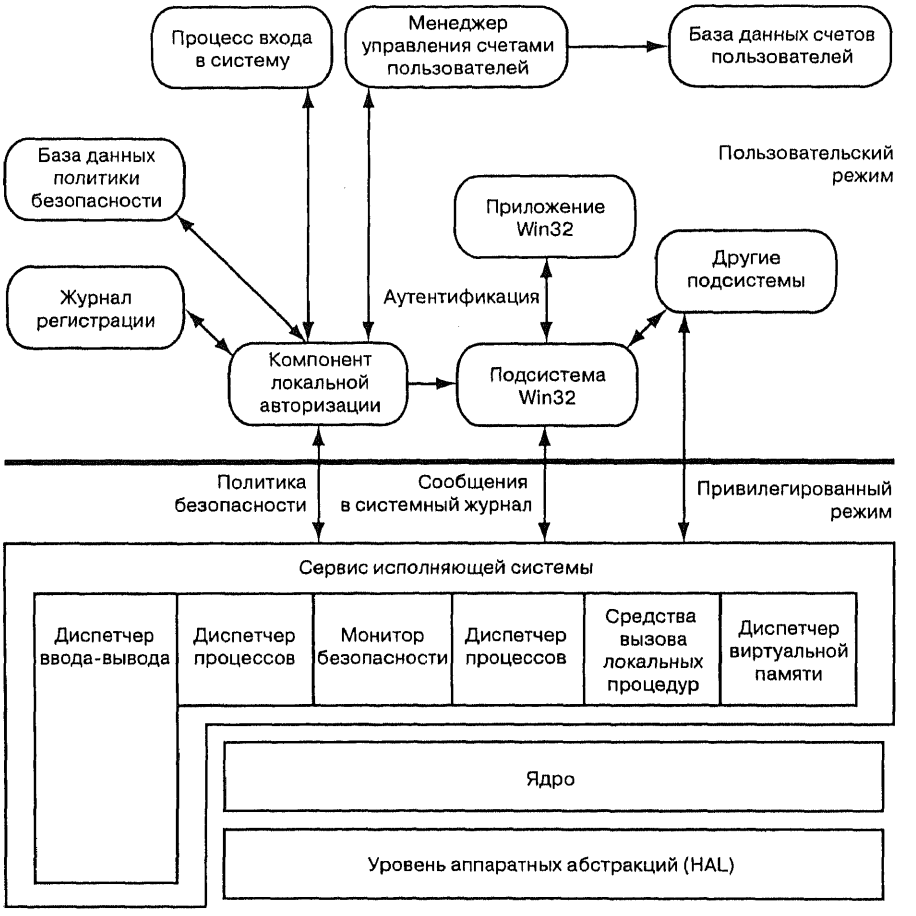

Рис. 3.3. Компоненты подсистемы безопасности

Windows NT позволяет одному потоку получать атрибуты безопасности другого процесса при помощи механизма представления (impersonation). Например, рабочий поток сервера обычно представляет процесс клиента (использует маркер доступа клиента) для получения разрешений на доступ к объектам, к которым начальный (управляющий) поток сервера доступа не имеет.

Хотя ОС этого типа имеет достаточно развитую систему обеспечения информационной безопасности, все равно существуют потенциальные уязвимости, приводящие к успешному проведению атак на Windows NT.

Имеет смысл в рамках документированных возможностей системы провести детальный анализ стратегий нападения, которые мог бы реализовать

потенциальный нарушитель, являющийся к тому же зарегистрированным пользователем в ОС. В этом случае злоумышленник в состоянии запустить любой процесс, заставить стартовать любой драйвер или редактировать базу реестра только на правах непривилегированного пользователя системы Windows NT.

Для определения потенциальных возможностей нарушителя рассмотрим механизмы Win32 API для Windows NT, позволяющие реализовать НСД. Хотя приложения в ОС Windows NT имеют свое виртуальное пространство памяти, доступ в которое ограничен со стороны других прикладных процессов, тем не менее в существующей версии Win32 API имеется целый ряд механизмов, позволяющих одному процессу получить доступ в виртуальную память, используемую другим процессом.

Итак, к этим возможностям относятся:

- наличие функций, позволяющих получать доступ на чтение и запись в память другого процесса (WriteProcessMemory и ReadProcessMemory);
- наличие функций, получающих контекст выполнения какого-либо пользовательского процесса (GetThreadContext и SetThreadContext), что позволяет одному процессу получить значения системных регистров другого, в том числе и регистра указателя стека; подобная особенность может позволить модифицировать стек процесса другим процессом;
- наличие механизма ловушек (hook), при помощи которых один процесс может реализовать перехват сообщений другого процесса;
- существует также способ запуска внешней процедуры в адресном пространстве другого процесса и использование процедуры удаленного запуска потока (CreateRemoteThread) с последующей модификацией области стека и загрузки необходимой DLL;
- программа нарушителя обладает способностью передавать в драйвер режима ядра команды вызова процедур. Драйвер верхнего уровня, как правило, выполняется в контексте вызываемого процесса, поэтому прикладному процессу для работы в режиме ядра достаточно передать вызываемому драйверу просто адрес вызываемой процедуры. Таким образом обеспечивается доступ из прикладного процесса ко всем ресурсам системы;
- особенностью ОС Windows NT является отсутствие механизмов предотвращения выхода за пределы локального стека потока, что может привести к проведению атак типа «переполнение буфера» (buffer overflow).

Использование тех или иных документированных возможностей API Windows NT для «взлома» штатных средств защиты данной ОС зависит от

уровня привилегий, полученных пользовательским процессом при регистрации. Можно выделить следующие группы привилегий (по убыванию уровня):

- права администратора, включающие полный контроль операционной системы;
- право перезапуска операционной системы и совокупность прав для установки драйвера режима ядра;
- право редактирования базы реестра;
- право запуска прикладной программы.

На сегодняшний день для пользователей, склонных к неправомерным поступкам, существует ряд потенциальных возможностей получать дополнительные привилегии, например:

- программа нарушителя обнаруживает адрес функции Win32 API OpenProcess, изменяет ее код так, чтобы результат выполнения был всегда положительным. Затем с помощью функции DebugActiveProcess один из системных процессов, работающих от имени ОС, в режиме отладки создает в его рамках поток, который вносит текущего пользователя в группу Administrator;
- использование ошибки в процедуре NetAddAtom (ntoskrnl.exe) или функции 4346 (win32k.sys). Указанные ошибки позволяют изменить переменные или код ядра ОС и получить рядовому пользователю права администратора системы;
- перехват трафика внутри сети и использование того факта, что после аутентификации пользователя в домене Windows NT в пакете NetLogonSamLogon, передаваемом контроллером домена компьютеру, сведения о группах, членом которых является пользователь, передаются в открытом виде. Установив в сети proxy-сервер, злоумышленник может заменить SID некоторой глобальной группы на SID группы DomainAdmins, после чего на локальном компьютере получит права члена группы DomainAdmins.

Таким образом, потенциальный нарушитель может получать практически неограниченные полномочия в ОС. Это позволит ему запускать не только прикладные программы в контексте любого зарегистрированного пользователя системы, реализующие вышеперечисленные стратегии использования особенностей Win32 API, но также устанавливать фильтр драйвера хранения ключей или фильтр файловой системы. Такие фильтры позволяют перехватывать или ключи, или открытую информацию.

Например, получая привилегии редактирования базы реестра можно включить нежелательную *динамически подключаемую библиотеку* (DLL) в список AppInit_DLLs. Он содержит множество DLL, которые отображаются при старте ОС на адресное пространство каждого процесса, использующего библиотеку User32.DLL, то есть фактически для каждой запускаемой программы, представляющей собой GUI-приложение.

Кроме документированных возможностей, предоставляемых нарушителю интерфейсом Win32 API, существует целый ряд уязвимостей, которые подразделяются на следующие категории:

- вызванные ошибками или недоработками в ПО Windows NT;
- вызванные особенностью функционирования Windows NT;
- вызванные некорректной настройкой Windows NT.

Таким образом, кроме корректной настройки ОС Windows NT и использования Service Pack для устранения ошибок при работе данной ОС, необходимо применение дополнительных средств защиты.

3.2.5. Аудит

Аудит связан с действиями (событиями), так или иначе затрагивающими безопасность системы. К их числу относятся:

- вход в систему (успешный или нет);
- выход из системы;
- обращение к удаленной системе;
- операции с файлами (открыть, закрыть, переименовать, удалить);
- смена привилегий или иных атрибутов безопасности (режима доступа, уровня благонадежности пользователя и т.п.).

Можно назвать и другие события, например смену набора регистрируемых действий. Полный перечень событий, потенциально подлежащих регистрации, зависит от избранной политики безопасности и от общей специфики системы.

Если фиксировать все совершаемые операции, объем регистрационной информации, скорее всего, будет расти слишком быстро, а эффективно проанализировать ее будет невозможно. Для обеспечения гибкости построения системы аудита должно быть предусмотрено наличие средств выборочного протоколирования, способных «следить» не только за пользователями (особенно за подозрительными), но и за совершаемыми событиями.

С помощью этого метода можно держать под контролем пользователей, имеющих специфическую репутацию, и реконструировать прошедшие

события. «Слежка» важна, в первую очередь, как профилактическое средство. Можно надеяться, что многие злоумышленники воздержатся от нарушений режима безопасности, поскольку знают, что их действия фиксируются. Реконструкция событий позволяет проанализировать случаи нарушений, понять, почему они произошли, оценить размеры ущерба и принять меры по недопущению подобных нарушений в будущем.

При протоколировании события необходимо, по крайней мере, записывать информацию такого рода:

- дату и время события;
- уникальный идентификатор;
- отмечать пользователя, являющегося инициатором действия;
- тип события;
- результат действия (успех или неудача);
- источник запроса (например, имя терминала);
- имена затронутых объектов (например, открываемых или удаляемых файлов).
- описание изменений, внесенных в базы данных защиты (например, новая метка безопасности объекта);
- метки безопасности субъектов и объектов события.

При такой организации система аудита может фиксировать все события, связанные с функционированием прикладного и системного ПО. Анализ реализации подсистем аудита в современных ОС показывает, что ядро системы (являющееся доверенным) взаимодействует с прикладным ПО (работа которого подлежит аудиту) через интерфейс обращений к данному сервису ОС и является наиболее удобным местом для осуществления мониторинга и аудита (рис. 3.4). Ядро ОС при обращении к нему прикладного ПО фиксирует события, подлежащие аудиту, а подсистема аудита заносит их в журнал событий, хранящийся на жестком диске.

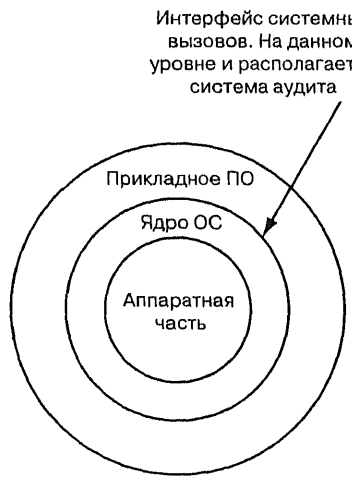

Рис. 3.4. Местоположение подсистемы аудита в архитектуре ОС

Генерировать события аудита может и прикладное ПО, но, в отличие от ядра ОС, данные действия не заносятся напрямую в журнал событий, а передаются в ядро системы, которое отправляет их в буфер (отведенный под запись событий) и впоследствии заносит в журнал событий. Примером событий,

генерируемых прикладным ПО, может служить информация о старте или останове Web-сервера. Буфер системы аудита находится в монопольном владении ядра ОС. При осуществлении записи или чтения в данный буфер производится его блокировка с целью запрещения одновременного чтения из него и записи в него. К каждому событию, помещенному в буфер, дописывается метка времени, последовательный номер и идентификатор процесса, породившего данное событие.

Общая схема архитектуры построения системы аудита представлена на рис. 3.5.

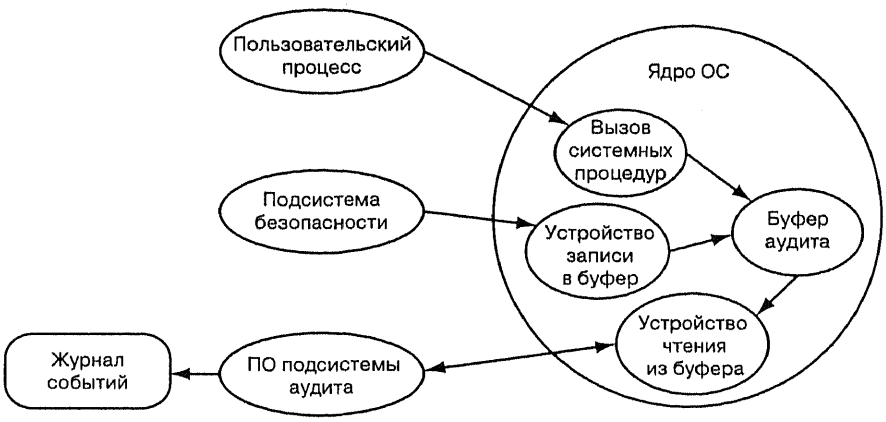

Рис. 3.5. Архитектура системы аудита

Необходимо подчеркнуть важность не только сбора информации, но и ее регулярного и целенаправленного анализа. В этом плане выгодное положение занимают средства аудита СУБД, поскольку к регистрационной информации могут естественным образом применяться произвольные SQL-запросы.

3.3. Защита в локальных сетях

Прежде чем перейти к рассмотрению раздела, связанного с безопасностью межсетевого взаимодействия, необходимо обратить внимание на то, что глобальные сети передачи данных представляют собой объединение локальных вычислительных систем (ЛВС), имеющих различную топологию построения и использующих разнообразные сетевые протоколы (TCP/IP, IPX/SPX, Netbios, DecNet и др.). Поэтому в данном разделе мы уделим внимание безопасности именно ЛВС. Отдельной проблемой является применение активных сетевых ресурсов – AD (в ОС Windows NT)

и NDS (в ОС Novell NetWare). (Информацию по аспектам безопасности и использования этих служб можно найти на сайтах Microsoft и Novell.)

Вопрос безопасности на примере конкретных сетевых ОС, таких как Windows NT и Novell NetWare. В этой части будут затронуты специфические аспекты их функционирования в рамках локальных вычислительных сетей (ЛВС), а также проблемы сетевой безопасности, возникающие в данных ОС. Особенности защиты информации в современных ОС были рассмотрены ранее, проблемы же межсетевого взаимодействия изложены в следующих разделах.

О безопасности в локальных сетях уже сказано достаточно много, поэтому в настоящем разделе мы рассмотрим только основные моменты, связанные с сетевой безопасностью ЛВС, которые построены на описанных выше ОС.

3.3.1. Общие вопросы безопасности в ЛВС

Безопасность ЛВС по сравнению с безопасностью межсетевого взаимодействия отличается тем, что в случае локальных вычислительных сетей на первое по значимости место выходят нарушения зарегистрированных пользователей ОС, поскольку в основном каналы передачи данных в ЛВС находятся на контролируемой территории, защита от несанкционированного подключения к которым реализуется административными методами.

Среди основных угроз, наиболее опасных для ЛВС, следует отметить следующие:

- анализ сетевого трафика с целью получения доступа к конфиденциальной информации, например к передаваемым в открытом виде по сети пользовательским паролям;
- нарушение целостности передаваемой информации. При этом может модифицироваться как пользовательская, так и служебная информация, например подмена идентификатора группы, к которой принадлежит пользователь;
- получение несанкционированного доступа к информационным ресурсам, например с использованием подмены одной из сторон обмена данными с целью получения доступа к файл-серверу от имени другого пользователя;
- попытка совершения ряда действий от имени зарегистрированного пользователя в системе, например злоумышленник, скомпрометировав пароль администратора, может начать общаться с ЛВС от его имени.

Причинами возникновения данных угроз в общем случае могут оказаться:

- наличие уязвимостей в базовых версиях сетевых протоколов. Так, при использовании стека протоколов TCP/IP нарушитель может внедрить в ЛВС ложный ARP-сервер (см. пример, приведенный далее);
- уязвимости специализированных защитных механизмов. Например, причиной возникновения подмены стороны информационного обмена может служить уязвимость процедур аутентификации клиентов при доступе к серверу;
- некорректное назначение уровня доступа;
- использование в качестве каналов передачи данных общедоступной среды, например применение топологии построения ЛВС с общей шиной. В данном случае злоумышленник может использовать ПО (например, Network Monitor или LanAnalyzer), позволяющее просматривать все передаваемые в сети пакеты;
- некорректное администрирование ОС, например задание не до конца продуманных разрешений на удаленное редактирование системного реестра (в ОС Windows NT);
- ошибки в реализации ОС;
- ошибки персонала.

Обеспечение защиты в соответствии с заданной политикой безопасности в ЛВС является комплексной задачей и осуществляется при помощи:

- корректного администрирования сетевых настроек ОС (в том числе и настроек, отвечающих в данной ОС за сетевую безопасность, которые являются штатными средствами большинства современных ОС);
- дополнительных защитных механизмов: шифрования, ЭЦП, аутентификации сторон и др.;
- организационных методов защиты и контроля за их неукоснительным соблюдением, например с использованием системы аудита.

Говоря о дополнительных защитных механизмах, следует отметить, что надежная защита в ЛВС возможна только при использовании средств обеспечения конфиденциальности и достоверности передаваемого по сети трафика, например при использовании ПК «Игла-П». В случае применения криптографических средств защиты информации в ЛВС проблем с распределением ключевой информации будет возникать гораздо меньше, нежели при межсетевом взаимодействии, поскольку число пользователей в ЛВС ограничено и появляется возможность применить эффективные

организационные методы распределения ключевой информации. Например, при использовании в ЛВС средств криптографической защиты можно создать центр распределения ключей, в функции которого входила бы генерация ключевых дискет и рассылка их конечным пользователям, что позволит отказаться от открытого распределения ключевой информации.

Необходимо отметить, что действенным методом поддержания надежного функционирования системы безопасности в ЛВС является использование системы аудита. Действенность аудита в данном случае заключается не только в своевременном реагировании администратора безопасности на нарушения выбранной политики безопасности в ЛВС, но и в возможности привлекать зарегистрированных пользователей к ответственности за совершенные проступки.

Определенную пользу может оказать применение особых топологий построения ЛВС, минимизирующих возможность применения нарушителем средств, которые позволяют прослушивать среду передачи данных. Это прослушивание обычно выполняется с целью получения доступа к сетевым пакетам. В подобном случае применение сетевой топологии типа «звезда» не позволит нарушителю получить доступ со своей рабочей станции к сетевым пакетам, ему не предназначенным. Правда, злоумышленник может осуществить несанкционированное подключение к одному из каналов, однако подобные посягательства должны предотвращаться либо организационными методами, либо шифрованием трафика, что сделает попытки нарушителя бессмысленными.

Кроме того, для исключения возможности неавторизованного входа в компьютерную сеть в последнее время используется комбинированный подход – пароль совместно с аутентификацией пользователя по персональному носителю «идентификационной информации». В качестве носителя может использоваться пластиковая карта, смарт-карта, ключевые дискеты или различные устройства для идентификации личности по биометрической информации (например, по радужной оболочке глаза или по специальному программному обеспечению). С помощью этих мер можно значительно повысить степень защиты от несанкционированного доступа. В этом случае для доступа к компьютеру пользователь должен вставить смарт-карту в устройство чтения и ввести свой персональный код. Программное обеспечение позволяет установить несколько уровней безопасности, которые управляются системным администратором. Этот подход значительно надежнее применения паролей, поскольку, если пароль скомпрометирован, пользователь об этом может не знать; если же пропала ключевая дискета, необходимые меры можно принять немедленно.

Говоря о топологии построения ЛВС и преимуществах использования тех или топологий с точки зрения обеспечения безопасности, необходимо отметить так называемые *виртуальные локальные вычислительные сети* (VLAN).

VLAN представляют собой группу компьютеров, серверов и других сетевых ресурсов, которые функционируют так, как будто они подключены к одному сегменту сети, хотя на самом деле этого может и не быть. За счет того, что VLAN реализуются на сетевом уровне, такое программное решение повышает производительность сети и улучшает ее управляемость, поскольку весь этот инструментарий при добавлении, перемещении или реорганизации узлов сети может быть быстро и просто перенастроен. VLAN, организуемые посредством установки коммутатора, не позволяют некоторым типам трафика, таким как пакеты протоколов ARP (address resolution protocol) и SAP (service advertising protocol), выходить за пределы тех участков сети, где они используются. Это же распространяется и на сетевые пакеты, которые должны оставаться внутри определенных доменов (например, пакеты данных, которыми обмениваются рабочая станция пользователя и его локальный сервер). Прочая информация, предназначенная для ресурсов вне локальной сети, может передаваться в другие VLAN.

Использование VLAN дает возможность повысить пропускную способность сети за счет ее эффективной сегментации. В отличие от обычной коммутации, в этом случае передача данных ограничена только необходимыми адресатами, что приводит к снижению общей загрузки сети.

Говоря об оборудовании, применяемом для создания VLAN, следует отметить, что, на первый взгляд, коммутаторы VLAN мало отличаются от сетевых маршрутизаторов. Разница, однако, заключается в том, что коммутаторы ethernet, применяемые в вычислительных сетях подобного вида, устанавливают виртуальные линии связи между узлами таким образом, чтобы гарантировать получение каждой станцией необходимой ей полосы и не мешать при этом другим станциям. Широковещательные пакеты, например ARP и SAP, остаются в пределах VLAN.

Кроме того, наиболее приспособленные для организации VLAN коммутаторы могут выполнять также функции мостов и маршрутизаторов. Действует ли такое устройство в качестве моста, маршрутизатора или коммутатора, зависит от получаемой команды и от собственных возможностей. Например, по MAC-адресу коммутаторы определяют передающие и принимающие станции сети, а некоторые из них используют этот адрес для фильтрации VLAN. Другие узнают, к какой VLAN подсоединена та или иная станция, по соответствующему порту.

В зависимости от используемого в сети протокола и от предписанного устройству порядка действий в отношении данной станции коммутатор

будет играть роль моста или маршрутизатора. Так, создавая виртуальную сеть между двумя устройствами, коммутатор не выполняет функций ни моста, ни маршрутизатора. Если же необходимо обеспечивать трафик между отдельными VLAN, он может действовать как маршрутизатор (если протокол допускает маршрутизацию IP или IPX) или как мост.

Таким образом, в рамках VLAN создаются виртуальные границы, которые могут быть пересечены только через маршрутизаторы. При этом условии для управления правами доступа можно использовать стандартные средства безопасности. Так, например, в качестве правил фильтрации могут выступать следующие протоколы: на основе широкого вещания и на основе защиты. В частности, создание VLAN на основе правил защиты означает, что трафик не может выходить за пределы VLAN, если он не проходит через маршрутизатор. Возможность создавать виртуальные соединения между внешней и защищенной VLAN отсутствует.

Кроме правил фильтрации можно воспользоваться маршрутизаторами со встроенными средствами шифрования проходящего через них трафика.

Более подробную информацию по этой проблеме можно найти в Internet (см. список Web-серверов).

Пример внедрения ложного ARP-сервера в ЛВС

В IP-сетях для адресации IP-пакетов, кроме непосредственно IP-адреса, необходимо знание физического адреса рабочей станции, например ethernet-адреса, находящегося в сетевой карте. Для определения соответствия IP-адреса физическому адресу используется ARP-протокол (Address Resolution Protocol). Принцип действия ARP-протокола заключается в следующем: при первом обращении к сетевым ресурсам рабочая станция отправляет широковещательный запрос, в котором указывается IP-адрес целевой рабочей станции, и просит ее сообщить свой ethernet-адрес. Эта процедура характерна тем, что данный запрос получат все станции. Полученный в ARP-ответе физический адрес будет внесен ОС запросившей рабочей станции в ARP-таблицу, содержащую соответствия IP и ethernet целевой рабочей станции.

Последовательность действий нарушителя в этом случае такова:

1. Нарушитель находится в ожидании ARP-запроса (осуществляется с помощью анализа широковещательных пакетов, передаваемых в ЛВС).
2. При получении конкретного запроса происходит передача на запрашивающую станцию ARP-ответа, в котором будет содержаться

физический адрес рабочей станции нарушителя вместо физического адреса запрашиваемой рабочей станции. Таким образом, на атакуемой рабочей станции будет создана запись в ARP-таблице, в которой в соответствии с IP-адресом запрашиваемой рабочей станции будет зафиксирован физический адрес нарушителя, следовательно, все сетевые пакеты, направляемые с атакуемого хоста целевой рабочей станции, будут попадать на рабочую станцию злоумышленника.

Очевидно, что для того, чтобы подобная атака принесла ощутимые результаты, ее нужно модифицировать, поскольку в данном случае нарушитель будет принимать только исходящие пакеты атакуемого хоста. Но смысл примера заключается в другом – здесь мы постарались продемонстрировать сам факт уязвимости ЛВС, построенных на основе стека TCP/IP.

3.3.2. Безопасность в сетях Novell NetWare

В этом подразделе рассказывается о вопросах безопасности ОС Novell NetWare, связанных с функционированием сетевой части данной ОС, причем вопросы корректной настройки штатных средств ОС такого типа здесь рассматриваться не будут, поскольку это материал для отдельной книги. Описываемая ОС представляет собой классический пример клиент-серверной архитектуры, в которой клиенты работают в активном взаимодействии с сервером, например при авторизации клиента на сервере.

Хотя данная ОС и проверена временем, тем не менее она не отличается особой «стройностью» защиты на сетевом уровне. Так, на сегодняшний день известно достаточное количество уязвимостей Novell NetWare, возникающих из-за недостаточного внимания, уделенного авторами данной ОС вопросам безопасности организации взаимодействия в сети.

Например, применение нарушителем программ типа sniffer (дает возможность осуществлять анализ сетевого трафика) позволяет получить пароль зарегистрированного пользователя при прохождении им процедуры login на сервере (подобная уязвимость была актуальна для ранних версий Novell NetWare – 2.x и 3.x), поскольку он передается в открытом виде.

Критичной точкой безопасности для многих серверов является контроль физического доступа к консоли управления. Вот почему так важно оградить управление сервером от всякой возможности физического посягательства, для чего следует ограничить доступ к консоли.

В этом смысле использование администратором процедуры удаленной консоли (rconsole.exe) является не только удобным средством, позволяющим

производить удаленное управление сервером, включая загрузку и выгрузку network loadable modules (NLMs), но и вполне доступной мишенью для нарушителя. Нарушитель в этом случае имеет возможность получить доступ к консоли с удаленного места, при этом ему совсем не нужен физический доступ к серверу, который может быть защищен организационными мерами.

В ходе процесса RCONSOLE пароль передается на сервер в зашифрованном виде. Если внимательно рассмотреть процедуру взаимодействия пользователя и сервера в рамках инициации процесса RCONSOLE, можно заметить, что передаваемые данные представляют собой NCP-пакеты (64 байта, 60 байт, 64 байта и 310 байт). И в одном IPX/SPX-пакете, имеющем длину 186 байт, содержится 8-байтный пароль по смещению 3Ah, которое легко найти, поскольку смещение 38h всегда FE, а смещение 39h всегда FF. Данный пароль, как уже говорилось, передается в зашифрованном виде, само шифрование осуществляется следующим образом:

- клиент запрашивает у сервера сеансовый ключ;
- сервер посылает клиенту 8-байтный ключ в открытом виде;
- клиент сначала зашифровывает пароль при помощи своего идентификатора, а затем зашифровывает полученное 16-байтное значение на ключе, результатом чего является 8-байтное значение, которое и вставляется в посылаемый пакет.

Очевидно, что данная схема шифрования паролей не может считаться стойкой по отношению к вероятному нарушителю, имеющему возможность просматривать сетевой трафик – в случае топологии ЛВС с общей шиной это не составляет труда. Нарушителю, перехватившему данные, которые передавались между клиентом и сервером, и получившему пароль для осуществления несанкционированной инициации процесса RCONSOLE, не составит труда получить сетевой адрес и адрес узла рабочей станции (эти данные можно отыскать с помощью USERLIST/A), на которой работал клиент. Теперь нарушитель будет осуществлять доступ с серверу от имени клиента. Далее нарушитель вставляет в сформированный им пакет, содержащий 8-байтный пароль по смещению 3Ah, сетевой адрес и адрес узла и посылает этот пакет серверу, после чего сервер воспринимает его как легального пользователя, а нарушитель получает доступ к удаленному управлению консолью.

Разработчики данной ОС, учитывая подобную уязвимость, начиная с версии 3.11, применили механизм выработки MAC на NCP-пакеты. Существует несколько видов использования механизма обеспечения достоверности NCP:

- 0 = не использовать;
- 1 = использование по требованию;
- 2 = использовать, если обе стороны сетевого обмена поддерживают данный механизм;
- 3 = обязательно использовать.

По умолчанию между клиентом и сервером используется тип 2 подтверждения достоверности, однако, получив доступ к файлу Net.cfg, нарушитель сможет прописать там строку Signature Level=0 или, имея доступ к консоли управления сервером, ввести строку SET NCP PACKET SIGNATURE=0 и тем самым добиться цели. Использование данного механизма позволит избежать не только приведенной выше атаки, но также обеспечить неизменность передаваемых по сети исполняемых модулей, например login.exe. Если же этого не сделать, то нарушитель может инфицировать login.exe при передаче его по сети.

В общем случае процедура login между клиентом и сервером выглядит следующим образом:

1. Клиент посылает запрос серверу.
2. Сервер проверяет имя клиента и отправляет ему пару (R, UID), где R – случайное число, а UID – идентификатор пользователя.
3. Клиент выполняет следующие вычисления: X = hash(UID, password) и Y = hash(X, R), а затем посылает серверу Y.
4. Сервер получает Y и вычисляет значения X1 = hash(UID, password) и Y_1 = hash(X1, R). После чего производит сравнение Y = Y_1. Если равенство выполнено, клиент получает доступ к серверу.
5. Клиент и сервер вычисляют Z = hash(X, R, C) (C является константой), которое затем используется в качестве ключа генерации MAC для данной сессии.

В Novell NetWare 4.x описанная процедура проходит с использованием асимметричного алгоритма RSA.

Однако на приведенную выше схему возможны атаки типа «человек в середине» (man in the middle) при условии, что нарушитель способен просматривать пакеты, пересылаемые между клиентом и сервером, и посылать пакеты быстрее, чем клиент и сервер. Для этого нарушителю, например, нужно находиться между клиентом и сервером или провести атаку типа «внедрение ложного объекта», например внедрив ложный ARP-сервер. Наиболее удобным способом для нарушителя будет подключение между сервером и клиентом, как показано на рис. 3.6. В этом случае нарушитель может не только анализировать трафик, но и блокировать передачу некоторых пакетов.

Подобные атаки, очевидно, не будут работать, если выполнится условие повсеместного использования MAC на NCP-пакеты в ЛВС. Поэтому его включение администратором ЛВС обязательно.

Атака осуществляется следующим образом:

Рис. 3.6. Подключение нарушителя

1. Нарушитель ожидает начала login со стороны клиента. Когда клиент посылает запрос серверу на прохождение login, нарушитель, в свою очередь, тоже направляет запрос серверу.
2. Сервер отвечает двумя значениями R1 и R2, предназначенными соответственно для клиента и нарушителя.
3. Когда нарушитель получает R2, он изменяет в данном пакете сетевой адрес сервера и посылает R2 клиенту.
4. Клиент вычисляет Y1 = hash(X, R2) и посылает его серверу, в то время как сервер ожидает значение Y2 = hash(X, R1).
5. Нарушитель перехватывает данный пакет и изменяет в нем сетевой адрес клиента на свой реальный.
6. После чего нарушитель начинает работать с сервером от имени клиента.

Учитывая сказанное, очевидно, что без дополнительных средств защиты, осуществляющих шифрование и подпись передаваемого трафика, невозможно обеспечить безопасность в данной ОС. Но все же некоторый уровень защиты при работе с этой ОС обеспечить можно, если придерживаться следующих рекомендаций (которые, правда, не претендуют на полноту):

- физически защитить сервер. Необходимо организовать локальную защиту сервера и ограничить к нему доступ (подробно методы защиты локальной рабочей станции рассматривались выше). Здесь будет уместно дать рекомендацию: используйте команду SECURE CONSOLE для предотвращения запуска NLM не из SYS: SYSTEM. Реализация этого пункта приведет к тому, что 75% потенциальных атак будут исключены;
- защитить критичные файлы системы. Например, вычислите контрольную сумму на файлы, находящиеся в SYS: LOGIN, SYS:PUBLIC и SYS:SYSTEM, и постоянно производите проверку полученных контрольных сумм;
- доступ пользователей к ресурсам должен быть организован в соответствии с выбранной политикой безопасности;

- использовать мониторинг работы с консолью; это можно осуществить с помощью запуска console.nlm;
- включить accounting, посредством которого можно регистрировать все попытки произвести login и logout к серверу;
- использовать механизмы подтверждения подлинности NCP-пакетов;
- стараться не применять процедуры RCONSOLE;
- обеспечить безопасность NCF-файлов;
- добавить команду EXIT в System Login Script;
- проверить местоположение rconsole.exe. Например, в Novell NetWare 3.12 rconsole.exe находится в SYS:SYSTEM и SYS:PUBLIC.

3.3.3. Безопасность в сетях Windows NT

Показатели сетевой безопасности Windows NT по сравнению с NetWare далеко не лучшие. У этой операционной системы тоже есть уязвимые места, позволяющие нарушителю не только дезорганизовать работу пользователей в ЛВС, но и осуществлять несанкционированный доступ к защищаемой информации. Уязвимости Windows NT связаны как с непроработанностью вопросов сетевой безопасности, ошибками при проектировании ее сетевой части, так и с некорректным администрированием. Правда, можно устранить большинство существующих на сегодняшний день уязвимостей путем отслеживания обновлений ОС, выпускаемых фирмой Microsoft в виде Service Pack, и корректно администрируя ОС, однако в этой системе существует целый класс уязвимостей, устранение которых возможно только при использовании дополнительных защитных механизмов, выраженных в реализации шифрования сетевого трафика или контроля доступа к каналам передачи данных в ЛВС.

В общем случае все вопросы сетевой безопасности Windows NT можно отнести к следующим разделам:

- безопасность процедур входа в домен Windows NT;
- безопасность протокола SMB;
- безопасность протокола PPTP;
- безопасность IIS, реализованного на базе ОС Windows NT;
- безопасность сервиса DCOM.

Перечислим некоторые уязвимости этой ОС и одновременно укажем методы их устранения:

- отправка атакуемой системе непрерывного потока произвольных UDP-пакетов (UDP flood) по порту 53 (DNS) приводит к краху DNS-сервера (Service Pack 3);

- атака, направленная на DNS-сервер. Атакующий хост отправляет на DNS-сервер запрос на поиск адреса. Если требуемого адреса нет в кэше атакуемого сервера, то сервер обращается с запросом к соседним серверам из дерева DNS. Атакующий хост в этот момент отсылает ответ от имени одного из соседних серверов DNS с заведомо ложным адресом искомого хоста. Для этого необходимо знать идентификатор запроса. Затем эта информация попадает в кэш DNS-сервера. Идентификатор запроса можно разгадать, так как он увеличивается на единицу при каждом новом запросе (Service Pack 3);
- атака заключается в посылке атакуемому сегменту сети непрерывного потока широковещательных UDP-пакетов по порту 19. Все компьютеры с Windows NT, на которых загружена служба Simple TCP/IP Services, будут отправлять ответные пакеты, создавая переполнение трафика UDP датаграммами (Service Pack 3);
- постоянная посылка UDP-пакетов по порту 137 (NetBIOS Name Service) приводит к краху WINS-сервера (в том случае, когда атака осуществляется из другой сети, подобную уязвимость локализует межсетевой экран; если же атака происходит из той же ЛВС, поможет только запрет на использование данной службы);
- атака заключается в посылке атакуемой системе двух специальным образом сформированных IP-фрагментов и заключенных в одну UDP-датаграмму. Windows NT неправильно обрабатывает такого рода фрагментированные IP-пакеты, что приводит к краху системы (Service Pack 3);
- посылка атакуемой системе OOB (Out of Band) пакета данных по порту 139 (NetBIOS) приводит к краху системы. Атака также действует на DNS-сервер (порт 53), работающий вместе с WINS-сервером, и приводит к краху DNS-сервера (здесь поможет запрет на использование данной службы либо фильтрация трафика по данному порту);
- посылка атакуемой системе ICMP-пакета длиной более 64 Кб с его неизбежной последующей фрагментацией, что приводит к краху системы. (Service Pack 3);
- посылка атакуемой системе SMB logon запроса с неверно указанной длиной данных тоже приводит к краху системы (Service Pack 3);
- посылка атакуемой системе IP-пакета с запросом на соединение (SYN), в котором адреса получателя и отправителя совпадают, также приводит к краху системы (Service Pack 3);
- атака основана на том факте, что исходный пароль учетной записи компьютера с Windows NT устанавливается равным имени этого

компьютера. Злоумышленник, прослушивая сеть, может рассчитать ключ сеанса в момент первоначальной аутентификации компьютера и все последующие ключи в процессе их смены (в этом случае поможет физическая защита сети либо шифрование сетевого трафика);

- атака типа «человек в середине» основана на том, что после аутентификации пользователя в домене Windows NT в пакете NetLogonSamLogon, передаваемом контроллером домена компьютеру, сведения о группах, членом которых является пользователь, отправляются открыто. Установив в сети proxy-сервер, злоумышленник может заменить SID некоторой глобальной группы на SID группы DomainAdmins. После этого пользователь на локальном компьютере получает права члена группы DomainAdmins (поможет физическая защита сети или шифрование сетевого трафика);
- следующий тип атаки использует уязвимость, основанную на том, что для аутентификации достаточно знать 16-байтный хэшированный пароль пользователя. Помимо этого SMB-протокол поддерживает возможность аутентификации с передачей пароля открытым текстом, что позволяет проводить атаки типа Downgrade (понижение уровня аутентификации), вынуждая клиента использовать именно этот режим аутентификации (здесь поможет установление Service Pack 3 и занесение в ключ реестра HKLM\SYSTEM\CurrentControlSet\Services\Rdr\ параметра EnablePlainTextPassword (типа REG_DWORD) со значением 1 или установление SMB-протокола с электронной подписью (SMB signing)).

Кроме перечисленных уязвимостей, причиной успешных атак на ОС Windows NT может стать некорректная реализация выбранной политики безопасности. Обычно это выражается в том, что разрешения отдельным пользователям или группам пользователей при работе с рабочей станцией из ЛВС задаются некорректно. Чтобы избежать этих неприятностей, следует:

- удалить учетную запись пользователя Guest (которая по умолчанию присутствует в ОС после ее установки);
- ограничить доступ пользователей к рабочей станции из ЛВС путем задания соответствующих прав, например запретить доступ к сети группе Everyone;
- по возможности не использовать удаленное редактирование системного реестра.

Выполнение перечисленных рекомендаций позволит обезопасить работу в ЛВС под управлением ОС Windows NT.

Результатом решения проблем, связанных с безопасностью ОС Windows NT, стало появление в Windows NT 5.0 целого ряда механизмов безопасности:

- применение *активных каталогов* (AD), построенных с использованием архитектур X.500 и DNS; службы безопасности могут использовать их для хранения учетных записей. ОС Windows NT определяет взаимодействие активного каталога и служб безопасности. В активном каталоге хранятся правила безопасности домена и учетные записи, причем доступ к этой информации надежно защищен. При определении прав доступа активный каталог, как и другие процессы, использует процедуру олицетворения. Это означает, что права доступа по запросам клиентов по протоколу LDAP (базовый протокол для взаимодействия с активным каталогом) проверяются ОС, а не самим каталогом. В свою очередь, компоненты системы безопасности ОС доверяют информации, хранящейся в активном каталоге;
- применение файловой системы с шифрованием. Файловая система подобного типа работает прозрачно для пользователя и обеспечивает шифрование данных на уровне файлов или папок. Эта система работает с использованием CryptoAPI и применяет для шифрования DES, а хранение ключей шифрования файлов осуществляется вместе с защищаемыми ресурсами (в виде отдельных полей данных в зашифрованном файле). При этом для шифрования ключей шифрования файлов используются асимметричные алгоритмы;
- поддержка многочисленных протоколов безопасности:
  - Kerberos 5.0 заменит в Windows NT 5.0 существовавший до этого протокол LAN Manager (NTML) и станет основным средством обеспечения доступа к ресурсам как внутри доменов, так и между ними. Сервер распределения ключей (KDC) будет реализован на каждом контроллере домена, которые становятся полностью эквивалентными кластерами в архитектуре Kerberos. Клиентское ПО реализовано в виде DLL-библиотеки *провайдера безопасности* (security provider), а процедура начальной аутентификации интегрирована в сервис WinLogon, который от имени пользователя получает первый билет на сервере Kerberos (то есть контроллере домена). Серверное ПО интегрировано в службы обеспечения безопасности контроллера домена. Для доступа к базе данных субъектов применяется служба каталогов (Windows NT Directory Service). Другие системные компоненты NT, например *редиректор* (Redirector), используют Kerberos для обращения к серверу SMB при удаленном доступе к файлам.

Windows NT будет поддерживать делегирование полномочий субъектов с помощью опций PROXY и FORWARDING в билетах. При этом серверы могут от имени клиентов получать билеты на доступ к другим серверам;

- NTLM-протокол проверки подлинности в Windows NT будет использоваться только для взаимодействия с предыдущими версиями ОС;
- протокол распределенной проверки подлинности паролей (DPA);
- SSL/PCT-протоколы, которые будут использоваться для обеспечения защиты в технологии «клиент-сервер». В качестве инфраструктуры, поддерживающей функционирование данного протокола, будут реализованы две службы: служба сертификации открытых ключей в домене NT и интерфейс CryptoAPI v2.0; последний будет поддерживать сертификаты открытых ключей в формате X.509.

Интеграция протоколов безопасности в прикладное ПО реализована за счет создания нового интерфейса для приложений Win32 – SSPI (Security Support Provider Interface), см. рис. 3.7. Применение этого интерфейса позволит унифицировать обращение к функциональным возможностям данных протоколов и изолировать прикладное ПО от выполнения функций безопасности. Вынесение этих функций за рамки прикладного ПО позволит минимизировать влияние ошибок или закладок в программах на уровень обеспечиваемой безопасности.

SSPI представляет собой набор доступных прикладному ПО функций, которые могут быть разбиты на следующие классы:

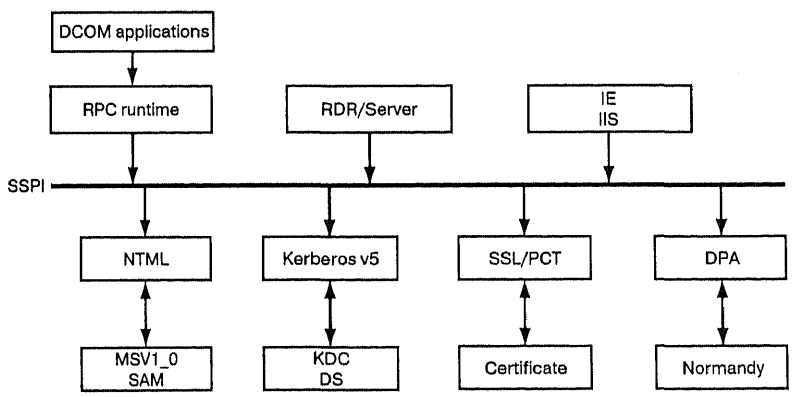

Рис. 3.7. Интерфейс SSPI

- управление мандатами пользователей (табл. 3.2);
- управление контекстом безопасности. Под контекстом безопасности следует понимать следующую структуру данных:

```
typedef struct_SecPkgInfo {
ULONG Fcapabilities; //битовая маска возможностей пакета
USHORT wVersion; //версия драйвера
USHORT wRPCID; //идентификатор RPC времени выполнения
ULONG  cbMaxToken; //размер маркера аутентификации (authentication
token)
SEC_CHAR *Name; //имя
SEC_CHAR *Comment; //комментарий
          } SecPkgInfo, *PsecPkgInfo;
```

Таблица 3.2. Функции, доступные в SSPI

Функция	Назначение
AcquireCredentialsHandle	Получение дескриптора данных авторизации
FreeCredentialsHandle	Освобождение дескриптора данных авторизации
QueryCredentialsAttributes	Получение информации о дескрипторе

При работе с контекстом безопасности прикладному ПО доступны следующие функциональные возможности: создание контекста, удаление контекста, имитация клиента сервером (изменение привилегий и прав доступа сервера на те, что есть у клиента), подавление имитации клиента сервером, инициализация контекста в начале соединения, создание дополнительного защитного сообщения, освобождение занимаемой контекстом памяти и получение информации о контексте;

- целостность передаваемых сообщений. Для обеспечения целостности передаваемых пакетов данных предусмотрены функции, перечисленные в табл. 3.3;
- управление пакетами.

Таблица 3.3. Функции, обеспечивающие целостность пакета данных

Функция	Назначение
MakeSignature	Создание подписи
VerifySignature	Проверка подписи

В заключение хотелось бы отметить, что развитая сетевая поддержка в ОС Windows NT существенно затрудняет обеспечение безопасности ЛВС на основе ОС Windows NT. Следовательно, при эксплуатации этой операционной системы целесообразно всемерно упрощать способы и протоколы взаимодействия компьютеров и прикладных систем между собой.

Аутентификация в CIFS

В ОС Windows NT для удобства доступа к файлам и сервису печати создан протокол CIFS (Common Internet File System), базирующийся на SMB-протоколе (Server Message Block). CIFS построен по архитектуре «клиент-сервер» и позволяет получать доступ к информационным ресурсам посредством механизмов, подобных заложенным в WWW.

Безопасность в CIFS строится на аутентификации сторон информационного обмена. При этом, исходя из идеологии создания CIFS, защиту трафика должны осуществлять механизмы безопасности в SMB-протоколе. (Базовая версия SMB-протокола не содержит криптографических механизмов, которые позволяют защищать передаваемый трафик, но существует модификация данного протокола – SMB-signing, позволяющая обеспечивать целостность передаваемых данных по SMB-протоколу.)

Установка соединения с использованием протокола SMB происходит следующим образом:

- клиент устанавливает сессию по протоколу NetBIOS, в которой сообщает серверу NETBIOS имя клиентского компьютера;
- клиент посылает серверу сообщение, называемое SMB_COM_NEGOTIATE, где перечисляет диалекты SMB, которые он «понимает». В этой фазе клиент и сервер «договариваются» о том, какой диалект протокола будет использоваться в дальнейшем. В частности, от этого зависит, в какой форме будут передаваться пароли;
- сервер отвечает сообщением, в котором указывает номер диалекта из тех, что перечислил клиент. Этот номер будет использоваться в дальнейшем. Сервер выбирает наиболее современный протокол из тех, которые поддерживаются и клиентом, и сервером. Если применяется диалект выше PC NETWORK PROGRAM 1.0, сервер устанавливает в ответном сообщении бит, предлагающий клиенту шифровать пароль, и передает в этом сообщении значение, называемое challenge, которое будет использовано клиентом для шифрования своего пароля;
- клиент посылает сообщение SMB_SESSION_SETUP_ANDX, в котором сообщает пользовательские имя и пароль, открытым текстом или в зашифрованном виде, в зависимости от используемого диалекта. При этом клиент передает серверу несколько чисел (UID – user ID, PID – process ID, MID – multiplexer ID), необходимых в дальнейшем для идентификации сессии;
- сервер отвечает на запрос об аутентификации, и при положительном ответе соединение считается установленным. Клиент может обращаться к файлам, каталогам и принтерам сервера в соответствии с правами доступа, установленными для объектов на сервере.

В случае использования алгоритма challenge-response происходит следующее:

- сервер посылает 8-байтный challenge;
- клиент добавляет пять нулевых байт к значению Lan Manager compatible password hash и пять нулевых байт к значению Windows NT password hash, шифрует оба полученных значения на ключе, переданном сервером (challenge), и отсылает их обратно;
- сервер производит ту же самую операцию со значениями, хранящимися в базе данных SAM, и сравнивает их с полученным от клиента показателем. Если они совпадают, то соединение считается установленным.

В описанной схеме существует несколько уязвимых мест, которые могут быть использованы для нарушения конфиденциальности:

- некоторые действия не требуют аутентификации, то есть могут проводиться анонимно (нуль-сессии). К ним относятся, в частности, получение у сервера списка разделяемых по сети ресурсов и даже редактирование системного реестра. Если к разделяемому ресурсу разрешен доступ группе Everyone, то к нему можно обратиться с пустым именем и паролем;
- при установлении соединения по протоколу NetBIOS сервер неспособен никоим образом проверить имя, переданное ему клиентом, и не запротоколирует соединение до тех пор, пока не произойдет установление SMB-сессии. В любом случае сервер протоколирует только имя, сообщенное ему клиентом, а не IP-адрес. Это может быть использовано, например, для обхода установленного ограничения на станциях, с которых может обращаться к серверу данный пользователь. Кроме того, это позволяет эффективно «заметать следы» несанкционированного доступа;
- клиент определяет используемый диалект протокола (сообщая, какие протоколы он поддерживает), а также решает, применять или нет шифрование. Это означает, что сервер не может потребовать шифрования или отказаться от соединения без шифрования. Потенциальный взломщик может использовать этот момент для того, чтобы, например, произвести атаку типа «понижение уровня» (downgrade);
- при установлении соединения Windows NT (так же как Windows 95, Windows for Workgroups и проч.) по умолчанию посылает имя и пароль, под которыми пользователь вошел в систему. Таким образом, появляется возможность получить password hash, который потом можно попытаться взломать/расшифровать. Атака такого типа остается

совершенно не замеченной пользователем и может быть весьма эффективно и просто использована для нарушения безопасности системы. Для защиты, во-первых, необходимы межсетевые экраны, не пропускающие SMB-трафик между корпоративной сетью и Internet, и, во-вторых, контроль внутренних Web-серверов;

- передаваемый сервером challenge и ответ клиента можно перехватить (например, сетевым анализатором) и попытаться взломать методом «грубой силы»;
- описанная схема чувствительна к различным атакам типа «человек в середине». Рассмотрим, например, такой сценарий: злоумышленник посылает серверу запрос на установление соединения. Сервер отвечает отправкой challenge. Злоумышленник ждет запроса на установление соединения от легального пользователя и в ответ посылает ему challenge, полученный от сервера. Легальный пользователь шифрует полученный challenge, используя в качестве ключа свой пароль. Нарушитель перехватывает пакет и отправляет его серверу уже от своего имени. Возможен также перехват сессии с подделкой номера последовательности на уровне TCP/IP и значений UID и MID;

Для борьбы с атаками, использующими криптографическую слабость LM password hash, Microsoft выпустила так называемые *заплатки* (hot-fix, lm-fix). Однако сейчас этот метод недоступен в связи с обнаруженными в нем ошибками. При этом следует учесть, что этот метод не сработает и в смешанной сети, включающей в себя, например, клиентов Windows 95 или Windows for Workgroups, поскольку они могут использовать только схему шифрования паролей LAN Manager.

Весьма серьезной угрозой является возможность использования снифферов для получения паролей в сети. Существует общедоступная утилита для Windows NT, позволяющая «прослушивать» трафик в сети и выделять из него пароли tcpdump. Вероятно, опытному программисту не составит большого труда написать фильтр, выделяющий из всего трафика сообщения SMB, которые содержат пароли.

Атаки типа «человек в середине», а также возможности перехвата и расшифровки паролей наводят на мысль, что безопасное использование Windows NT в сетях, таких как Internet, требует применения дополнительных мер защиты (еще одного уровня) для борьбы с подобными угрозами. Оригинальная версия протокола аутентификации в CIFS разработана достаточно давно, однако в ходе ее эксплуатации были обнаружены серьезные уязвимости. Результатом устранения подобных дефектов стала новая версия данного протокола. Усовершенствование протокола аутентификации

проводилось не только устранением логических уязвимостей, но и применением новых криптографических алгоритмов. При этом пользователям следует учесть, что замена ранних версий протокола аутентификации не приводит к необходимости менять свои пароли или сервер ключей.

Аутентификация (доработанная версия) в CIFS имеет следующие подразделы:

- двусторонняя аутентификация сессии между сервером и клиентом при помощи протокола типа запрос-ответ с использованием разделения знания пользовательского пароля. Ответ вычисляется на основе DES-шифрования случайного числа и/или метки времени, содержащихся в запросе. Ключи для шифрования вычисляются на основе пароля пользователя;
- аутентификация сообщений реализуется при помощи вычисления MAC на каждое сообщение. MAC вычисляется с использованием ключевой MD5 (ключи вычисляются из пользовательского пароля и случайных значений клиента и сервера).

При описании протокола аутентификации будем считать, что CIFS-сервер либо взаимодействует с ключевым сервером, хранящим базу данных хэш-значений пользовательских паролей, либо имеет свою собственную базу данных хэшированных паролей (табл. 3.4).

Таблица 3.4. Обозначения, применяемые в CIFS

U	Имя пользователя
P (U)	*Пароль пользователя* U
KS	128-битный сеансовый ключ
KA	56-битный DES-ключ, полученный из первых 7 байт KS
KB	56-битный DES-ключ, полученный из вторых 7 байт KS
SN	32-битный номер шага
KM	40-битный ключ, используемый для вычисления MAC
[S]<N:M>	N байт строки S, начиная с байта с номером M (первый байт является нулевым)
[S]<N>	Первые N байт строки S
{M}K	Шифрование строки
a, b, ..., z	Конкатенация байтовых строк a, b, ..., z
MD4 (M)	Вычисление хэш-кода от строки M в соответствии с алгоритмом MD4
MD5 (M)	Вычисление хэш-кода от строки M в соответствии с алгоритмом MD5
Z (N)	Строка нулей длиной N
CS	Уникальное для сессии 8-байтное значение

Аутентификация сессии состоит из следующих шагов:

1. Клиент вычисляет сеансовые ключи на основе пароля пользователя, инициализирует номер сессии, создает список поддерживаемых алгоритмов и их параметров, а также другие параметры сессии (на них мы подробно останавливаться не будем) и посылает серверу запрос на прохождение аутентификации со списком параметров сессии (Context C).

   C: K_S = MD4(P(U))
   K_A = [K_S] < 7 >
   K_B = [K_S] < 7 : 7 >
   K_C = [K_S] < 2 : 14 >, Z(5)
   C→S: Context C.

2. Сервер выбирает параметры сессии (Context S) и посылает клиенту:

   S→C: Context S, CS.

3. Клиент вычисляет ответ серверу, MAC-ключ, MAC от предполагаемого на данном шаге сообщения и посылает серверу свое имя, вычисленные значения и параметры запроса (идентификатор алгоритма и др. – Msess):

   C: R = {CS}K_A, {CS}K_B, {CS}K_C
   K_M = K_S, R
   SN = 0
   MC = [MD5(K_M, SN, Msess, U, R)] < 8 >
   SN = 1
   C→S: Msess, U, R, MC;

4. Сервер посылает имя пользователя, CS и R (из предыдущего шага) ключевому серверу (KS) по защищенному каналу.

   S→KS: U, CS;

   Ключевой сервер, получив имя пользователя, вычисляет сеансовый ключ $K_{S'}$, после чего вычисляет R' и проверяет R = R'; если оно выполняется, посылает $K_{S'}$. В противном случае серверу возвращается отрицательный результат сравнения и аутентификация заканчивается.

   KS: KS' = MD4(P(U))
   $K_{A'}$ = [$K_{S'}$] < 7 >
   $K_{B'}$ = [$K_{S'}$] < 7 : 7 >
   $K_{C'}$ = [$K_{S'}$] < 2 : 14 >, Z(5)
   R' = {CS}$K_{A'}$, {CS}$K_{B'}$, {CS}$K_{C'}$
   KS→S: $K_{S'}$

5. Сервер вычисляет MAC-ключ и MAC на сообщение, полученное от клиента (MC'), и производит сравнение MC = MC'. В случае его выполнения клиент аутентифицирован успешно, и клиенту отправляется ответ сервера вместе с параметрами сессии (Msessr).

   S: $K_{M'} = K_{S'}$, K
   MC' = [MD5(K_M, SN, Msess, U, R)] < 8 >
   MS = [MD5($K_{M'}$, SN, Msessr)] < 8 >
   S→C: Msessr, MS

6. Клиент проверяет равенство MS' = MS, и, если оно верно, считается, что сервер аутентифицирован клиентом, после чего клиент устанавливает свой номер шага равным 2 и производит следующие вычисления:

   C: MS' = [MD5(K_M, SN, Msessr)] < 8 >
   SN = 2

Для последующих взаимодействий клиента и сервера используется аутентификация сообщений, которую можно описать так:

1. Клиент посылает серверу запрос совместно с MAC на данный запрос с использованием текущего значения номера шага (SN), после чего увеличивает SN на единицу.

   C→S:
   Mreq, [MD5(K_M, SN, Mreq)] < 8 >
   Mreq – параметры сессии со стороны клиента
   C: SN + SN + 1

2. Сервер проверяет MAC и посылает клиенту ответ, после чего увеличивает свой SN' на 2.

   S→C: Mrsp, [MD5($K_{M'}$, SN' + 1, Mrsp)] < 8 >
   S: SN' = SN' + 2

3. Клиент проверяет MAC, и, если он корректен, считается, что аутентификация завершена успешно.

   C: SN = SN + 1

На практике варианты протокола могут изменяться:

- сеансовый ключ K_S может вычисляться различными способами;
- протокол аутентификации сообщений может не применяться;
- пароль может передаваться в открытом виде.

Безопасность PPTP-протокола

Этот раздел тесно связан с разделом, посвященным созданию *виртуальных частных сетей* (VPN), однако здесь рассматривается и безопасность

конкретного протокола, используемого для создания VPN, – *туннельного протокола* типа *точка-точка* (PPTP). Общие сведения по вопросам создания VPN и защите информации и информационных ресурсов в рамках VPN можно найти в разделе 3.4.2.

PPTP – протокол, который позволяет выполнять туннелирование PPP-соединений по IP-сети для создания VPN. Таким образом, удаленный хост в сети X может туннелировать трафик на шлюз в сети У и имитировать подключение с внутренним (в сети X) IP-адресом к сети У. Шлюз получает трафик для внутреннего IP-адреса и передает его удаленной машине в сети X. Существует два основных способа применения PPTP: с использованием Internet и с использованием коммутируемых линий. Функционирование PPTP заключается в инкапсулировании пакетов виртуальной сети в пакеты PPP, которые, в свою очередь, инкапсулируются в пакеты GRE (Generic Routing Incapsulation), передаваемые по IP от клиента к шлюзу-серверу PPP и обратно. Совместно с каналом инкапсулированных данных существует управляющий сеанс на базе TCP. Пакеты управляющего сеанса позволяют запросить статус и сопровождать сигнальную информацию между клиентом и сервером. Канал управления инициируется клиентом на сервере на TCP-порте 1723. В большинстве случаев это двунаправленный канал, по которому клиент посылает запросы на сервер и наоборот. В рамках PPTP не оговариваются конкретные алгоритмы аутентификации и шифрования, протокол обеспечивает интерфейс для применения различных алгоритмов.

В данном разделе речь идет о реализации PPTP от Microsoft, который является частью ОС Windows NT Server (данное ПО можно бесплатно получить с Web-сайта Microsoft). Сервер Microsoft PPTP может существовать только для Windows NT, хотя клиентское ПО есть и для Windows NT/95/98. В протоколе PPTP фирмы Microsoft реализовано несколько вариантов аутентификации:

- с использованием текстового пароля. Клиент передает серверу пароль в открытом виде;
- хэшированный пароль. Клиент передает серверу хэш-код пароля;
- аутентификация типа «запрос-ответ».

Необходимо отметить, что последний вариант аутентификации шифрует передаваемые данные с использованием 40- или 128-битных ключей, но не все ОС Windows поддерживают шифрование трафика. Так, ОС Windows NT его поддерживает, в то время как базовые версии ОС Windows 95/98 не имеют такой возможности.

Несмотря на широкое распространение данной реализации PPTP, в ней существует достаточное количество уязвимостей. Брюс Шнеер и Питер

Мадж провели криптоанализ данной реализации протокола и обнаружили в ней ряд существенных уязвимостей.

Их можно разделить на:

- уязвимости в реализации и применении функций хэширования паролей в ОС Windows NT;
- уязвимости протокола аутентификации (CHAP), используемого в PPTP;
- уязвимости, связанные с протоколом шифрования в одноранговых сетях (MPPE), используемым в PPTP;
- уязвимости реализации протокола PPTP, обусловленные некорректной реализацией логики работы самого протокола.

Уязвимости, связанные с применением хэш-функций

В ОС Windows NT применяются две однонаправленные хэш-функции – Lan Manager (с использованием алгоритма DES) и Windows NT (с использованием алгоритма MD4).

Хэш-функция Lan Manager вычисляется следующим образом:

1. Пароль превращается в 14-символьную строку. Для этого можно отсекать более длинные пароли либо дополнять короткие пароли нулевыми элементами.
2. Все символы нижнего регистра заменяются символами верхнего регистра; цифры и специальные символы остаются без изменений.
3. 14-байтная строка разбивается на две 7-байтные половины.
4. Каждая половина строки используется в роли ключа DES, осуществляется шифрование фиксированной константы с помощью каждого ключа, получаются две 8-байтные строки.
5. Две строки объединяются для создания одного 16-разрядного значения хэш-функции.

Хэш-функция Windows NT вычисляется так:

1. Осуществляется преобразование пароля длиной до 14 символов с различением регистров.
2. Происходит хэширование пароля с помощью MD4 и получение 16-символьного значения хэш-функции.

Причины успешного проведения словарных атак на приведенные здесь хэш-функции заключаются в следующем:

- преобразование символов верхнего регистра в нижний, что уменьшает число возможных паролей (Lan Manager);
- отсутствие индивидуальной привязки хэш-значения пароля к пользователю. Два пользователя с одинаковыми паролями будут иметь одинаковые хэш-значения (Lan Manager и Windows NT);

- разбиение пароля на две половины и необходимость работы с каждой из них в отдельности приводит к тому, что атаку методом «грубой силы» можно провести на каждую половину (это выполнить значительно легче, нежели провести атаку на весь хэш-код в целом (Lan Manager)).

Очевидно, что хэш-функция Windows NT обладает более высокими показателями, чем хэш-функция Lan Manager. Вторая используется для совмещения с ранним версиями ОС Windows, и применять ее в сетях ОС Windows NT нет необходимости, тем не менее хэш-код Lan Manager передается совместно с хэш-кодом Windows NT в сетях ОС Windows NT. Эта качественная характеристика приводит к тому, что появляется возможность проводить атаки на более слабую хэш-функцию Lan Manager даже в случае использования хэш-функции Windows NT.

Уязвимости протокола аутентификации

Аутентификация в PPTP-протоколе очень похожа на аутентификацию в CIFS, поэтому все уязвимости, описанные для протокола аутентификации в CIFS, имеют место и при аутентификации в протоколе PPTP. Следует добавить, что применение хэш-функций в ходе аутентификации делает протокол аутентификации более уязвимым к возможным атакам, поскольку используемые хэш-функции являются уязвимыми.

Кроме того, данный протокол аутентифицирует только клиента. Атакующий злоумышленник, выполняющий подмену соединения, может тривиально замаскироваться под сервер. Если шифрование разрешено, атакующий не сможет посылать и принимать сообщения (пока не взломает шифр), однако, используя старое значение вызова, он в состоянии получить данные двух сессий, зашифрованные одним ключом.

Уязвимости протокола MPPE

Формально MPPE не может считаться полноценным протоколом, в соответствии с которым происходит шифрование пакетов PPTP (с использованием поточного шифра RC4 с 40- или 128-битным ключом). Необходимо, чтобы при включенном режиме шифрования данных шифровались только те пакеты, чьи номера протоколов лежат в диапазоне 0x0021 – 0x00fa. Типы пакетов, шифрование которых осуществляется/не осуществляется, регламентируются в RFC 1700.

В MPPE степень защиты ключа не превышает степени защиты пароля, поскольку ключи генерируются с использованием хэш-кода пароля и случайного числа, пересылаемого от сервера клиенту (только для 128-битных ключей). Большая часть паролей имеет длину существенно меньше 40 бит и раскрывается с помощью словарных атак.

Общая же степень защиты определяется не длиной 40 или 128 бит, а количеством битов энтропии пароля, поэтому любая программа, которая использует словарь слабых паролей, способна прочитать зашифрованный трафик. Кроме того, наличие стилизованных участков PPTP-пакета позволяет набрать статистику для облегчения последующего вскрытия кода.

128-битный RC4 использует в процессе генерации ключей 64-битную случайную величину. Такой подход делает словарную атаку непрактичной. Тем не менее метод «грубой силы» применительно к паролям более эффективен, нежели тот же метод по отношению к ключам. Случайное число также означает, что для двух сессий с одним паролем будут использованы разные 128-битные ключи RC4, хотя для шифрования текста в обоих направлениях будет применен один и тот же ключ.

Поскольку RC4 является поточным алгоритмом шифрования, атакующий имеет возможность изменять передаваемый трафик, при этом изменения могут быть не обнаружены. Так, например, замена или частичная подмена управляющего бита или байта в зашифрованном пакете может привести к самым неприятным последствиям.

Еще одной проблемой при употреблении MPPE является механизм синхронизации ключей. Так, атакующий может либо подавать запросы на рассинхронизацию, либо выбрасывать MPPE-пакета с неверным значением счетчика пакетов. Если перед обменом 256-го пакета, когда происходит смена сеансового ключа, постоянно выполнять эту операцию, атакующий может добиться успеха – сеансовый ключ не будет изменен.

Уязвимости протокола PPTP

Существует несколько возможных путей проведения атак на протокол PPTP, а именно:

- использование снифферов позволяет получать некоторое количество служебной информации о пользователе, а также набрать статистику для последующего подбора паролей. Так, если атакующий регулярно передает пакеты PPTP_START_SESSION_REQUEST, он может наблюдать создание новых и закрытие существующих соединений. Таким образом, злоумышленник может собрать информацию о системе и шаблонах ее использования, при этом ему не нужно быть поблизости;
- открытость для атаки этапа конфигурации соединения. Дело в том, что пакеты согласования параметров PPP передаются до начала шифрования и после его окончания и реальная аутентификация пакетов не производится. В этом случае возможна так называемая атака методом «внедрение ложного сервера»;

- успешное проведение атак типа «отказ в обслуживании» с использованием управляющего канала PPTP-протокола.

Таким образом, протокол PPTP от Microsoft весьма уязвим при реализации и обладает серьезными недостатками с точки зрения теории протоколов. С учетом перечисленных недостатков в Microsoft была произведена его доработка, но, как утверждают Брюс Шнеер и Питер Мадж, изменения носили «косметический» характер, так что обновленная версия PPTP-протокола унаследовала большинство недостатков предыдущей версии.

3.3.4. Система Secret Net NT

Система Secret Net NT (Информзащита) предназначена для организации защиты информации в ЛВС на основе сетевой ОС Windows NT. Она позволяет организовать эффективную защиту отдельного *домена локальной сети Windows NT*.

Система Secret Net NT не подменяет собой систему разграничения доступа сетевой ОС Windows NT, а дополняет ее в части защиты рабочих станций сети, позволяя тем самым повысить защищенность всей автоматизированной системы обработки информации в целом. Иначе говоря, система Secret Net NT является специализированным программно-техническим продуктом, дополняющим ОС Windows NT функциями защиты от несанкционированного доступа к различным ресурсам рабочих станций и серверов домена локальной сети Windows NT, который позволяет централизованно управлять этой защитой в рамках данного домена.

Дополнительно к стандартным механизмам защиты, реализованным в ОС Windows NT, система Secret Net NT обеспечивает:

- опознание (идентификацию) пользователей при помощи специальных аппаратных средств (Touch Memory и Smart Card). Возможна организация усиленной аутентификации пользователя сервером управления доступом;
- полномочное (мандатное) управление доступом пользователей к данным. Дополнительно к избирательному (дискреционному) управлению доступом, реализованному в Windows NT, возможно также разграничение доступа к файлам (на локальных и удаленных дисках) в соответствии со степенью конфиденциальности сведений, хранящихся в файле, и уровнем допуска пользователя;
- возможность подключения и использования средств криптографической защиты данных, передаваемых по сети, и данных, хранящихся в файлах на внешних носителях (жесткие и гибкие магнитные диски и т.д.). В сети возможно определение рабочих станций, обмен

данными между которыми будет осуществляться в криптографически защищенном виде;
- централизованное оперативное управление доступом пользователей к совместно применяемым ресурсам (каталогам и принтерам) как в одноранговой, так и в доменной сети;
- оперативный контроль работы пользователей сети;
- оповещение администратора безопасности о событиях несанкционированного доступа;
- централизованный сбор и анализ содержимого журналов;
- контроль целостности программ, используемых ОС и пользователем.

Система Secret Net NT состоит из двух основных частей (рис. 3.8), которые при установке системы защиты в домене локальной сети Windows NT размещаются на различных серверах и рабочих станциях, входящих в состав домена. Серверная часть системы Secret Net NT всегда устанавливается на основной контроллер домена (Primary Domain Controller).

Клиентская часть системы Secret Net NT может быть установлена на любом сервере или рабочей станции домена, включая и основной контроллер. При этом на некоторые рабочие станции клиентская часть может не устанавливаться. В подобном случае эти рабочие станции останутся незащищенными. В дальнейшем, для простоты изложения, все серверы и рабочие станции домена, отличные от основного контроллера домена, будем называть рабочими станциями.

Серверная часть системы Secret Net NT состоит из следующих компонентов и подсистем:

- сервер управления доступом;
- программа управления NetAdmin;
- подсистема ведения базы данных системы защиты.

Клиентская часть системы Secret Net NT состоит из следующих компонент и подсистем:

- компонент обеспечения загрузки легальной копии ОС;
- агент сервера управления доступом;
- подсистема опознавания пользователя;
- подсистема контроля целостности;
- подсистема полномочного управления доступом;
- подсистема криптографической защиты;
- локальная база данных системы защиты.

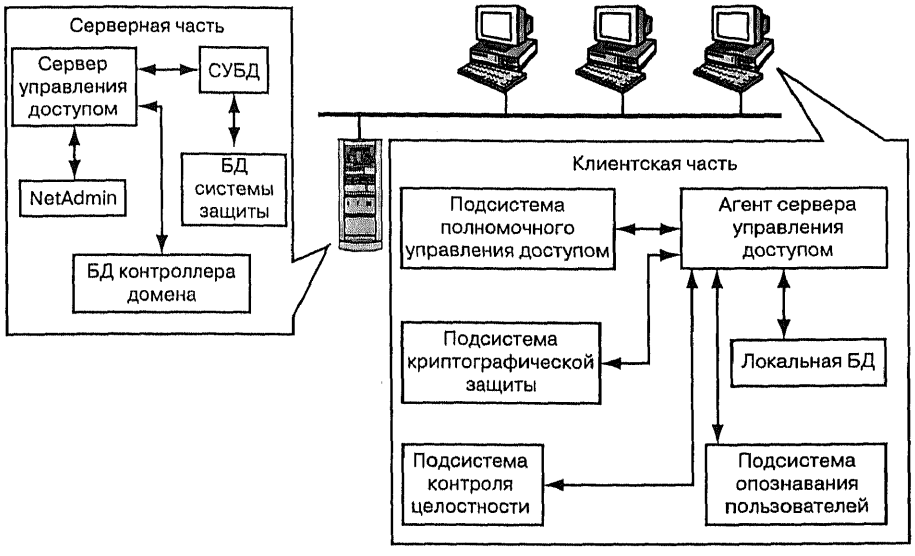

Рис. 3.8. Архитектура Secret Net NT

При правильном выборе конфигурации системы и режимов использования ОС Windows NT, а также при соответствующей настройке имеющихся средств защиты и корректно осуществляемой организационно-административной поддержке их применения ОС Windows NT позволяет создавать защищенные распределенные автоматизированные системы обработки данных.

Для применения ОС Windows NT с целью обеспечения защиты в ЛВС необходимо соблюдать следующие условия:

- использовать ОС Windows NT 4.0 Service Pack 3. После инсталляции ОС Windows NT необходимо установить необходимые права доступа пользователей к различным каталогам на жестких дисках и ключам реестра;
- система ОС Windows NT должна быть установлена на раздел с файловой системой NTFS. Необходимо, чтобы на диске отсутствовали иные операционные системы. В ЛВС должен присутствовать основной домен-контроллер. Обмен данными между рабочими станциями ЛВС должен осуществляться с использованием протокола TCP/IP;
- система Secret Net NT должна быть установлена с платой защиты от загрузки и настроена следующим образом:
  - для всех пользователей системы должны быть подготовлены ключевые носители (дискеты или touch memory), и для всех пользователей

системы должен быть установлен режим усиленной аутентификации;
- на всех компьютерах локальной сети (за исключением основного контроллера домена) должен быть установлен режим шифрования трафика;
- должно быть установлено принудительное затирание файла подкачки страниц, а также затирание данных на жестком диске в один проход;

• организационными мерами защиты должна быть обеспечена сохранность ключевой информации пользователей, а также управление настройками компьютера с основного контроллера домена.

3.4. Защита информации при межсетевом взаимодействии

3.4.1. Общие сведения

Появление локальных и глобальных сетей передачи данных предоставило пользователям удивительные возможности для оперативного обмена информацией. Если до недавнего времени подобные сети создавались только в специфических и узконаправленных целях (академические сети, сети военных ведомств и т.д.), то развитие Internet и аналогичных систем привело к использованию глобальных сетей передачи данных в повседневной жизни практически каждого человека.

Повсеместное распространение Internet привело к необходимости применять в телекоммуникационном оборудовании и информационных услугах открытые и универсальные решения. Сама жизнь требовала обеспечить совместимость и интегрируемость предоставляемых продуктов и услуг в рамках Internet, но для реализации такого подхода пришлось пожертвовать безопасностью создаваемых систем. Недостатки в защите отправляемых по Internet данных являются неизбежным следствием недоработанности вопросов сохранения конфиденциальности в базовом стеке протоколов Internet – TCP/IPv4. Поскольку существующая версия стека протоколов TCP/IP была разработана в 70-х годах, соответственно его создатели даже предположить не могли, что через некоторое время их детище получит статус международного промышленного стандарта. Первые нарушения безопасности в сети Internet были зафиксированы

в 1987 году. С тех пор и по сегодняшний день их число стремительно растет. Дело в том, что количество хостов в сети Internet имеет экспоненциальную зависимость, и уже в 1996 году их было порядка 12 млн, но одновременно с этим возрастает и число атак, которым ежедневно подвергаются компьютеры. По статистике института CSI (Computer Security Institute), каждая пятая машина уже подвергалась тому или иному виду подобных атак. С 1991 года число незаконных вторжений возросло на 498%, а число пострадавших узлов на 781%. По результатам опроса, проведенного CSI среди 500 наиболее крупных организаций, компаний и университетов, потери, вызванные этими атаками, оцениваются в 66 млн долларов, и все это происходит несмотря на то, что существует большое количество средств защиты информации различного уровня: межсетевые экраны, средства шифрования трафика, криптографические протоколы и т.д. Так, специалисты компании Internet Security Systems считают, что в любой сети, основанной на протоколе TCP/IP, насчитывается примерно 135 потенциальных лазеек для нарушителей.

В этом разделе особое внимание уделяется вопросам обеспечения информационной безопасности, возникающим при построении информационно-телекоммуникационных систем на основе сети Internet, поскольку стек протоколов TCP/IP является на сегодняшний день наиболее распространенным. Более того, основные принципы обеспечения безопасности, рассмотренные применительно к TCP/IP, могут быть перенесены на другие протоколы (например, DECnet и др.).

Для начала проведем анализ уязвимостей стека протоколов TCP/IP, угроз, возникающих при использовании Internet в качестве транспортной среды, и основных принципов противодействия атакам. Следует заметить, что все существующие атаки в сети Internet можно условно разделить на два типа:

- атаки на стек протоколов TCP/IP. Internet в данном случае выступает как пример распределенной системы, а основным и самым опасным типом атак на подобные системы являются удаленные;
- атаки на телекоммуникационные службы, предоставляемые в рамках сети Internet. К данному типу атак относятся и учитывающие уязвимости реализации стека протоколов TCP/IP в конкретных ОС, а также атаки, направленные на прикладные телекоммуникационные службы. Этот вид осуществляемых угроз, очевидно, связан прежде всего

с ошибками и уязвимостями в реализации сетевого программного обеспечения и сетевых частей современных ОС.

В рамках этой книги рассматриваются только атаки первого типа, поскольку именно они носят общий и, если выразиться точно, базовый характер.

Очевидно, что атаки второго типа имеют частный характер, их качественный состав постоянно меняется как в общем смысле, так и применительно к отдельным решениям. Например, обнаруженные ошибки и уязвимости ОС Windows NT постоянно устраняются с выходом новой версии Service Pack, но при этом обнаруживаются все новые и новые, вот почему рассмотрение атак на телекоммуникационные службы может носить исключительно частный, прикладной характер. Атаки данного типа и связанные с ними уязвимости постоянно освещаются в Internet на специализированных сайтах, к примеру на CERT.

Необходимо отметить, что в последнее время активно развивается и находит применение идея распределенных вычислений. К сожалению, рамки одной книги не позволяют в полном объеме осветить проблемы использования и безопасности распределенных вычислений, поэтому рекомендуется посетить, например, сайт destributed.net.

Угрозы в распределенных IP-сетях

С точки зрения безопасности распределенные системы характеризуются прежде всего наличием удаленных атак, поскольку компоненты распределенных систем обычно используют открытые каналы передачи данных и нарушитель может не только проводить пассивное прослушивание передаваемой информации, но и модифицировать передаваемый трафик (активное воздействие). И если активное воздействие на трафик может быть зафиксировано, то пассивное воздействие практически не поддается обнаружению. Но поскольку в ходе функционирования распределенных систем обмен служебной информацией между компонентами системы осуществляется тоже по открытым каналам передачи данных, то служебная информация становится таким же объектом атаки, как и данные пользователя.

При этом нарушитель, осуществляя атаку, обычно ставит перед собой следующие цели:

- нарушение конфиденциальности передаваемой информации;
- нарушение целостности и достоверности передаваемой информации;
- нарушение работоспособности системы в целом или отдельных ее частей.

Выявление фактов проведения удаленных атак является чрезвычайно затруднительным и зависит от:

- расположения нарушителя по отношению к объекту атаки; оно может быть внутрисегментное или межсегментное;
- уровня воздействия согласно модели ISO/OSI.

Именно трудность выявления факта проведения удаленной атаки выводит этот вид неправомерных посягательств на первое место по степени опасности, поскольку необнаруживаемость препятствует своевременному реагированию на осуществленную угрозу, в результате чего у нарушителя увеличиваются шансы успешной реализации атаки.

Проанализируем типовые сценарии действий нарушителя по реализации удаленных атак и на этой основе выявим главные причины успешного проведения удаленных атак и сформулируем перечень рекомендаций по минимизации вероятности их проведения нарушителем.

Среди наиболее распространенных атак на распределенные системы, построенные на основе IP-протоколов, следует отметить следующие:

- анализ сетевого трафика. Целью атак подобного типа является прослушивание каналов связи и анализ передаваемых данных и служебной информации с целью изучения топологии и архитектуры построения системы, получения критической пользовательской информации (например, пароли пользователей или номера кредитных карт, передаваемых в открытом виде). Для реализации данной атаки существует большое количество программных средств, например IP-Watcher; также с успехом могут использоваться такие средства, как LanAnalyzer или NetworkMonitor. При анализе сетевого трафика нарушитель может находиться как внутри сегмента, так и между сегментами. Атакам данного типа подвержены такие протоколы, как FTP или Telnet, особенностью которых является то, что имя и пароль пользователя передаются в рамках этого протокола в открытом виде;
- подмена одной из сторон информационного обмена в распределенных системах. Существование такого типа атак является следствием недостатков в аутентификации сторон информационного обмена в стеке протоколов TCP/IP. При этом следует отметить, что в рамках стека протокола TCP/IP существует два типа протоколов транспортного уровня: TCP (с установлением виртуального канала передачи данных); UDP (без установления виртуального канала передачи данных). Виртуальный канал устанавливается за счет подтверждения приема

сообщений в ходе его формирования; по нему сообщения принимаются и передаются с регистрацией их последовательности. Здесь же осуществляется управление потоком сообщений, организовывается повторная передача искаженных пакетов и т.д. В том случае, когда стороны информационного обмена используют протокол UDP, проведение атак значительно облегчается по сравнению с протоколом TCP. В сети Internet атаки, основанные на подмене одной из сторон TCP-соединения, называются hijacking. С использованием данной атаки нарушитель может начать работу с FTP- или Telnet-сервером от имени легального пользователя. Пример проведения подобной атаки приводится ниже;

- внедрение ложного объекта распределенной системы. Внедрение ложного объекта возможно из-за недостаточной защищенности маршрутизирующих и управляющих протоколов, что позволяет осуществлять навязывание ложных маршрутов прохождения сообщений при помощи удаленной модификации служебных пакетов, передаваемых в рамках протоколов маршрутизации (например, в Internet широко применяются протоколы RIP, OSPF, ICMP и SMNP). С другой стороны, при таких атаках используется тот факт, что в глобальных распределенных системах часто приходится сталкиваться с недостатком информации, необходимой для адресации сообщений. В IP-сетях данную проблему решают протоколы удаленного поиска, такие как DNS и ARP. Внедрив ложный объект, нарушитель изменяет путь прохождения трафика между легальными участниками информационного обмена, получая тем самым полный контроль над проходящим через него потоком данных, который он может модифицировать (удалять, изменять и вставлять новые сообщения). Наиболее распространенными в сети Internet угрозами являются: внедрение ложного ARP-сервера, внедрение ложного DNS-сервера (данная атака может реализовываться как при помощи перехвата DNS-запроса, так и с помощью создания направленного шторма ложных DNS-ответов на атакуемую рабочую станцию или на атакуемый DNS-сервер), а также создание ложного маршрутизатора с использованием протокола ICMP;
- отказ в обслуживании. Возможность проведения атак подобного рода осуществляется из-за неправильной обработки некорректных данных программным обеспечением атакуемой стороны или вследствие того, что сетевая ОС может иметь только ограниченное число виртуальных сетевых соединений. Подобные удаленные атаки также могут проводиться за счет генерации максимально возможного числа запросов на

соединение (направленный шторм запросов) или просто пакетов с данными. Результатом подобной атаки может являться либо временный выход из строя атакуемого хоста, либо полное «замирание» системы с необходимостью дальнейшей перезагрузки. В ситуациях, когда необходимо обеспечить оперативность обработки данных информационно-вычислительной системой, защита от такого рода атак приобретает особую актуальность. Типичным примером атак данного типа в сети Internet являются нарушения работоспособности хоста посредством направленного шторма ложных TCP-запросов на создание соединения.

Пример проведения типовой удаленной атаки в IP-сетях

Здесь приводится типовая удаленная атака, направленная на подмену одной из сторон TCP-соединения. Для начала рассмотрим схему установления TCP-соединения (рис. 3.9). В ходе установления соединения в TCP-пакете для его идентификации используются два 32-разрядных поля, выполняющие функции счетчиков пакетов: Sequence Number (SN) и Acknowledgement Number (AN), а также 6-битное поле Control Bits, содержащее управляющие флаги.

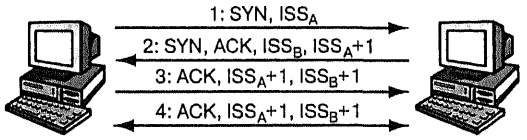

Рис. 3.9
Установление TCP-соединения

Последовательность действий для установления TCP-соединения такова:

1. Участник A посылает участнику B сообщение с установленным управляющим битом SYN, содержащее в поле SN некоторое начальное значение (ISS_A).
2. Участник B отвечает сообщением с установленными контрольными битами SYN и ACK, содержащим в поле SN свое начальное значение (ISS_B), а в поле AN – $ISS_A + 1$.
3. Участник A, получая сообщение 2, проверяет значение поля AN и в случае успешной проверки отправляет участнику B сообщение с установленным управляющем битом ACK, содержащее в поле SN значение $ISS_A + 1$, а в поле AN – $ISS_B + 1$.
4. Теперь участники A и B могут обмениваться TCP-сообщениями, содержащими полноценные данные, и при этом в подобных сообщениях установлен управляющий бит ACK, а в полях SN и AN должны быть значения $ISS_A + 1$ и $ISS_B + 1$ соответственно.

Атакующему для проведения атаки на конкретное TCP-соединение необходимо только получить текущие начальные значения ISS_A и ISS_B. Если нарушитель имеет возможность анализировать трафик, очевидно, что задача получения начальных значений представляется предельно простой – они берутся из перехваченных сообщений. В случае, если атакующий находится в другом сегменте и не имеет возможности анализировать трафик атакуемой стороны, задача подмены сводится к угадыванию начальных значений. Решение этой задачи облегчается тем, что разработчики сетевых компонентов ОС почему-то отказались от использования принципа случайности данных значений. Более того, даже в рекомендациях RFC 793 предложено получать начальные значения по счетчику, который должен увеличиваться каждые 4 мкс на 1 (единицу). В большинстве ОС закон генерации начальных значений зависит от системного времени, поэтому угадать начальные значения очень легко.

Теперь рассмотрим пример удаленной атаки подобного типа. Пусть сервер A находится в доверительных отношениях с хостом B. Целью злоумышленника в данном случае может служить установление TCP-соединения с сервером A от имени хоста B. Нарушитель, прежде чем провести атаку на сервер A, должен установить с ним легальное соединение с целью узнать закон генерации начальных значений. При проведении атак происходит следующая последовательность действий:

1. Хост нарушителя отправляет запрос на сервер A для установления TCP-соединения от имени хоста B. Запрос имеет следующий вид: SYN, ISS_X.
2. Получая данный запрос, сервер A анализирует IP-адрес отправителя (нарушитель вставляет IP-адрес хоста B) и, полагая, что запрос пришел от хоста B, отправляет ему следующий ответ: SYN, ACK, ISS_B, $ISS_X + 1$. Данное сообщение никогда не дойдет до нарушителя, поскольку оно будет направлено на IP-адрес хоста B, но для того, чтобы хост B, получив ответ на запрос, которого он не посылал, не направил серверу A сообщение о закрытии соединения (с установленным битом RTS), нарушитель должен заблокировать на время работу хоста B (проведя атаку типа «отказ в обслуживании»).
3. Нарушитель, имея представление о законе формирования начальных значений, может произвести угадывание ISN_B. Затем он отправляет серверу A следующее сообщение: ACK, $ISS_X + 1$, $ISS_B + 1$.

После чего сервер A решает, что он установил соединение с хостом B, но сообщения от сервера A не дойдут до нарушителя, поскольку направляются

на IP-адрес хоста В, однако нарушитель сможет выполнять на сервере А удаленные команды.

Возможности проведения перечисленных выше удаленных атак в распределенных системах базируются на:

- использовании общедоступных каналов передачи данных. Например, большинство локальных сетей в Internet имеют топологию с общей шиной, что облегчает задачу по перехвату передаваемых сообщений. Следует отметить, что топология построения локальной сети типа «звезда» устойчива к некоторым удаленным атакам, например к внедрению ложного ARP-сервера;
- уязвимости в процедурах идентификации, реализованных в стеке протоколов TCP/IP. Идентифицирующая информация на уровне IP передается в открытом виде, что приводит к неконтролируемому ее искажению. Дальнейшая идентификация хостов в сети Internet возложена на транспортный уровень, однако протокол UDP не содержит дополнительных средств идентификации, и как следствие – протоколы более высокого уровня, применяющие UDP, потенциально подвержены удаленным атакам, использующим уязвимости в идентификации. Единственным протоколом, предназначенным для решения подобных задач, является TCP (о недостатках процедуры идентификации в протоколе TCP сказано в примере удаленной атаки в Internet);
- отсутствии возможности контроля за маршрутом прохождения сообщений в сети Internet. Это приводит к следующему: истинный IP-адрес отправителя нельзя определить, что делает удаленные атаки практически ненаказываемыми, поскольку проследить маршрут передачи IP-пакета является в существующей версии протокола IPv4 серьезной проблемой;
- необходимости использования механизмов удаленного поиска в сети Internet;
- отсутствии в базовой версии стека протоколов TCP/IP механизмов, обеспечивающих конфиденциальность и целостность передаваемых сообщений. На сегодняшний день инструментарий, обеспечивающий шифрование и подписание передаваемых сообщений или всего трафика, является всего лишь дополнением к существующим технологиям обеспечения электронной связи, но их применение требут введения некоторых ограничений на работу пользователей или на работу информационной системы, куда встраиваются данные механизмы.

Атаки подобного рода, очевидно, возможны только на сетевом и транспортном уровнях Internet, но, как уже упоминалось, каждый вышележащий протокол будет наращивать потенциал нарушителя для проведения виртуальных нападений.

Реализация средств, обеспечивающих информационную безопасность, должна обязательно учитывать тот факт, что информационно-вычислительные системы, построенные с использованием TCP/IP, изначально уязвимы для удаленных атак. Например, Internet Information Server (фирмы Microsoft) использует для разграничения доступа идентификацию пользователей на основе IP-адреса, но в силу того, что IP-адрес может быть изменен, этот тип разграничения доступа к информационным ресурсам обладает уязвимостью, выражающейся в возможности несанкционированного получения сведений из базы данных.

Поэтому защита основной версии стека протоколов TCP/IP является важнейшей задачей при реализации защиты межсетевого взаимодействия, причем реализация защиты протоколов верхних уровней должна учитывать уязвимость базовой модели.

С учетом недостатков в безопасности IPv4, а также необходимости модификации базовой версии IP-протокола в рамках обеспечения транспортной компоненты информационных систем подготовлена новая версия IP-протокола – IPv6.

Протокол IP v.6 и рекомендации IPSec

Расширение инфраструктуры сети Internet приводит к устареванию протоколов стека TCP/IP. Современный уровень развития межсетевого взаимодействия выдвигает новые требования к базовым протоколам, заключающиеся в обеспечении передачи информации в режиме реального времени, возможности контроля загрузки каналов передачи данных, наличии механизмов защиты информации. Базовые протоколы семейства TCP/IP не удовлетворяют ни одному из этих требований. Следовательно, близок момент, когда четырехмиллиардная емкость адресного пространства TCP/IP будет практически полностью исчерпана.

Для решения указанных проблем с июля 1992 года тематическая группа по технологии Internet (IETF) приступила к разработке основных требований к протоколам семейства TCP/IP нового поколения, названным IP Next Generation (IPng). В январе 1995 года данный проект был опубликован под общим заголовком «Рекомендации для протокола IP нового поколения». В этом документе были представлены основные требования к IPng, как-то: форматы заголовков пакетов, подходы к организации адресного

пространства и маршрутизации, а также принципы построения средств обеспечения безопасности в современных IP-сетях.

IPv6 должен обеспечивать следующие характеристики, отсутствующие у IPv4:

- расширенное адресное пространство. IPv6 использует 128-битные адреса вместо 32-битных. В результате адресное пространство увеличивается в 2^{96} раза, что будет достаточно даже в случае неэффективного распределения сетевых адресов;
- улучшенные возможности маршрутизации. В связи с увеличением межсетевого трафика, связанного с обработкой больших объемов мультимедийной информации и использованием сети Internet в различных сферах деятельности, весьма существенной является необходимость обеспечения высоких скоростей маршрутизации. Без применения эффективных алгоритмов обработки пакетов данных невозможно повысить скорость работы маршрутизаторов до уровня, сравнимого со скоростью передачи информации по каналам связи;
- управление доставкой информации. IPv6 позволяет отмечать соответствие конкретного пакета определенным условиям его передачи, заданным отправителем. В результате достигается регулирование скорости передачи конкретных потоков данных, что позволяет обеспечивать эффективную поддержку специальных протоколов (например, в режиме реального времени и др.). За счет назначения приоритетов передачи данных по определенным протоколам появляется возможность гарантировать первоочередность обработки наиболее критической информации и предоставления важным данным всей полосы пропускания канала связи;
- средства обеспечения безопасности. IPv6 предоставляет возможности защиты от атак, связанных с подменой исходных адресов пакетов, и от несанкционированного доступа к полям данных пакетов. Эти возможности достигаются за счет применения алгоритмов аутентификации и шифрования. При создании средств обеспечения безопасности в рамках IPv6 учитывалось, что изменение базовых протоколов семейства TCP/IP вызвало бы полную перестройку сети Internet, таким образом, спецификация IPSec является дополнением по отношению к протоколам IPv4 и IPv6. Гарантии целостности и конфиденциальности данных в спецификации IPSec обеспечиваются за счет использования механизмов аутентификации и шифрования соответственно. Последние, в свою очередь, основаны на предварительном согласовании

сторонами информационного обмена, так называемого контекста безопасности – применяемых криптографических алгоритмов, алгоритмов управления ключевой информацией и их параметров. Спецификация IPSec предусматривает возможность поддержки сторонами информационного обмена различных протоколов и параметров аутентификации и шифрования пакетов данных, а также схем распределения ключей. При этом результатом согласования контекста безопасности является указатель на определенный элемент внутренней структуры стороны информационного обмена, описывающей возможные наборы параметров безопасности.

Общая идея построения средств безопасности имеет схожий со SKIP-протоколом характер: для обеспечения аутентификации используются AH-заголовки, а для обеспечения конфиденциальности – ESP-заголовки (см. раздел «Протокол SKIP»).

Поскольку спецификации IPSec могут применяться как в IPv4, так и в IPv6, их развитие осуществляется независимо от распространения протоколов IPv6.

3.4.2. Обеспечение защиты информации при построении VPN

Широкое распространение глобальных сетей передачи данных предоставляет возможность объединять территориально разбросанные локальные сети организаций для создания так называемых частных виртуальных сетей (VPN). Глобальные сети в этом случае выступают как транспортный компонент, объединяющий локальные сети в единую информационно-вычислительную систему. Создание VPN велось и до стремительного роста глобальных сетей, однако для их объединения служили выделенные каналы передачи данных, что приводило:

- к высокой стоимости аренды выделенных каналов связи;
- к жесткой привязанности к местоположению. Например, в случае переезда офиса компании с развернутым сегментом локальной сети, связанным выделенным каналом с общей сетью предприятия, возникали дополнительные проблемы с последующим подключением локальной и общей сетей.

Использование коммутируемых каналов глобальных сетей передачи данных позволило добиться гибкости, масштабируемости и универсальности при построении VPN. Кроме того, расширение Internet позволило многим компаниям перевести часть персонала на домашний режим работы. В этом случае сотрудник имеет постоянную связь с фирмой в рамках,

например, системы электронного документооборота, при этом проблемы связи его с сетью офиса решаются посредством провайдеров. Все это выглядит так, будто он работает на компьютере, установленном в кабинете. По мнению аналитиков, перевод 30% персонала, работающего в офисе, на домашнюю работу способствует повышению производительности труда всего коллектива в целом. В последнее время все более популярным становится использование портативных компьютеров совместно с сотовыми телефонами, обеспечивающими работу сотрудника не только дома, но и в дороге.

В связи с этим VPN бывают двух видов: объединяющие территориально удаленные сети в одну информационно-вычислительную сеть и позволяющие подключаться удаленным рабочим станциям, расположенным в пространстве глобальных сетей передачи данных, к корпоративной сети предприятия (рис. 3.10).

Широкое распространение VPN, построенных на основе глобальных сетей передачи данных, делает их уязвимыми по отношению к атакам потенциального нарушителя, поскольку такая конфигурация наследуют все уязвимости стека используемых протоколов, о которых было сказано выше. Следует отметить, что при использовании выделенных каналов связи обеспечение безопасности информации сводится к шифрованию и имитозащите трафика, передаваемого по выделенному каналу. Однако наиболее актуально вопросы безопасности стоят для VPN, где в качестве транспортного протокола используется TCP/IP. (Под глобальной сетью передачи данных здесь будем подразумевать Internet.)

Сети такого типа часто используют Internet для передачи конфиденциальных данных, пересылаемых в виде информационных пакетов по

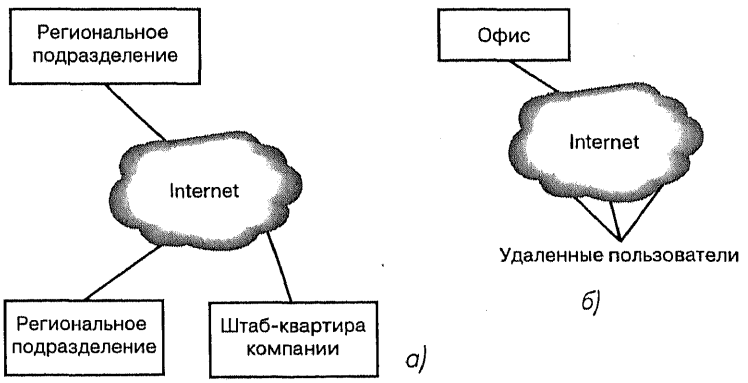

Рис. 3.10. Схемы построения VPN

протоколам IP и/или IPX. Средства обеспечения информационной безопасности трансформируют IP-пакеты, встраивая их внутрь других пакетов (так называемая инкапсуляция), которые затем маршрутизируются через Internet. Таким образом, информационный поток трансформируется в другой информационный поток (так называемое *туннелирование*). При этом шифруются не только поля данных передаваемого IP пакета, но и адресная часть, и служебные поля данных.

Говоря о протоколах, используемых для передачи информации по сетям VPN, следует отметить, что основными в этих случаях являются четыре протокола: Layer 2 Forwarding Protocol (протокол трансляции канального уровня), Layer 2 Tunneling Protocol (протокол туннелирования канального уровня), Point-to-Point Tunneling Protocol (протокол туннелирования между узлами), а также протокол IP Security (IPSec), предложенный комитетом IETF.

Первые три спецификации известны под общим названием протоколов трансляции канального уровня, поскольку в соответствии с ними пакеты протоколов сетевого уровня (AppleTalk, IP и IPX) сначала инкапсулируются в другие пакеты протокола канального уровня (PPP), а уже затем передаются адресату по IP-сети. Хотя эти спецификации и претендуют на решение проблемы безопасности в сетях VPN, они не обеспечивают шифрования, аутентификации, проверки целостности каждого передаваемого пакета, а также средств управления ключами. На сегодняшний день наиболее современным решением защиты сетей VPN является спецификация IPSec. Поэтому в случае использования первых трех спецификаций для обеспечения безопасности информации при передаче в рамках VPN необходимо применение дополнительных средств ее защиты.

Основные подходы к обеспечению защиты информации в VPN

Тема информационной безопасности при построении VPN освещает вопросы безопасности при объединении нескольких локальных сетей (то есть защита информационного канала и информационных ресурсов для соединения типа *локальная сеть – локальная сеть* (ЛС-ЛС)) и вопросы обеспечения информационной безопасности при подключении к локальной сети удаленного пользователя (ЛС-УП). Каждый из представленных вопросов будет иметь свой спектр угроз и соответственно методов и средств защиты информации.

Для VPN, построенных по принципу ЛС-ЛС, характерны следующие угрозы:

- изменение, удаление или добавление части трафика или отдельных IP-пакетов или их полей;

- анализ передаваемого трафика;
- инициация информационного обмена злоумышленником от имени одного из сегментов VPN.

При этом целями нападения могут быть:

- получение доступа к конфиденциальной информации, передаваемой между сегментами VPN;
- получение несанкционированного доступа к информационным ресурсам, предоставляемым в сегментах VPN;
- нарушение бесперебойности работы VPN;
- нарушение достоверности и целостности передаваемой информации и предоставляемых информационных ресурсов;
- наработка статистики функционирования сегментов VPN между собой.

Таким образом, при обеспечении защиты информации в рамках соединения типа ЛС-ЛС необходимо применять средства, которые смогли бы обеспечить:

- шифрование трафика или отдельных пакетов; при этом должна быть обеспечена защита информации разного уровня конфиденциальности;
- двустороннюю аутентификацию участников информационного обмена;
- выработку ЭЦП или контрольных сумм на передаваемые IP-пакеты;
- имитозащиту передаваемой информации;
- маскировку передаваемой информации путем сжатия трафика при передаче, а также генерирование ложного трафика в ходе приостановки обмена информации в VPN.

Когда же безопасность информации обеспечивается в рамках соединения ЛС-УП, очевидно, что большинство приведенных угроз и целей нарушителя будут сохраняться и в данном случае. При этом дополнительно следует обеспечить:

- разграничение доступа к предоставляемым локальной сетью информационным ресурсам;
- аутентификацию передаваемой информации на уровне пользователя.

Выбираемые решения по обеспечению защиты информации в VPN должны учитывать следующие требования, выдвигаемые современным уровнем построения информационных систем, а также специфическими аспектами построения VPN:

- защищаемая информация должна корректно маршрутизироваться и обрабатываться промежуточными системами, такими как маршрутизаторы и т.д.;

- система защиты должна быть масштабируема, то есть должны учитываться потребности защищаемой системы при ее росте и усложнении архитектуры построения;
- система защиты не должна накладывать ограничений на производительность и обрабатываемую информацию прикладной системы; при этом интеграция защиты информации с подобной системой не должна приводить к ее существенной доработке;
- учитывая большой объем передаваемой информации между сегментами VPN, необходимо реализовывать эффективные процедуры распределения и управления ключевой информацией;
- построение системы защиты должно обеспечивать ее централизованное администрирование и аудит;
- защищаемая локальная сеть в случае использования стандартных средств межсетевого обмена должна быть недоступна из глобальной сети.

Большинство систем защиты информации в VPN работают на сетевом уровне, это объясняется тем, что:

- применение защиты на сетевом уровне является прозрачным для прикладных систем;
- средства защиты сетевого уровня не ограничивают общую структуру и стратегию развития прикладной системы.

Среди применяемых решений защиты информации в VPN следует отметить межсетевые экраны (в том числе и шифрующие), программно-аппаратные комплексы (ПАК) шифрования IP-трафика (далее – *криптомаршрутизаторы*) и программные средства, реализующие прозрачное шифрование IP-трафика.

Использование межсетевых экранов и ПАК шифрования IP-трафика целесообразно для защиты соединений типа ЛС-ЛС в силу того, что для прохождения большого объема информации через данные соединения требуется высокая производительность подобных систем. Но если межсетевые экраны применяются в основном для фильтрации трафика с целью обеспечения защиты информационных ресурсов от несанкционированного доступа и от негативного воздействия из внешних сетей передачи данных, то в криптомаршрутизаторе функции фильтрации являются вспомогательными, а упор делается на обеспечение конфиденциальности, целостности и имитозащиты передаваемой информации. То есть основным является применение средств шифрования, выработки контрольных сумм и имитовставок с одновременным решением вопросов управления и распределения ключевой информации.

При выборе ПАК для создания защищенных VPN следует учитывать, что шифрование сообщений требуется не всегда. Часто это очень дорогостоящая процедура, поскольку без использования специальных аппаратных приставок большинство маршрутизаторов не могут одновременно с шифрованием обеспечивать приемлемый уровень быстродействия. Чтобы оценить необходимость шифрования сообщений в VPN, придется определить и классифицировать все типы данных в интрасети, которые не должны попасть к посторонним. Затем следует оценить размер возможного ущерба от потери таких данных или от их попадания в чужие руки. Далее нужно определить, кто заинтересован в получении ваших конфиденциальных данных, и подумать о мотивах и финансовых возможностях похитителей информации.

В этом разделе мы остановимся на рассмотрении ПАК шифрования IP-трафика, а межсетевым экранам посвящен самостоятельный раздел. Учитывая широкий спектр архитектур построения VPN, существует три основных типа подключения локальной сети к глобальной с использованием криптомаршрутизатора (рис. 3.11):

- простая схема с использованием маршрутизатора (рис. 3.11а). В этом случае маршрутизатор применяется как для защищаемой локальной сети, так и для открытой части локальной сети. Необходимым условием обеспечения безопасности при использовании данной схемы подключения является отсутствие незащищенных каналов обмена информацией между защищенной и открытой локальными сетями;

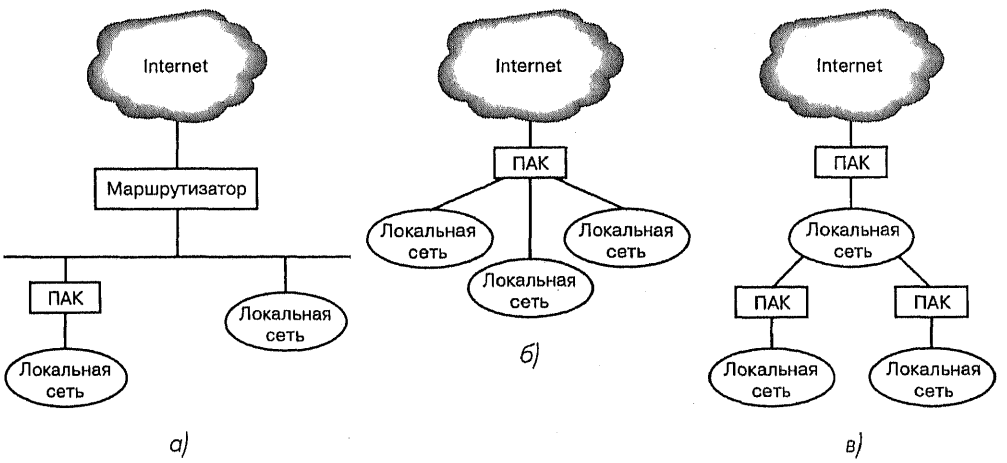

Рис. 3.11. Схемы подключения криптомаршрутизатора

- схема с непосредственным подключением (рис. 3.11б). Один ПАК может поддерживать несколько локальных сетей с разными политиками безопасности и разным уровнем конфиденциальности обрабатываемой информации;
- иерархическая схема (рис. 3.11в). Используется при необходимости обеспечения защиты информации и разграничения доступа для больших локальных сетей с разными уровнями безопасности. Администрирование данной системы защиты также носит иерархический характер.

Программные же средства защиты информации применяются в основном для защиты соединений типа ЛС-УП. В идеальном случае продукты защиты информации при построении VPN должны быть совместимы между собой, чтобы средства защиты информации различного уровня (удаленные пользователи, локальные сети произвольного масштаба и т.д.) могли быть объединены в рамках одной корпоративной сети. Совместимость большинства средств защиты информации в VPN обеспечивается за счет применения при их создании унифицированных решений. В качестве такого решения можно отметить криптографический протокол SKIP (Simple Key management for Internet Protocol), имеющий официальный статус проекта стандарта Internet (Internet draft) и подвергшийся многочисленным испытаниям и апробации в ряде коммерческих продуктов. SKIP в рамках средств, обеспечивающих информационную безопасность, может быть применен и при построении VPN, поскольку его реализация удовлетворяет требованиям, выдвинутым к продуктам подобного рода, кроме того, протокол SKIP является IP-совместимым, поэтому защищаемая информация может беспрепятственно обрабатываться и передаваться системами, использующими реализацию стека протоколов TCP/IP.

SKIP реализован в таких продуктах, как «Шифратор IP-потоков» (ШИП), программный комплекс (ПК) «Игла», межсетевой экран SunScreen SPF 100 и в ряде других.

Протокол SKIP

SKIP является типичным примером протокола, обеспечивающего аутентификационное распределение сеансовых ключей. В основе его создания лежит идея открытого распределения ключей Диффи-Хэлмана, что позволяет строить средства защиты, рассчитанные на применение в глобальных сетях передачи данных, поскольку реализация идеи Диффи-Хэлмана не требует проведения сложных организационно-технических мероприятий по распределению ключевой информации; пользователю нужно иметь только пару ключей – секретный/открытый – и справочник открытых ключей других пользователей.

Основная идея SKIP заключается в том, что ключ, вычисленный при помощи алгоритма Диффи-Хэлмана (Kij, называемый *мастер-ключом*), используется для шифрования пакетных ключей, создаваемых для каждого IP-пакета, при этом сами пакетные ключи добавляются к зашифрованному пакету в зашифрованном виде. Смысл применения пакетных ключей состоит в том, чтобы нарушитель не смог получить достаточного статистического материала для проведения криптоанализа; при этом избыточность, вносимая механизмами управления ключами, только для определенного вида трафика, не превышает 1–2%. При использовании трафика с короткими пакетами SKIP-протокол вносит большую избыточность в передаваемый трафик, и в некоторых случаях производительность канала может падать в четыре раза.

Сама схема работы протокола выглядит следующим образом:

- исходный IP-пакет шифруется с использованием пакетного ключа, и на его основе формируется SKIP-пакет – зашифрованный IP-пакет и SKIP-заголовок (табл. 3.5);
- SKIP-пакет включается в новый IP-пакет, другими словами, производится инкапсулирование; при этом заголовок нового IP-пакета может существенно отличаться от заголовка исходного (например, может быть изменена адресная часть IP-заголовка). Изменение адресной части позволяет реализовывать дополнительные механизмы защиты информационных ресурсов сети. Например, указывая в адресной части IP-адрес шифрующего ПАК, можно сделать локальную сеть недоступной для обращения к ней из глобальной сети с помощью стандартных средств. Фактически машину с реализованным на ней SKIP-протоколом не будет видно из общей сети, в то время как в рамках защищаемой подсети она будет доступна. На полученный пакет вычисляется контрольная сумма с использованием пакетного ключа, при этом пропускаются только динамически изменяемые поля IP-заголовка.

В результате получается стандартный IP-пакет, который можно обрабатывать и маршрутизировать в ходе его передачи по сети. Очевидным преимуществом программной реализации SKIP-протокола является возможность производить (реализовывать) обработку как зашифрованной информации, так и открытой информации.

Существует три варианта применения SKIP-протокола:

- использование SKIP для аутентификации передаваемых IP-пакетов (AH);
- использование SKIP для шифрования передаваемых IP-пакетов (ESP);

- использование SKIP для аутентификации и шифрования передаваемых IP-пакетов (AH/ESP).

Таблица 3.5. Структура SKIP-заголовка

0 1 2 3 4 5 6 7	0 1 2 3 4 5 6 7	0 1 2 3 4 5 6 7	0 1 2 3 4 5 6 7
Ver	Rsvd	Source NSID	Dest NSID	NEXT HEADER
Counter n			
Kij Alg	Crypt Alg	MAC Alg	Comp Alg
Kp, зашифрованное на K_{ij}... (8–16 байт)			
Source Master Key-ID (если значение поля Source NSID ненулевое)			
Destination Master Key-ID (если значение поля Dest NSID ненулевое)			

Ver – номер версии (текущая версия имеет номер 1); Rsvd – зарезервированное поле для будущих версий протокола; Source NSID – указывается пространство имен для мастер-ключа источника; Dest NSID – указывается пространство имен для мастер-ключа адресата; Next Header – идентифицирует тип заголовка, следующего за данным; Counter n – данное поле содержит локальное время источника, системное время на приемной стороне должно быть синхронизировано с точностью не меньше одного часа (значения данного поля используются для исключения проведения атак типа «повторная передача»); Kij Alg – идентифицирует алгоритм, использующийся для шифрования пакетных ключей; Crypt Atg – идентификатор алгоритма, использующегося для шифрования; Mac Alg – идентификатор алгоритма, использующегося для аутентификации; Comp Alg – идентификатор алгоритма, использующегося для сжатия данных перед зашифрованием; Kp – содержит зашифрованное значение пакетного ключа (Kp); Source Master Key-ID – идентификатор мастер-ключа источника; Destination Master Key-ID – идентификатор мастер-ключа адресата.

Использование протокола SKIP для аутентификации IP-пакетов обеспечивает достоверную их доставку за счет вычисления MAC на инкапсулированный IP-пакет. Целостность в этом случае обеспечивается за счет контрольных сумм или имитовставок, которые вместе с дополнительными параметрами находятся в AH-заголовках, содержащихся в пакете в случае использования аутентификации (табл. 3.6).

Таблица 3.6. Структура IP-пакета при использовании аутентификации

Заголовки			Содержимое пакета
IPv4 Hdr	SKIP Hdr	AH Hdr	Инкапсулированный протокол (например, IP, TCP, UDP)

Использование этого протокола в режиме ESP предполагает шифрование инкапсулируемого IP-пакета с одновременным обеспечением защиты от навязывания повторных сообщений (табл. 3.7).

Таблица 3.7. Структура IP-пакета при использовании шифрования

Заголовки			Содержимое пакета
IPv4 Hdr	SKIP Hdr	ESP Hdr	Инкапсулированный протокол (например, IP, TCP, UDP)

Применение совместного режима обеспечивает не только шифрование, но и аутентификацию передаваемого IP-пакета, при этом после SKIP-заголовка будет следовать AH-заголовок, а за ним – ESP-заголовок. Но в данном случае по соображениям безопасности пакетный ключ Kp не может использоваться одновременно для шифрования и выработки MAC, поэтому на основе исходного Kp вычисляются пакетный ключ для аутентификации A_kp и пакетный ключ для шифрования E_kp.

E_kp = h(Kp | 01h) | h(Kp | 00h), где h – хэш-функция
A_kp = h(Kp | 03h) | h(Kp | 02h)

Использование SKIP-протокола для защиты многоадресной доставки (multicast IP)

Эта операция может быть рассмотрена в двух контекстах:

- многоадресная доставка для динамически меняющихся групп (имеются в виду группы, в которых пользователи меняют свои адреса со временем, например видеоконференции);
- многоадресная доставка для постоянных групп, пользователи в которых имеют фиксированные адреса.

Чтобы реализовать SKIP для первого контекста, применяется следующий подход: для динамической группы выделяется общий для всех участников IP-адрес, создается групповой ключ Kg, который имеет то же значение, что и мастер-ключ в случае применения SKIP для одиночной доставки. Но теперь, поскольку группы могут динамически создаваться и удаляться, существуют эффективные механизмы выработки и обновления общего для группы ключа Kg.

В случае постоянных групп применение SKIP-протокола может быть сведено к описанному выше примеру с той только разницей, что группа будет иметь фиксированный Kg и обновление данного ключа может происходить лишь при его компрометации. Поэтому для данного случая не требуется наличие эффективных процедур обновления общего для группы ключа.

Рекомендации по распределению открытых ключей, используемых в SKIP-протоколе

SKIP использует открытые ключи, и в основе его создания – идея Диффи-Хэлмана, для которой уже продемонстрирована потенциальная уязвимость, приводящая к получению нарушителем доступа к конфиденциальной информации. При этом злоумышленнику нужно только иметь доступ к каналам передачи информации и знать спецификации используемых протоколов.

Защитой от данной атаки может служить сертификация открытых ключей. Поэтому значительная часть стандарта на SKIP посвящена организации служб сертификации, причем если для корпоративной сети может использоваться один центр сертификации, то для применения SKIP в глобальных сетях передачи данных необходимо создание иерархических структур подобных центров. В проекте стандарта на SKIP в качестве процедур сертификации и форматов представления сертификатов взяты рекомендации X.509.

Средства, применяемые для защиты информации при построении VPN

Данный раздел рассказывает о применении и основных принципах построения средств защиты VPN на примере четырех криптографических продуктов, которые, по мнению автора, являются наиболее типичными представителями этого семейства, а именно:

- «Шифратор IP-потоков» (далее – «Шип») и ФПСУ-IP, предназначенные для шифрования IP-трафика;
- программное средство «Игла-П», обеспечивающее защиту пользовательских соединений;
- продукты «Застава».

Результаты тестирования «Шифратора IP-потоков», ФПСУ-IP и «Заставы», предоставленные коллективом авторов, в числе которых И. Гвоздев, В. Зайчиков, Н. Мошак, М. Пеленицин, С. Селезнев, Д. Шепелявый, – см. в приложении «Сравнительные характеристики отечественных средств построения VPN».

Криптографический комплекс «Шифратор IP-потоков»

Этот комплекс выполнен в виде отдельного устройства, реализующего шифрование исходящего и входящего трафика локальной сети. КК (или криптомаршрутизатор) «Шифратор IP-потоков» (МО ПНИЭИ) построен на реализации SKIP-протокола. Зарубежные аналоги этого устройства, сконструированные на сходных принципах, широко используются для защиты Internet-трафика. Среди подобных средств следует отметить SWAN (в 1996 году данное устройство обеспечивало защиту 5% трафика в Internet), в качестве протокольной части которого реализованы рекомендации IPSec.

КК «Шип» предназначен:

- для обеспечения конфиденциальности и целостности информации, передаваемой в сетях общего пользования, построенных на основе протоколов IP, X.25, Frame Relay;
- для создания защищенных подсетей передачи конфиденциальной информации;
- для объединения локальных сетей, включая сети Novell (IPX), в единую защищенную сеть;
- для закрытия доступа к ресурсам сети и отдельным компьютерам из сети общего доступа;
- для организации единого центра управления защищенной подсетью.

КК «Шип» представляет собой распределенную систему шифраторов, средств управления шифраторами, средств хранения, распространения и передачи ключевой информации, а также средств оперативного мониторинга и регистрации происходящих событий.

Комплекс обеспечивает:

- закрытие данных на основе использования функций шифрования;
- контроль целостности передаваемой информации;
- аутентификацию абонентов (узлов сети);
- защиту доступа к локальной сети и сокрытие IP-адресов подсети;
- создание защищенных подсетей в сетях общего пользования;
- передачу контрольной информации в центр управления безопасностью защищенной IP-сети;
- поддержку протоколов маршрутизации RIP II, OSPF, BGP в виртуальной сети и локальных сетях;
- фильтрацию IP-, ICMP-, SKIP-пакетов и TCP-соединений на этапе маршрутизации и при приеме/передаче в канал связи;
- поддержку и преобразование различных сетевых протоколов;
- защиту от НСД ресурсов самого шифратора.

При разработке КК «Шип» были учтены международные рекомендации по защите протокола IP. Формат IP-пакетов, содержащих зашифрованную информацию, соответствует IPSEC (RFC 1825, 1826, 1827 1995 года) и предложениям SKIP-07 (фирма SUN Microsystems) с односторонней аутентификацией абонентов.

В состав КК «Шип» входят:

- шифратор IP-потоков – аппаратно-программный комплекс «Шип» (АПК «Шип»);

- центр управления ключевой системой (ЦУКС) «Шип».

АПК «Шип» и ЦУКС «Шип» обеспечивают:

- защиту (шифрование и проверку целостности с использованием имитовставки) данных, передаваемых между узлами сети, согласно ГОСТ 28147-89;
- одностороннюю аутентификацию узлов защищенной сети на основе имитовставки согласно ГОСТ 28147-89;
- управление ключевой системой защищенной сети из одного или нескольких центров управления.

В рамках КК «Шип» реализуется развитая ключевая система, а также эффективные процедуры управления ключами. Ключевая система построена на основе парных симметричных ключей шифрования, которые формируются в виде полной матрицы (серии) на заранее определенное число абонентов системы с учетом необходимого запаса. В состав механизмов управления ключами входит механизм распространения справочника соответствия (номер ключа – IP-адрес – имя компьютера) с использованием стандартного протокола Domain Name Service (DNS).

В зависимости от размеров создаваемой защищенной VPN и от требований к системе защиты по надежности, бесперебойности и работоспособности можно выделить три основных архитектуры построения защищаемых с использованием КК «ШИП» и подобных ему устройств VPN (рис. 3.12):

- обеспечение защиты VPN небольших размеров (а). Данная схема с одним ЦУКС «Шип» актуальна при организации защиты VPN небольшого размера и при отсутствии жестких требований ко времени оповещения абонентов о компрометации какого-либо абонента и жестких требований к полноте собираемых протоколов об ошибках доступа;
- построение защиты VPN с резервированием ЦУКС (б). При организации защиты VPN среднего размера и при наличии жестких требований ко времени оповещения абонентов о компрометации какого-либо абонента и к полноте собираемых протоколов об ошибках доступа следует использовать схему защиты VPN с несколькими ЦУКС «Шип»: один – главный, остальные – резервные. При этом желательно, чтобы ЦУКС «Шип» имели по возможности независимые друг от друга каналы подключения к глобальной сети передачи данных.
- построение региональных защищенных VPN (в). При организации защиты VPN большого размера (более 1024 КК «Шип») либо с целью минимизации межрегионального обмена, либо с целью структурирования управления доступом к VPN следует использовать региональную

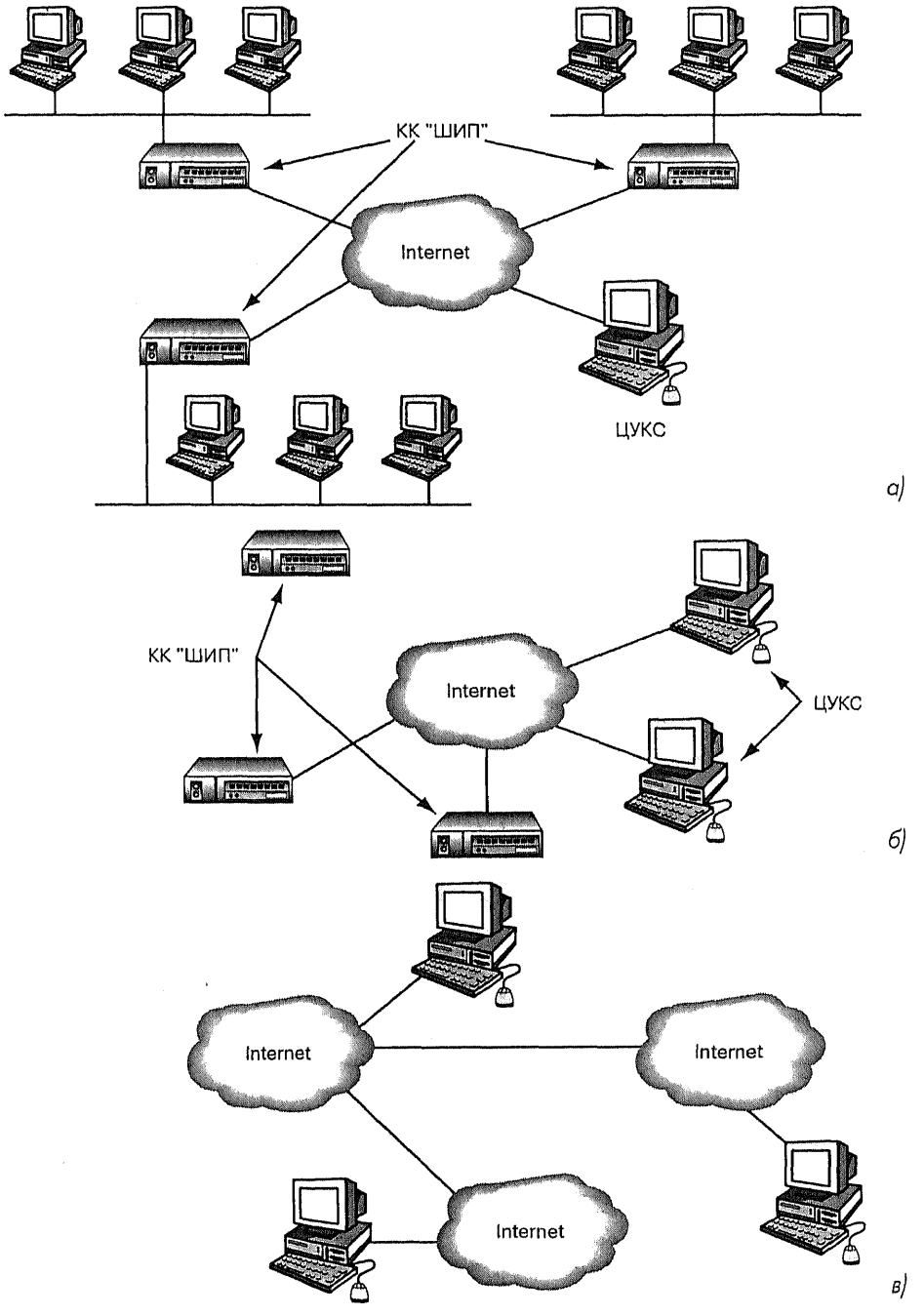

Рис. 3.12. Схемы организации защиты VPN

структуру защиты VPN, в которой главный ЦУКС «Шип» хранит описания и осуществляет сбор протоколов и действия, выполняемые при компрометации ключей, только для региональных ЦУКС «Шип», хранящих описания абонентов.

Использование криптографического комплекса «Шип» позволяет решать ряд задач по обеспечению конфиденциальности и целостности информации, передаваемой в сетях общего пользования, построенных на основе протоколов IP, X.25, Frame Relay, то есть наиболее распространенных в настоящее время типах сетей, а также внутри организаций. Решение заключается в создании защищенных подсетей и закрытии доступа из внешних сетей к ресурсам локальных сетей и отдельным компьютерам. К сожалению, кроме несомненных достоинств КК «Шип» имеет ряд существенных недостатков. На некоторых видах трафика вследствие использования SKIP-протокола происходит падение производительности канала передачи данных. Кроме того, при отсутствии встроенных механизмов синхронизации системного времени необходимо ее обеспечивать, поскольку рассинхронизация приводит к отказу в работе.

Система защиты информации «Застава»

Базовой технологией, используемой при построении продуктов «Застава» (Элвис+), является протокол SKIP (Simple Key Internet Protocol).

Программное обеспечение «Заставы» работает на операционных системах Windows 95/98/NT, Solaris (SPARC и Intel).

Все продукты, реализующие протокол SKIP, имеют открытый интерфейс для подключения любых модулей преобразования данных, в том числе и криптобиблиотек, и не содержат криптокомпонентов в своем составе. Это дает возможность выбирать криптобиблиотеки от разных производителей по собственному усмотрению и позволяет разработчику избежать необходимости сертификации SKIP-продуктов в ФАПСИ. К числу разработчиков криптомодулей для SKIP-продуктов относятся и несколько отечественных компаний, в частности «Анкад», обладающая необходимыми лицензиями и сертификатами на свою продукцию и предоставляющая реализацию алгоритма шифрования ГОСТ 28147-89, а также фирма «ЛанКрипто», разработавшая модуль преобразования данных, реализующий сразу несколько алгоритмов разной степени стойкости.

Система «Застава» имеет сертификаты Гостехкомиссии и состоит из следующих компонентов:

1. «Застава-Персональный клиент»
   Продукт является средством защиты рабочей станции (шифрование и аутентификация IP-трафика), находящейся в персональной эксплуатации. Эта программа содержит все средства администрирования и конфигурирования, необходимые для ее взаимодействия с любыми другими SKIP-совместимыми средствами защиты (в частности, средствами сертификации).
2. «Застава-Корпоративный клиент»
   Продукт является средством защиты рабочей станции корпоративной сети. От предыдущего продукта эта программа отличается тем, что пользователь защищаемой рабочей станции лишен возможности определять политику безопасности (следовательно, структуру сетевых соединений) для своей станции. Политика безопасности полностью контролируется администратором безопасности корпоративной сети и выдается пользователю как целостная структура данных на некотором носителе.
3. «Застава-Сервер»
   Продукт является функциональным аналогом продуктов семейства «Застава-Клиент» для серверных платформ. Он отличается прежде всего расширенными ресурсами для поддержания множественных соединений с клиентскими программными агентами, а также способностью работать с узлами, не имеющими постоянного IP-адреса.
4. «Застава-Офис»
   Это программный комплекс, предназначенный для коллективной защиты входящего и исходящего трафика сегмента локальной сети; защиты этого сегмента от несанкционированного доступа из внешних сетей; обеспечения путем туннелирования трафика защищенного взаимодействия с другими сегментами локальных сетей, защищенными при помощи аналогичных систем, и с отдельными рабочими станциями, защищенными при помощи программных систем «Застава-Клиент», а также с серверами, защищенными «Заставой-Сервером». Работает под управлением ОС Solaris Intel/Sparc. Продукт может управляться как из командной строки, так и удаленно с использованием графической оболочки, написанной на Java. ПО SKIP имеет сертификат ГТК № 46 для Windows 3.11/95 и № 48 для Solaris 2.4 («Застава-Офис» и «Застава-Клиент»).

Межсетевой экран «Застава»

Продукт представляет собой *межсетевой экран* (МЭ) уровня TCP/UDP со всеми стандартными возможностями экранов подобного класса.

МЭ обеспечивает дистанционное управление системными компонентами, в том числе и конфигурирование фильтров, проверку взаимной согласованности всех фильтров, анализ регистрационной информации.

Межсетевой экран включает средства контроля за целостностью своей программной и информационной части по контрольным суммам.

В МЭ предусмотрена процедура восстановления после сбоев и отказов оборудования, которая обеспечивает восстановление свойств МЭ. Также имеется возможность управления по SNMP.

«Застава» имеет сертификат Гостехкомиссии об удовлетворении требованиям к третьему классу защищенности МЭ в соответствии с РД «Средства вычислительной техники. Межсетевые экраны. Защита от несанкционированного доступа к информации. Показатели защищенности». Сертификат № 145 от 9 января 1998 года.

В настоящее время разработчик не занимается усовершенствованием этого продукта, предлагая в составе комплексной защиты информации межсетевой экран Firewall-1 от CheckPoint.

Продукт Certificate Discovery Server

Продукт обеспечивает распределение сертификатов открытых ключей под управлением стандартного и совместимого по ряду SKIP-продуктов различных производителей протокола CDP (Certificate Discovery Protocol). Типовая схема построения защищенной сети на основе продуктов «Застава» изображена на рис. 3.13.

ФПСУ-IP

Продукт ФПСУ-IP (АМИКОН) представляет собой межсетевой фильтр уровня TCP/UDP с возможностями организации криптографически защищенных по ГОСТу на уровне межсетевых соединений IP и сжатия сетевого трафика. Основные криптографические возможности комплекса реализованы с участием фирмы «ИнфоКрипт». Комплекс выполнен под собственной защищенной операционной средой на базе ОС DOS 5.0 и в качестве сетевых протоколов канального уровня поддерживает только Ethernet. Система рассчитана на наличие двух сетевых карт 10/100 Мбит/с. ФПСУ-IP реализован по принципу ложного ARP-сервера, который устанавливается в физический разрыв, разделяющий два сегмента сети, перехватывает ARP-запросы сетевых устройств, в ответ предоставляя свой MAC-адрес. Приходящие на него IP-пакеты подвергаются фильтрации и, если задано, дополнительно обрабатываются для организации VPN (компрессия данных и/или криптографическая обработка с применением

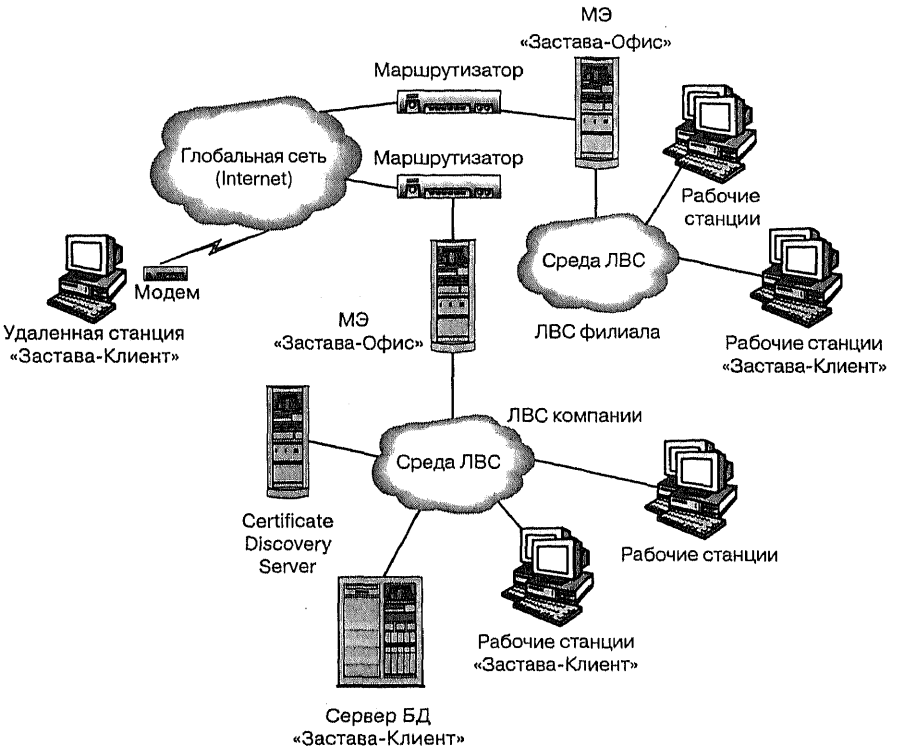

Рис. 3.13. Типовая схема использования «Заставы»

имитовставок), затем транслируются на другой интерфейс или сбрасываются в соответствии с результатами обработки.

Данный подход к реализации продукта представляется весьма разумным, так как в данной ситуации внутренний интерфейс межсетевого экрана может не иметь IP-адреса и таким образом быть неуязвимым к атакам уровня IP изнутри сети. Кроме того, такой МЭ полностью прозрачен для клиентов сети, пользователи могут даже не знать о его наличии. Выбор DOS в качестве операционной системы межсетевого экрана делает ее более безопасной и производительной по сравнению с Windows NT и UNIX из-за однозадачности DOS и отсутствия в ней встроенных сетевых возможностей. Однако такой выбор налагает на разработчика дополнительные обязанности по реализации удаленного доступа администратора и организации мониторинга и управления системой защиты.

Защита от НСД к операционной среде и рабочей информации осуществляется с помощью платы «Аккорд», функционирующей по принципу

расширения BIOS. Эта система обеспечивает разграничение доступа обслуживающего персонала по уровням доступа к системе и защиту программного обеспечения на жестком диске на ключе, находящемся на идентификаторе touch memory, что делает бессмысленным несанкционированное изъятие платы защиты от НСД и кражу межсетевого экрана (или жесткого диска) для получения доступа к рабочим материалам.

ФПСУ-IP имеет сертификат Гостехкомиссии, удовлетворяющий требованиям к третьему классу защищенности МЭ в соответствии с РД «Средства вычислительной техники. Межсетевые экраны. Защита от несанкционированного доступа к информации. Показатели защищенности». Сертификат № 233 от 29 апреля 1999 года.

ФПСУ-IP обладает следующими функциональными возможностями:

- фильтрация сетевых пакетов уровня IP и TCP в соответствии с задаваемыми администраторами правилами на основе IP-адресов отправителя и получателя, фреймов, инкапсулированных в IP-протоколы, времени и даты передачи пакета, разрешенных портов абонентов (для TCP/UDP-пакетов), а также пар адресов абонентов, для которых разрешено соединение;
- трансляция сетевых адресов отправителя и получателя в межсетевых туннелях, скрывающая внутренние адреса субъекта и объекта информационного взаимодействия (NAT);
- осуществление при туннелировании двусторонней аутентификации;
- применение «проходного» сжатия данных;
- строгая защита передаваемых данных на уровне IP, в том числе от навязывания ранее переданных;
- сокрытие факта использования защитных свойств комплекса. При нарушении правил фильтрации и сбросе пакета на межсетевом экране клиент получает ICMP-сообщение о том, что маршрут не найден.
- регистрация в защищенном хранилище записей статистической информации о функционировании комплекса (в том числе суточный биллинг);
- визуальное отображение на экране попыток нарушения правил фильтрации;
- обеспечение защиты от несанкционированного доступа к информации и ресурсам комплекса посредством идентификации администратора по идентификатору и содержимому памяти электронной таблетки touch memory, а также по паролю условно постоянного действия;
- разделение прав на доступ к работе комплекса для различных классов администраторов с обеспечением учета их активных действий.

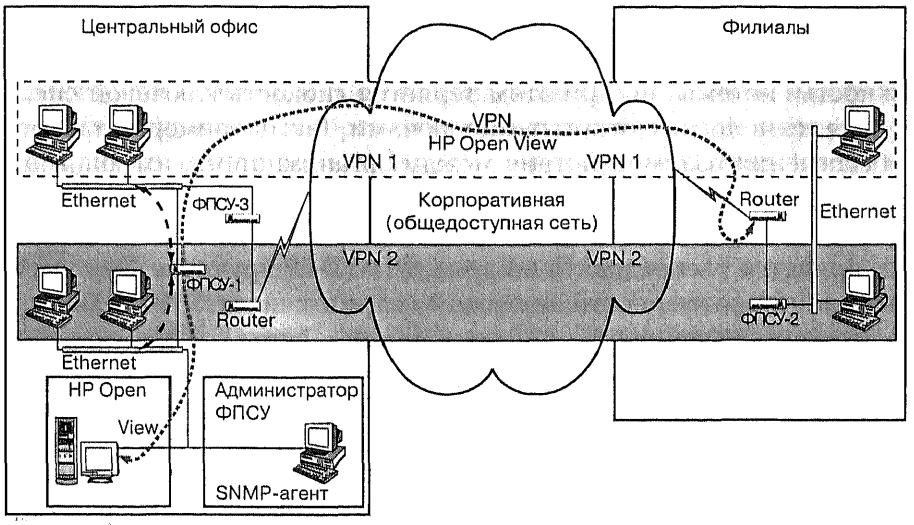

Рис. 3.14. Типовая схема использования ФПСУ-IP

Типовая схема использования ФПСУ-IP приведена на рис. 3.14.

Алгоритмы защиты информации базируются на использовании ГОСТа, но протоколы идентификации, аутентификации, нелинейного преобразования данных и их компрессии, а также обмена аутентификационными данными (ключами) являются собственными разработками АМИКОНа, выполненными при участии фирмы «ИнфоКрипт», и не соответствуют стандарту IPSec, что, конечно, можно оценивать неоднозначно. В фирменной реализации АМИКОНа не предусмотрена передача пакетного ключа вместе с каждым IP-пакетом, а защита трафика осуществляется на сеансовом ключе, который периодически меняется. При этом объем служебной информации в IP-пакете составляет 18–20 байт. Возможность эффективного сжатия информации также снижает количество передаваемых данных.

Ключевая система ФПСУ-IP является симметричной, то есть на каждом ФПСУ-IP имеется таблица данных парно-выборочной связи, первоначально загружаемая вручную локально. Одновременно в комплексе ФПСУ-IP может находиться по четыре комплекта собственных ключей и все необходимые данные для осуществления парно-выборочной связи для групп (максимально 32) генерации (в каждой группе генерации может быть до 9999 комплектов). Как только производится изменение номера комплекта на одном из ФПСУ-IP системы, все взаимодействующие

ФПСУ-IP автоматически осуществляют переход на новые комплекты. Использование технологии парно-выборочной связи, конечно, повышает надежность системы, но при этом теряется гибкость ключевой системы, присущая технологии с открытыми ключами. Так, например, остается вопрос обеспечения взаимодействия между организациями, имеющими разные подразделения генерации ключей и соответственно разные ключевые таблицы. По окончании использования всех комплектов ключей новые ключи придется распределять по всем ФПСУ-IP вручную. При увеличении за границы размерности ключевой матрицы числа ФПСУ-IP в системе необходимо генерировать новую матрицу и опять-таки вручную переустанавливать ее на все ФПСУ-IP.

Программный комплекс «Игла-П»

ПК «Игла-П» (МО ПНИЭИ) предназначен для организации защищенных VPN на базе общедоступных сетей передачи данных (создан на основе реализации SKIP-протокола). Применяя ПК «Игла-П», любой абонент защищенной VPN может обмениваться данными с любым другим абонентом VPN, причем шифрование передаваемых данных для абонентов является прозрачным. В ходе организации защиты информации обеспечивается не только шифрование IP-пакетов (инкапсулирование IP-пакета), но и аутентификация сторон информационного обмена.

Кроме того, «Игла-П» обеспечивает фильтрацию входящего трафика, что характерно для средств защиты данного типа. Это функциональное свойство аналогично принципу, лежащему в основе межсетевых экранов, но очевидные ограничения, которые присутствуют при построении средств защиты ПК «Игла-П», не позволяют полномасштабно развернуть систему фильтрации трафика, поэтому задаваемые в нем правила фильтрации носят слишком общий характер. Тем не менее такая функция необходима для надежной эксплуатации рабочей станции, особенно для предотвращения негативных последствий при поступлении некорректной информации от каналов связи или при передаче сообщений абоненту локальной сети.

ПК «Игла-П» является программным средством защиты информации и устанавливается непосредственно на рабочую станцию пользователя. Выполнение перечисленных выше функций обеспечивается за счет работы с трафиком до самого сетевого уровня. Например, ПК «Игла-П» встраивается в сетевую архитектуру Windows NT 4.0 на уровне драйвера сетевой оболочки ndis.sys и перехватывает обмен сетевой информацией между сетевыми протоколами и адаптерами. ПК «Игла-П» производит регистрацию установленных в системе компонентов сетевого ПО и препятствует

выполнению незарегистрированных компонентов (драйверов, протоколов и сетевых адаптеров).

Характерной особенностью подобного средства защиты является то, что с его помощью можно организовывать защищенный информационный обмен как между пользователем и защищаемой локальной сетью, так и между двумя пользователями (рис. 3.15). Организация взаимодействия пользователь-защищаемая локальная сеть обеспечивается за счет совместимости между ПАК шифрования IP-трафика, стоящего на входе в локальную сеть, и пользовательским средством защиты. Пакеты, отправляемые ПК «Игла-П» для локальной сети, вначале попадают в ПАК шифрования IP-трафика, где обрабатываются (фильтруются и расшифровываются), а затем маршрутизируются внутри локальной сети. Например, ПК «Игла-П» совместим с КК «Шип» за счет реализации в обоих SKIP-протоколов.

При этом, например, ПК «Игла-П» может осуществлять защищенный обмен информацией как с использованием сетевых плат, так и с использованием модемных соединений, что обеспечивает гибкость при обеспечении защиты.

Поскольку современные ОС являются многопользовательскими, то при реализации программных средств защиты, ориентированных на применение в подобных ОС, необходимо обеспечить индивидуальные настройки для каждого пользователя, а также перекрыть возможности одного пользователя оказывать негативное влияние на режим работы другого. ПК «Игла-П» позволяет разграничить сферы доступа к настройке параметров конфигураций для различных категорий пользователей.

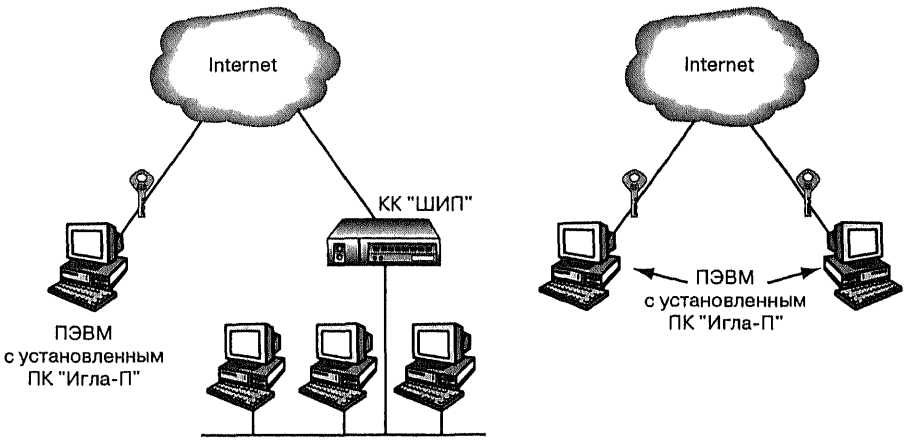

Рис. 3.15. Схемы применения ПК «Игла-П»

Перед началом работы с «Игла-П» каждый пользователь Windows NT проходит регистрацию, которую производит администратор безопасности программного комплекса. При этом пользователю присваивается одно из следующих прав доступа к настройкам конфигурации:

- право просмотра и внесения изменений;
- право просмотра;
- запрет доступа.

3.5. Защита технологии «клиент-сервер»

Одной из самых распространенных на сегодняшний день моделей построения распределенных вычислительных систем является технология «клиент-сервер». Не вдаваясь в детали, опишем общую идею построения подобного вида технологий, в основе которой – разделение субъектов информационного взаимодействия на серверы, предоставляющие информационные услуги, и на клиентов, потребляющих эти услуги. Среди услуг, предоставляемых серверами, можно выделить: электронный почтамт, база данных, распределенная обработка данных, файл-сервер, сервер печати и т.д. Общая структура построения информационно-вычислительных систем (ИВС), использующих технологию «клиент-сервер», представлена на рис. 3.16.

Программное обеспечение, построенное по этой технологии, работает на прикладном уровне ISO/OSI, поэтому ИВС, сконструированные с использованием технологии «клиент-сервер», независимы от архитектуры транспортной среды передачи данных.

Поскольку в основе любых типов построения распределенных систем лежит взаимодействие удаленных друг от друга субъектов информационного обмена, очевидно, что протокольная часть программного обеспечения является краеугольным камнем любой клиент-серверной технологии. Собственно эта часть и реализует заданный подход к построению ИВС. В рамках данных ИВС могут работать пользователи с разными уровнями привилегий и обрабатываться информация, имеющая разный уровень секретности.

Наряду с функциями передачи сведений подобные системы выполняют задачи по разграничению доступа клиентов к информационным ресурсам сервера, а также по аудиту работы ИВС. К тому же в рамках подобных систем реализуется сложная система администрирования и поддержания работоспособности.

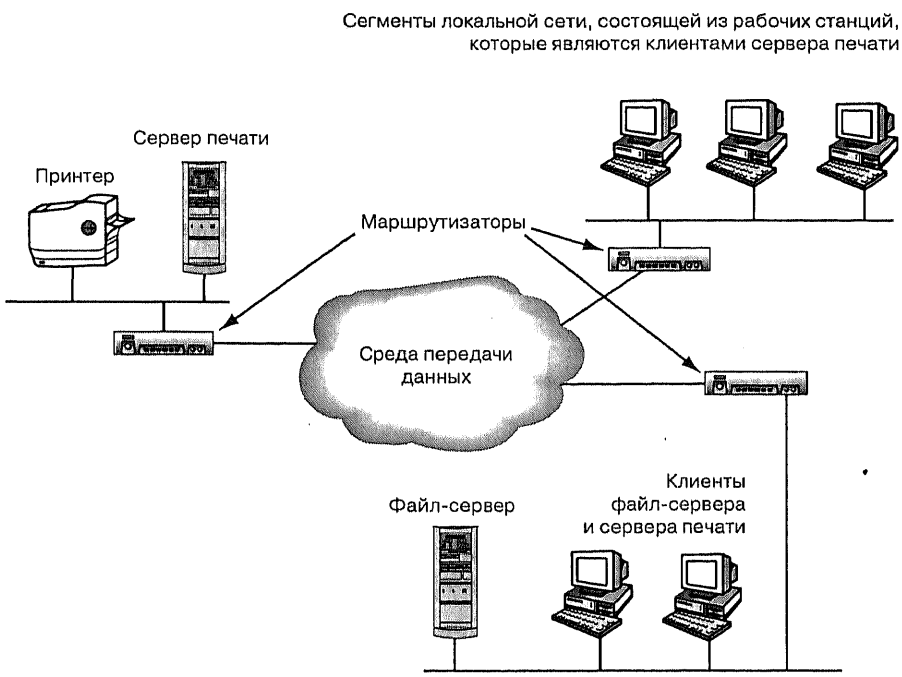

Рис. 3.16. Общая структура ИВС

Из всех протоколов, применяемых в клиент-серверных технологиях, чаще других пользуются протоколами прикладного уровня. К ним относятся:

- протокол HTTP (HyperText Transfer Protocol), являющийся основой Web-технологии (или WWW) и использующийся для доставки гипертекстовых сообщений. В рамках Web-технологии применяются еще три механизма, без которых ее существование было бы немыслимо: язык гипертекстовой разметки документов (HTML), универсальный способ адресации (URL) и универсальный межсетевой интерфейс CGI;
- Telnet-протокол – протокол удаленного доступа. Позволяет клиенту подключиться к любому серверу и работать с ним так, как если бы он был удаленным терминалом этого сервера;
- протокол FTP (File Transfer Protocol), позволяющий клиенту осуществлять удаленный доступ к файлам, расположенным на FTP-сервере;
- Gopher-протокол, предназначенный для поиска файлов на серверах Gopher и копирования найденных файлов на рабочую станцию клиента.

В качестве современного образца программного обеспечения, построенного по принципу клиент-серверных технологий, можно привести Internet Information Server (IIS) фирмы Microsoft, реализующий серверную часть протоколов HTTP, FTP и Gopher и использующий в качестве интерфейса взаимодействия со стеком протокол TCP/IP Windows Sockets. Клиентской частью перечисленных выше протоколов может служить, например, Internet Explorer и Netscape Navigator. Пример архитектуры взаимодействия клиентской части протоколов с серверной частью, реализованной в IIS, представлен на рис. 3.17.

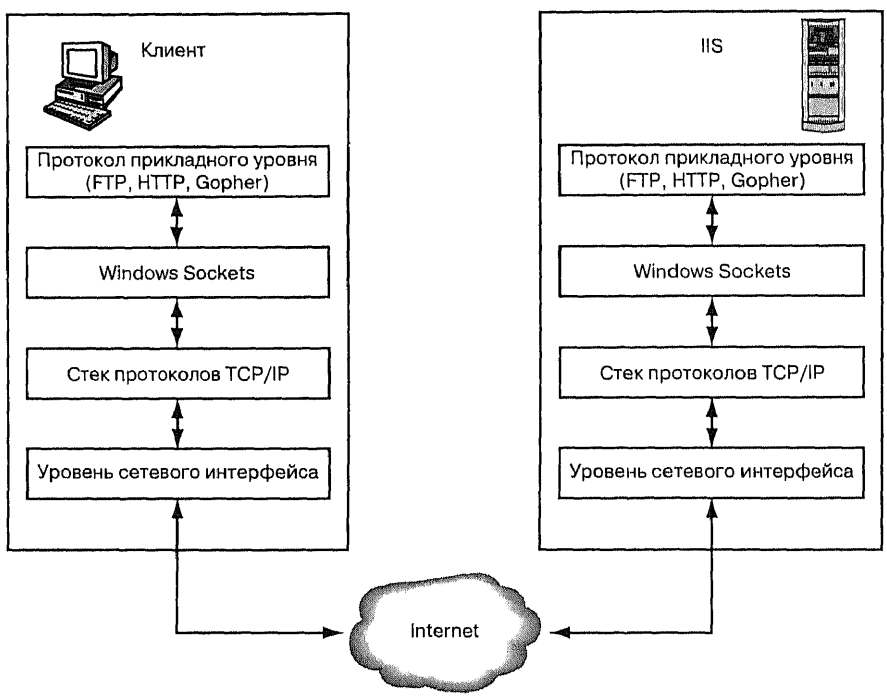

Рис. 3.17. Архитектура типового взаимодействия в рамках Internet

3.5.1. Типовые угрозы и обеспечение информационной безопасности при использовании технологии «клиент-сервер»

Учитывая, что Web-технологией пользуются практически повсеместно, важно напомнить, что именно безопасность клиент-серверных технологий играет основополагающую роль в современном развитии Internet. Особую актуальность этой проблеме придает тот факт, что в последнее время резко

увеличилось количество компаний, применяющих Internet в коммерческих целях; оборот денежных средств через Internet тоже имеет тенденцию к стремительному росту. Поскольку подавляющее большинство продуктов прикладного уровня, представленных в Internet, построены по технологии «клиент-сервер», понятно, что защите информации в подобных технологиях следует уделить особое внимание.

Вот почему обзор конкретных решений по защите технологии «клиент-сервер» представлен в виде анализа средств защиты информации Web-технологий.

В общем случае факты нарушения безопасности существующих ИВС могут быть следствием:

- отсутствия или недостаточности применения необходимых средств защиты;
- недостатков в системе администрирования;
- уязвимости в существующих средствах защиты;
- ошибок в программном обеспечении;
- ошибок и преднамеренных действий обслуживающего персонала.

При использовании архитектуры «клиент-сервер» реализация политики безопасности включает:

- защиту передаваемого между серверной и клиентской стороной трафика (под защитой в данном случае подразумевается обеспечение конфиденциальности, целостности, достоверности и имитозащищенности передаваемой информации);
- разграничение доступа клиентов к информационным ресурсам, предоставляемым сервером. Например, доступ определенной группы пользователей к определенным ресурсам Web-сервера должен быть ограничен;
- обеспечение работоспособности ИВС. Информационные ресурсы сервера должны быть доступны клиентам за приемлемое для них время, например, это требование носит особую актуальность в случае предоставления через Internet результатов торгов на бирже, когда недоступность информации в течение часа уже может носить убыточный характер;

Учитывая специфику построения клиент-серверных ИВС, можно выделить следующие типовые угрозы и уязвимости информационной безопасности (список конкретных угроз и уязвимостей, приведенный ниже, не претендует на полноту, а служит исключительно для иллюстрации примеров возможных нарушений политики безопасности):

- угрозы и уязвимости, связанные с недостатками в системе разграничения доступа клиентов к ресурсам или с некорректным администрированием системы разграничения доступа. К ним относятся:

- несанкционированный доступ к пользовательским данным и нарушение работы ИВС, возникающие из-за отсутствия механизмов обеспечения конфиденциальности и целостности передаваемых данных и бесперебойности обмена данными. Это часто приводит к успешным атакам на узлы сети, а также к эффективному анализу трафика;
- отсутствие механизмов контроля за применением пользовательских учетных записей как на серверной части, так и на клиентской может привести к незаконному использованию учетных записей;
- доступ к информационному ресурсу с применением не предназначенных для этого средств, что может привести к несанкционированному доступу к защищаемой информации. Во избежание подобных угроз необходимо контролировать использование специально предназначенных для этой цели средств;
- нарушитель может исключить легального клиента из установленного сеанса доступа к ресурсу и затем воспользоваться его именем;
- возможность пользователям (не входящим в группу администраторов) установить некорректные атрибуты безопасности у защищаемого ресурса, что приведет к несанкционированному доступу к данному ресурсу;
- снижение эффективности функционирования подсистемы разграничения доступа из-за отсутствия механизмов динамического управления значениями атрибутов безопасности;

• угрозы и уязвимости, возникающие вследствие недостатков системы аудита работы подсистемы безопасности клиент-серверной ИВС, а именно:

- невозможность возложения ответственности на нарушителя политики безопасности, возникающая вследствие того, что система аудита оставляет незарегистрированными некоторые события, связанные с безопасностью;
- в случае недостатка памяти, отведенной под журнал событий, может возникнуть ситуация, когда данные подсистемы аудита будут утеряны до того, как начнут использоваться;
- нарушитель может получить несанкционированный доступ к журналу системы аудита и повредить его целостность;

• угрозы и уязвимости, возникающие вследствие недостатков в подсистеме идентификации/аутентификации:

- получение нарушителем доступа к информации, применяющейся в ходе аутентификации пользователей, например ключевым носителям и т.д.;
- компрометация параметров аутентификации, которая возникает в силу отсутствия механизмов, запрещающих проведение аутентификации в случае нескольких неудачных попыток;
- выбор некорректных параметров аутентификации, который приведет к ухудшению стойкости схемы аутентификации, возникающий в силу отсутствия механизмов контроля корректности устанавливаемых значений;
- компрометация параметров аутентификации, повышения уязвимости подсистемы идентификации/аутентификации, возникающая вследствие использования неуникальных идентификаторов пользователей;

• угрозы и уязвимости, возникающие из-за непроработанности организационных аспектов реализации политики безопасности:

- разрешение клиентам устанавливать большое или неконтролируемое количество сеансов связи с сервером может привести к потере работоспособности сервера;
- нарушение безопасности ИВС в силу того, что нарушитель может начать работу в ИВС, когда отсутствует требуемый контроль безопасности;
- нарушение информационной безопасности ИВС вследствие некорректных действий пользователя, причиной которых явилась неосведомленность о запрещенных в рамках ИВС действиях;

• угрозы и уязвимости, возникающие вследствие отсутствия или недостатков системы контроля функционирования подсистемы защиты информации ИВС:

- в случае отсутствия контроля целостности программной части средств защиты информации и ядра операционной системы возможно изменение алгоритма функционирования данного программного обеспечения, что может послужить причиной нарушения системы безопасности;
- сбой в работе ИВС по внешним или внутренним причинам может вызвать нарушение работы средств защиты и при неспособности средств контроля работоспособности средств защиты обнаружить сбой и после этого продолжить функционирование;
- отсутствие действующих автоматически или при участии администраторов механизмов восстановления безопасного состояния средств защиты, а также механизмов проверки целостности данных средств

защиты может привести к переходу в небезопасное состояние при возникновении сбоев;

- угрозы и уязвимости, возникающие вследствие непроработанности системы информационной безопасности в целом:
  - отсутствие механизмов обеспечения конфиденциальности и целостности передаваемых данных средств защиты и бесперебойности обмена данными. При успехе удаленных атак на узлы локальной сети или проведении анализа трафика сети это может привести к несанкционированному доступу к данным средствам защиты и нарушениям в их работе;
  - при отсутствии механизмов обнаружения физических воздействий и противодействия техническим средствам, производящим удаленное воздействие на систему, повышается вероятность осуществления атак на физическом уровне;
  - если существует возможность использования не контролируемых средствами защиты механизмов удаленного доступа или межмашинного взаимодействия, возникает угроза нарушения политики безопасности путем обхода средств защиты;
  - недостатки в организации разделения доменов безопасности могут привести к несанкционированному доступу к адресным пространствам субъектов, к коду или данным средств защиты, а также к посторонним воздействиям на работу средств защиты;
  - применение разных политик управления доступом в условиях существования в ИВС двух классов информации – конфиденциальной и общедоступной – оставляет возможность для передачи информации между объектами, относящимися к разным классам;
  - отсутствие средств защиты, контролирующих импорт и экспорт данных во внешние системы, может привести к экспорту несоответствующей информации из ИВС или несанкционированному доступу к полученным документам до задания их атрибутов безопасности;
- угрозы и уязвимости, возникающие вследствие отсутствия специализированных механизмов, поддерживающих функционирование средств защиты информации, или при наличии недостатков в них:
  - состояния различных средств защиты ИВС, правильное функционирование которых зависит от точности отсчета времени, могут противоречить друг другу из-за неспособности верно определять значения времени;
  - согласованность функционирования средств защиты на различных компонентах ИВС может быть нарушена из-за недостатков

протоколов удаленной идентификации/аутентификации пользователей на различных компонентах ИВС.

Говоря о безопасности клиент-серверных технологий, встречающихся в Internet, а именно о базовых версиях Web-приложений, необходимо выделить следующие уязвимости ИВС:

- передача трафика в открытом виде, что позволяет производить не только анализ передаваемой информации, но и ее модификацию;
- использование парольной аутентификации, причем пароли в данном случае могут передаваться в открытом виде;
- отсутствие защиты служебной информации позволяет применять атаки типа Web spoofing (подмена имени ресурса), основанные на замене нарушителем URL, содержащихся в передаваемых HTTP-запросах и ответах;
- возможность запуска CGI-скриптов и Java-апплетов, способных негативно влиять на безопасность ИВС в целом.

Как видно из приведенного выше списка, современные ИВС, построенные по принципу «клиент-сервер», обладают рядом уязвимостей, которые могут негативно сказаться на безопасности ИВС. Очевидно, что наряду с уже существующими в современных клиент-серверных приложениях механизмами защиты информации необходимо использовать дополнительные средства безопасности.

Примеры атак в Web-технологиях

Здесь рассматривается достаточно простая атака на рабочую станцию с установленными ОС Windows NT 4.0 и браузером (browser) типа Internet Explorer. Идея нападения заключается в том, что нарушитель создает HTML-страницу, содержащую ссылку вида: file://\server\share\image.gif. Она относится к ресурсам в формате CIFS, и очевидно, что данный ресурс уже находится на рабочей станции нарушителя. Пользователь, желающий просмотреть данный ресурс, должен зарегистрироваться на предложенном ему сервере (обычно типа Lanman). При этом регистрация пользователя производится ОС автоматически, то есть его имя и хэшированный пароль пересылаются на сервер. После чего злоумышленник, проводя подбор пароля с применением атаки по словарю, может получить пароль пользователя или в дальнейшем использовать его хэшированный вариант.

Еще один пример атаки в Web-технологиях – Web spoofing. Для осуществления этой операции злоумышленник должен сначала привлечь внимание пользователя к ложному Web-узлу. Это может быть сделано

несколькими способами: путем проникновения на существующий узел и подмены локаторов URL, путем внесения имитируемого узла в список механизма поиска или даже с помощью электронной почты, по которой можно оповестить пользователей о существовании адреса, который может их заинтересовать. Имитируемый узел затем помещает свой собственный адрес перед любым указателем ресурсов URL, запрашиваемым пользователем, так что адрес http://www.anyurl.com превращается в http://www.spoofserver.com/http://www.anyurl.com. Правильная Web-страница затем отсылается назад пользователю через сервер, где эту информацию можно изменить, а любую информацию, которую передает пользователь, – перехватить. Процесс может быть продлен, в результате чего все другие оперативные связи будут иметь пристыкованный впереди адрес; как следствие, все запросы других локаторов URL будут нарушаться. Те два признака, которые должны насторожить пользователя и подсказать, что его соединения осуществляются через другой сервер, – статусная строка в низу экрана и адрес назначения наверху – могут быть изменены путем использования Java-апплетов.

3.5.2. Подходы, применяемые к обеспечению информационной безопасности в клиент-серверных ИВС

На сегодняшний день большинство Web-серверов обеспечивают защиту информационных ресурсов с использованием идентификации пользователей на основе ввода ими своих идентификаторов и паролей. Эта информация часто передается на сервер в открытом виде по общедоступным каналам передачи данных. Сервер проверяет идентификатор и пароль пользователя, находящиеся в запросе, на соответствие уровню доступности информации, содержащейся в списке контроля доступа (ACL) для данного ресурса. ACL содержит информацию, указывающую, какой пользователь имеет доступ к данному ресурсу и какой при этом используется вид доступа, например: в качестве ресурса может выступать URL. Данный подход содержит ряд уязвимостей и недостатков, а именно:

- информация, применяемая системой разграничения доступа, передается по сети в открытом виде, что позволяет нарушителю, имеющему доступ к каналу передачи данных, перехватывать ее и затем успешно пользоваться этими данными от имени легального пользователя;
- пользователям доставляет определенное неудобство помнить различные идентификаторы и пароли при доступе к разным Web-серверам;
- данный подход затруднителен для мониторинга, поскольку пользователь может применять разные идентификаторы в сети;

- администрация привилегий пользователей в глобальных сетях может потребовать больших временных затрат со стороны администратора.

Все потенциальные угрозы требуют привлечения дополнительных средств защиты информации в Web-технологиях. Подходы к обеспечению безопасности в ИВС, построенных по технологии «клиент-сервер», можно условно разделить на следующие категории:

- создание средств защиты прикладного уровня, направленных на установление защищенного канала передачи данных между клиентом и сервером. В качестве применяемых решений можно отметить использование криптографических протоколов SSL и PCT и криптографических ядер типа «Верба-О» и «Верба» (о них уже говорилось ранее). В этом разделе в качестве конечного продукта, предназначенного для установления защищенного канала передачи данных между Web сервером и Web клиентом, рассматривается СКЗИ «Форт»;
- средства защиты, реализующие наряду с установлением защищенного канала передачи данных прикладного уровня разграничение доступа клиентов к защищаемым ресурсам сервера. Оптимальным в данном случае является использование криптографического протокола Kerberos, которому в этом разделе уделено особое внимание. В качестве конечного продукта защиты информации рассматривается система TrustedWeb;
- применение средств защиты нижележащих уровней. Например, можно использовать туннелирование трафика при помощи средств защиты информации, являющихся реализацией SKIP-протокола. Такое решение позволяет эффективно устанавливать защищенный канал передачи данных, однако реализовать гибкую систему разграничения доступа к защищаемым ресурсам с использованием средств защиты информации подобного типа не представляется возможным.

Эффективность применения перечисленных средств зависит от того, насколько их реализация удовлетворяет требованиям масштабируемости и интегрируемости с существующей инфраструктурой.

В качестве дополнительных механизмов, которые должны быть эффективно реализованы (здесь под эффективностью будет пониматься следующее: данные механизмы не должны накладывать существенных ограничений на оперативность, полнофункциональность и эффективность администрирования архитектуры «клиент-сервер»), в рамках системы безопасности следует отметить:

- ключевую инфраструктуру;
- систему аудита;
- систему администрирования средств защиты.

Если говорить о конкретных видах средств, то при использовании средств защиты первого типа наиболее проблематичной представляется организация ключевой инфраструктуры. В случае применения симметричной ключевой системы управление ключами решается организационными методами, то есть создаются специальные центры безопасности, которые и оформляют распределение симметричных ключей, записанных на ключевые носители – дискеты, смарт-карты и touch memory. Очевидно, что данная ключевая структура эффективна только в небольших организациях. Широкое применение криптографических средств защиты информации с симметричными ключами возможно только в случае дополнительного использования асимметричного распределения ключей. Поэтому проблема организации ключевой системы сводится к организации разветвленной структуры центров сертификации открытых ключей.

В случае применения средств защиты второго типа в добавление к ключевой системе следует организовать эффективные системы аудита и администрирования. Система администрирования необходима для того, что порой приходится организовывать доступ нескольких групп пользователей к нескольким защищаемым серверам. Классический подход в этих обстоятельствах требует проведения процедур администрирования для каждого сервера в отдельности. Эффективным решением здесь является объединение процедур администрирования всех защищаемых серверов в единую систему, то есть администрирование системы безопасности, объединяющей несколько серверов, должно по возможности исходить из единого центра безопасности системы.

3.5.3. Криптографические протоколы, используемые для защиты технологии «клиент-сервер»

В этом разделе описывается архитектура построения, принципы действия, вопросы интеграции и практической стойкости таких известных криптографических протоколов, как Kerberos и SSL. Завершает раздел сравнительный анализ практического применения этих протоколов для обеспечения защиты информации в технологии «клиент-сервер».

Протокол Kerberos v5

Протокол Kerberos используется для аутентификации и обмена ключами, предназначенными для установки защищенного канала связи между абонентами, работающими как в локальной сети, так и глобальных сетях. Kerberos разработан для сетей TCP/IP и построен на основе доверия участников

протокола к третьей стороне, роль которой выполняет серверная часть Kerberos. Создание Kerberos велось в рамках проекта «Athena» в Массачусетском технологическом университете в середине 80-х годов, и с тех пор этот протокол претерпел ряд принципиальных изменений, которые отразились в существующих ныне пяти версиях.

Основу Kerberos составляет протокол Нидхэма-Шредера с третьей доверенной стороной. Служба Kerberos, находящаяся в сети, действует как арбитр, которому доверяют все участники. Kerberos обеспечивает безопасную сетевую аутентификацию пользователей (ресурсов) сети с последующей авторизацией доступа клиента (клиентского приложения) к ресурсам сети. Защищенность установленных в рамках сессии Kerberos соединений обусловливается применением симметричных алгоритмов шифрования (DES и ряда других алгоритмов, которые могут быть взаимозаменяемы). Основу аутентификации составляет знание абонентом своего секретного пароля (что является результатом хэширования секретного ключа пользователя).

В модели протокола Kerberos существуют два основных элемента – клиенты и серверы. Клиентами могут быть пользователи, а также независимое программное обеспечение, которое нуждается в авторизованных услугах по загрузке удаленных файлов, отправке сообщений или доступу к принтерам либо в получении каких-либо привилегий у администратора. Kerberos хранит базу данных о клиентах и их секретных ключах. Наличие в базе данных секретных ключей каждого пользователя и ресурсов сети, поддерживающих данный протокол, позволяет создавать зашифрованные сообщения, направляемые клиенту или серверу; их успешное расшифрование и является гарантией прохождения аутентификации всеми участниками протокола.

Секретные ключи, хранящиеся в базе данных, используются только с целью аутентификации и выработки *сеансового ключа* (session key) для зашифрования сообщений, передающихся в ходе сессии.

Описание работы протокола

Клиент, желающий пройти аутентификацию и выработать сеансовый ключ для установки защищенного соединения с ресурсом сети, посылает запрос на получение билета предоставления билета (ticket-granting ticket) для службы предоставления билета (Ticket-Granting Service, TGS). Запрошенный билет пересылается пользователю в зашифрованном на секретном ключе клиента виде. Для получения доступа к конкретному серверу в сети клиент посылает полученный билет предоставления билета к серверу TGS. Далее сервер TGS после успешного прохождения билетом предоставления билета необходимых проверок высылает клиенту билет на доступ

к интересующей его службе. Завершающая фаза состоит в пересылке клиентом билета вместе с аутентификатором (authenticator) целевому серверу и после успешного прохождения проверок клиент может получить от сервера доступ к службам, которые данный сервер предоставляет.

Протокол Kerberos поддерживает два типа вверительных грамот – билеты и аутентификаторы. Билет используется для обеспечения безопасности при доступе клиента к целевому серверу. Существует большое количество типов билета: начальные, обновляемые, предварительные и недействительные. Принадлежность билета к тому или другому типу определяется параметрами, установленными в нем. В общем виде билет можно представить так:

Tc,s = s, {c,a,v,Kc,s}Ks (см. табл. 3.8)

Таблица 3.8. Перечень сокращений

c	Идентификатор (имя) клиента
s	Идентификатор (имя) сервера
a	Сетевой адрес клиента
v	Начальное и конечное время действия билета
t	Метка времени
Kx	Секретный ключ, принадлежащий x
Kx,y	Сеансовые ключи x и y
{m}Kx	Сообщение m, зашифрованное на ключе Kx
Tx,y	Билет, принадлежащий x, для предоставления y
Ax,y	Аутентификатор, принадлежащий x, для предоставления y

Билет предоставляется для доступа к строго определенному серверу и на строго определенное время, причем время начала действия билета может быть указано в будущем (предварительные билеты). Он содержит имя клиента, его сетевой адрес, начальное и конечное время действия клиента и сеансовый ключ, зашифрованные на секретном ключе сервера. Однажды получив билет, клиент может использовать его для доступа к серверу в течение промежутка времени, отведенного для данного билета. Клиент не имеет возможности расшифровать билет в силу того, что он зашифрован на секретном ключе сервера, к которому необходимо получить доступ.

Аутентификатор, используемый в Kerberos, имеет следующую форму:

Ac,s = {c,t,key}Kc,s

Аутентификатор создается клиентом всякий раз, когда тот хочет получить доступ к целевому серверу. Аутентификатор содержит имя клиента, метку времени и сеансовый ключ, зашифрованные на сеансовом ключе,

выработанном между клиентом и сервером. Основное отличие аутентификатора от билета заключается в том, что он может использоваться только один раз. Находящаяся в аутентификаторе метка времени не позволяет злоумышленнику, перехватившему данный аутентификатор и билет, использовать их спустя некоторое время для успешного прохождения процедуры аутентификации.

В рамках протокола Kerberos v5 используются следующие сообщения (рис. 3.18):

- от клиента аутентификационному серверу: c,tgs;
- от аутентификационного сервера клиенту: {Kc,tgs}Kc,{Tc,tgs}Ktgs;
- от клиента к серверу TGS: {Ac,s}Kc,tgs,{Tc,tgs}Ktgs,s;
- от сервера TGS клиенту: {Kc,s}Kc,tgs,{Tc,s}Ks;
- от клиента целевому серверу: {Ac,s}Kc,s,{Tc,s}Ks.

После приема первого сообщения аутентификационный сервер Kerberos пытается найти пользователя в своей базе данных и в случае, если поиск завершен удачно, создает сеансовый ключ (1 шаг), который будет использоваться клиентом и сервером TGS, отправляет его клиенту зашифрованным на секретном ключе клиента вместе с билетом предоставления билета (ticket-granting ticket (TGT)) (2 шаг). TGT зашифровывается на секретном ключе TGS-сервера и содержит идентификаторы клиента и сервера, сеансовый ключ TGS-клиент, а также начальное и конечное время действия TGT.

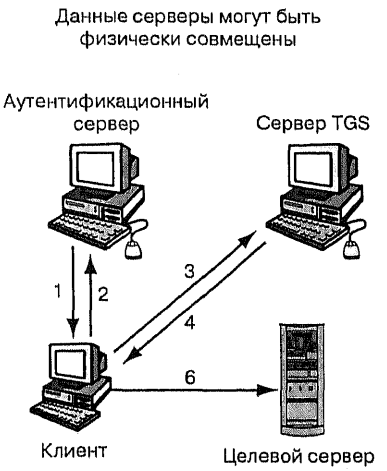

Рис. 3.18. Схема работы протокола

После чего клиент, принимая данное сообщение, расшифровывает его и получает сеансовый ключ. Клиент сохраняет сеансовый ключ и TGT в памяти компьютера (что может привести в случае использования недоверенной программной среды к их компрометации). Теперь клиент имеет возможность пройти аутентификацию у TGS-сервера при помощи полученного TGT и на протяжении времени жизни TGT, указанного в нем.

Далее клиент может получить отдельный билет для каждой службы, в которой он нуждается. С этой целью он должен послать запрос в службу TGS (на практике программное обеспечение посылает запрос автоматически, то есть невидимо для пользователя), который состоит из TGT

и аутентификатора (третий шаг). Служба TGS при приеме TGT расшифровывает его при помощи своего секретного ключа. Затем TGS из расшифрованного TGT получает сеансовый ключ и расшифровывает аутентификатор. В завершение производится сравнение информации, содержащейся в аутентификаторе, с информацией в TGT. Точнее, сетевой адрес клиента в билете сличается с сетевым адресом, указанным в запросе, а также сравнивается метка времени с текущим временем.

Проверка метки времени обусловливает одно из требований к поддержке протокола Kerberos – синхронизацию часов компьютеров в сети с точностью до нескольких минут. Если время в запросе указано далеко в будущем или уже прошло, то в этом случае TGS пытается восстановить предыдущий запрос. Также служба TGS должна следить за всеми действительными аутентификаторами (время жизни которых не истекло), поскольку в прошедших запросах может быть указана метка времени, которая будет оставаться действительной еще некоторое время. Хотя при этом другие запросы, содержащие различные билеты и аутентификаторы, будут игнорироваться.

В том случае, если запрос содержит отвечающий требованиям билет, сервер предоставляет клиенту билет для доступа к целевому серверу. TGS также создает сеансовый ключ для сервера и клиента, зашифрованный на сеансовом ключе клиент-TGS. Оба этих сообщения передаются клиенту, после чего он расшифровывает сообщение, содержащее сеансовый ключ (шаг 4).

Для успешного прохождения аутентификации у целевого сервера клиент создает аутентификатор, зашифрованный на сеансовом ключе «клиент-сервер», и отправляет его вместе с билетом, зашифрованным на секретном ключе целевого сервера, который был передан от службы TGS.

Приняв данные от клиента, сервер проводит проверку аутентификатора. Он расшифровывает его при помощи своего секретного ключа и извлекает из него сеансовый ключ «клиент-сервер». Билет также подвергается проверке. Процедура проверки схожа с процедурой, проводимой в случае сессии клиент-TGS, то есть проверяется соответствие сетевых адресов и временной метки.

В том случае, если приложение (клиент) требует аутентификации сервера, он возвращает клиенту сообщение, содержащее метку времени, которая зашифрована на сеансовом ключе «клиент-сервер». После чего клиент и сервер имеют возможность пересылать зашифрованные сообщения.

Вопросы безопасности Kerberos v5

Несмотря на то что данный протокол получил широкое распространение среди пользователей и подвергался многочисленным доработкам, в результате чего и появилась его пятая версия, обеспечиваемый уровень безопасности не в полной мере удовлетворяет неоходимым требованиям, и в некоторых случаях данный протокол может стать объектом успешной атаки нарушителем.

Рассмотрение вопросов практической безопасности следует начать с напоминания, что Kerberos, как и любое другое программное средство криптографической защиты, работает в недоверенной программной среде. Недокументированные возможности или неправильная конфигурация данной среды может привести к тому, что произойдет утечка критической информации, например выход в среду передачи сеансовых ключей, хранящихся на жестком диске, или данных открытой информации. Даже в случае хранения ключей на время сеанса работы пользователя только в оперативной памяти сбой в ОС может привести к тому, что ключи будут скопированы на жесткий диск.

При использовании на рабочей станции с установленным ПО Kerberos многопользовательского режима или при отсутствии контроля доступа к рабочей станции создается потенциальная возможность внесения программ-закладок или модификации криптографического ПО. Поэтому безопасность Kerberos во многом зависит от надежности защиты рабочей станции, на которой установлен данный протокол.

Что касается самого протокола, здесь необходимо указать на конкретные уязвимости, которые могут стать причиной успешной удаленной атаки. К ним относятся:

- повторное использование перехваченной информации. Злоумышленник имеет возможность перехватывать аутентификаторы и использовать их повторно. Это возможно в течение всего времени действия билета. В аутентификаторе имеется поле метки времени. Кроме того, для защиты от повторного использования следует кэшировать билеты, что, к сожалению, не всегда осуществимо;
- синхронизация часов. Защитные свойства аутентификаторов основаны на том, что часы субъектов системы жестко синхронизированы. Если сервер может быть введен в заблуждение относительно текущего системного времени, появляется возможность повторного использования перехваченных аутентификаторов. Ввиду того, что протоколы синхронизации часов (NTP и др.) не обладают высокой степенью надежности, это иногда превращается в серьезную проблему;

- восстановление паролей. Применительно к системе Kerberos возможен целый класс атак, направленных на восстановление паролей пользователей. В самом простом случае злоумышленник может собирать перехваченные из сети билеты и пытаться дешифровать их. С учетом того, что пользователи обычно выбирают плохие пароли, атакующий, получив достаточное количество материала, имеет хорошие шансы восстановить пароль. Для успешного осуществления этой атаки сбор билетов не является обязательным. В силу того, что начальный запрос посылается в открытом виде, злоумышленник может генерировать его любое число раз и получать зашифрованные на ключе пользователя TGT-билеты;
- повторное использование идентификаторов субъектов. Теоретически возможна ситуация, когда новый субъект системы получит тот же идентификатор, которым был снабжен уже выбывший субъект. Если этот идентификатор остался в списках управления доступом какого-либо сервера, новый субъект унаследует права доступа выбывшего. Контроль за корректным удалением субъектов из системы целиком возлагается на администраторов;
- сеансовые ключи. Важное их свойство – многосеансовость: каждый такой ключ используется в нескольких сеансах связи между сервером и клиентом, что делает возможным подстановку сообщений, перехваченных в течение одного сеанса, в информационный поток другого;
- область действия билетов и механизм доверия. Реализованный в версии 5 механизм межкластерной аутентификации также порождает проблемы для безопасности. Специфика этого механизма такова, что все участвующие в доверительных отношениях кластеры должны иметь «таблицы маршрутизации» для перенаправления ответов на запросы к серверам аутентификации. Эти таблицы должны быть статическими, и для их настройки требуется какая-то другая защищенная технология; что делает Kerberos зависимым от другой системы безопасности;
- файлы билетов. В системе UNIX файл с текущими билетами пользователя доступен только владельцу (маска доступа 0600). Однако хранение билетов непосредственно на файловой системе открывает доступ к ним любому процессу с соответствующими привилегиями. Альтернатива – хранение билетов в памяти – также не является удачным выходом, так как память подлежит свопингу на диск. Еще более опасной эта ситуация становится при использовании бездисковых машин, осуществляющих доступ к файлам по сети. При реализации Kerberos

для Windows NT это ограничение может быть легко устранено, так как данная ОС обеспечивает механизм блокирования страниц в оперативной памяти, а также позволяет создавать и использовать невыгружаемые области памяти.

В заключение хотелось бы отметить, что протокол Kerberos является перспективным средством аутентификации. На сегодняшний день он поддерживается в таких крупных проектах, как OSF DCE, Windows NT 5.0 и FreeBSD 2.0. Kerberos может использоваться в сочетании с различными криптографическими схемами, включая шифрование с открытым ключом.

Протокол Secure Socket Layer

Основная цель SSL-протокола – обеспечить конфиденциальность и достоверность между взаимодействующими приложениями. Данный протокол располагается между стеком протоколов TCP/IP и прикладным ПО, обеспечивая тем самым инкапсуляцию трафика прикладного уровня. Основу SSL-протокола составляет протокол аутентификации (Handshake Protocol), который позволяет не только удостовериться клиенту и серверу в аутентичности друг друга, но и согласовать используемые алгоритмы, и распределить общий сеансовый ключ. Основное достоинство SSL-протокола – независимость от прикладного ПО.

Таким образом, SSL-протокол обеспечивает безопасность сетевого соединения, которая основана на использовании следующих криптографических механизмов:

- алгоритмов шифрования, обеспечивающих конфиденциальность соединения. С их помощью зашифровываются данные, передаваемые сторонами информационного взаимодействия после успешного прохождения аутентификации и ключевого обмена. В рамках SSL могут использоваться такие алгоритмы блочного шифрования, как DES, RC4 и др., и поддерживаться сразу несколько алгоритмов. Выбор конкретного алгоритма, на котором будет производиться зашифрование, осуществляется в ходе согласования контекста безопасности;
- двусторонней аутентификации с одновременным распределением сеансовых ключей. При этом используются методы асимметричной криптографии (RSA, DSS и др.);
- хэш-функций (MD5, SHA и др.), которые используются для обеспечения целостности и достоверности передаваемых сообщений на основе генерации MAC-кодов.

Описание протокола аутентификации

В ходе протокола аутентификации (SSL Handshake protocol) клиент и сервер согласуют версию протокола, используемые алгоритмы, а также проходят двустороннюю аутентификацию и вырабатывают общий сеансовый ключ.

Протокол аутентификационного обмена ключами содержит следующие шаги:

1. От клиента серверу (Client hello). Клиент пересылает случайное число (Client.param), идентификатор сессии (SessionID), список поддерживаемых криптографических алгоритмов и список поддерживаемых методов сжатия сообщений.
2. От сервера клиенту (Server hello). Сервер выбирает свое случайное число (Server.param), проверяет SessionID из сообщений Client hello (на предмет уникальности, для этого сервер должен хранить список ранее использованных с данным клиентом идентификаторов сессии) и посылает Server.param и SessionID вместе с выбранным идентификатором криптографического алгоритма и методом сжатия из списков, предоставленных клиентом.
3. От сервера клиенту (Certificate). Сервер передает клиенту свой сертификат, который кроме открытого ключа подписи может содержать открытый ключ шифрования, а также ряд параметров, используемых для последующего распределения сеансовых ключей, например параметры сервера, в соответствии с протоколом Диффи-Хэлмана.
4. От сервера клиенту (Server key exchange). Данное сообщение используется, если оно не имеет сертификата или если сертификат сервера содержит только открытый ключ подписи. В нем находятся подписанные параметры, использующиеся для обмена ключами: Sign(Client.param+Server.param+Param) (здесь и далее под знаком + будем подразумевать конкатенацию строк). Содержание Param зависит от выбранного типа обмена ключами (RSA, Fortezza или Диффи-Хэлмана), например: в случае обмена ключами типа Диффи-Хэлмана в нем присутствует значение $X = g^x \bmod n$.
5. От сервера клиенту (Certificate request). В данном сообщении сервер запрашивает у клиента его сертификат.
6. От сервера клиенту (Server helloDone). Данное сообщение означает конец передачи данных в рамках протокола SSL Handshake со стороны сервера.
7. От клиента серверу (Certificate). Серверу пересылается сертификат клиента в случае, если было передано сообщение (Certificate request).

Если у клиента нет сертификата, он может ответить серверу предупреждающим сообщением; в этом случае сервер в зависимости от выбранной политики безопасности может продолжить обмен данными или ответить клиенту сообщением об ошибке и закончить сессию.

8. От клиента серверу (Client key exchange). Содержимое данного сообщения зависит от выбранного типа открытого распределения ключей, например: в случае выбора RSA клиент выбирает случайное число (pre_master_secret) и зашифровывает его на открытом ключе сервера, полученного из сертификата сервера.
9. От клиента серверу (Certificate verify). Данное сообщение используется для проверки сертификата. В нем передается значение Hash(master_secret + pad2 + Hash(handshake_messages + master_secret + pad1)), где master_secret – значение, получаемое из pre_master_secret, а pad1 и pad2 – случайные значения, которые проверяются на сервере.
10. От клиента серверу (Finished). Данное сообщение используется для завершения протокола SSL Handshake. В нем передается значение Hash(master_secret + pad2 + Hash(handshake_messages +Sender + + master_secret + pad1)), которое проверяется на сервере.

В SSL-протоколе используется два типа аутентификации – двусторонняя аутентификация и аутентификация только сервера. Существует также третий тип установления защищенного канала передачи данных – без проведения аутентификации клиента и сервера.

Прежде чем приступить к рассмотрению безопасности SSL-протокола, следует отметить, что основным секретным параметром, знание которого распределяется между клиентом и сервером, является pre_master_secret.

Вопросы безопасности протокола SSL

Основным недостатком SSL является то, что большинство его реализаций поставляется с учетом экспортных ограничений в длине ключа равной 40 бит. При сегодняшнем уровне развития вычислительной техники это позволяет проводить на данный протокол атаки методом «грубой силы».

Еще одним негативным моментом в SSL являются возобновляемые сессии, суть которых заключается в том, что, если клиент и сервер разорвали соединение, они могут возобновить его, проведя минимальный обмен данными, и использовать старый параметр SessionID. Злоумышленник, скомпрометировав одну из предыдущих серий, может провести с сервером процедуру ее восстановления. В результате будут скомпрометированы все последующие данные, передаваемые в данной сессии.

Кроме того, в SSL для аутентификации и шифрования используются одинаковые ключ, что при определенных условиях может привести к потенциальной уязвимости (в начале второй главы этой книги рассматривался вопрос о последствиях использования одинаковых ключей для подписи и для асимметричного шифрования). К тому же подобное решение дает возможность собрать больше статистического материала, чем при аутентификации и шифровании разных ключей.

Как и другие программные продукты, SSL подвержен атакам, связанным с недоверенной программной средой, внедрением программ-закладок и др.

Из сравнительного анализа данного протокола можно сделать вывод, что Kerberos является более полнофункциональным механизмом защиты данных. Наряду с реализацией защищенного канала передачи данных прикладного уровня в нем разрешается осуществлять разграничение удаленного доступа к защищаемым ресурсам; для этого, правда, протокол Kerberos требует небольшой доработки. SSL не позволяет реализовывать разграничение доступа к ресурсам, но легче интегрируется в конечные ОС – поддержка клиентской части SSL-протокола уже исполнена в большинстве современных браузеров, таких как Internet Explorer или Netscape Navigator. Чтобы они начали работать, необходимо инсталлировать библиотеки CryptoAPI и получить в СА сертификат открытого ключа. Серверная часть SSL-протокола может интегрироваться, например в рамках IIS, как ISAPI-фильтр. Интеграция Kerberos рассматривается ниже. Если Kerberos может работать как с открытыми ключами, так и с применением предварительного распределения ключевой информации, то SSL может применяться только при наличии сертификатов открытых ключей (должен использоваться хотя бы сертификат ключа подписи).

3.5.4. Решения по защите информации в Web-технологиях

Этот раздел посвящен продуктам TrustedWeb (SSE) и «Форт» (МО ПНИЭИ), которые имеют существенные различия как по составу сервисных функций, так и по сложности администрирования.

Проект TrustedWeb

TrustedWeb является продолжением проекта SESAME, который, в свою очередь, был основан на протоколе Kerberos. При создании проекта TrustedWeb были учтены результаты маркетинговых исследований, показавших, каким требованием должны удовлетворять средства защиты в современных ИВС:

- средство защиты должно легко интегрироваться в существующую инфраструктуру ИВС, а также масштабироваться с минимальными затратами;
- затраты по конфигурированию и администрированию данного средства защиты должны быть минимальны;
- средство защиты должно реализовывать или быть рассчитано на использование ИВС с разным уровнем секретности защищаемой информации.

TrustedWeb обеспечивает:

- аутентификацию сторон информационного обмена. В TrustedWeb используется два типа аутентификации пользователей: на основе паролей и на основе сертификатов открытых ключей;
- выработку сеансового ключа;
- обеспечение конфиденциальности и целостности передаваемой информации;
- разграничение доступа к защищаемым ресурсам. При этом реализация системы разграничения доступа является дружественной для пользователей и удобной в управлении;
- аудит системы безопасности;
- инсталляцию с минимальными изменениями существующей инфраструктуры.

Организационная структура данного продукта представлена на рис. 3.19. Структурные компоненты, изображенные на нем, выполняют следующие задачи:

- сервер безопасности домена (устанавливается ПО Domain Security Server и TrustedWeb Server). Его роль основная; все серверы, предоставляющие информационные услуги, и клиенты с инсталлированным продуктом TrustedWeb должны пройти процедуру администрирования на данном сервере. Все клиенты и серверы, зарегистрированные на одном сервере безопасности, составляют домен безопасности (данное понятие является основополагающим во всей концепции построения TrustedWeb – см. рис. 3.19). Клиенты и серверы из разных доменов безопасности используют механизм междоменных отношений для взаимодействий друг с другом. Сервер безопасности выполняет следующие функции:
  - аутентификацию клиентов и серверов;
  - разграничение доступа к защищаемым ресурсам;
  - аудит системы безопасности;
  - администрирование и управление системой безопасности;

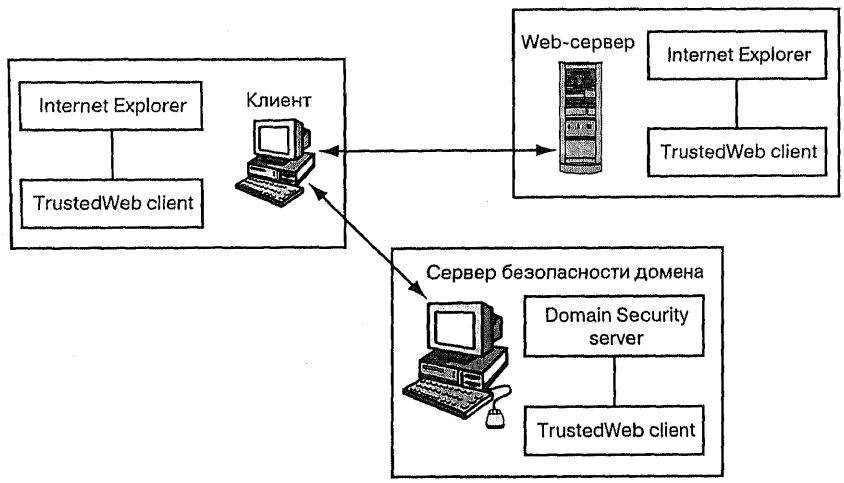

Рис. 3.19. Организационная структура TrustedWeb

- Web-сервер (установленное ПО TrustedWeb Server-TWS). ПО TWS работает как proxy-сервер по отношению к ПО Web-сервера и обеспечивает контроль доступа к ресурсам Web-сервера, например к URL. Разрешение или запрещение доступа к ресурсам сервера происходит на основе правил доступа, определенных в ACL:
  - общедоступные URL;
  - пользовательские URL. Доступ к содержимому данных URL возможен только при наличии у пользователя прав доступа к конкретному ресурсу. Данный тип URL виден пользователям, не имеющим права доступа к содержимому данного URL;
  - скрытые URL. Данный тип ресурсов является наиболее защищенным. Они не видны из сети, для того чтобы получить к ним доступ, пользователь должен установить с TWS контекст безопасности и иметь соответствующие привилегии;
- клиент (TrustedWeb Client). Рабочая станция с установленным ПО Internet Explorer или Netscape Navigator; данная версия TrustedWeb поддерживает ОС Windows 95/NT. ПО TrustedWeb Client (TWC) взаимодействует с ПО Domain Security Server (DSS) и TWS в рамках реализации системы разграничения доступа. ПО клиента запрашивает необходимые пользователю привилегии (роль или атрибуты) для доступа к Web-серверу и передает их TWS. Клиент также получает от DSS информацию, необходимую для обеспечения конфиденциальности и целостности передаваемых между TWC и TWS сведений.

Система разграничения доступа

Доступ к ресурсам компьютера зачастую основан на проверке идентичности пользователей, осуществляемой с предоставлением пользователем своего пароля для доступа к определенному серверу. Данный подход достаточно уязвим по отношению к потенциальным атакам злоумышленникови неприемлем в тех организациях, которые хотят иметь возможность управлять доступом пользователей к нескольким серверам в режиме реального времени. При этом отрицается возможность проведения громоздких операций по администрированию и переконфигурированию защищаемых серверов. TrustedWeb позволяет решить эту проблему.

В основе системы разграничения доступа лежит два типа контроля доступа: а) с использованием ролей и б) с использованием атрибутов. В первом случае роль назначается каждому пользователю домена безопасности и определяет права доступа к конкретным защищаемым ресурсам. Пользователи, обладающие одинаковыми ролями, соответственно имеют одинаковые права доступа к одним и тем же ресурсам. Понятие *роль* распространяется на все защищаемые серверы домена, поэтому пользователь получает доступ к ресурсам любого сервера, где действует понятие роли. При этом используется одинаковый механизм аутентификации, то есть для всех доступных ему ресурсов он должен помнить только один пароль и идентификатор или иметь один сертификат открытого ключа. Разграничение доступа на основе атрибутов доступа (вариант «б») используется в TrustedWeb в том случае, если роль неоправданна, например, когда пользователю нужно выделить уникальные права доступа.

Архитектура открытых ключей

Как уже говорилось, в рамках TrustedWeb используются сертификаты открытых ключей для использования в качестве одного из типов аутентификации. С целью обеспечения надежного функционирования всей системы безопасности в комплект поставки TrustedWeb входит ПО центра сертификации. Данное ПО допускается использовать опционально. Подобное решение принято потому, что, во-первых, организация может не использовать сертификаты открытых ключей или уже иметь свою архитектуру сертификации.

Формат представления сертификатов в TrustedWeb – X. 509 v3. Архитектура открытых ключей представлена на рис. 3.20.

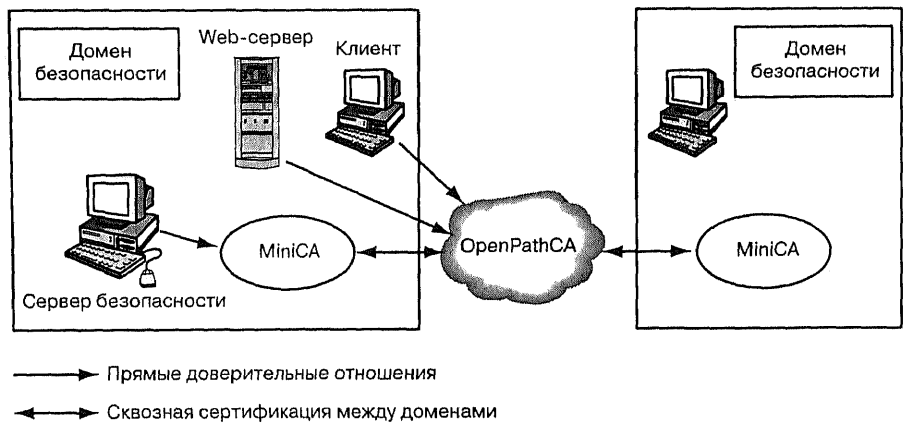

Рис. 3.20. Архитектура открытых ключей

«Форт»

В настоящее время решения для работы в Web являются зеркальным отражением современных методов управления инфраструктурами кредитно-финансовой сферы. Благодаря использованию Web-технологий компании предоставляют возможность коллегам, партнерам и клиентам получить доступ к необходимой для них информации. При этом возникают проблемы управления этим доступом и надежной защиты сайта или корпоративной информационной системы от попыток несанкционированного доступа к защищаемым ресурсам.

Одним из решений подобных проблем является система «Форт». Целью разработчиков этой системы заключалась в необходимости поддержать корпоративные интересы в бизнесе. Система «Форт» – это совместный продукт компании АНКЕЙ и МО ПНИЭИ, интегрированный со средством криптографической защиты информации «Верба-0», сертифицированным ФАПСИ.

Система «Форт» – это программное решение, обеспечивающее криптографическую защиту информации, передаваемой между браузером и Web-сервером по протоколу HTTP, и позволяющее организовать защищенный доступ множества пользователей к выделенному серверу.

Данное средство – способ обеспечения криптографической защиты информации, передаваемой по протоколу HTTP, с применением в качестве клиентского программного обеспечения браузера Microsoft Internet Explorer, что позволяет пользователю быстро, без дополнительного обучения и адаптации применять установленную у него систему. При этом использование СКЗИ «Верба-OW» гарантирует надежность и безопасность передачи данных по сетям общего доступа. Средства криптографической

защиты информации, реализованные в системе «Форт», являются «прозрачными» для пользователя.

Функции шифрования HTTP-трафика и обеспечения целостности передачи данных (при помощи ЭЦП) вызываются пользователем по мере необходимости, с помощью соответствующих функций, встроенных в Internet Explorer.

Система «Форт» способна работать «прозрачно» как с защищенными Web-серверами, так и с обычными. Она имеет открытый интерфейс, что позволяет интегрировать ее с другими программными продуктами. Система проста в инсталляции на компьютеры с операционными системами Windows NT 4.0, Windows 95 и начинает реализовывать функции шифрования автоматически сразу после установки.

«Форт» состоит из подсистемы ЭЦП и подсистемы шифрования HTTP-трафика. Они обе выполнены в виде открытых прикладных интерфейсов. В серверной части системы используется Microsoft Internet Information Server. Для обеспечения функций ЭЦП, проверки подписи, шифрования HTTP-трафика применяется СКЗИ «Верба-OW», а для генерации ключей – СКЗИ «Верба О».

3.6. Применение межсетевых экранов

В последнее время наиболее популярными среди средств защиты информационных ресурсов в Internet становятся *межсетевые экраны* (Firewall или брандмауэры). Понятно, что доступ к Internet расширил возможности проникновения посторонних пользователей в хранилища важной для той или иной организации информации. Поэтому следует помнить, что при наличии упреждающей политики защиты существенно легче оградить свои ресурсы от несанкционированных посягательств. Большинство экспертов в этой области рекомендуют многоуровневый подход для обеспечения безопасности сетевых ресурсов, включая шифрование и аутентификацию, однако при выборе любой архитектуры системы защиты ее ключевым компонентом является брандмауэр. Межсетевой экран размещается на шлюзе между локальной сетью и Internet. Помимо других функций брандмауэр может просматривать IP-пакеты и в зависимости от адресов отправителя и получателя пропускать или не пропускать пакеты, пытающиеся проникнуть в систему.

Межсетевой экран (МЭ) располагается на границе сети и регулирует доступ к корпоративным ресурсам. Это устройство анализирует и собирает информацию о внешних по отношению к сети пакетах и сеансах (в зависимости от типа брандмауэра). Отвечающий реализуемой политике

безопасности МЭ в соответствии с принятыми правилами пропустит или не пропустит конкретный пакет и позволит или не позволит организовать конкретный сеанс.

МЭ делятся на три основных класса:

- фильтры пакетов;
- шлюзы сеансового уровня;
- шлюзы уровня приложений.

Системы фильтрации пакетов просеивают каждый IP-пакет через сито определенных пользователем правил и определяют права пакета на проход во внутреннюю часть сети. Шлюзы уровня приложений в ответ на каждый поступающий запрос о предоставлении сервиса организуют внешний сетевой сеанс; они же открывают соответствующий внутренний сеанс для санкционированного доступа и передают пакеты между внешними и внутренними соединениями. Вообще говоря, шлюзы приложений, по сравнению с фильтрами пакетов, обеспечивают более тщательный контроль за сеансом, но, как следствие, они требуют применения и более мощных вычислительных мощностей. Системы обоих типов предназначены для того, чтобы защитить сеть от таящихся вовне опасностей.

По мнению экспертов, брандмауэры должны обладать тремя важными особенностями, а именно:

- весь трафик должен проходить через одну точку;
- брандмауэр обязан контролировать и регистрировать весь проходящий трафик;
- платформа МЭ должна быть неприступна для атак.

Рынок МЭ сформировался в начале 90-х годов, хотя такие компании, как Digital Equipment и Cisco Systems, включали аналогичные технологии в свои продукты и ранее. Развитие шло настолько быстро, что уже в 1992 году технологии МЭ достигли расцвета; с этого момента рынок просто изобилует разнообразными продуктами подобного типа.

3.6.1. Пакетные фильтры

Фильтры пакетов производят оценку данных на основе IP-информации, содержащейся в заголовке пакета, а точнее в адресе отправителя и получателя пакета. Фильтр не только считывает IP-заголовок, но и сопоставляет полученную информацию со списком правил фильтрации для разрешения или запрещения передачи пакета (табл. 3.9). В правилах фильтрации содержатся поля IP-адресов, типы протоколов, номера портов отправителя и получателя.

Таблица 3.9. Пример правил фильтрации

Правила	Адрес отправителя	Адрес получателя	Действие
A	192.168.0.3	192.168.0.5	Разрешить
B	0.0.0.0	0.0.0.0	Запретить
C	192.168.0.4	192.168.0.5	Разрешить

Прежде чем разрешить пакету продолжение предполагаемого для него маршрута, фильтры пакетов сравнивают указанные в нем данные с предопределенными значениями. В целом фильтры пакетов представляют наименее дешевые решения МЭ, но, благодаря своему умению проверять пакеты различных протоколов, являются и самыми гибкими инструментами решения поставленной задачи. Кроме того, фильтры работают быстро, поскольку для принятия решения они просто просматривают информацию о пакете. Однако фильтры пакетов имеют несколько существенных недостатков: они не в состоянии отслеживать конкретный сетевой сеанс и не в силах предотвратить атаки с имитацией IP-адресов.

Имитация IP-адресов имеет место, когда хакер присваивает IP-адрес законного пользователя – зачастую им является внутренний адрес того, кто имеет доступ к ресурсам. Так как фильтры пакетов «просматривают» информацию об IP-адресе, то они допускают пакет с разрешенным адресом в сеть вне зависимости от того, откуда инициирован сеанс и кто скрывается за адресом. Однако усовершенствованная версия этого механизма, известная как динамическая фильтрация пакетов, позволяет анализировать адрес, с которого некто пытается осуществить доступ, и производит «пингование» (ping) для проверки этого адреса. Очевидно, если злоумышленник использует внутренний IP-адрес компании извне, то ping не достигнет отправителя пакета и сеанс не получит продолжения. Динамическую фильтрацию пакетов поддерживают продукты типа WatchGuard Security System компании Seattle Software Labs и BorderWare Firewall Server компании Secure Computing (данный продукт был приобретен Secure вместе с компанией Border Network Technologies из Торонто в начале этого года).

Компании Seattle Software Labs, Cisco и CheckPoint Software Technologies также поддерживают технологию преобразования сетевого адреса, которая обеспечивает обычную фильтрацию пакетов с искажением. При прохождении пакета через брандмауэр его IP-адрес заменяется каким-то иным, выбранным из пула адресов. Такая замена позволяет скрыть внутренние адреса от злоумышленника за пределами сети. Другие типы брандмауэров, например шлюзы уровня приложения и шлюзы уровня канала, обладают этим же свойством по умолчанию.

Когда дело касается протоколирования сетевого трафика, продукты, содержащие только фильтры пакетов, оказываются не в состоянии выполнить эту задачу – в результате администраторы не могут определить, что их сеть была взломана. Сети, имеющие несколько точек доступа извне, или сети, содержащие чрезвычайно важную информацию, наряду с фильтрами пакетов должны использовать и другие продукты.

3.6.2. Шлюзы сеансового уровня

Шлюз сеансового уровня представляет собой транслятор TCP-соединения. Пользователь устанавливает соединение с определенным портом на брандмауэре, который и производит дальнейшее соединение с местом назначения по другую от себя сторону. Во время сеанса этот транслятор, действуя как провод, копирует байты в обоих направлениях. Как правило, пункт назначения задается заранее, в то время как источников может быть много (соединение типа «один-много»). Используя разные порты, можно создавать различные конфигурации соединений. Данный тип шлюза позволяет создавать транслятор для любого определенного пользователем сервиса, базирующегося на TCP, осуществлять контроль доступа к этому сервису и сбор статистики по его использованию.

Чтобы определить, является ли запрос на сеанс связи допустимым, шлюз сеансового уровня выполняет следующую процедуру (схема не претендует на исчерпывающую полноту). Когда авторизованный клиент запрашивает некоторую услугу, шлюз принимает этот запрос, проверяя, удовлетворяет ли клиент 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 внешним хостом и следит за выполнением процедуры квитирования связи по протоколу TCP. Эта процедура состоит из обмена TCP-пакетами, которые помечаются флагами SYN (синхронизировать) и ACK (подтвердить).

Первый пакет сеанса TCP, помеченный флагом SYN и содержащий произвольное число, например 1000, является запросом клиента на открытие сеанса. Внешний хост, получивший этот пакет, посылает в ответ пакет, помеченный флагом ACK и содержащий число, на единицу больше, чем в принятом пакете (в нашем случае 1001), подтверждая таким образом прием пакета SYN от клиента. После этого осуществляется обратная процедура: хост посылает клиенту пакет SYN с исходным числом (например, 2000), а клиент подтверждает его получение передачей пакета ACK, содержащего число 2001. На этом процесс квитирования связи завершается.

Шлюз сеансового уровня считает запрошенный сеанс допустимым только в том случае, если при выполнении процедуры квитирования связи флаги SYN и ACK, а также числа, содержащиеся в TCP-пакетах, оказываются логически связанными между собой.

После того как шлюз определил, что доверенный клиент и внешний шлюз являются авторизованными участниками сеанса TCP, и проверил допустимость данного сеанса, он устанавливает соединение. Начиная с этого момента шлюз просто копирует и перенаправляет пакеты туда и обратно, не проводя никакой фильтрации. Он поддерживает таблицу установленных соединений, пропуская данные, относящиеся к одному из сеансов связи, которые зафиксированы в этой таблице. Когда сеанс завершается, шлюз удаляет соответствующий элемент из таблицы и разрывает цепь, использовавшуюся в данном сеансе.

Шлюзы сеансового уровня не имеют уязвимых мест, однако после установления связи такие шлюзы фильтруют пакеты только на сеансовом уровне модели OSI, то есть не могут проверять содержимое пакетов, передаваемых между внутренней и внешней сетью на уровне прикладных программ, и, поскольку эта передача осуществляется вслепую, хакер, находящийся во внешней сети, имеет возможность протащить свои пакеты через шлюз. После этого хакер может уже напрямую обратиться к внутреннему Web-серверу, который сам по себе не способен выполнять функции МЭ.

Иными словами, если процедура квитирования связи успешно завершена, шлюз сеансового уровня установит соединение и будет копировать и перенаправлять все последующие пакеты независимо от их содержимого. Чтобы фильтровать пакеты, генерируемые определенными сетевыми службами в соответствии с их содержимым, необходим шлюз прикладного уровня.

3.6.3. Шлюзы уровня приложений

Вместо анализа IP-пакетов шлюзы этого типа изучают данные на уровне приложений. Часто шлюзы приложений используют уполномоченное приложение для создания отдельного сеанса. В отличие от фильтра пакетов, этот сеанс не допускает прямого соединения между двумя сетями, другими словами, он не может выступать в качестве посредника для трафика пакетов

Обнаружив сетевой сеанс, шлюз приложений останавливает его и вызывает уполномоченное приложение для оказания запрашиваемой услуги, пусть это будет telnet, ftp, World Wide Web или электронная почта.

Инициировав уполномоченный сеанс, брандмауэр, по существу, ограничивает доступ к определенным приложениям. Многие шлюзы приложений предоставляют также полномочия для доступа к HTTP, Network News Transfer Protocol (протокол для управления группами новостей Usenet), Simple Mail Transfer Protocol и SNMP. Некоторые продукты, например Gauntlet компании Trusted Information Systems, помимо вышеупомянутых сервисов, поддерживают и такие, как rlogin, TN3270, POP-3, gopher, X Window, finger и whois.

По большей части популярные МЭ представляют собой шлюзы приложений, хотя они в состоянии осуществлять также фильтрацию пакетов и некоторые другие функции. По самой своей природе шлюзы приложений имеют много преимуществ над фильтрами пакетов. Они выполняются на стандартном оборудовании, в частности на рабочей станции UNIX, имеющей больше, чем у маршрутизатора, возможностей для настройки. В январе 1996 года Raptor Systems выпустила Eagle NT – первый брандмауэр на базе Windows NT, с помощью которого администраторы сетей могут сконфигурировать надежную защиту без знания UNIX. С тех пор такие компании, как CheckPoint и NetGuard, тоже выпустили программные МЭ для Windows NT.

Ввиду того, что шлюзы приложений функционируют на уровне приложений, контроль доступа может быть отрегулирован значительно точнее, нежели при использовании фильтров пакетов. Однако один из недостатков такого подхода заключается в том, что поток трафика в этом случае существенно замедляется, поскольку инициация уполномоченного сеанса требует времени. Многие шлюзы приложений поддерживают скорости вплоть до уровня T-1, но, если компании развертывают несколько МЭ сразу или число сеансов увеличивается, заторы становятся настоящей проблемой.

Шлюзы приложений, кроме того, требуют отдельного приложения для каждого сетевого сервиса, иначе МЭ, не имеющие соответствующего приложения, не позволят осуществить к нему доступ. С технической точки зрения это означает, что при появлении новой версии какого-либо приложения она должна быть загружена на брандмауэр. В CheckPoint эта проблема решена следующим образом: заказчики могут скачать с узла Web компании макросы с поддержкой последних редакций популярных приложений.

Другой тип МЭ, шлюзы уровня канала, напоминает шлюзы приложений, за исключением того, что уполномоченный сеанс организуется на уровне TCP (или UDP), а не на уровне приложения. Приложение выполняется не на МЭ, а в настольной системе внутреннего пользователя. Шлюзы каналов располагаются на хостах и как таковые действуют аналогично

транслирующей программе, получая пакеты от одного хоста и передавая их другому. Такая схема менее надежна, чем в случае шлюза приложений, поскольку пакеты анализируются на сеансовом (пятый уровень модели OSI), а не на прикладном уровне (седьмой уровень модели OSI). Кроме того, шлюзы уровня канала упрощают задачу инициирования внутренним пользователям незаконного сеанса и редко применяются в автономных устройствах, однако если они используются совместно со шлюзами приложений, то такой МЭ более надежен.

3.6.4. Использование межсетевых экранов для создания VPN

Ряд МЭ позволяют также организовывать виртуальные корпоративные сети VPN, объединяющие несколько локальных сетей, включенных в Internet, в одну виртуальную сеть. BorderWare Firewall Server 4.0 позволяет передавать информацию через Internet по шифрованным каналам с применением закрытых и открытых ключей. Продукт обеспечивает шифрование по алгоритму RSA Data Security, стандарт шифрования данных DES с 56-разрядным ключом, Triple DES в три раза мощнее DES, и стандарт шифрования RC5.

Eagle 4.0 компании Raptor Systems также поддерживает VPN с возможностью фильтрации внутри канала виртуальной пользовательской сети. Обычно канал открыт для всех протоколов, но Eagle может пропускать только указанные приложения. Например, сеансы telnet могут быть запрещены, а электронная почта и WWW разрешены. Раньше все сервисы необходимо было разрешать или запрещать одновременно, теперь же некоторые функции можно блокировать по выбору.

Соединения через VPN (менее дорогостоящие, чем выделенная линия) работают, только если на обоих концах канала установлены брандмауэры одного и того же поставщика. При установке нескольких брандмауэров их неполное взаимодействие между собой может создать некоторые проблемы. Однако поставщики брандмауэров, и не только они, стремятся выработать стандарты, чтобы заказчики получили свободу выбора, а узлы гладко взаимодействовали друг с другом. Недавно отдел бизнес-приложений для Internet компании NEC Technologies продемонстрировал VPN с протяженностью от США до Японии. Ввиду того, что многие стандарты шифрования запрещены к экспорту из США, в демонстрации NEC использовала шифрование DES и Triple DES на конце канала в Сан-Хосе и японскую версию DES на другом конце канала в Токио.

Сеть VPN может применяться не только для связи между двумя офисами. Часто мобильному или удаленному пользователю нужен аналогичный

уровень защиты и надежности при обмене информацией или доступе к сервисам Internet. CheckPoint Software использует клиентское программное обеспечение шифрования, благодаря чему удаленные пользователи имеют возможность инициировать надежное соединение либо через коммутируемые линии, либо через Internet.

Средство CheckPoint FireWall-1 SecuRemote совместимо с Firewall-1 позволяет мобильным или удаленным пользователям производить защищенную передачу данных и применять сервисы Internet, даже когда прямой защищенный канал с главным офисом установить невозможно. При установке на портативную или другую удаленную машину SecuRemote дает возможность позвонить по локальному номеру Internet вне зависимости от местоположения пользователей, инициировать соединение VPN, отправить и получить шифрованную информацию.

3.6.5. Proxy-серверы

Proxy-сервер управляет и контролирует трафик Internet между сетью компании и внешним миром. В его функции входит руководство всем исходящим трафиком и проведение его через одну точку, кэширование страниц Web и осуществление контроля над разрешенными сервисами Internet внутри и вне сети. Но даже самое подробное описание proxy-сервера не дает ясной и полной картины его возможностей, в результате до сих пор существует путаница между МЭ и proxy-серверами. Proxy-сервер может или размещаться на той же машине, что и МЭ, или быть установленным за ним (в зависимости от имеющегося на диске места).

На одном уровне proxy-сервер скрывает внутренние IP-адреса и обеспечивает централизацию управления и защиты исходящего трафика IP. При передаче всего трафика через proxy-сервер внутренние IP-адреса остаются скрытыми от внешнего мира.

Некоторые из МЭ, представленных на рынке, наделены функцией разделяемого шлюза, но многие брандмауэры, главным образом фильтры пакетов, такой функции не имеют. В этих случаях добавление proxy-сервера к шлюзу позволит компании выступать перед всем внешним миром под единственным IP-адресом. Данный подход имеет несколько преимуществ. Одно из них в том, что хакеры не могут получать действительные внутренние IP-адреса и использовать их против компании (метод, известный как «подделка IP-адресов» или IP-spoofing). Также компаниям не придется регистрировать каждый внутренний IP-адрес, что стоит иногда довольно дорого.

Proxy-протокол CERN, реализованный в наиболее популярных браузерах, поддерживает протоколы HTTP, ftp и gopher. Клиентская программа должна быть специальным образом сконфигурирована для Internet, вот для чего используется модифицированная версия HTTP. Proxy Server компании Microsoft поддерживает proxy-протокол CERN, так что любой совместимый с этим стандартом браузер может применить протокол без дополнительного клиентского программного обеспечения.

При работе с такими браузерами, как Navigator компании Netscape и Internet Explorer компании Microsoft, пользователь может через меню **Options** задать имя proxy-сервера, с которым затем браузер будет взаимодействовать автоматически. Пользователи могут также определить различные полномочия для сеансов Internet разного типа. Браузеры, не совместимые со стандартом CERN, требуют специальной конфигурации на стороне клиента.

Поддержка proxy-стандарта CERN реализована практически на любом клиенте Proxy Server (компании Microsoft) в составе посредника для Winsock, не требующего никаких изменений на стороне клиента и поддерживающего широкий круг протоколов, в том числе протоколы, не ориентированные на Web, в частности аудио- и видеотрафик. Посредник для Winsock аналогичен Socks (шлюзу TCP для связи авторизованных клиентов с Socks-совместимым сервером или брандмауэром).

После открытия сеанса Internet в задачу proxy-сервера входит обеспечение защиты коммуникаций. Proxy Server компании Netscape использует протокол SSL, разработанный для шифрования сообщений между клиентом и удаленным сервером. Proxy-серверы обеих компаний используют этот протокол посредством организации SSL-туннелей – технологии для защищенного обмена сообщениями, например HTTP и NNTP (Network News Transport Protocol).

Клиенты, использующие Netscape Navigator, уже поддерживают SSL и могут взаимодействовать с удаленным сервером непосредственно через брандмауэр. Proxy Server компании Netscape также позволяет клиентам, не ориентированным на SSL, осуществлять защищенные коммуникации с удаленными хостами через Internet. При таком подходе proxy-сервер запрашивается об организации защищенного соединения по поручению незащищенного клиента. Клиент связывается с посредником с помощью стандартного HTTP, а посредник взаимодействует с удаленным узлом, используя SSL, так что канал через Internet в любом случае оказывается защищенным.

Для проверки входящего трафика на предмет наличия вирусов можно использовать и дополнительные модули независимых производителей.

Данные продукты должны заниматься сканированием входящего трафика HTTP и ftp, а также вложений в почтовые сообщения, благодаря этому инфицированный трафик Internet будет остановлен прежде, чем он сможет попасть на настольные системы в частной сети.

Как и многие другие виды сетевых серверов, proxy-серверы требуют особой конфигурации и постоянного управления. Требования к системе компании Netscape довольно незначительны, и продукт работает с самыми разными серверами Web. Proxy Server этой компании работает как на платформе Intel, так и с некоторыми видами UNIX: OSF компании Digital, HP-UX, AIX, IRIX компании Silicon Graphics, а также SunOS и Solaris компании Sun Microsystems. Поскольку многие брандмауэры лучше приспособлены для работы под UNIX, применение proxy-сервера с аналогичной операционной системой облегчает установку и администрирование.

Управлять продуктом Netscape можно из любого SSL-совместимого браузера, кроме того, продукт поддерживает SNMP-1, SNMP-2 и возможность автоматической конфигурации. Параметры сервера поддаются тонкой настройке. Например, пользователи, которым требуются URL с расширением .com, могут быть направлены к одному proxy-серверу, а тем, кому нужны адреса, оканчивающиеся на .edu, – к другому. В случае возникновения сбоя на одном сервере его место занимает другой proxy-сервер. Чтобы снизить трафик в Internet, оператору лучше установить proxy-сервер в каждом локальном центре и на каждом шлюзе Internet. Операторы Internet наиболее заинтересованы в предлагаемой proxy-серверами возможности кэширования.

Таким образом, proxy-серверы позволяют при доступе к защищаемому сегменту сети осуществлять идентификацию и аутентификацию удаленного пользователя и являются основой для создания VPN.

Брандмауэры и proxy-серверы имеют много общего, но для создания защищенной системы в сети должны быть установлены и те, и другие. Сетей с абсолютно надежной защитой не существует, но, по мнению автора, для обеспечения безопасности сочетание брандмауэра и proxy-сервера оптимально.

3.6.6. Виды подключения межсетевых экранов

Учитывая многообразие практических вопросов обеспечения безопасности, решаемых с помощью брандмауэров, существует несколько схем их подключения к защищаемой сети. Как уже говорилось, брандмауэр – это система или комбинация систем, позволяющих разделить сеть на две или более части и реализовать набор правил, определяющих условия прохождения

пакетов из одной части в другую. Пример типовой схемы подключения брандмауэра при организации VPN представлен на рис. 3.21.

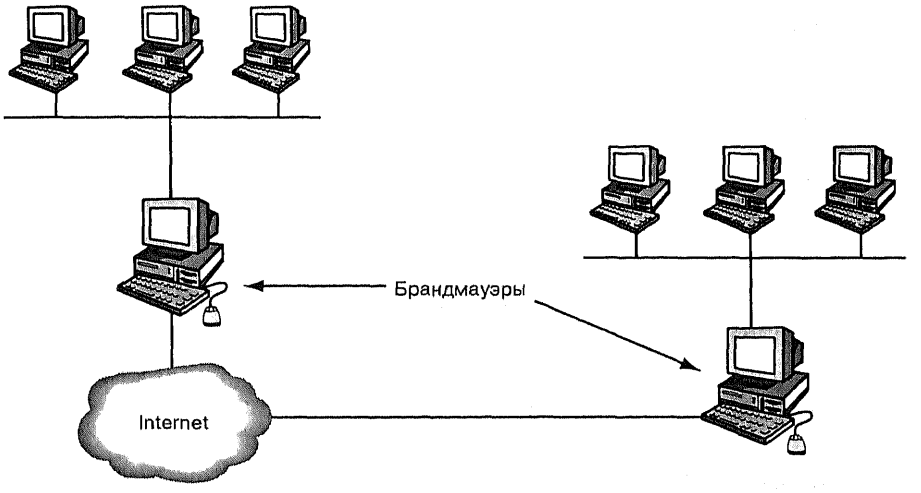

Рис. 3.21. Типовая схема подключения брандмауэра

В принципе МЭ может работать в качестве внешнего маршрутизатора, используя поддерживаемые типы устройств для подключения к внешней сети (см. рис. 3.21). Но иногда работает схема, изображенная на рис. 3.22, однако пользоваться ей следует только в крайнем случае, поскольку требуется очень аккуратная настройка маршрутизаторов, и даже небольшие ошибки могут стать причиной серьезных нарушений в защите.

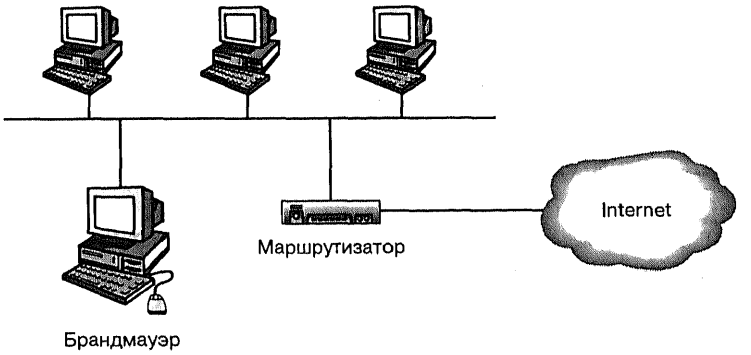

Рис. 3.22 Схема с внутренним маршрутизатором

В случае использования МЭ с несколькими сетевыми интерфейсами чаще всего подключение осуществляется через внешний маршрутизатор (рис. 3.23). При этом между внешним маршрутизатором и брандмауэром

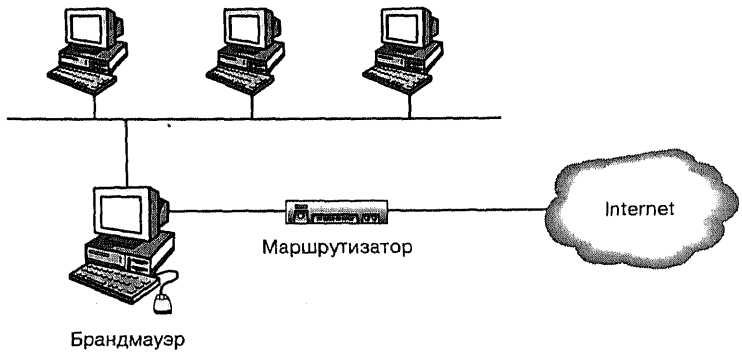

Рис. 3.23 Схема подключения с внешним маршрутизатором

имеется только один путь, по которому идет весь трафик. Обычно маршрутизатор настраивается таким образом, что брандмауэр является для него единственной видимой снаружи машиной. Эта схема наиболее предпочтительна с точки зрения безопасности и надежности защиты.

3.6.7. Использование межсетевых экранов

Говоря об использовании брандмауэров, следует отметить, что этот вид средств защиты информации не нужно воспринимать как панацею от любой беды, поскольку межсетевые экраны в первую очередь предназначены для реализации политики разграничения уделенного доступа к ресурсам сети. Проблема заключается в том, что если, например, брандмауэр запрещает доступ данной группы IP-адресов с Web-сервера, то злоумышленник будет пытаться получить доступ к незащищенным информационным ресурсам.

Если же говорить о том, какие удаленные атаки способен отразить брандмауэр, необходимо выделить следующие:

- противоправный анализ сетевого трафика. Очевидно, что только брандмауэр с реализованными функциями шифрования трафика может противостоять данным атакам;
- подмена одной из сторон информационного обмена. МЭ в данном случае не помощник;
- внедрение ложного объекта распределенной системы. В этом случае МЭ показывает неоднозначные результаты, поскольку внедрение ложного ARP-сервера он сможет предотвратить, как, впрочем, и навязывание ложного маршрута при помощи ICMP-протокола, однако внедрение ложного DNS-сервера он предотвратить не в состоянии;
- отказ в обслуживании. Вмешательство МЭ во время этой атаки только ухудшит ситуацию, так как нарушитель выведет из строя МЭ

и все рабочие станции защищаемой сети будут отрезаны от внешнего мира.

В заключение следует отметить, что МЭ является рабочей станцией с установленной ОС, например FreeBSD, у которой определенным образом переделано ядро, поэтому очевидно, что он будет наследовать все уязвимости, присущие данной ОС, и большинство атак для этой системы будут применимы и для брандмауэра.

3.6.8. Применение криптографии в межсетевых экранах на примере CheckPoint Firewall-1

Межсетевой экран FireWall-1 – это программный продукт компании CheckPoint Software Technologies (распространяемый также некоторыми другими компаниями, например SunSoft), обеспечивающий комплексное решение технических проблем информационной безопасности в intranet. FireWall-1 предоставляет следующие возможности:

- межсетевое экранирование;
- поддержку виртуальных частных сетей;
- защиту коммуникаций между сервером и удаленным клиентом.

FireWall-1 – это проверенный продукт, завоевавший около 40% рынка межсетевых экранов, удостоенный многочисленных наград компьютерных изданий и независимых центров тестирования. В основе идеи создания FireWall-1 – новаторская технология многоуровневой фильтрации с формированием и анализом состояния сетевых соединений, предоставляющая надежную защиту для всех без исключения сетевых протоколов в сочетании с высокой эффективностью и полной прозрачностью для легальных пользователей. Эта технология дополнена средствами централизованного администрирования (конфигурации, анализа и аудита) защитных механизмов, а также логически замкнутым набором криптографических средств.

Только FireWall-1 позволяет организации создать единую интегрированную политику безопасности, которая распространялась бы на множество межсетевых экранов и управлялась бы с любой выбранной для этого точки сети предприятия. Продукт имеет и массу дополнительных возможностей, таких как управление списками доступа аппаратных маршрутизаторов, балансировка сетевой нагрузки на серверы, а также элементы для построения систем повышенной надежности, которые тоже полностью интегрируются в глобальную политику безопасности. Работа CheckPoint FireWall-1 открыта для пользователей и обеспечивает рекордную производительность практически для любого IP-протокола и высокоскоростной технологии передачи данных.

FireWall-1 обеспечивает максимальный уровень безопасности, его высокоэффективные характеристики основываются на технологии инспекции пакетов с учетом состояния протокола. На сегодняшний день это самый передовой метод контроля сетевого трафика, разработанный и запатентованный компанией CheckPoint. Данный метод обеспечивает сбор информации из пакетов данных как коммуникационного, так и прикладного уровня, что достигается сохранением и накоплением ее в специальных контекстных таблицах, которые динамически обновляются. Такой подход обеспечивает полный контроль даже за уровнем приложения без необходимости введения отдельного приложения-посредника (proxy) для каждого защищаемого сетевого сервиса.

Набор продуктов, называемых CheckPoint открытой платформой безопасного взаимодействия предприятий (OPSEC – Open Platform for Secure Enterprise Connectivity), основывается на концепции объединения технологий защиты информации вокруг единого средства представления информационной безопасности предприятия.

CheckPoint FireWall-1 состоит из двух основных модулей:

- модуль управления, который включает графический интерфейс пользователя и собственно компоненты управления. Графический интерфейс позволяет работать с базами данных CheckPoint FireWall-1: базой правил, сетевыми объектами, услугами, пользователями и т.д.;
- экранирующий модуль FireWall-1, установленный на всех компонентах корпоративной сети, производящих фильтрацию сетевых потоков данных. Это могут быть серверы с одним сетевым интерфейсом (тогда модуль защищает только данный сервер) или шлюзы с несколькими интерфейсами (тогда модуль экранирует подсеть). Экранирующий модуль решает три основные задачи:
  - фильтрацию сетевого трафика;
  - протоколирование информации и возбуждение сигналов тревоги;
  - шифрование/дешифрование информации.

Экранирующий модуль встраивается в ядро операционной системы между канальным и сетевым уровнями. Он перехватывает все пакеты, поступающие из/в сетевой/ую карты/у, и обрабатывает их в соответствии со специфицированной базой правил. Такая обработка производится для каждого сетевого интерфейса на серверах и шлюзах. При фильтрации все правила, заданные для данного интерфейса, просматриваются по очереди. Если подходящее правило найдено, выполняются заданные в нем действия (протоколирование, шифрование и т.д.), после чего пакет пропускается или отвергается. Таким образом, экранирующий модуль FireWall-1

функционирует независимо от сетевых механизмов операционной системы. Работа в пространстве ядра позволяет избежать многочисленных переключений контекстов процессов, что существенно повышает эффективность фильтрации.

Экранирующий модуль FireWall-1 способен выполнять еще одну очень важную функцию – трансляцию сетевых адресов. Она не только уменьшает потребность в легальных IP-адресах (в пределах защищаемой сети могут использоваться произвольные адреса, которые при выходе во внешние сети преобразуются и попадают в диапазон, выделенный данной организации), но и скрывает топологию защищаемой сети, что существенно затрудняет действия злоумышленников.

Шифрование в CheckPoint FireWall-1

По мере становления и развития VPN в системе Internet и использования этой информационной магистрали для совершения кредитных операций, продаж и электронного документооборота резко усилились попытки несанкционированного доступа в сети подобного вида. При ведении торговли через Internet, включая пересылку денежных средств, получение и проверку кредитной информации, продажу и даже поставку, требуется надежная и эффективная защита. Новинка для семейства CheckPoint FireWall-1 версии 2.X – это отдельная линия продуктов с расширенными возможностями шифрования. Теперь все продукты CheckPoint FireWall-1 поддерживают работу с шифрованными данными (поставляется как дополнение).

CheckPoint FireWall-1 предоставляет безопасную двунаправленную связь через Internet и механизм шифрования, а также подписи для гарантии целостности данных и конфиденциальности при соединении виртуальных частных сетей. CheckPoint FireWall-1 обеспечивает все потребности защиты предприятия:

- конфиденциальность (подслушивание на линии невозможно);
- подлинность (невозможны подстановки сетевых адресов);
- целостность (подстановка и модификация данных на лету невозможна).

CheckPoint FireWall-1 реализует единую интегрированную стратегию защиты с распознаванием клиентов и шифрованием данных. Предлагая различные схемы шифрования и интегрированные системы распределения ключей, CheckPoint FireWall-1 позволяет предприятию полностью использовать возможности Internet для ведения бизнеса и обеспечения связи.

CheckPoint FireWall-1 поддерживает скорость шифрования более 10 Мб/с на обычных настольных рабочих станциях. Управление полностью интегрировано в редактор базы правил графического интерфейса пользователя

CheckPoint FireWall-1 и охватывает средства просмотра регистрационных записей. Продукт версии 2.1 поставляется с алгоритмом шифрования FWZ1 и реализует выборочное шифрование для широкого спектра сетевых протоколов. Кодирование, декодирование и распределение ключей интегрировано с другими продуктами CheckPoint.

На рис. 3.24 изображена общая корпоративная сеть, где две частных сети соединены через Internet, а также через шлюз FireWall. HQ – станция управления для обеих сетей. Частные сети (hq-сеть и DMZnet) защищены шлюзом FireWall, но внешняя часть сети Internet рассматривается как открытая и небезопасная.

Шлюзы выполняют шифрование для каждой из своих областей: либо для локальной вычислительной сети, либо для группы сетей, которые защищаются шлюзом. За шлюзом, во внутренних сетях, пакеты не шифруются. Область шифрования HQ состоит из HQnet и DMZnet, а область шифрования в London из London-net. Если администратор системы определил, что связь между HQ и London должна быть зашифрована, CheckPoint FireWall-1 начнет шифровать пакеты, проходящие через шлюзовой вход (gateway) в Internet.

Таким образом, шифрование может использоваться в гетерогенной сети без установки и конфигурирования сети на каждом отдельном компьютере.

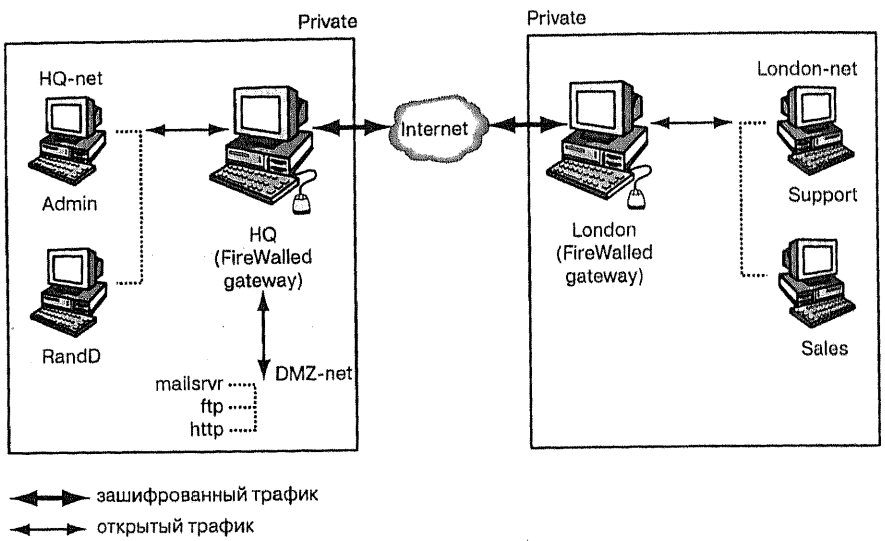

Рис. 3.24. Пример корпоративной сети

Аутентификация в CheckPoint FireWall-1

CheckPoint FireWall-1 обеспечивает пользователям, в том числе удаленным и клиентам dial-up, защищенный доступ к сетевым ресурсам организации с установлением подлинности пользователя при помощи различных схем ее проверки.

Прежде чем соединение пользователя будет разрешено, механизм установления подлинности FireWall-1 определит, какой именно пользователь пытается установить соединение и как он себя авторизует. Заметим, что для этого не потребуется каких-либо изменений на серверах и в клиентских приложениях.

Средства установления подлинности пользователей полностью интегрированы в средства работы с политикой безопасности масштаба предприятия и соответственно могут централизованно управляться посредством графического интерфейса администратора безопасности. Используя программу просмотра статистики, можно отслеживать любые сессии установления подлинности клиента.

FireWall-1 предоставляет три метода установления подлинности пользователя:

- аутентификация пользователя;
- клиентская аутентификация;
- проверка подлинности установленного сеанса связи.

Метод User Authentication

Метод установления подлинности пользователя системы FireWall-1 позволяет определять привилегии доступа для каждого пользователя в отдельности (даже если это многопользовательская ЭВМ) для протоколов FTP, TELNET, HTTP и RLOGIN независимо от IP-адреса клиентского компьютера. Например, если пользователь вынужден работать с серверами организации удаленно, то администратор безопасности может разрешить ему доступ во внутреннюю сеть без того, чтобы его привилегии распространялись на всех других пользователей его рабочего компьютера.

FireWall-1 выполняет проверку подлинности пользователя при помощи специального сервера безопасности, функционирующего на шлюзовом компьютере. FireWall-1 перехватывает все попытки авторизации пользователя на сервере и перенаправляет их соответствующему серверу безопасности. После того как подлинность пользователя установлена, сервер безопасности FireWall-1 открывает второе соединение на необходимый сервер приложения. Все последующие пакеты сессии также перехватываются и инспектируются FireWall-1 на шлюзе.

Пример HTTP Authenticating Proxy

HTTP Authenticating Proxy представляет собой механизм идентификации тех пользователей, которые обращаются к услугам HTTP. Он выполняется на шлюзе и может защищать любое число серверов HTTP во внутренней сети.

Предположим, что в конфигурации, отображенной на рис. 3.25, имеются серверы HTTP на всех компьютерах (то есть на Tower, Palace и BigBen), которые защищены HTTP Authenticating Proxy на шлюзе London.

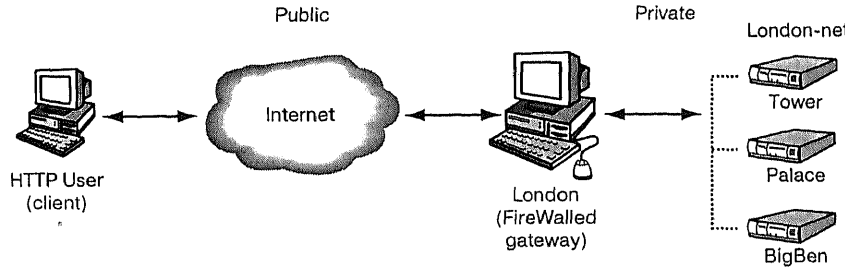

Рис. 3.25. Схема защиты в Internet

URLs должен быть задан следующим образом:
http://<gateway>/<logical server name>/<resource name>, где <resource name> указывает на ту часть URL, которая относится непосредственно к WWW-серверу. Обычно это имя файла.

Предположим, что следующие серверы HTTP имеются в сети и их конфигурация выглядит следующим образом (см. табл. 3.10).

Таблица 3.10. Таблица настроек

Имя сервера	Адрес	Хост	Порт
info	http://www.London.com/info/info.html	Palace	80
tickets	http://www.London.com/tickets/ordtick.html	BigBen	80
actors	http://www.London.com/actors/bios.html	Tower	8000
reviews	http://www.London.com/reviews/clips.html	Tower	8080

В этом случае шлюз C (www.London.com), а также имена серверов определены в поле имени в окне Control Properties/Authentication, где определяется связь между именем сервера, именем компьютера и портом. Для всех внешних соединений известно только одно имя C (www.London.com).

Пользователь, который пытается обратиться к серверам HTTP, например определяя URL http://www.London.com/actors/bios.html, будет

задержан HTTP Authenticating Proxy на шлюзе London. Затем на его экране отобразится окно, в котором нужно ввести идентификатор и пароль для доступа к серверам HTTP.

Клиентская аутентификация

Client Authentication позволяет администратору предоставлять привилегии доступа определенным IP-адресам, пользователи которых прошли соответствующие процедуры установления подлинности. В противовес User Authentication клиентская аутентификация не ограничена только определенными службами, она может обеспечить аутентификацию любого приложения, как стандартного, так и специфичного.

Client Authentication системы FireWall-1 не является очевидным для пользователя, но в то же время какого-либо дополнительного программного обеспечения или модификации существующего не требуется. Для установления подлинности администратор может указать, как каждый из пользователей должен будет авторизоваться, какой сервер и какие службы будут доступны, в какие часы и дни и сколько сессий может быть открыто и как долго они продлятся.

Проверка подлинности установленного сеанса связи

Механизм Transparent Session Authentication можно использовать для любых служб. При этом установление подлинности будет происходить для каждой сессии в отдельности.

После того как пользователь соединился непосредственно с сервером, шлюз с установленным FireWall-1 (в случае необходимости установления его подлинности) инициирует соединение с агентом авторизации сессий. Агент производит нужную авторизацию, и, если подлинность клиента установлена, FireWall-1 разрешает данное соединение.

Поддерживаемые схемы авторизации пользователя

FireWall-1 поддерживает разнообразные варианты авторизации пользователей:

- метод SecurID (пользователь набирает номер, высвечивающийся на электронной карточке Security Dynamics SecurID);
- метод S/Key (от пользователя требуется набрать соответствующую запрашиваемому номеру комбинацию ключа S/Key);
- метод OS Password (пользователь должен набрать пароль операционной системы. Internal-пользователь набирает специальный пароль, хранящийся в FireWall-1 шлюза);

- метод Axent (требуется ввод в соответствии с инструкциями сервера Axent);
- метод RADIUS (требуется ввод в соответствии с инструкциями сервера RADIUS).

Для систем на базе RADIUS-серверов существует несколько сертифицированных реализаций, предлагаемых для использования официальными партнерами.

Распределение ключей в центре CheckPoint FireWall-1

Управляющая станция, отвечающая за централизованное администрирование конфигурации из нескольких межсетевых экранов, выполняет также роль центра распределения криптографических ключей. Другими словами, для всех экранов она генерирует пару ключей (открытый/секретный), чтобы применить алгоритм Диффи-Хэлмана, а также служит сертификационным центром, обслуживающим запросы на их получение. На управляющей станции можно генерировать и новые ключи. Другие шлюзы, как правило, просто получают ключи из сертификационного центра. Станция управления обеспечивает каждый шлюзовой вход (gateway), на котором она установлена:

- парой ключей Диффи-Хэлмана;
- сертифицированной системой авторизации;
- шифрованной передачей ключей на другие шлюзы;

В начале шифрованного сеанса шлюз вычисляет сессионный ключ по алгоритму Диффи-Хэлмана и затем начинает шифрованную связь.

3.7. Защита электронной почты

Электронная почта на сегодняшний день является одной из наиболее применяемых технологий обмена данными между пользователями. В настоящее время это способ повсеместного общения пользователей в Internet. В корпоративных системах подобный метод находит свое применение в качестве средства обеспечения электронного документооборота.

В электронной почте сообщения могут быть простыми записками, передаваемыми в соседнюю комнату, и письмами со сложной структурой вложений, пересылаемыми на другой континент. Преимущество электронной почты для пользователей заключается в следующем: этот вид связи очень похож на обычную почту, однако следует иметь в виду, что он не сводится только к пересылке сообщений по электронным каналам связи. На сегодняшний какой-либо единый общепринятый стандарт

в этой области отсутствует. Дело в том, что системы электронной почты функционируют на различном оборудовании и основываются на различных концепциях, однако главные принципы их построения и работы аналогичны.

Система электронной почты является одним из примеров сетей, построенных по принципу клиент-серверных приложений. Здесь, как и в других подобных распределенных механизмах, пользователь взаимодействует с клиентским программным обеспечением, а администратор – с серверным. Серверы различаются уровнями производительности и надежности, совместимостью с различными стандартами электронной почты, устойчивостью к ошибкам, возможностью расширения.

Говоря об архитектуре построения серверного программного обеспечения, следует отметить, что обычно она состоит из трех основных частей: подсистема хранения сообщений, транспортная подсистема и служба каталогов. Подсистема хранения сообщений отвечает за получение сообщений и хранение до момента прочтения их пользователем. Хранящиеся в подсистеме сообщения могут также содержать присоединенные к ним файлы. Они обычно занимают много места, поэтому часто их количество или размер ограничиваются. Транспортная подсистема, называемая также подсистемой маршрутизации сообщений, осуществляет пересылку сообщений от одного почтового ящика к другому. Сообщение электронной почты бесполезно, если оно не поступит в нужный почтовый ящик. Служба каталогов располагает списком имен всех пользователей в системе и обеспечивает пересылку почты адресатам. Каталоги могут содержать списки сетевых имен или более развернутую информацию, с помощью которой можно объединить пользователей по фирмам, рабочим группам, отделам или по географическому признаку. Для того чтобы найти пользователя в системе электронной почты, нужно знать только его адрес.

3.7.1. Принципы защиты электронной почты

Основными угрозами в системах электронной почты являются следующие:

- несанкционированный доступ к *почтовым сообщениям* (ПС), то есть нарушение конфиденциальности;
- преднамеренное изменение получателем ПС с целью нарушения его достоверности или целостности;
- выдача себя за другого пользователя, чтобы снять с себя ответственность или же использовать его полномочия с целью формирования ложного ПС; изменение законного ПС; санкционирование ложных обменов ПС или же их подтверждение;

- отказ от факта передачи ПС;
- утверждение о том, что ПС получено от некоторого пользователя, хотя на самом деле оно сформировано самим злоумышленником;
- несанкционированные изменения полномочий других пользователей на отправку и получение ПС (ложная запись других лиц, ограничение или расширение установленных полномочий и т.п.);
- набор статистики обмена ПС (отслеживание: кто, когда и к каким ПС получает доступ);
- заявление о сомнительности протокола обеспечения безопасности доставки ПС из-за раскрытия некоторой конфиденциальной информации;
- заявление о ложном времени получении ПС.

К этому списку можно добавить еще две проблемы, тесно связанные с обеспечением безопасности работы электронной почты:

- анонимность;
- атаки потенциальных нарушителей, учитывающие уязвимости в реализации и построении систем электронной почты;

Решение первой проблемы может осуществляться двумя путями:

- применением криптографических средств защиты информации не только к данным, находящимся в ПС, но и к служебной информации ПС;
- использованием *анонимайзера* – почтового сервера, играющего роль proxy-сервера, но только для почтовых сообщений; то есть реальная служебная информация ПС (адрес электронной почты отправителя, IP-адрес отправителя и др.) заменяются соответствующей служебной информацией анонимайзера;

Очевидно, что решение второй проблемы связано с практической реализацией той или иной почтовой системы, и здесь трудно дать какие-либо конкретные рекомендации, кроме разве что тщательного выбора системы связи, применение которой не приводит к большим количествам уязвимостей. (Информацию о подобных уязвимостях можно оперативно находить в Internet, например на сайте CERT.)

Поэтому остановимся на типовых задачах информационной безопасности в системах электронной почты.

Одними из основополагающих документов в части требований защиты информации являются Рекомендации ISO 7498-2-89 и Стандарты МККТТ/ISO для безопасной обработки ПС (Рекомендации серий X.400, X.500 МККТТ). Стандарты и рекомендации для безопасной передачи, приема и обработки сообщений в системе определяют следующие принципы защиты информации:

- конфиденциальность содержания (позволяет отправителю быть уверенным, что никто не прочитает ПС, кроме определенного получателя);
- конфиденциальность последовательности сообщений (позволяет отправителю ПС скрыть последовательность сообщений);
- целостность содержания (позволяет получателю убедиться, что содержание ПС не модифицировано);
- целостность последовательности сообщений (позволяет получателю убедиться в том, что последовательность ПС не изменена);
- аутентификация источника ПС (отправитель получает возможность аутентифицироваться у получателя как источник ПС, а также у любого устройства передачи ПС, через которое они проходят);
- доказательство доставки (отправитель может убедиться в том, что ПС доставлено неискаженным нужному получателю);
- доказательство передачи (отправитель может убедиться в идентичности устройства передачи ПС, на которое оно было подано);
- безотказность поступления (позволяет отправителю сообщения получить от устройства передачи ПС, на которое оно поступило, доказательство того, что сообщение предназначено для доставки определенному получателю);
- безотказность доставки (позволяет отправителю получить доказательство поступления ПС);
- управление контролем доступа (позволяет двум компонентам системы обработки ПС установить безопасные соединения);
- защита от попыток расширения своих законных полномочий (на доступ, формирование, распределение и т.д.), а также изменения (без санкции на то) полномочий других пользователей;
- разметка по уровню защиты ПС (обеспечивает возможность определить уровень защиты ПС в соответствии с принятой политикой безопасности);
- защита от модификации программного обеспечения путем добавления новых функций.

Надежная защита при передаче, обработке и хранении конфиденциальной информации базируется на применении современных криптографических программных и аппаратно-программных средств, реализующих механизмы шифрования информации, а также подтверждения ее целостности и подлинности с использованием электронной цифровой подписи.

Защита информации в почтовых системах должна состоять из комплекса организационно-технических мероприятий по применению программных и аппаратно-программных криптографических методов и средств

защиты с целью предотвращения несанкционированного доступа к конфиденциальной информации, обрабатываемой и хранимой абонентами системы и передаваемой по коммуникационной сети с использованием незащищенных линий связи.

В соответствии с требованиями по безопасности конфиденциальной информации СКЗИ, встроенное в обрабатывающую систему, должно обеспечивать защиту информации на всех этапах, начиная с момента ее ввода в систему вплоть до обработки и хранения. При этом в составе СКЗИ должны быть реализованы средства, осуществляющие:

- зашифрование (расшифрование) файлов и блоков конфиденциальной информации при передаче их по телекоммуникационным каналам, а также при хранении на внешней памяти ЭВМ;
- генерацию ЭЦП (проверку ЭЦП) ПС;
- управление ключевой системой;
- защиту от НСД.

Работы по встраиванию СКЗИ и обеспечению надлежащего уровня защиты системы требуют доработок ее прикладной части. В первую очередь доработке подлежат части системы, ответственные за принятие решения о соответствии ЭЦП под документом ЭЦП абонента-отправителя, система оповещения о компрометации ключей абонента, интерфейсы взаимодействия между компонентами системы.

3.7.2. Средства защиты электронной почты

Почта Privacy-Enhanced Mail

Почта с повышенной секретностью (Privacy-Enhanced Mail, PEM) представляет собой стандарт Internet, одобренный Советом по архитектуре сети (Internet Architecture Board, IAB), для обеспечения безопасности электронной почты в Internet. Первоначальный вариант был создан группой секретности и безопасности (Privacy and Security Research Group, PSRG) Internet Resources Task Force (IRTF), а затем разработка была передана в рабочую группу PEM (PEM Working Group) при IETF. Протоколы PEM предназначены для шифрования, проверки подлинности и целостности сообщения и управления ключами.

PEM является расширяемым стандартом. Процедуры и протоколы PEM разработаны так, чтобы быть совместимыми со множеством подходов к управлению ключами, включая симметричную схему и использование открытых ключей для шифрования ключей шифрования данных.

Симметричная криптография применяется для шифрования текста сообщений. Для контроля целостности сообщения используются криптографические способы хэширования. Другие документы поддерживают механизмы управления ключами с помощью сертификатов открытых ключей, алгоритмов, режимов и связанных идентификаторов, а также электронные подробности, инфраструктуру и процедуры управления ключами.

PEM поддерживает только определенные алгоритмы, но позволяет добавлять и алгоритмы более поздних версий. Сообщения шифруются алгоритмом DES в режиме CBC. Проверка подлинности, обеспечиваемая средством *проверки целостности сообщения* (Message Integrity Check, MIC), использует MD2 или MD5. Симметричное управление ключами может применять либо DES, либо тройной DES с двумя ключами (так называемый режим EDE). Для управления ключами PEM также поддерживает сертификаты открытых ключей, используя RSA (длина ключа до 1024 бит) и стандарт X.509 для структуры сертификатов.

PEM обеспечивает три сервиса повышения секретности: конфиденциальность, проверку подлинности и контроль целостности сообщений. К электронной постовой системе не предъявляется никаких специальных требований. PEM может быть встроена выборочно, в определенные узлы или у конкретных пользователей, при этом ее установка не влияет на работу остальной сети.

PEM раскрывается в следующих четырех документах:

- RFC 1421: часть I. Процедуры шифрования и проверки подлинности сообщений. В этом документе определяются процедуры шифрования и проверки подлинности сообщений, которые должны обеспечить функции почты с повышенной секретностью для передачи электронной почты в Internet;
- RFC 1422: часть II. Управление ключами с помощью сертификатов. В этом документе определяется архитектура и инфраструктура управления ключами, которые основаны на использовании сертификатов открытых ключей, предоставляющих информацию о ключах отправителям и получателям сообщений;
- RFC 1423: часть III. Алгоритмы, режимы и идентификаторы. Этот документ содержит определения, форматы, ссылки и цитаты для криптографических алгоритмов, режимов использования и связанных идентификаторов и параметров;
- RFC 1424: часть IV. Сертификация ключей и родственные функции. В этом документе описываются три типа функций, поддерживаемых PEM: сертификация ключей, хранение и CRL.

Сертификаты

Сертификат PEM представляет собой надмножество X.509, которое определяет процедуры и соглашения для инфраструктуры управления ключами, используемой с PEM, а в будущем и с другими протоколами (в том числе стеками TCP/IP и OSI).

Инфраструктура управления ключами использует общий корень для всей сертификации Internet. Центр регистрационной политики (Internet Policy Registration Authority, IPRA) определяет глобальную стратегию для всей иерархии. Ниже корня IPRA находятся центры сертификационной политики (Policy Certification Authorities, PCA), каждый из которых определяет и опубликовывает свою стратегию регистрации пользователей и организаций. Каждый PCA сертифицирован IPRA. Следом за PCA идут CA, сертифицирующие пользователей и управляющие организационными подразделениями (департаментами, офисами, дочерними компаниями). Первоначально предполагалось, что большинство пользователей будет регистрироваться в качестве членов организаций.

Как ожидается, ряд PCA обеспечит сертификацию пользователей, не входящих ни в одну организацию. Предполагается выделить один или несколько PCA для регистрации клиентов, желающих воспользоваться преимуществами секретности PEM и сохранить анонимность. Стратегия этих PCA позволит регистрировать пользователей, которые не хотят обнародовать конфиденциальные сведения.

Сообщения PEM

Сердцем PEM является специальный формат сообщений. В примере 1 приводится общий вид зашифрованного сообщения при симметричном управлении ключами. В примере 2 – общий вид подписанного и зашифрованного сообщения при управлении ключами на базе открытых ключей. В примере 3 показано подписанное (но незашифрованное) сообщение при управлении ключами на базе открытых ключей.

Пример 1. Встроенное сообщение (симметричный случай)

```
-----BEGIN PRIVACY-ENHANCED MESSAGE-----
  Proc-Type: 4,ENCRYPTED
  Content-Domain: RFC822
  DEK-Info: DES-CBC,F8143EDE5960C597
  Originator-ID-Symmetric: any@icn.com,,
  Recipient-ID-Symmetric: to@chinet.com,ptf-kmc,3
  Key-Info :
DES-ECB,RSA-MD2,9FD3AAD2F2691B9A,B70665BB9BF7CBCDA60195DB94F727D3
```

```
  Recipient-ID-Symmetric: penl-dev@tis.com,ptf-kmc,4
  Key-Info :
DES-ECB,RSA-MD2,161A3F75DC82EF26,E2EF532C65CBCFF79F83A2658132DB47
LLrHBOeJzyhP+/fSStdH8okeEnv47jxe7SJ/iN72ohNcUk2jHEUSoHlnvNSIHE9M
8tEjmF/zxB+bATMtPjCUHbz8Er9wloxIkjHUlBEpvXROUrUzYbkNpkOagV2IzUpk
J6UiRRGcDSvzrsoK+oNvqu6z7Xs5Xfz5rDqUcMlKlZ6720dcBHGGsDLpTpSCnpot
dXd/H5LMDHnonNvPCmQUHt==
-----END PRIVACY ENHANCED MESSAGE-----
```

Пример 2. Встроенное шифрованное (encrypted) сообщение на базе открытых ключей (асимметричный случай)

```
-----BEGIN PRIVACY-ENHANCED MESSAGE-----
  Proc-Type: 4,ENCRYPTED
  ContentDomain: RFC822
  DEK-Info: DESCBC,BFF968AA74691AC1
  Originator-Certificate :
MIIBlTCCAScCAWuwDQYJKoZIhvcNAQECBQAwUTELMAkGAlUEBhMCVVMxIDAeBgNV
BAoTFIJTQSBEYXRhIFNIY3VyaXR5LCBJbmMuMQ8wDQYDVQQLEmZCZXRhIDExDzAN
BgNVBAsTBk5PVEESWTAeFw05MTA5MDQxODM4MTdaFm05MzA5MDMxODM4MTZaMEUx
CzAJBgNVBAYTAlVTMSAmHgYDVQQKExdSUOEgRGFOYSBTZWNlcmlOeSwgSW5jLjEU
MBIGAIUEAxMLVGVzdCBVc2VyIDEmHTAKBgRVCAEBAglCAANLADBIAkEAwHZHI7i+
yJcqDtjJComzTdBJrdAiLAnSC+CnnjOJEEyuQiBgkGrgIh3j8/xOfM+YrsyFlu3F
LZPVtzlrldhYFJQIDAQABMAOGCSqGSIb3DQEBAgUAAIkACKrOPqphJYwlj+YPtc
iWlFPuN5jJ79Khfg7ASFxskYkEMjRNZV/HZDZQEhtVaU7Jxfzs2mfX5byMp2X3U/
5XIJXGx7qlJsDgHQGs7Jk9H8CHlfuSHUgN4w==
  Key-Info:RSA,
I3rRIGXUGWAF8js5wCzRTkdh034PTHdRZY9Tuvm03M+NM7fx6qc5uIxps2LrlgO+
  wGrtiUm/ovtKdlnzeZQ/aQ==
  Issuer-Certificate:
MIIB3DCCAUgCAQowDQYJKoZIhvcNAQECBQAwTzELMAkGAIUEBhMCVVMxIDAeBgNV
BAoTFIJTQSBEYXRhIFNIY3VyaXR5LCBJbmMuMQ8mDQYDVQQEEwZCZXRhIDExDTAE
BgNVBAsTBFRMQOEwHhcNOTEmOTAxMDgwMDAwWhcNOTlwOTAxMDclOTU5HjBRMQsw
CQYDVQQGEwJVUzEgMB4GAIUEChMXUINBIERhdGEgU2VjdXJpdHksIEIuYy4xDzAN
BgNVBAsTBkJIdGEgMTEPMAOGAIUECxMGTk9UQVJZMHAmCgYEVQgBAQICArmDYgAm
XwJYCsnp61QCxYykNIODwutF7jMJ3kE+3PjYyHOwk+79rLg6X65B/ED4bJHt05XH
cqAz/7R7XhjYCmOPcqbdzoACZtIIETrKrcJiDYoP+DkZ8klgCk7hQHpbImIDAQAB
MAOGCSqGSIb3DQEBAgUAA38AAICPv4f9Gx/tY4+p+4DB7MV+tKZnvBoy8zgoMGOx
dD2jMZ/3HsyWKHgSFOeH7AJB3qr9zosG47pyMnTf3aSy2nBO7CMxpUHRBcXUpE+x
EREZd9++32ofGBIXaIaInOgVUnOOzSYgugiQO77nJLDUjOhQehCizEs5wUJ35a5h
  MIC-Info:RSA-MD5,RSA,
UdFJR8u/TIGhfH651eeme210H4tooa3vZCvVNGBZirf/7nrgzHDABz8m9NsXSexv
  AjRFbHoNPzBuxwnlOAFeAOHJszE4yBvhG
  Recipient-ID-Asymmetric :
```

```
MFExCzAJBgNVBAYTAIVTMSAmHgYDVQQKExdSUOEgRGFOYSBTZHNlcmIOeSwgSH5j
LjEPMAOGAIUECxMGQftiVOYSAxMQ81tfDQYDVQQLEwZOTIRBUIk=,
  66
  Key-Info:RSA,
06BSIww9CTyHPtS3bMLD+EOhejdvX6QvlHK2ds2sQPEaXhX8EhvVphHYTjmekdHv
7xOZ3Jx2vTAhOYHMcqqCjA==
qeWlj/YJ2Uf5ng9yznPbtDOmYloSwIuV9FRYx+gzY+81Xd/NQrXHfi6/MhPfPF3d
jIqCJAxvld2xgqQimUzoSla4r7kQQ5c/Iua4LqKeq3clFzEv7MbZhA==
-----END PRIVACY ENHANCED MESSAGE-----
```

Первым полем является **Proc-Type**, идентификатор типа обработки, которой подверглось сообщение. Существует три возможных типа сообщений. Спецификатор **ENCRYPTED** обозначает, что сообщение зашифровано и подписано. Спецификатор **MIC-ONLY** и **MIC-CLEAR** указывают, что сообщение подписано, но не зашифровано. Сообщения **MIC-CLEAR** не кодируются и могут быть прочитаны с помощью другого, не входящего в PEM программного обеспечения. Для преобразования сообщений **MIC-ONLY** в удобочитаемую форму необходимо программное обеспечение PEM. Сообщение PEM подписывается всегда, а шифрование не является обязательным.

Следующее поле, **Content-Domain**, задает тип почтового сообщения. Оно не влияет на безопасность. Поле **DEK-Info** содержит информацию о *ключе обмена данными* (Data Exchange Key, DEK), алгоритме, используемом для шифрования текста, и параметрах, связанных с алгоритмом шифрования. В настоящее время определен единственный алгоритм, им является DES в режиме CBC (DES-CBC) Второе подполе содержит **IV**. В будущем для PEM могут быть определены и другие алгоритмы, их использование планируется запротоколировать в поле **DEK-Info** и других полях, определяющих алгоритм.

В сообщениях с симметричным управлением ключами следующим полем будет **Originator-ID-Symmetric** с тремя подполями. Первое подполе с помощью уникального адреса электронной почты определяет отправителя. Второе поле не является обязательным и определяет орган, выдавший заменяемый ключ. Третьим является необязательное подполе **Версия/Окончание срока**.

Далее, при использовании симметричного управления ключами, у каждого получателя есть два поля **Recipient-ID-Symmetric** и **Key-Info**. Поле **Recipient-ID-Symmetric** содержит три подполя, которые определяют получателя так же, как подполя поля **Originator-ID-Symmetric** определяют отправителя.

Поле **Key-Info** задает параметры управления ключами. У этого поля четыре подполя. Первое определяет алгоритм, использованный для шифрования DEK. Так как в рассматриваемом сообщении применяется симметричное управление ключами, то отправитель и получатель используют общий ключ. Он называется *заменяемым ключом* (Interchange Key, IK) и используется для шифрования DEK, который может быть зашифрован либо с помощью DES в режиме ECB (этот способ обозначается DES-ECB), либо тройным DES (DES-EDE). Второе подполе определяет алгоритм MIC. Может использоваться MD2 (обозначается RSA-MD2) или MD5 (RSA-MD5). Третье подполе, DEK, и четвертое подполе, MIC, шифруются с помощью IK.

В примере 3 представлено сообщение, в котором используется управление ключами с помощью открытых ключей (в перечне PEM такой способ называется асимметричным). В сообщениях **ENCRYPTED** после поля **DEK-Info** идет поле **Originator-Certificate**. Форма сертификата соответствует стандарту X.509. Следующим полем является **Key-Info** с двумя подполями. Первое подполе определяет алгоритм с открытым ключом, использованный для шифрования DEK, в настоящее время поддерживается только RSA. Следующее подполе – DEK, зашифрованное открытым ключом отправителя. Это необязательное поле, которое позволяет отправителю расшифровать свое собственное сообщение, возвращенное почтовой системой. Следующим полем является **Issuer-Certificate** – сертификат организации, подписавшей сертификат отправителя (**Originator-Certificate**).

Далее при асимметричном управлении ключами следует поле **MIC-Info**. Первое подполе задает алгоритм вычисления MIC, а второе – алгоритм, использованный для подписи MIC. Третье подполе содержит MIC, подписанный закрытым ключом отправителя.

Пример 3. Встроенное (незашифрованное) MIC-ONLY-сообщение (асимметричный случай)

```
-----BEGIN PRIVACY-ENHANCED MESSAGE-----
  Proc-Type: 4,MIC-ONLY
  Content-Domain: RFC822
  Originator-Certificate :
MIIBITCCAScCAHUwDQYJKoZIhvcNAQECBQAwTzELMAkGAIUEBhMCVVMxlDAeBgNV
BAoTFIJTQSBEYXRhIFNIY3VyaXR5LCBJbmMuMQ8wDQYDVQQLEwZCZXRhIDExDzAN
BgNVBAsTBk5PVEFSHTAeEw05MTA5MDQxODM4MTdaFw05MzA5MDMxODM4MTZaMEUx
CzAJBgNVBAYTAIVTMSAwHgYDVQQKExdSUOEgRGFOYSBTZMNlcmlOeSwgSW5jLjEU
MEIGAIUEAxMLVGVzdCBVc2VylDEwHTAKBgRVCAEBAgICAANLADBIAkEAmHZHI71+
```

```
yJcqDtjJCowzTdBJrdAiLAnSC+CnnjOJELyuQiBgkGrgIh3j8/xOfM+YrsyFlu3F
LZPVtzlndhYFJQIDAQABMAOGCSqGSIb3DQEBAgUAAIkACKrOPqphJYwlj+YPtcIq
ItJIFPuN5jJ79Khfg7ASFxskYkEMjRNZV/HZDZQEhtVaU7Jxfzs2mfX5bMp2X3U/
5XUXGx7qusDgHQGs7Jk9W8CW1fuSI4UgN4w==
  Issuer-Certificate:
MIIB3DCCAUgCAQowDQYJKoZIhvcNAQECBQAkTzELMAkGAIUEBhMCVVMxlDAeBgNV
BAoTFIJTQSBEYXRhIFNIY3VyaXR5LCBJbmMuMQ8wDQYDVQQLEIZCZXRhIDExDTAE
BgNVBAsTBFRMQOEwHhcNOTEwOTAxMDgwMDAmkmcNOTImOTAxMDclOTU5HjBRMQsw
CQYDVQQGEwJVUzEgMB4GAIUEChMXUINBIERhdGEgU2VjdXJpdHksIEIuYywxDzAN
BgNVBAsTBkJIdGEgMTEPMAOGAIUECxMGTk9UQVJZMHAwCgYEVQgBAQICArwDYgAw
XwJYCsnpelQCxYkNIODwlltF/jMJ3kE+3PjYjHOwk+/9rEg6X65B7LD4bJHt05XH
cqAz77R7XhjYCmOPcqbdzoACZIlETrKrcJIDYoP+DkZ8klgCk7hQHpbltflDAQAB
MAOGCSqGSIb3DQEBAgUAA38AAICPv4f9Gx7tY4+p+4DB7MV+tKZrlvBo8zgoMGOx
dD2jMZ73Hs*kl4gSFOeH7AJB3qr9zosG47pyMnTf3aS2nBO7CMxpUkfRBcXUpE+x
EREZd9++32otGBIXaIalnOgVUnOOzSYgljglQO77nJEDUOhQehCizEs5mUJ35a5h
  MIC-Info:RSA-MD5,RSA,
jV20fH+nnXHU8bnE8kPAad7mSQITDZIbVuxvZAOVRZ5q5+EjI5bQvqNeqOUNQjr6
  EtE7K2QDeVMCj/XsdJIA8fA==
LSBBIGIIc3NhZ2UgZrti9lHVzZSBpbBOZXNOaH5nLgOKESBGb2xsb3dpbmcgaXMg
YSBibGFuaj/Bsakf51OgOKDQpUaGIzIGIzIHRoZSBIbrnQuDQo=
-----END PRIVACY-ENHANCED MESSAGE-----
```

Следующие поля связаны с получателями. Каждому получателю соответствуют два поля – **Recipient-ID-Asymmetric** и **Key-Info**. У поля **Recipient-ID-Asymmetric** два подполя. Первое определяет орган, выдавший открытый ключ получателя, а вторым является необязательное подполе **Версия/Окончание срока**. Поле **Key-Info** задает параметры управления ключами: первое подполе определяет алгоритм, использованный для шифрования сообщения, а вторым подполем служит DEK, зашифрованный открытым ключом получателя.

Безопасность PEM

Длина ключей RSA, используемых в PEM, может меняться в диапазоне от 508 до 1024 бит. Этого достаточно практически для любого уровня безопасности. Более вероятно, что вскрытие будет направлено против протоколов управления ключами. Для обеспечения конфиденциальности никогда не записывайте на бумажный носитель свой закрытый ключ! Процедуры сертификации ключей в PEM, если все пользователи строго следуют соответствующим указаниям, делают это невозможным но, как известно, люди часто неаккуратны.

Нарушитель может поступить хитрее и модифицировать реализацию PEM, работающую в вашей системе. Эта измененная версия может тайком пересылать нарушителю всю вашу почту, зашифровав ее открытым ключом;

ему может быть послана даже копия вашего закрытого ключа. Если измененная реализация будет работать хорошо, то вы никогда не узнаете, что случилось.

Реального способа предотвратить такое вскрытие не существует. Вы можете использовать однонаправленную хэш-функцию и получить контрольную сумму исполняемого кода PEM. Затем, при каждом запуске программного обеспечения, можно проверять контрольную сумму, чтобы вовремя обнаружить изменения. Но нарушитель точно так же может изменить и код контрольной суммы при изменении кода PEM. Если у атакующего есть доступ к вашему компьютеру, он может разрушить безопасность PEM.

Вывод заключается в том, что нельзя доверять никакому элементу ПО, если вы не доверяете всей аппаратуре, на которой оно работает. Для многих такие опасения покажутся необоснованными, но для некоторых пользователей эта угроза вполне реальна.

Разновидности PEM

Средство TIS/PEM

Доверенные информационные системы (Trusted Information Systems, TIS), частично поддерживаемые Управлением по передовым научным проектам правительства Соединенных Штатов, являются особой, весьма специфической реализацией PEM (TIS/PEM). Они были разработаны для платформ UNIX, затем их также перенесли на VMS, DOS и Windows.

Хотя спецификации почты с повышенной секретностью определяются для Internet одним главным сертификационным центром, TIS/PEM поддерживает существование нескольких иерархий сертификации. Для того чтобы пользоваться услугами TIS/PEM, узлу не нужно присоединяться к иерархии Internet.

Программа RIPEM

RIPEM – это программа Марка Риордана (Mark Riordan), реализующая протоколы PEM. Свободный доступ к этой программе закрыт, однако для частного, некоммерческого применения ей можно воспользоваться бесплатно. Лицензия на право работать с этой программой входит в документацию.

Код не может быть экспортирован. Конечно, законы правительства США не действуют за пределами Соединенных Штатов, и кое-кто из пользователей игнорирует экспортные ограничения. Код RIPEM доступен по всему миру на электронных досках объявлений. Для экспорта разрешена версия, называемая RIPEM/SIC, реализующая только цифровые подписи.

Протокол MSP

Протокол безопасности сообщений (Message Security Protocol, MSP) – это военный аналог PEM. Он был создан NSA в конце 80-х годов во время работы над программой создания безопасной системы передачи данных по сети (Secure Data Network System, SDNS-program). Это совместимый с X.400 протокол уровня приложения для закрытия электронной почты. MSP планируется использовать в разрабатываемой сети оборонных сообщений (Defense Message System, DMS).

Предварительный протокол безопасности сообщений (Preliminary Message Security Protocol, PMSP), который предполагается использовать для «несекретных, но важных» сообщений, представляет собой адаптированную для применения с X.400 и TCP/IP версию MSP. Этот протокол также называют Mosaic.

Как и PEM, программные реализации MSP и PMSP достаточно гибки, их конструкция позволяет подстроиться под использование различных алгоритмов для осуществления функций безопасности, таких как подпись, хэширование и шифрование.

Программа электронной почты Pretty Good Privacy

Pretty Good Privacy (PGP) – это свободно распространяемая программа безопасной электронной почты, разработанная Филипом Циммерманном (Philip Zimmermann). Для шифрования данных она использует IDEA, для управления ключами и цифровой подписи – RSA (длина ключа до 2047 бит), а для однонаправленного хэширования – MD5.

Для получения случайных открытых ключей PGP использует вероятностную проверку чисел на простоту, используя для получения стартовых последовательностей интервалы между нажатиями клавиш на клавиатуре. PGP генерирует случайные ключи IDEA с помощью метода, описанного в ANSI X9.17 (приложение C), используя IDEA вместо DES. PGP также шифрует закрытый ключ пользователя с помощью хэшированной парольной фразы, а не пароля непосредственно.

Сообщения, зашифрованные PGP, имеют несколько уровней безопасности. Единственная вещь, известная криптоаналитику о зашифрованном сообщении, – это получатель сообщения и только при том условии, что криптоаналитику известен ID ключа получателя. Лишь расшифровав сообщение, получатель узнает, кем оно подписано, если оно подписано. Это качество резко отличает сообщения почты PGP от сообщений PEM, в заголовках которых немало информации об отправителе, получателе и самом сообщении в незашифрованном виде.

Самой интересной особенностью PGP является распределенный подход к управлению ключами. В данном продукте не предусмотрено наличие центров сертификации открытых ключей, вместо этого в PGP поддерживается «сеть доверия». Каждый пользователь сам создает и распространяет свой открытый ключ. Пользователи подписывают ключи друг друга, создавая взаимосвязанное сообщество пользователей PGP.

Например, Алиса может каким-то физическим способом передать Бобу свой открытый ключ. Боб лично знает Алису, поэтому он подписывает ее открытый ключ. Одну подписанную копию он возвращает Алисе, а другую оставляет. Когда Алисе нужно связаться с Кэрол, она посылает Кэрол подписанную Бобом копию ключа. Кэрол, которая каким-то образом уже получила ключ Боба и доверяет ему заверить ключ другого человека, проверяет его подпись под ключом Алисы и убеждается, что она правильна. Таким образом, Боб знакомит Алису и Кэрол.

PGP не определяет стратегию установки доверительных связей, пользователи сами решают, кому верить, а кому нет. PGP обеспечивает механизмы для поддержки ассоциативного доверия открытым ключам и для использования доверия. Каждый пользователь хранит набор подписанных открытых ключей в виде файла *кольца открытых ключей* (public-key ring). Каждый ключ кольца обладает полем законности ключа, определяющим уровень доверия к ключу конкретного пользователя. Чем выше уровень доверия, тем больше пользователь уверен в законности ключа. Поле доверия к подписи измеряет, насколько пользователь верит тому, кто подписал открытые ключи других пользователей. Другими словами, поле доверия к владельцу ключа задает уровень, определяющий, насколько конкретный пользователь верит его владельцу, подписавшему другие открытые ключи. Это поле вручную устанавливается пользователем. PGP непрерывно обновляет эти поля по мере появления новой информации.

На рис. 3.26 показано, как выглядит эта модель для конкретного пользователя, например Алисы. Ее ключ находится в самом верху иерархии, это означает, что владелец ключа абсолютно надежен. Алиса подписывает ключи Боба, Кэрол, Дейва, Элен и Фрэнка. Она доверяет Бобу и Кэрол подписывать открытые ключи других участников, кроме того, она частично доверяет Дейву и Элен подписывать открытые ключи других пользователей. Наконец, она доверяет Гейл подписывать открытые ключи других людей, хотя сама не подписывала ключ Гейл.

Двух частично доверяемых подписей может оказаться достаточно для сертификации ключа. Алиса считает, что ключ Курта законен, так как Дейв и Элен подписали его. Уровень доверия устанавливается в PGP вручную.

Алиса не должна автоматически доверять ключам других людей только потому, что они подписаны ключом, который она считает правильным.

Ключ Оуэна вообще не входит в сеть. Вполне возможен случай, что Алиса получила его от сервера. Так как PGP не считает ключ автоматически правильным, Алиса должна либо объявить о правильности ключа, либо решиться поверить одному из тех, кто подписал ключ. Конечно, ничто не мешает Алисе использовать ключи, которым она не доверяет. Задача PGP не в том, чтобы помешать ей установить соединения, а в том, чтобы информировать Алису о появлении подозрительного ключа.

Самым слабым звеном этой системы является отзыв ключей: гарантировать, что кто-нибудь не воспользуется скомпрометированным ключом, невозможно. Если закрытый ключ Алисы украден, она может послать некий *сертификат отзыва ключа* (key revocation certificate), но, так как распределение ключей уже произошло, нельзя гарантировать, что это сообщение

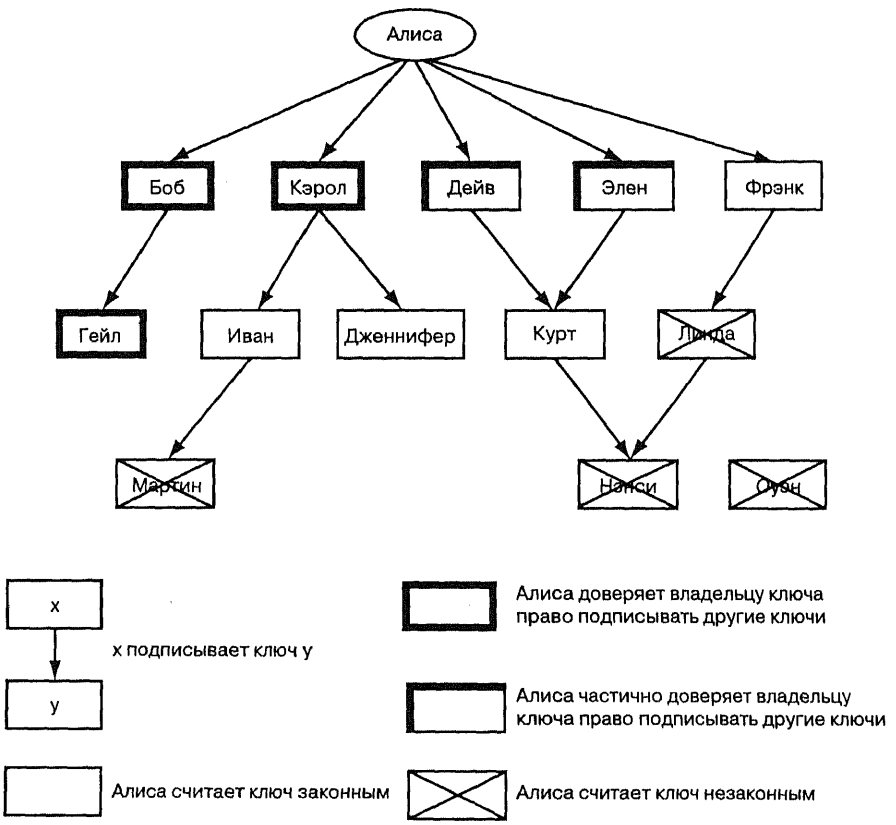

Рис. 3.26. Модель доверительных отношений в PGP

будет получено всеми, кто использует ее открытый ключ в своем кольце ключей. И так как Алиса должна будет подписать свой сертификат отзыва ключа своим закрытым ключом, то потеряв ключ, она не сможет его отозвать.

В PGP версии 3.0 включены опции тройного DES, SHA, другие алгоритмы с открытыми ключами, разделение пар открытый ключ/закрытый ключ для шифрования и для подписи, расширенные процедуры отзыва ключей, улучшенные функции управления кольцом ключей, API для интегрирования PGP в другие программы и полностью переписанные исполняемые модули.

PGP доступна для MS DOS, UNIX, Macintosh, Amiga и Atari. В личных некоммерческих целях ее можно использовать свободно, переписав со многих узлов ftp в Internet. Чтобы скопировать PGP с узла MIT с помощью telnet, подключитесь к net-dist.mit.edu, войдите в систему как getpgp, ответьте на вопросы, затем используйте ftp для соединения с net-dist.mit.edu и перейдите в каталог, указанный в сессии telnet. Эту программу также можно получить на ftp.ox.ac.uk, ftp.dsi.unimi.it, ftp.funet.fi, ftp.demon.co.uk, CompuServe, AOL и др.

3.7.3. Защита в архитектуре X.400

Архитектура X.400

Рекомендации X.400 призваны обеспечить стандартизацию обработки сообщений. Передача сообщений по электронной почте состоит из пересылки и хранения сообщений. Главными «участниками» обмена сообщениями являются:

- система обработки сообщений (Message Handling System – MHS);
- пользователи;
- списки рассылки.

Система обработки сообщений

Согласно стандарту X.400 (см. рис. 3.27) системой обработки сообщений (Message Handling System, MHS) является любая система, которая позволяет вычислительным машинам обмениваться сообщениями. Она, в свою очередь, состоит из нескольких логических элементов:

- системы передачи сообщений (Message Transfer System – MTS);
- агентов пользователя (User Agents – UAs);
- агентов удаленного пользователя (Remote User Agents – RUAs) и хранилища сообщений (Message Store – MS);
- модулей доступа (Access Units – AUs) и шлюзов X.400.

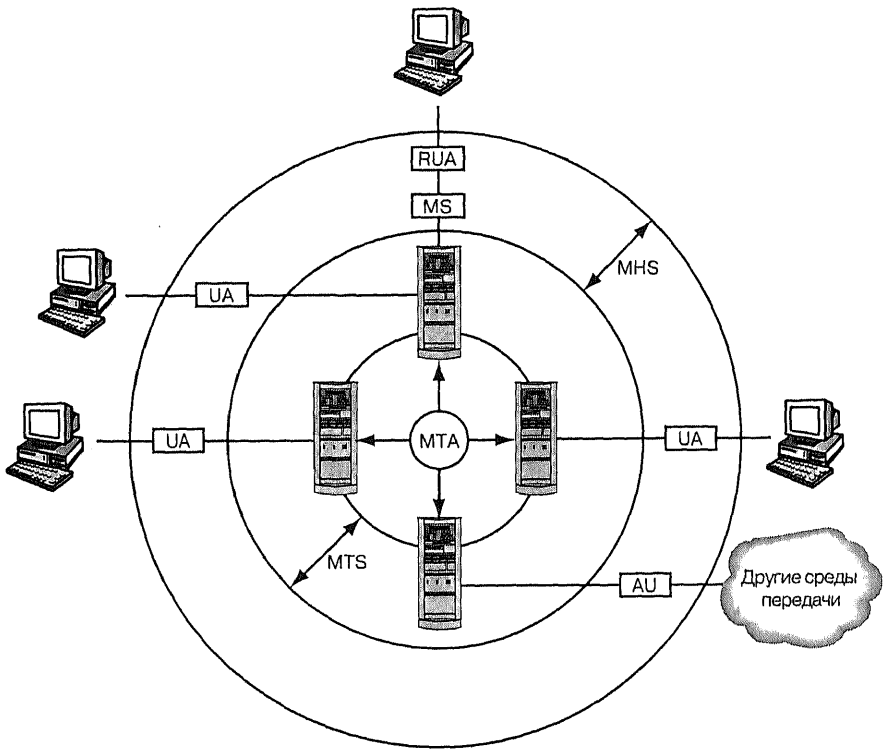

Рис. 3.27. Архитектура X.400

Система передачи сообщений

Система передачи сообщений состоит из агентов передачи сообщений (Message Transfer Agents, MTA) и обеспечивает средства взаимодействия между агентами пользователя (UA) для обмена сообщениями. Система передачи сообщений (Message Transfer System, MTS) отвечает за пересылку сообщений до конечного пункта назначения.

Агенты пользователя и агенты передачи сообщений

При составлении, отправке и получении писем пользователь взаимодействует с системой обработки сообщений (MHS) через своего агента пользователя (User Agents, UA). Агенты пользователя обеспечивают возможность редактирования, временного хранения, отправки и приема сообщений. Готовое к отправке письмо пересылается агенту передачи сообщений (Message Transfer Agents, MTA). MTA, так же как и обычное почтовое отделение, отвечает за сортировку и доставку электронных писем конечным получателям или пересылку на другие почтовые отделения. Она включает

в себя подсистему транспортировки сообщений, которая определяет маршрут движения сообщения. Чтобы прочитать доставленное письмо, пользователь обращается к своему UA, который, в свою очередь, получает письма от MTA. Каждый пользователь взаимодействует только с одним агентом пользователя (UA), и каждый агент пользователя (UA) взаимодействует только с одним агентом передачи сообщений (MTA).

UA обычно представляет собой приложение, запускаемое пользователем. MTA – более универсальная программа, отслеживающая интересы всех своих пользователей. MTA должна иметь доступ ко всем сообщениям своих локальных пользователей, иметь средства хранения сообщений до их доставки пользователю и средства передачи сообщений по истечении срока хранения. Кроме того, для доставки сообщений MTA должна обладать доступом к описаниям пользователей и основных соединений.

Агент удаленного пользователя и хранилище сообщений

Для удаленного пользователя процесс взаимодействия с системой обработки сообщений внешне выглядит так же, как и для локального пользователя. Рекомендации X.400 определяют эту процедуру через специализированного агента пользователя с промежуточным хранением писем.

Агент удаленного пользователя (Remote User Agent, RUA) можно рассматривать как клиентскую программу, взаимодействующую с сервером (хранилищем сообщений – Message Store, MS), который, в свою очередь, взаимодействует с агентом передачи сообщений (MTA). Хранилище сообщений (MS) размещается на том же компьютере, что и MTA, и работает только с удаленными пользователями.

MS обеспечивает хранение сообщений и передачу их по требованию RUA, представляя собой автоматическое, безопасное, постоянно доступное средство управления сообщениями удаленных пользователей.

Модули доступа и шлюзы X.400

Модули доступа (Access Units) и шлюзы предназначены для обеспечения взаимодействия с другими коммуникационными системами или другими системами электронной почты.

Модули доступа представляют собой интерфейс между MTS X.400 и другими средствами связи. Примером может служить факс-шлюз (FAX Access Unit – FAU) или шлюз с обычной почтой (Physical Delivery Access Unit – PDAU), позволяющий посылать сообщения с помощью средств факсимильной связи или средств физической доставки, а также телеграфных и телетайпных аппаратов.

Шлюзы можно рассматривать как трансляторы, принимающие сообщения от одной системы электронной почты и передающие и транслирующие их таким образом, чтобы они были понятны другой системе.

Почтовое сообщение (ПС) в системе X.400 состоит из конверта и содержимого. Конверт электронного письма содержит всю необходимую информацию для того, чтобы письмо дошло до адресата. Обычно он включает адреса получателей X.400, адрес отправителя X.400, тип содержания письма, уровень приоритета, в соответствии с которым письмо должно быть доставлено, уникальный идентификатор самого письма и сведения о маршруте, по которому оно шло от одного MTA к другому.

Рис. 3.28
Структура ПС X.400

В системе X.400 можно посылать текстовые, двоичные или специальным образом закодированные данные. Содержимое электронного сообщения имеет такую же структуру, как и обычное письмо:

- заголовок сообщения. В заголовке указываются имена или адреса получателей и отправителей, время отправки или получения письма, тема письма;
- тело сообщения. Тело сообщения содержит ту информацию, которая должна быть передана. Поскольку конверт запечатан и система передачи сообщений не знает содержимого письма, можно пересылать любые типы данных, например зашифрованный текст, исполняемый программный код, графику, звук. Тело письма может быть составным и иметь несколько вложений, которые могут быть скомпонованы различным образом.

Выше были описаны сообщения, которыми обмениваются пользователи системы обработки сообщений (MHS). В системе X.400 определены также сообщения, которые формируются внутри нее – извещения (квитанции) и пробные письма.

Квитанции используются для передачи информации о сообщениях, которые были посланы при помощи системы передачи сообщений: об успешной доставке, о недоставке, о прочтении и т.д.

Квитанции бывают двух типов:

- квитанции о доставке. Они формируются агентами передачи сообщений (MTA) в зависимости от того, было ли сообщение доставлено или нет. Например, если MTA не может доставить электронное письмо нужному адресату, он составляет квитанцию о невозможности доставки и посылает ее обратно отправителю;

- квитанции о состоянии. Их часто называют «извещениями о прочтении»; они информируют отправителя, действительно ли указанный адресат получил со своего MTA письмо и прочитал его.

Пробные письма – это текстовые сообщения, с помощью которых обмениваются MTA, чтобы проверить, могут ли сообщения быть отправлены назначенному MTA.

Система обработки сообщений Messenger 400

В качестве практической реализации X.400 рассмотрим систему Messenger 400, которая разработана фирмой Infonet Software Solutions (ISS) и является программной реализацией системы обработки сообщений X.400. Messenger 400 позволяет пользователям и приложениям обмениваться сообщениями с другими системами, поддерживающими стандарты X.400 1984, 1988 и 1992 годов по ряду транспортных протоколов, включая X.25, TCP/IP и асинхронные линии.

Messenger 400 является модульной реализацией MHS X.400 ITU-T.

Все продукты системы Messenger 400:

- основаны на стандартах;
- поддерживают различные платформы: Intel (Microsoft Windows NT), Intel (SCO UNIX), Sun SPARC (Solaris), Stratus FTX, Hewlett Packard (HP/UX), IBM RS/6000 (AIX), Tandem S Series with Tandem (Non-stop UX), DEC Alpha (DEC-UNIX);
- широко применяются и легко настраиваются.
- основаны на проверенной и надежной технологии;
- предоставляют расширенные возможности локального и удаленного управления;
- защищены.

В основе системы электронной почты Messenger 400 лежит хорошо отработанная технология хранения и пересылки информации. Система передачи данных Messenger 400 умеет идентифицировать пользователя и определять пути к нему, несмотря на различие почтовых систем и географическое положение. Развитая система шлюзов обеспечивает соединение с многочисленными и разнообразными коммуникационными системами в единую сеть. Шлюзы поддерживают все наиболее распространенные системы электронной почты, основанные на локальных сетях, и системы для рабочих групп, включая:

- cc:Mail;
- Microsoft Mail;

- NetWare Global MHS;
- Lotus Notes;
- Novell GroupWise.

Кроме того, шлюз SMTP/MIME предоставляет возможность обмениваться почтой с пользователями сети Internet и любой другой почтовой системы, поддерживающей протокол SMTP.

Принципиально систему Messenger 400 можно представить как набор рабочих агентов или почтовых приложений, которые приводятся в действие при поступлении сообщений в локальный почтовый ящик и последовательно выполняются до момента доставки сообщения к конечному пользователю. Функциональные возможности, обеспечиваемые рабочими агентами, позволяют рассматривать их как почтовые приложения. Messenger 400 предоставляет также заранее определенные для этих почтовых приложений специальные службы, такие как списки рассылки, запросы к службе каталогов и почтовые уведомления.

MTA в системе Messenger 400 представляет собой базу данных, которая должна содержать информацию обо всех пользователях и сведения о маршрутах передачи сообщений пользователям других MTA.

Почтовые ящики пользователей являются, по сути, файлами очереди. Каждый пользователь или удаленный MTA представлен в виде записи в базе данных MTA с соответствующим файлом очереди и соответствующим ПО агента пользователя. Как только MTA получил сообщение, предназначенное для отправки на другой MTA или другому пользователю, оно ставится в очередь. Это отмечается в файле очереди, следом вызывается соответствующее приложение по обработке помещенных в очередь сообщений.

Messenger 400 включает:

- сервис пользователя. Применяя соответствующие исполняемые модули, пользователь может создавать, отправлять, получать и управлять электронными письмами;
- маршрутизацию и передачу сообщений. Система обеспечивает хранение отправляемых и принимаемых сообщений, маршрутизацию и гарантированную доставку;
- службу каталогов. Служба каталогов запрашивает данные об организациях и пользователях в сетях Messenger 400, помогая правильно адресовать сообщения;
- почтовые приложения и шлюзы. Messenger 400 предлагает ряд заранее настроенных приложений, которые предоставляют пользователю дополнительные услуги, а также обеспечивает обмен сообщениями пользователей X.400 с пользователями других систем электронной

почты. Messenger 400 позволяет использовать дополнительные модули физической доставки;
- интегрирование и настройку приложений. Messenger 400 предлагает полный набор интерфейсов прикладного программирования (API), позволяющего для новых задач создавать почтовые и иные приложения, включая X/API в соответствии с рекомендациями X/OPEN;
- оптимизацию. Для оптимизации использования системы и повышения ее характеристик Messenger 400 содержит программу-планировщик. Определяя параметры планировщика, можно выбирать различные стратегии планирования для разных типов и приоритетов сообщений.

Messenger 400 позволяет пользователю взаимодействовать с локальными агентами (UA) и агентами удаленного пользователя (RUA). Агент локального пользователя, входящий в состав ПО Messenger 400, размещается на том же компьютере, что и агент передачи сообщений (MTA) Messenger 400. Удаленная версия агента пользователя, устанавливаемая отдельно от MTA на другом компьютере, предполагает ее совместную работу с хранилищем сообщений (MS). Администратор может настраивать базу данных хранилища сообщений и управлять ей. Хранилище сообщений может взаимодействовать с агентами удаленных пользователей различных производителей.

Локальный агент пользователя Messenger 400 предназначен для работы под управлением операционной системы UNIX.

Messenger 400 предоставляет заранее настроенные почтовые приложения, которые:

- уведомляют пользователя о приходе электронной почты;
- автоматически отправляют электронную почту в другой почтовый ящик;
- автоматически посылают пользователю ответ или принимают и сохраняют поступающую электронную почту;
- рассылают сообщения по спискам рассылки;
- дают возможность пользователям и системным интеграторам быстро создавать почтовые приложения.

Обеспечение информационной безопасности в X.400

Существуют два аспекта защиты при передаче и обработке ПС X.400: а) управление и администрирование системы защиты; б) защита обмена ПС. Обеспечение защиты в архитектуре X.400 включает в себя защиту ПС, непосредственно предоставленных MTS UA, MS и AU. В общем случае многие элементы службы защиты ПС обеспечивают возможность защиты в направлении *отправитель-получателю* и требуют использования UA.

При этом они не требуют использования MTS, обладающей возможностями защиты (например, секретность содержимого может обеспечиваться путем шифрования содержимого ПС отправителем и его расшифрования получателем с различными параметрами защиты, передаваемыми внутри конверта ПС). Такое сообщение может передаваться MTS, которая будет обрабатывать формат содержимого (неформатированные октеты) и поля защиты в конверте. Некоторые из элементов службы защиты ПС взаимодействуют с MTS и требуют применения MTA с возможностями защиты.

Защиту почтовой системы X.400 целесообразнее осуществлять, используя разработанную в МО ПНИЭИ почтовую систему. Преимущество подобного подхода заключается в использовании защищенной электронной почты, соответствующей стандарту X.400, основным достоинством которого является защита почтового сервера от несанкционированного доступа. Для предоставления абонентам услуг защищенной электронной почты в МО ПНИЭИ разработан электронный почтамт на базе программного продукта Messenger 400. У каждого пользователя электронной почты устанавливается ПО *абонентского пункта* (АП) защищенной электронной почты X.400, в котором криптографическая защита информации и процедура аутентификации выполнены на встроенном в АП ПО «Верба-О». Особенностью данного метода является обязательное наличие криптосервера и АП центра управления ключевой системой (ЦУКС), присоединяемых к электронному почтамту. Но для конкретной реализации данного метода требуется дополнительная проработка вопросов, связанных с согласованием перехода ИС ТКП на защищенную почту X.400. В данной книге рассмотрены типовые решения по построению защиты почтовых систем на базе X.400.

Архитектура построения защищенной почтовой системы с использованием разработки МО ПНИЭИ представлена на рис. 3.29.

Основными компонентами защищенной почтовой системы являются:

- защищенный абонентский пункт или защищенный АП. На нем как раз и реализуются все функции криптографической защиты информации, серверная часть использует только аутентификацию пользователей почтовой системы. Таким образом, защита реализована на уровне пользователей, то есть почтовые сообщения (ПС) на серверную часть приходят уже зашифрованными и подписанными, почтовый сервер обеспечивает лишь транспортные функции по доставке ПС;
- криптосервер, на который вынесены с UNIX-платформы электронного почтамта средства криптографической защиты информации, используемые для аутентификации;
- центр управления безопасностью (ЦУБ) – это выделенный АП, который позволяет рассылать административные письма-команды и на

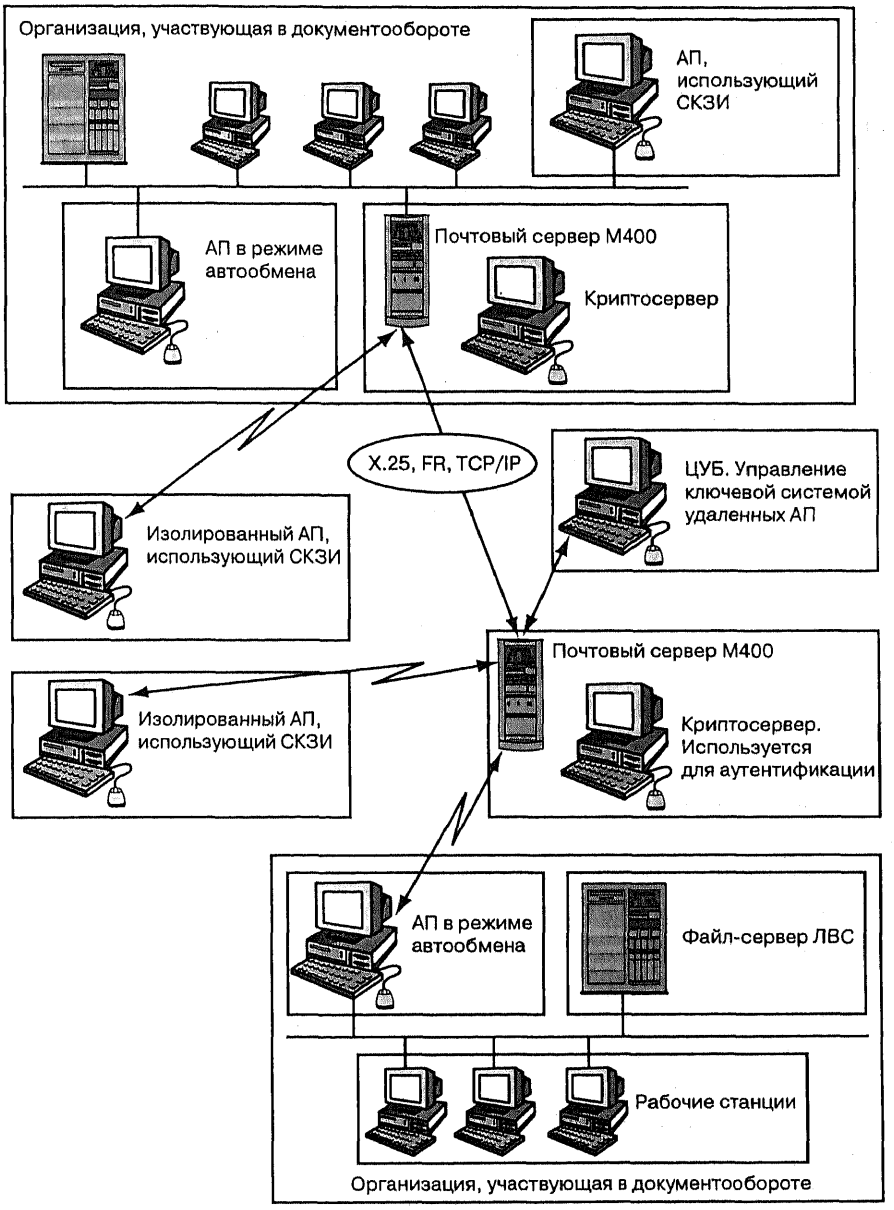

Рис. 3.29. Защищенная почтовая система

который возложены функции по управлению ключами пользователей. Оператор ЦУБ является администратором безопасности. Для пользователя ЦУБ эквивалентен защищенному АП;

- шлюз ELINK, используемый АП для связи с системой передачи сообщений Messenger 400. Шлюз ELINK обеспечивает передачу сообщений с абонентского пункта на почтовый сервер Messenger 400 по протоколам LAPS и ELINK и строгую аутентификацию абонентов на основе системы криптографической защиты информации. При подсоединении абонента к почтовому серверу производится аутентификация пользователя с помощью криптосервера. Может производится как односторонняя, так и двусторонняя аутентификация. Шлюз ELINK реализован в составе почтового сервера и представляет с одной стороны интерфейс обращения к криптосерверу, а с другой – к Messenger 400 (рис. 3.30).

Защищенный АП обеспечивает:

- обмен защищенной информацией через систему электронной почты Messenger 400 с использованием процедур абонентского шифрования электронной подписи сообщений;
- аутентификацию абонентов на почтовом сервере с целью предотвращения попыток НСД;
- обмен данными с почтамтом по протоколам SPX/IPX, TCP/IP, X.28/X.25, а также по специализированному помехоустойчивому протоколу LAPS;
- обмен частными сообщениями от абонента к абоненту с обеспечением сохранности частной почты;
- многоадресную доставку сообщений;
- передачу «электронных посылок», то есть присоединение документированного (текстового, бинарного) файла к сообщению. Файл затем будет отправлен вместе с сообщением получателю;
- выдачу извещений о доставке или невозможности доставки сообщений;
- прямой обмен защищенными сообщениями между двумя АП по коммутируемым телефонным линиям и аутентификацию абонентов при установлении соединения;

Рис. 3.30. Взаимодействие компонентов защищенной почтовой системы

- защиту от НСД с регистрацией попыток НСД, реализованной на базе системы «Аккорд»;
- регистрацию протокола сеанса связи и ведение журналов входящих и исходящих сообщений на НЖМД с возможностью вывода данной

информации на печать за любой период времени (в течение установленного срока хранения);
- хранение передаваемых и принятых сообщений в специальных папках данных;
- обмен факсимильными, голосовыми и видеосообщениями (в соответствующей комплектации).

Защита информации и процедура аутентификация основаны на использовании встроенного в АП ПО СКЗИ «Верба». При этом обеспечивается:

- совместная работа с криптографическим сервером;
- шифрование (зашифрование и расшифрование) данных и файлов по ГОСТ 28147-89;
- контроль целостности данных от случайных или преднамеренных изменений посредством вычисления имитовставки по ГОСТ 28147-89;
- вычисление значения хэш-функции произвольных блоков данных по ГОСТ Р 34.11-94;
- вычисление и проверка значения ЭЦП по ГОСТ Р 34.10-94.

Программное обеспечение (ПО) абонентского пункта (АП) разработано в соответствии с рекомендацией МККТТ X.420 (межперсональные сообщения) и функционирует под управлением операционной системы MS DOS 5.0 и выше.

АП подключается к линиям и каналам связи по физическому стыку типа RS-232 (протокол X.3, X.28) с использованием асинхронных модемов, соответствующих рекомендациям V.22, V.22bis, V.32, V.32bis со встроенными протоколами коррекции ошибок типа MNP-5, V.42, V.42bis.

3.8. Корпоративные системы и опыт обеспечения информационной безопасности в них

В этом разделе рассмотрены корпоративные решения по построению информационно-вычислительных систем. В качестве примеров представлены разработки SWIFT, SmartCity и UEPS. Здесь же обобщен практический опыт по их использованию и обеспечению информационной безопасности в сложных системах.

3.8.1. Система S.W.I.F.T.

Наиболее известной из всех ныне существующих международных платежных систем является S.W.I.F.T., получившая широкое распространение в сфере международных межбанковских расчетов. Кроме того, система

S.W.I.F.T. может применяться для обмена информацией и осуществления взаиморасчетов при операциях с ценными бумагами и дорожными чеками. В перспективе предполагается использование данной технологии и в других сферах экономики, где необходима оперативная, качественная среда для передачи финансово значимой информации, требующая высокого уровня обеспечения конфиденциальности.

С технической точки зрения S.W.I.F.T. представляет собой международную телекоммуникационную сеть, позволяющую финансовым организациям из разных стран, используя компьютеры и терминалы различных типов, подключиться к ней для передачи банковской и финансовой информации. В системе принят особый формат банковских сообщений – стандарт, который развивается с помощью рабочей группы специалистов банков и организации S.W.I.F.T. В системе S.W.I.F.T. используются как международные стандарты, разработанные ISO, так и стандарты Международной торговой палаты (ICC). В результате развития S.W.I.F.T. образовалась новая сеть – S.W.I.F.T. II, которая базируется на четырехуровневой сетевой архитектуре и на системе управления процессорами, находящимися в операционных центрах S.W.I.F.T.

Логическая архитектура системы S.W.I.F.T. II подчиняется основным принципам, установленным ISO (Международная организация стандартизации) для взаимодействия открытых систем. Каждый активный компонент архитектуры S.W.I.F.T. II называется узлом. Узлы могут быть связаны между собой:

- прямыми выделенными линиями;
- местными (международными) коммутируемыми линиями;
- локальными сетями;
- спутниковыми каналами связи.

Архитектура системы состоит из четырех основных компонентов:

- процессор управления системой (SCP);
- коммутационный процессор (SP);
- региональный процессор (RP);
- процессор передачи (CP).

Фактически вся система S.W.I.F.T. II сосредоточена в двух центрах управления системой (SCC), которые расположены в Зетервуде недалеко от Лейдена (Нидерланды) и в Калпепере (США). SCC состоит из двух ключевых компонентов системы, а именно: SCP и SP. Для улучшения работоспособности и защиты от сбоев в системе S.W.I.F.T. II применяется дублирование

каждого SCP и резервирование работы каждого SP. В любое время только один SCP является активным и осуществляет непосредственное управление системой. Остальные три SCP постоянно находятся в резерве и непрерывно обновляют свое состояние по данным конфигурации активного SCP.

Процессор управления системой SCP отвечает за функционирование всей системы в целом. Он постоянно контролирует и управляет всеми активными компонентами системы, так же как и всем доступом к системе в целом. К функциям управления SCP относятся:

- разрешение открытия нового сеанса и хранение данных сеанса;
- распространение нового программного обеспечения по системе;
- контроль всех технических и программных средств;
- сбор диагностической информации о неисправностях;
- управление процессом восстановления после ошибки;
- динамическое распределение системных ресурсов.

Коммутационные процессоры SP управляют маршрутизацией и хранением сообщений. Основные функции SP:

- маршрутизация сообщений между пользователями через RP;
- надежное хранение двух копий всех обработанных данным SP сообщений (на двух разных носителях) и соответствующей им предыстории доставки;
- формирование подтверждений о хранении, доставке обработанных данным SP сообщений или их недоставки;
- обработка выборки сообщений.

Региональный процессор RP осуществляет логическое подключение пользователей к сети S.W.I.F.T. II и, по сути, является входной и выходной точкой системы. Программное обеспечение RP, взаимодействуя с программами пользователя, осуществляет точное и безопасное логическое подключение к S.W.I.F.T. II. В его функции входит:

- проверка входных сообщений до пересылки в SP;
- обработка протоколов прикладного уровня;
- контроль и проверка номеров входной последовательности (ISN) всех сообщений;
- верификация контрольных сумм сообщений;
- формирование положительных (ACK) и отрицательных (NAK) подтверждений приема сообщений.

Каждый RP обслуживает конкретную страну или территорию и расположен в безопасных (с контролем доступа) центрах. Для каждого

пользователя системы, известного по его физическому адресу, назначается основной RP, который и будет производить обслуживание данного пользователя.

Процессор передачи CP обеспечивает связь между RP и другими узлами системы, тем самым позволяя RP, подключенному к собственному SP, принимать информацию от других SP.

Для того чтобы получить физический доступ к системе S.W.I.F.T. II, индивидуальные пользователи должны иметь компьютерный терминал (CBT), который подключается к системе S.W.I.F.T. II через ряд местных узлов подключения, известных как точки доступа к S.W.I.F.T. II (SAP) или удаленные точки доступа (RAP). В состав SAP/RAP входят:

- процессор, выполняющий функции управления линиями пользователя и линиями подключения SAP/RAP к транспортной сети S.W.I.F.T. II (STN);
- порты, предоставляемые пользователям.

Доступ к услугам S.W.I.F.T. II через точки доступа (SAP) или удаленные точки доступа (RAP) обеспечивается с помощью транспортной сети STN, работающей под коммуникационным протоколом X.25. Различие между SAP и RAP заключается в обеспечении уровня безопасности, хотя они обладают одинаковыми операционными возможностями при работе с несколькими отдельно подключенными пользователями. Если из-за проблем на линии связи или неисправности точек доступа SAP (RAP) пользователь не может войти в систему в свою основную SAP (RAP), альтернативный вход производится в другой точке.

Подключение пользователей к сети S.W.I.F.T. II возможно по выделенным линиям связи, через *общие сети передачи данных* (PDN) или через PSTN (*коммутируемые линии*), подсоединенные к точке доступа.

Подключение выделенных линий возможно во всех SAP со скоростью передачи данных по линиям 2400, 4800 и 9600 бит/с. Для данного типа подключения характерно, что пользователю выделяется отдельный порт на точке доступа. Для данного типа подключения по желанию пользователя может использоваться шифрование.

Подключение через PDN возможно только со скоростями, эквивалентными скоростям выделенных линий. Подключение пользователя к PDN обеспечивается при помощи выделенных линий с применением протокола X.25. Для данного типа подключения предполагается обязательное шифрование данных согласно протоколу X.25.

В системе S.W.I.F.T. II имеется два типа подключения через коммутируемые линии (PSTN):

- через порты PSTN совместного использования, к которым все пользователи имеют доступ на основе строгой конкуренции. Скорость работы через эти порты не более 2400 бит/с, и средства шифрования не применяются;
- через выделенные порты (для каждого пользователя свой) со скоростью передачи данных до 9600 бит/с и возможностью (по желанию пользователя) применять средства шифрования информации.

Безопасность в системе S.W.I.F.T. II

Все вопросы, связанные с безопасностью в системе S.W.I.F.T. II, условно можно отнести к следующим разделам:

- физическая безопасность;
- безопасность логического доступа к системе S.W.I.F.T. II;
- обеспечение безопасности сообщений, передаваемых и хранящихся в системе;
- безопасность обмена сообщениями «пользователь-пользователь».

Средства безопасности, обеспечиваемые системой S.W.I.F.T. II, состоят из:

- процедуры входа в систему;
- процедуры выбора приложения;
- нумерации сообщений;
- проверки ошибок:
- криптозащиты, пока сообщение находится в части сети S.W.I.F.T. II;
- контроля доступа к сообщениям в SAP, региональных процессорах, коммутационных процессорах, центрах управления системой.

Отдел главного инспектора системы S.W.I.F.T. II (CIO) управляет всеми вопросами, связанными с безопасностью работы сети S.W.I.F.T. II. Пользователям рекомендуется обеспечивать надлежащую безопасность процедур, осуществляемых в их собственных организациях, например контроль доступа к терминалам S.W.I.F.T. II, управление их подключением и использованием.

Физическая безопасность

Осуществляется на основе разграничения и контроля доступа ко всем операционным и административным узлам S.W.I.F.T. II путем использования электронных средств и средств обнаружения несанкционированного доступа. Применяется также дистанционное управление для узлов S.W.I.F.T. II,

которые управляются автоматически. Если пользователь запрашивает центр о доступе к SAP, то в обязательном порядке должен быть сделан запрос к CIO и без его санкции никому не будет дано разрешение на доступ к SAP.

Безопасность логического доступа к системе S.W.I.F.T. II

Как уже говорилось выше, пользователи могут получить физический доступ к системе S.W.I.F.T. II только через CBT, работающее с одним или более логическим терминалом (LT). Каждому LT назначаются уникальные таблицы безопасности для процедур LOGIN и SELECT (выбор финансового приложения – FIN), которые представляют собой последовательности ключей в табличном виде. Каждый ключ в таблице может применяться только один раз и связан с последовательными номерами процедур, использующих эти ключи. Эти таблицы формируются и отсылаются пользователю до их подключения к системе S.W.I.F.T. II на основе запроса на их применение, причем новые таблицы безопасности создаются сразу после высылки очередных таблиц и пересылаются пользователю только по мере необходимости. Пользователи, активно применяющие таблицы безопасности, могут запросить в отделе главного инспектора таблицы с 2400 ключами вместо обычных таблиц (1200 ключей).

Доступ LT к системе S.W.I.F.T. II производится с помощью команды LOGIN. До того как будет послан запрос LOGIN, пользователю необходимо ввести ключ запроса и ключ ответа из таблицы безопасности LOGIN. Цель запроса LOGIN:

- определить логический путь для связи LT с системой;
- ограничить доступ в систему несанкционированных пользователей;
- позволить пользователям проверить, что они подключились к подлинной системе S.W.I.F.T. II;
- указать размер окна, которое должно быть открыто для сеанса GPA.

При запросе процедуры LOGIN система S.W.I.F.T. II производит следующие действия:

1. Проверяет заголовок и текст сообщения, отправляемого процедурой LOGIN.
2. Проверяет подлинность концевика MAC, сформированного с использованием ключа из таблицы безопасности, но не содержащего информацию о самом ключе.
3. Если подтверждение подлинности пользователя прошло успешно, порядковый номер запроса LOGIN (LSN) сравнивается с ожидаемым системой LSN. Если LSN находится в допустимом диапазоне, система проверяет, что запрос LOGIN выдан после дня, указанного в последней

команде LOGOUT (указывает временные рамки для LT, в течение которых от данного LT не будут приниматься запросы). Если же LSN не совпадает с ожидаемым, то в поле подтверждения подлинности системы указывается следующий ожидаемый LSN.

4. Подтверждает запрос LOGIN, возвращая либо положительное подтверждение LOGIN (LAK), либо отрицательное подтверждение LOGIN (LNK). Подтверждение будет содержать концевик MAC, основанный на ключе ответа, но не содержащий информацию о нем и позволяющий пользователю проверить подлинность системы.
5. Записывает попытку LOGIN вместе с ответом системы в предысторию LT.

Попытка произвести запрос LOGIN на линии связи, по которой LT уже вошел в систему, рассматривается как серьезная ошибка и игнорируется системой.

Доступ к приложению FIN (из которого отправляются сообщения «пользователь-пользователь» и ряд системных сообщений) производится с помощью команды SELECT, которая проходит процедуру подтверждения для гарантии того, что:

- только проверенные пользователи могут получить доступ к системе S.W.I.F.T. II;
- пользователь связался с подлинной системой S.W.I.F.T. II.

Алгоритм подтверждения подлинности сообщения формирует концевик MAC, используя при этом произвольный ключ защиты и ключ ответа, связанные последовательным номером с запросами LOGIN или SELECT, а также другие элементы данных (день/время отметки).

Этот процесс отличается от процесса подтверждения подлинности сообщения «пользователь-пользователь». Он не требует обмена «ключами достоверности», но в нем используются уникальные таблицы безопасности, созданные для каждого пользователя.

Помимо произвольных ключей защиты и ответа, известных только системе S.W.I.F.T. II и конечному пользователю, элементы данных, применяемые для формирования MAC, посылаются в составе MAC как часть сообщений LOGIN и SELECT. В ответ система S.W.I.F.T. II формирует новый концевик MAC с тем же ключом защиты, но с другими элементами данных и включает его в SAK или LNK, давая возможность пользователю

подтвердить достоверность системы S.W.I.F.T. II. Далее происходит приостановка подключения пользователя к системе и формируется запрос с новым концевиком, используя новый ключ доступа, для гарантии того, что сеанс будет возобновлен санкционированным LT с одновременной проверкой подлинности системы S.W.I.F.T. II

В S.W.I.F.T. II разработана и рекомендована Советом директоров для повсеместного использования улучшенная архитектура системы обеспечения безопасности, которая соответствует в широком смысле современному уровню развития телекоммуникационных технологий и криптографических методов. Основой нового подхода стало применение интеллектуальных карт (ICC), изменение алгоритма проверки достоверности и увеличение длины двусторонних ключей, которыми обмениваются пользователи.

Для обеспечения безопасности логического доступа к системе S.W.I.F.T. II в рамках нового подхода была разработана служба безопасного входа в систему и выбора режима (SLS), которая позволяет пользователям получить доступ к услугам системы S.W.I.F.T. II с помощью ICC вместо использования бумажных таблиц Login и Select. При применении ICC требуются *считыватели карт*. Необходимо заметить, что на данном этапе предлагается два различных типа считывателей карт. Первый – упрощенный считыватель карт (BCR), который поддерживает только службу SLS. Второй – считыватель карт с модулем защиты (SCR), в котором кроме функций считывателя реализована также функция модуля аппаратной защиты, выполняющего генерирование ключей и шифрование секретной информации (применяется как для поддержки SLS, так и других служб, созданных в рамках нового подхода). Так как в SCR должны храниться секретные данные, это устройство выполнено защищенным от вскрытия – любая попытка добраться до его внутренних частей вызывает автоматическое уничтожение секретной информации, хранящейся в SCR.

Кроме того, при обслуживании SCR или BCR предусмотрено несколько различных режимов работы (отключенный от CBT или подключенный к CBT), широкий перечень услуг по обучению персонала организации и варианты конфигурирования этих устройств специально выделяемыми людьми (офицерами безопасности).

Служба SLS и операции в рамках этой службы

Как уже говорилось, основное назначение SLS – замена бумажных таблиц Login/Select механизмом, способным генерировать сеансовые ключи доступа к системе, которые при использовании бумажных таблиц приходилось считывать операторам CBT вручную. Необходимо отметить, что ICC

не содержит самих ключей доступа, но хранит алгоритм, который может сгенерировать требуемый сеансовый ключ для любого запроса Login/Select. Так как данный алгоритм не совпадает с применяемым в системе S.W.I.F.T. II в настоящее время алгоритмом для генерации бумажных таблиц, ключи доступа, получаемые из ICC, отличаются от своих эквивалентов в бумажных таблицах.

Для обеспечения логического доступа к услугам системы S.W.I.F.T. II необходимо вставить соответствующим образом сконфигурированную ICC в считыватель карт и ввести PIN-код на клавиатуре считывателя. При выборе функции Login (Select) на CBT необходимые коды автоматически генерируются ICC и передаются в CBT, к которому подключен считыватель. Затем CBT продолжает обрабатывать запрос Login (Select) обычным образом. Для большинства пользователей считыватель карт будет оставаться подключенным к CBT с целью получения максимальной выгоды от службы SLS. Но есть возможность использования и неподключенного считывателя карт (например, для удаленных терминалов или в случае аварии), когда необходимые коды доступа хотя и генерируются в ICC, но отображаются на дисплее считывателя карт, а затем вручную вводятся в CBT.

Обеспечение безопасности сообщений, передаваемых и хранящихся в системе

Безопасность обмена сообщениями в системе S.W.I.F.T. II заключается в следующем:

- обеспечении безопасности передачи;
- проверке сообщений;
- обеспечении безопасности доставки.

Учитывая, что сеть защищена от несанкционированного доступа, поток сообщений при передаче и хранении должен иметь защиту от:

- утраты, повреждения, ошибочной доставки или задержки сообщений;
- ошибок при передаче и хранении;
- утраты конфиденциальности;
- внесения в сообщение ложных изменений.

Обеспечение безопасности передачи

Во все сообщения приложений GPA и FIN системы S.W.I.F.T. II добавляется обязательное окончание – так называемый концевик CHK, который содержит контрольную сумму данного сообщения, пересчитываемую в узлах ввода/вывода сети. Если произошло искажение сообщения во время

передачи (это устанавливается путем проверки контрольной суммы принятого сообщения, являющейся уникальной для каждого сообщения, с вычисленной контрольной суммой) и это не было зафиксировано в протоколе проверок низкого уровня, то на поступившую информацию будет передан отрицательный ответ и он повторится.

Проверка сообщений

Все входные сообщения проверяются соответствующим RP до того, как передать их SP. Только сообщения, отвечающие стандартам S.W.I.F.T. II и синтаксису, принимаются к доставке. Результаты непрерывных проверок постоянно сохраняются, и из-за очень строгих стандартов, установленных в S.W.I.F.T. II, любая серьезная ошибка протокола приводит к закрытию сеансов FIN или GPA.

Безопасность доставки

После строгих проверок, проведенных для всех входных потоков сообщений и высококачественных методов обеспечения безопасности, использованных для передачи сообщений, все сообщения, положительно подтвержденные системой S.W.I.F.T. II, рассматриваются правильными и, следовательно, доставленными системе.

Обязательный концевик CHK используется получающим LT для проверки того, что ни одной ошибки не появилось при передаче между входным RP и реципиентом. Обязательное использование подтверждений приема пользователем сообщения (UAK/UNK) дает возможность системе S.W.I.F.T. II ответить, принял ли LT посланное ему сообщение или нет. Система S.W.I.F.T. II не будет считать сообщение доставленным до тех пор, пока положительное подтверждение приема пользователем сообщения (UAK) не будет получено от LT-реципиента. Система S.W.I.F.T. II будет пытаться доставить сообщение 11 раз, после чего доставка сообщения прекращается и отправитель извещается, что сообщение не может быть доставлено. Каждая следующая попытка доставки после первой будет содержать соответствующее количество концевиков PDM. Проверка сообщений S.W.I.F.T. II гарантирует, что сообщения для подготовки (которые имеют концевик TNG) не могут быть адресованы к действующим местам назначения, и наоборот – действующий поток сообщений не может быть адресован к местам назначения по подготовке.

Безопасность обмена сообщениями «пользователь-пользователь»

При обмене сообщениями между пользователями для обеспечения конфиденциальности и подлинности, а также для контроля над целостностью сообщений система S.W.I.F.T. II рекомендует применять алгоритм

проверки достоверности. Проверка достоверности – важная часть системы обеспечения безопасности S.W.I.F.T. II, она заключается в обмене между пользователями ключами и проверке того, что достоверный результат представлен в определенных типах сообщений.

Принимающий терминал проверяет текст полученного сообщения при помощи стандартного алгоритма SA/2 и согласованного ключа достоверности. И если в ходе проверки получен отрицательный результат, то это может, скорее всего, произойти из-за ошибок передачи или неверного ключа достоверности.

Ключ достоверности состоит из 32 шестнадцатеричных символов, разделенных на две части по 16 знаков, и может быть использован как для передачи, так и для приема или в обоих направлениях. Для формирования ключа необходимо придерживаться следующих правил:

- первая и вторая половина должны быть различны;
- в каждой половине любой разрешенный символ может появиться только один раз.

Здесь необходимо отметить, что ключи достоверности передаются между корреспондентами по почте, и для обеспечения безопасности ключевой информации всем пользователям системы S.W.I.F.T. II рекомендуется поддерживать корреспондентские отношения только с известными пользователями и в организации – инициаторе обмена выбирать тип ключа достоверности в соответствии с проводимой политикой безопасности этой организации.

Как уже говорилось, в связи с переходом на новые технологии обеспечения безопасности в системе S.W.I.F.T. II, использующие ICC, был усовершенствован и процесс обмена ключами достоверности между пользователями, результатом чего стало появление службы обмена двусторонними ключами (BKE). Назначение BKE – заменить утомительную систему ручного обмена двусторонними ключами подтверждения подлинности между корреспондентами по открытой почте на систему, которая использует новые сообщения S.W.I.F.T. II и считыватель карт с модулем защиты, специально разработанные для этой цели. Система позволит полностью автоматизировать процесс обмена ключами. По новой технологии каждый двусторонний ключ подтверждения подлинности создается внутри SCR и зашифровывается перед передачей в CBT, к которому SCR подключен. Ключи подтверждения подлинности бывают либо двунаправленными (один и тот же ключ служит для проверки передаваемых и принимаемых сообщений), либо однонаправленными (для приема и передачи сообщений используются отдельные ключи). Служба

BKE основана на стандарте ISO по обмену ключами (ISO 11166 – Banking – Key Management by Means of Asymmetric Algorithms). В этом стандарте определено использование асимметричных алгоритмов для шифрования и цифровой подписи двусторонних ключей, которыми обмениваются корреспонденты. Специально для обеспечения распределения открытых ключей в системе S.W.I.F.T. II был создан Центр управления безопасностью (SMC), в составе которого работает Центр сертификации ключей, выдающий сертификаты открытых ключей пользователей системы S.W.I.F.T. II

Следующий за процедурами перехода и начальной установки реальный обмен двусторонними ключами подтверждения подлинности по сети S.W.I.F.T. II состоит из передачи четырех специальных сообщений S.W.I.F.T. II между корреспондентами, один из которых выступает как инициатор обмена, а другой – как получатель. Первые два сообщения используются исключительно для целей установления сеанса обмена двусторонними ключами. В третьем сообщении инициатор обмена посылает ключ, созданный и зашифрованный внутри SCR, используя открытый ключ получателя. Таким же образом SCR создает цифровую подпись инициатора обмена ключами. После получения третьего сообщения участник обмена проверяет цифровую подпись, и если она действительно принадлежит отправителю, то ему высылается подтверждение, ключ признается верным и заносится в файл двусторонних ключей.

Непосредственно после обмена новый ключ становится «будущим» ключом для этих корреспондентов и будет использоваться для проверки финансовых сообщений, начиная с взаимно согласованных даты и времени. При использовании ключей приема/передачи каждый корреспондент является инициатором обмена для своего ключа передачи.

3.8.2. Технология SmartCity

SmartCity представляет собой комплексное решение на основе применения смарт-карт с использованием концепции электронного кошелька, являющееся вместе с тем платформой для разработки множественных приложений и инструментарием для системных интеграторов. На основе технологии SmartCity реализуется система, которая является объектно-ориентированным приложением в архитектуре «клиент-сервер». В такой системе используются смарт-карты нескольких типов: карты типа электронного кошелька, в том числе совместимые со стандартом EMV (Europay, MasterCard и Visa), с возможностью неоднократного пополнения баланса карты, и карты памяти, использование которых

ограничивается первоначально определенной суммой средств (непополняемые карты). Гибкая и в то же время защищенная архитектура системы, построенной на основе SmartCity, позволяет с легкостью адаптировать ее к требованиям каждого эмитента карт. Учет и обработка всех транзакций, многоэмитентность, поддержка нескольких электронных кошельков и разных видов валют на одной карте, дебетовые и кредитные кошельки, многоязычная поддержка являются характерными особенностями системы.

В зависимости от предполагаемого количества эмитентов смарт-карт на основе технологии SmartCity могут быть реализованы системы двух типов:

- SmartCityBack Office – открытая система, разработанная для нескольких эмитентов смарт-карт. Открытая архитектура системы обеспечивает обработку транзакций по картам различных эмитентов либо напрямую, через систему SmartCity, либо через расчетно-клиринговые системы;
- SmartCity Lite – закрытая платежная система, рассчитанная на одного эмитента карт. Примером такой системы является корпоративная карточка для безналичных расчетов внутри одного банка или предприятия, городская карточка или карточка для расчетов на автозаправочных станциях. Закрытая система обладает сокращенным набором функций открытого варианта.

Архитектура системы, выполненной по технологии SmartCity (рис. 3.31), имеет четыре уровня:

- терминальный уровень, представленный устройствами обслуживания держателей карточек: POS-терминалами, торговыми автоматами, информационными киосками, банкоматами, рабочими местами для операционистов банка и работников почтово-банковских отделений;
- уровень front-end, включающий сервер авторизации, сервер телекоммуникации, сервер безопасности, станцию управления online-устройствами, станцию мониторинга сети банкоматов и ряд автоматизированных рабочих мест (администратора, операциониста, графической персонализации и удаленной персонализации);
- SmartCity, представляющий собой программный комплекс, реализованный в архитектуре «клиент-сервер», ядром которого является программный комплекс SmartCity Back Office.
- уровень бухгалтерского компонента, представленный системой Smart-Retail, также реализованный в трехуровневой архитектуре «клиент-сервер».

Рис. 3.31. Транзакция денег в электронном кошельке

В ходе работы данной системы производятся следующие операции:

- выпуск карт для пользователей;
- начисление средств на карту. Поддерживается несколько способов первоначального начисления средств на смарт-карты и дальнейшего их кредитования с использованием POS-терминалов, работающих в режиме online, терминалов операционистов банка, станций просмотра, банкоматов, устройств Account-to-Card, предназначенных для перевода средств со счета кредитной карты или дебетовой карты, устройств Cash-to-Card, предназначенных для начисления средств на карту в соответствии с суммой введенных в устройство банкнот;
- покупка на терминале продаж;
- сбор транзакций. Транзакции, совершенные в режиме online, записываются в формате данных системы SmartCity в специальные файлы транзакций, которые хранятся в энергонезависимой памяти каждого устройства либо на локальном или сетевом диске для устройств, организованных на базе ПЭВМ. В системе поддерживаются несколько методов сбора карточных транзакций с устройств и их передачи модулю обработки транзакций при помощи модемного соединения с телекоммуникационным сервером, транспортных смарт-карт, портативных компьютеров и компьютерных сетей;
- обработка транзакций с использованием криптографических методов и средств;
- контроль над транзакциями. Все транзакции в системе SmartCity отслеживаются и полностью контролируются с помощью программы

просмотра базы данных и модуля создания отчетов; в результате появляется возможность быстро заменять потерянные, украденные или испорченные карты без утраты информации;
- работа со стоп-листами;
- взаиморасчеты. Процедура взаиморасчетов в SmartCity обеспечивает проведение выборки данных по взаиморасчетам между продавцами и организациями-эквайерами в системе и выгрузку этих данных в основную финансовую систему. Экспортируемые данные по взаиморасчетам формируются в соответствии с требованиями, предъявляемыми к формату данных в основной системе;
- клиринг и взаиморасчеты между эмитентами – участниками открытой системы. Стандартная система, реализованная по технологии SmartCity, поддерживает проведение непосредственного клиринга и взаиморасчетов между эмитентами карт – участниками так называемой полуоткрытой системы. При использовании в расширенной системе технология SmartCity с легкостью поддерживает существующие системы клиринга и взаиморасчетов, разработанные другими компаниями или самими эмитентами;
- отчетность.

Наиболее важной характеристикой SmartCity является система безопасности, состоящая из:

- криптографической защиты информации. Встроенные в смарт-карты алгоритмы шифрования и ЭЦП могут использоваться для защиты финансовых транзакций, операций дебетования и кредитования смарт-карт и безопасной передачи дополнительных файлов. В системе используется также ряд других средств криптографической защиты информации, но о них будет сказано далее и более подробно;
- контроля доступа на уровне администратора системы ко всем приложениям и функциям SmartCity в дополнение к мерам безопасности, предоставляемым средствами Sybase и Oracle;
- контроля над процессом выпуска карт клиентов при помощи карт администратора;
- контроля над всеми транзакциями в системе: восстановление утерянных и украденных карт без потери суммы, разрешение споров между держателями карт и магазинами;
- использования стоп-листов или списков утерянных и украденных карт;

- аутентификации владельца смарт-карты при использовании смарт-карты на основе PIN-кодов;
- идентификации пользователей с применением биометрических способов.

Подсистема криптографической защиты информации SmartCity

Криптографическая система безопасности включает:

- реализацию криптографических алгоритмов на смарт-картах;
- защищенные криптопроцессоры, являющиеся основой сервера безопасности, в функции которых входит:
  - генерация, диверсификация, хранение и распределение ключей;
  - генерация криптопоследовательности, использующейся при кредитовании смарт-карт;
  - проверка подлинности всех транзакций;
  - защищенная передача информации между узлами системы;
  - защита любых критических в плане безопасности операций;
- защищенные модули безопасности SAM, использующиеся в устройствах, обслуживающих смарт-карты. Удерживая в памяти до 32 ключей, эти модули при помощи встроенного криптопроцессора осуществляют следующие криптографические операции: шифрование PIN-кода по стандарту ANSI X9.8, создание MAC (стандарт ANSI X9.19).

Криптографические механизмы позволяют реализовать в SmartCity:

- целостность, конфиденциальность и подлинность информации при проведении транзакций;
- подлинность операций клиринга и взаиморасчетов;
- аутентификацию в системе смарт-карта-устройство чтения смарт-карт;
- целостность и конфиденциальность дополнительной информации, передаваемой в системе.

Во время создания каждой из транзакций смарт-карта и SAM-модуль считывателя смарт-карт выполняют следующие процедуры:

- шифрование наиболее важной информации по транзакции (сумма платежа, баланс карты и т.д.);
- формирование ЭЦП к сумме кредита/дебета для подтверждения подлинности данной суммы. Ключи, используемые для создания кредитной или дебетовой карты ЭЦП, известны только карте и системе SmartCity, выпустившей ее;

- создание MAC-кодов к транзакции для подтверждения подлинности транзакции и целостности данных. Ключи, используемые для создания MAC, известны только SAM-модулю и системе SmartCity, которой принадлежит данное устройство.

Любая транзакция начинается с проведения аутентификационной выработки сеансового ключа. Этот процесс выполняется в следующем порядке:

1. Устройство работы со смарт-картой генерирует случайное число и передает его смарт-карте.
2. Смарт-карта увеличивает счетчик транзакций и с использованием ключа аутентификации и счетчика транзакций генерирует сеансовый ключ, после чего зашифровывает на сеансовом ключе случайное число и передает его устройству.
3. Устройство подобным образом генерирует сеансовый ключ, расшифровывает ответ смарт-карты и производит проверку соответствия полученного числа отправленному; в случае успешного результата проверки пользователь допускается для проведения транзакции.

Для кредитования смарт-карты требуется сервер безопасности, который генерирует необходимую криптопоследовательность. Сервер безопасности защищен организационно-техническими мерами от НСД и оснащен криптопроцессором. Работа с сервером безопасности происходит в режиме online. При отсутствии возможности организовать режим работы online для кредитования карт клиентов на торговых терминалах используются SAM-модули. Помимо обычной защиты, предоставляемой самим модулем, на нем указывается также максимальная сумма кредитования, которую можно провести с помощью данного SAM-модуля.

При загрузке транзакций модуль Transaction Protection в первую очередь устанавливает подлинность каждой транзакции, вычисляя уникальное значение MAC и сравнивая со значением, созданным SAM-модулем устройства к данной транзакции. Если транзакция совершена по карте, выпущенной этим же банком или эмитентом, далее проводится проверка подлинности ЭЦП кредитования/дебетования. Недействительные или повторяющиеся транзакции отклоняются, что фиксируется в журнале загрузок. Действительная транзакция сохраняется в базе данных SmartCity. Если в поле MAC система обнаружила, что транзакция совершена по карте, выпущенной другим эмитентом в SmartCity, транзакция помечается как внешняя и экспортируется через клиринговую систему тому эмитенту, по чьей карте проводилась транзакция. Внешние транзакции проверяются

при помощи MAC, подписываются сертификационной подписью открытой системы (Open system MAC) и экспортируются в систему эмитента, выпустившего карты, по которым были проведены собранные транзакции. Система эмитента импортирует эти транзакции и проверяет их подлинность с помощью ЭЦП транзакции. Поскольку проверить подлинность ЭЦП в транзакции может только система SmartCity, выпустившая смарт-карту, транзакции разрешается передавать между эмитентами любыми доступными средствами, в том числе по Internet.

Положительным моментом в подсистеме криптографической защиты информации SmartCity является развитая ключевая система.

Организация ключевой системы и управление ключами

Для каждой открытой системы выбирается организация, являющаяся центром открытой системы. Этой организации принадлежат мастер-криптоплаты, и только эта организация может добавлять участников открытой системы. Мастер-криптоплата генерирует общие ключи системы и переносит их в криптоплаты эмитентов.

В дальнейшем эмитент самостоятельно инициализирует свои криптоплаты на основе криптоплат, переданных ему из центра. Каждый участник открытой системы создает часть ключей для использования только своей системой. Это обязательно должны быть ключи кредитования и ЭЦП транзакции.

Сгенерированные ключи системы хранятся в зашифрованном виде в аппаратно-защищенной памяти самого процессора безопасности. Для шифрования ключей используется RSA алгоритма.

Ключи системы делятся на ключи открытой системы и ключи закрытой системы. Первые получают все участники открытой системы. Общие ключи используются для идентификации карты как карты открытой системы и дебетования карты на POS-терминале. Кроме ключей открытой системы создается также Kinit1-ключ, идентифицирующий эмитента в открытой системе.

Для каждого нового эмитента создается пара контрольных карт (контрольная и инициализационная), предназначенных для использования при выпуске карт клиентов.

Ключи закрытой системы уникальны для каждого участника системы и создаются модулем управления ключами с использованием процессора безопасности эмитента. В процессе генерации ключей эмитент имеет возможность «переписать» ключ открытой системы Kdebit2. В этом случае он оказывается уникальным ключом эмитента и кошелек смарт-карты,

защищенный ключом Kdebit2, становится закрытым криптографическим средством. «Закрытые» кошельки нельзя дебетовать на торговых терминалах других эмитентов.

В системе используются следующие ключи:

- ключи клиента;
- ключи эмитента;
- ключи транспортной карты;
- ключи устройств, работающих со смарт-картами.

Каждая карта клиента имеет свой уникальный набор ключей, создающихся на основе парных ключей эмитента. Необходимо отметить, что при инициализации карты клиента в процессинговом центре эмитента все ключи, записываемые на карту, генерируются на основе пары мастер-ключей системы и серийного номера карты. Даже если ключ карты известен нарушителю, подделать ее невозможно, поскольку у поддельной карты будет другой серийный номер. При этом используется следующий алгоритм: K = DES((DES(Серийный номер карты, Мастер-ключ 1) XORСерийный номер карты), Мастер-ключ 2).

Карта клиента содержит следующие ключи:

- аутентификации;
- дебетования карты (один из них может быть ключом закрытой системы);
- кредитования карты;
- для доступа к файлам на карте (файл транзакции, дополнительные файлы);
- для подписи транзакции при пересылке между участниками системы.

Для каждого эмитента создаются его уникальные ключи закрытой системы:

- кредитования кошельков;
- создания ЭЦП к операциям дебетования и кредитования кошельков;
- доступа к дополнительным файлам на картах клиентов;
- шифрования PIN-кода;
- для подписи транзакции данного эмитента.

Ключи системы загружаются в карточки и во все устройства, работающие с карточками. Далее ключи используются в процессах кредитования/дебетования карт и при проверке подлинности транзакций во время передачи от терминала к процессинговому модулю и от одного процессингового модуля к другому.

Секретные ключи транспортной карты вычисляются и записываются на карту в момент ее инициализации в процессинговом центре. Набор

секретных ключей уникален для каждой карты, они являются производными серийного номера карты и парных ключей системы, хранящихся в базе данных. При этом используется алгоритм, аналогичный алгоритмам генерации для ключей клиента. PIN-код, общий для всех транспортных карт, хранится в базе и автоматически прописывается на карту.

К ключам транспортной карты относятся:

- ключ, идентифицирующий эмитента;
- ключ для доступа на чтение/запись транзакций на транспортной карте.

Ключи устройств, работающих со смарт-картами (например, PINpad CM 450/SC 455/ SC 552), загружаются в устройство посредством ПО процессингового модуля (CMS). Набор загружаемых ключей выбирается в зависимости от назначения устройства (POS-терминал, терминал кредитования и т.д.). Устройство PINpad оснащено модулем SAM, где хранятся ключи в открытом виде, и при попытке вскрытия модуля они сбрасываются из памяти терминала.

3.8.3. Система UEPS

Универсальная система электронных платежей (Universal Electronic Payment System, UEPS) представляет собой банковское приложение, использующее интеллектуальные карточки, первоначально разработанное для Южной Африки, но позднее было принято к использованию основными банковскими группами этой страны. К началу 1995 года в ЮАР было выпущено около 2 млн карточек. Эта система также действует в Намибии и в настоящее время распространяется по крайней мере одним российским банком. Она позволяет использовать безопасные дебетовые карточки, подходящие для регионов, в которых плохая телефонная сеть делает невозможной диалоговую проверку. Карточки есть и у покупателей, и у продавцов; покупатели могут использовать свои карточки для перевода денег продавцам. Продавец может воспользоваться своей карточкой, чтобы позвонить в банк и поместить деньги на свой банковский счет; покупатель может воспользоваться своей карточкой, чтобы позвонить в банк и перевести деньги на свою карточку. Нет необходимости заботиться об анонимности, нужно обеспечить только защиту от мошенничества.

Вот как выглядит протокол связи между покупателем Алисой и продавцом Бобом (В действительности Алиса и Боб просто вставляют свои карточки в машину и ожидают выполнения транзакции.) Когда Алиса впервые получает свою карточку, она получает и пару ключей, K_1 и K_2, банк вычисляет их, используя ее имя и некоторую секретную функцию. Только

в карточки продавцов встроены секретные программы, необходимые для вычисления ключей пользователей:

1. Алиса посылает Бобу свое имя, A, его имя, B, и случайное число, R_A, шифруя их с помощью DES: сначала ключом K_2, затем K_1. Она также посылает свое имя открытым текстом.

   $A((A, B, R_A))$

2. Боб вычисляет K_1 и K_2 по имени Алисы. Он расшифровывает сообщение, убеждается, что A и B правильны, затем шифрует вторую половину сообщения Алисы ключом K_2.

   $E_{K_2}(A, B, R_A)$

3. Боб не посылает это сообщение Алисе, 56 бит шифротекста становятся ключом K_3. Боб посылает Алисе свое имя, ее имя и случайное число R_B, шифруя их с помощью DES сначала ключом K_3, затем K_1.

   $((B, A, R_B))$

4. Алиса аналогичным образом вычисляет K_3 и расшифровывает сообщение Боба, убеждаясь, что A и B правильны; затем шифрует вторую половину сообщения Боба ключом K_3.

   (B, A, R_B)

5. Алиса не посылает это сообщение Бобу, 56 бит шифротекста становятся ключом K_4. Затем Алиса посылает Бобу свое имя, его имя и проверочное значение C, которое содержит имена отправителя и получателя, дату, контрольную сумму, количество и два MAC. Все это шифруется DES: сначала ключом K_4, затем K_1. Один из MAC может быть проверен банком Алисы, а второй может быть проверен только расчетно-кассовым центром. Алиса уменьшает свой счет на соответствующее значение.

   $((A,B,C))$

6. Боб аналогичным образом вычисляет K_4. При условии, что все имена совпадают и проверка выполнена правильно, он принимает платеж.

Нововведением в этом протоколе является то, что каждое сообщение зависит от предыдущего и выступает удостоверением *всех* предыдущих сообщений. Это означает, что повторить старое сообщение никому не удастся, получатель просто никогда не расшифрует его.

Другая разумная идея в этом протоколе – навязывание правильной реализации. Если разработчик приложения неправильно реализует протокол, он просто не будет работать.

Обе карточки сохраняют записи каждой транзакции. Когда карточки рано или поздно установят диалоговое соединение с банком (продавец – положить деньги на счет, а покупатель – снять со счета), он извлечет эти записи для последующего контроля.

Аппаратура изготавливается в форме, устойчивой к взлому, чтобы помешать любому из участников испортить данные. Алиса не сможет изменить значение своей карточки. Подробная запись содержит данные для обнаружения и запрещения мошеннических транзакций. В карточках используются универсальные секреты – ключи MAC, функции для преобразования имен пользователей в K_1 и K_2, но считается, что решение обратной задачи для этих секретов достаточно трудное.

Эта схема, конечно же, несовершенна, но она безопаснее бумажных чеков и обычных дебетовых карточек. Источником угрозы мошенничества являются не военные враги, а покупатели и продавцы. UEPS предоставляет защиту от таких злоупотреблений.

Обмен сообщениями является прекрасным примером устойчивого протокола: в каждом сообщении присутствуют имена обеих сторон, включая информацию, уникальную для сообщения; каждое сообщение явным образом зависит от всех предыдущих.

3.9. Электронные платежные системы и Internet

Компьютерные технологии все глубже и глубже проникают в повседневную жизнь и практическую деятельность человека. Тенденции, способствовавшие появлению так называемой электронной банковской системы, зародились приблизительно двадцать лет назад. Наряду с развитием электронных платежных систем появляются все новые и новые инструменты по проведению операций – электронные деньги (e-cash), смарт-карты и т.д. При этом усовершенствованные технологии находят применение не только в деятельности финансовых институтов, но и в повседневной деятельности обычного покупателя.

Появление новых систем оплаты стало следствием того, что традиционные системы, основанные на наличных деньгах, чеках, кредитных картах, EFT/POS и банковских переводах средств с одного счета на другой, несмотря на свою простоту и удобство в обращении, имеют высокие затраты.

В этой главе рассмотрены платежные системы, системы, позволяющие организовать электронную коммерцию в Internet, а также вопросы безопасности и используемые средства защиты информации.

3.9.1. Классификация платежных систем

Новые платежные механизмы должны заменить традиционные методы оплаты наличными деньгами, чеками и денежными переводами их электронными аналогами (рис. 3.32).

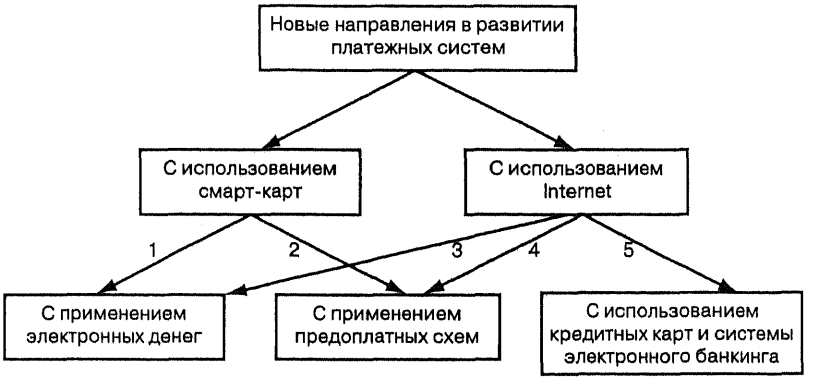

Рис. 3.32. Классификация платежных систем

Платежные системы, построенные на основе понятий «электронные деньги» и «смарт-карты»

Представителем практической реализации является система Mondex, разработанная компанией Mondex International (рис. 3.33). Система использует смарт-карты (см. раздел 3.9.2) и электронные деньги (см. раздел 3.9.3) как форму представления денежных средств, хранящихся на смарт-картах.

Плательщик, имеющий смарт-карту, загружает на нее электронные суммы через специальные устройства, в качестве которых могут выступать банковские или телефонные аппараты. Дальнейшие операции оплаты, перевода денежных средств на другой носитель и т.д. будут проходить минуя расчетные системы. Фактически электронные суммы, используемые в данной технологии, являются аналогом наличных денег.

Преимущество данного подхода заключается не только в том, что денежные средства могут быть переведены с использованием общедоступных телефонных каналов или глобальных сетей передачи данных, но и в неотслеживаемости действий клиента.

При существующей системе Mondex необходимо иметь один организационный центр, обеспечивающий эмиссию электронных денег в валюте страны. Такой эмитент осуществляет в системе Mondex функции распространителя и снабжает банки-участники соответствующими электронными

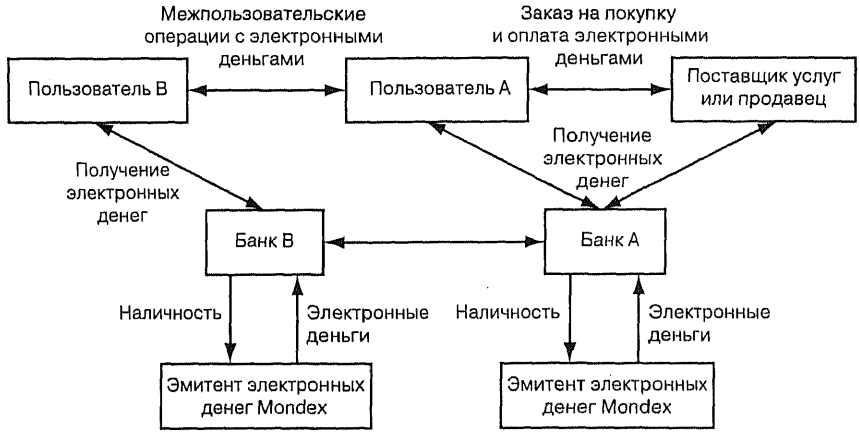

Рис. 3.33. Архитектура системы Mondex

суммами в обмен на оборотные средства, имеющие себестоимость, и наличные.

На сегодняшний день первые успешные испытания данной системы прошли по инициативе Hong Kong Shanghai Bank в двух торговых центрах: в г. Тайку (о. Гонконг) и на полуострове Коулун.

Хотя организационная структура системы оплаты Mondex может видоизменяться в зависимости от распространения данной системы, ее применения и ряда других факторов, однако основа ее представляется рациональной, поскольку расчеты производятся не предоплатными данными, а электронными эквивалентами наличных денег.

Платежные системы на основе предоплатных схем

На сегодняшний день эти продукты и реализации финансового обмена являются наиболее распространенными платежными системами. Впервые они появились в Европе в начале 90-х годов. Идея состояла в использовании особой смарт-карты, способной хранить данные о денежных суммах, которые заносятся на нее при помощи кард-ридеров, обладающих функциями записи. При осуществлении покупки или оплаты услуг количество денежных средств, записанных на карте, уменьшается на величину требуемой суммы.

Других принципиально новых идей в таких системах, кроме идеи использования специальных интеллектуальных устройств для хранения данных, называемых еще электронными бумажниками, не применяется.

Данные о денежных средствах, загружаемые на смарт-карту, соответственно списываются со счета пользователя, или пользователь кредитуется

банком на указанную сумму. В подобных системах свободного обращения денежных средств между пользователями не происходит. Каждая сделка обязательно должна быть зафиксирована в том или ином операционном центре финансового учреждения, обслуживающего данные системы оплаты. Поэтому они являются лишь одной из форм безналичных расчетов применительно к широкому кругу физических лиц.

Примером подобной формы расчетов может служить система VisaCash фирмы Visa International (смарт-карта VisaCash). Первые испытания этой системы проведены в 1995 г. во время летних Олимпийских Игр в Атланте. В ходе опробования было задействовано более 1500 пунктов общественного питания и автозаправочных станций, правда, был установлен верхний предел на суммы, загружаемые в электронный кошелек. Он составлял 100 долларов.

Архитектура данной системы представлена на рис. 3.34.

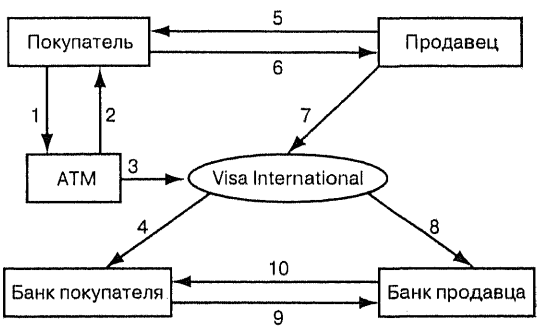

Рис. 3.34
Архитектура VisaCash

Платеж осуществляется в следующем порядке:

1. Сначала покупатель использует кредитную карту для запроса данных об оплате.
2. Данные предоплаты загружаются на смарт-карту.
3. Информация об оплате поступает в финансовую сеть Visa.
4. Информация о оплате поступает в банк покупателя, и с его счета списывается необходимая сумма.
5. Продавец поставляет услугу или товар.
6. От покупателя к продавцу передаются данные по оплате товара или услуги.
7. Предоплатные данные поступают в финансовую сеть Visa.
8. Данные поступают в банк продавца.
9. и 10. Осуществляется взаиморасчет между банками продавца и покупателя, результатом которого является перевод денежных средств на счет продавца.

Системы с электронной наличностью, использующие Internet

Первой полноценной системой, вобравшей в себя все достоинства финансового обмена, совершаемого в электронной форме, стала e-cash фирмы Digicash (рис. 3.35). Подобные системы основаны на использовании электронных денег, передаваемых на рабочую станцию пользователя через Internet, при этом, как и в системе Mondex, применяются именно электронные деньги, а не данные предоплаты. Существенное отличие этой системы от Mondex заключается в том, что выпускаемые в e-cash электронные монеты являются неразменными, а это во многом ограничивает возможность непрерывного осуществления денежных переводов. В существующем варианте организационной структуры системы все пользователи должны быть зарегистрированы в Mark Twain Bank, осуществляющем ее обслуживание.

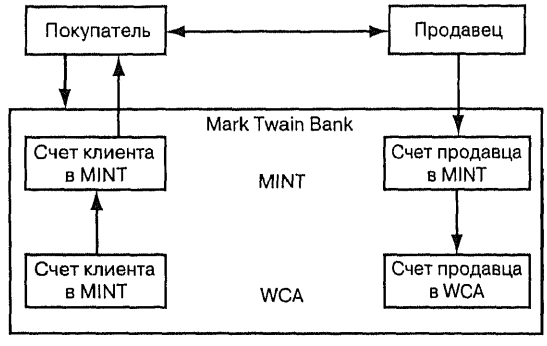

Рис. 3.35
Принципы работы системы Digicash

В системе предполагается использование двух типов счетов пользователей – WCA (счета для «реальных» денег) и MINT (счета для электронных денег). Конвертация денег из наличных в электронные происходит путем перевода денежных средств со счета WCA на счет MINT.

Предоплатные схемы в Internet

Примером является система компании CyberCash, которая по своим функциональным возможностям сходна с системой VisaCash. CyberCoin предназначена для осуществления мелких платежей, включая оплату разменной монеты через Internet. В начале операции покупатель должен открыть счет, называющийся CyberCash Account, затем со своего сберегательного счета перевести на него деньги, где они уже будут представлены как деньги CyberCoin, и, используя ПО SuberCash Consumer Wallet, открыть доступ к электронным деньгам, которые передаются на рабочую станцию

пользователя через Internet. При этом следует учесть, что, хотя в данной системе и используются электронные деньги, расчеты фактически осуществляются переводом денежных средств со счета покупателя на счет продавца с применением банковской сети и предоплаты электронных денег путем резервирования денежных средств на счете пользователя. Операции в CyberCoin не требуют открытия дополнительных специализированных счетов в других банках, поскольку осуществляются через существующие банковские сети, что отличает эту систему от Digicash.

Сама покупка в данной системе начинается с приобретения кредитной карты для оплаты и выбора предлагаемых на Web-сайте продавца товаров или услуг, после чего покупателю предлагается нажать на кнопку **Pay** (Оплатить). Это событие активизирует ПО CyberCash, осуществляющее обработку и передачу электронных денег. (Если данное ПО недоступно покупателю, то предлагается загрузить его из Internet.)

Далее ПО зашифровывает финансовую информацию, подписывает ее и посылает продавцу, который проверяет целостность переданной информации, но при этом расшифрование информации, содержащей номер кредитной карты покупателя, не осуществляется. Данные передаются на сервер CyberCash с подписью продавца, после чего производятся необходимые проверки и в случае успешного выполнения осуществляется расшифрование номера кредитной карты. Затем информация о кредитной карте отправляется на сервер процессинговой компании, обслуживающей данный тип кредитных карт, на котором производятся завершающие проверки (проверяется номер кредитной карты, наличие на счете покупателя запрошенной суммы и т.д.). Если все в порядке, производятся переводы денежных средств со счета покупателя на счет поставщика и продавец информируется об успешном проведении платежа.

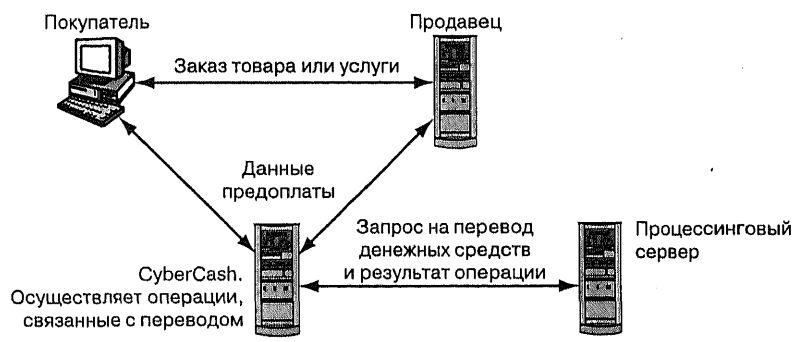

Рис. 3.36. Осуществление платежных операций в CyberCash

Использование чеков, кредитных карт и электронный банкинг

К системам этого типа относятся аналоги традиционных систем оплаты чеком или кредитной картой, но с использованием сети Internet.

К данному типу относятся также системы, позволяющие пользователям проводить финансовые операции не выходя из дома; подобного рода системы называются home banking. В последнее время перед владельцами банковских счетов и кредитных карт возникла необходимость оперативно управлять своими личными финансами. При этом подразумевается, что клиенту нет нужды использовать стандартные средства общения с банком, например рассылку платежных поручений по почте или через операционное отделение, что, очевидно, накладывает определенные временные издержки на проведение подобных операций.

Первая система домашнего управления личными финансами была разработана фирмой Intuit и называлась Quicken. Она позволяла пользователям осуществлять перевод денежных средств по своим счетам, оплачивать товары и услуги, получать выписки и отчеты по проведенным операциям и др.

Система Decart Home Bank

Отечественным аналогом формы электронных расчетов стала система удаленного обслуживания клиентов Decart Home Bank, являющаяся совместным проектом таких компаний, как «АйТи», «Арсенал» и МО ПНИЭИ. Она обладает не только широким набором функциональных возможностей (сведения о движении денежных средств по счету, о курсах валют, об управлении финансами и т.д.), но и высоким уровнем обеспечения информационной безопасности.

Система Decart Home Bank позволяет проводить подобные операции с использованием Internet и построена с применением стандартных протоколов (SMTP/POP3) и решений (Miscrosoft Exchange Server), что обеспечивает этому продукту высокий уровень масштабируемости и интегрируемости.

Decart Home Bank (рис. 3.37) состоит из клиентской части, которая представляет собой рабочую станцию с применением ПО «Декарт», подключенную к Internet, и банковской части.

К функциям банковской части системы относятся:

- регистрация пользователей системы (клиентов банка);
- обработка запросов пользователей (клиентов банка);

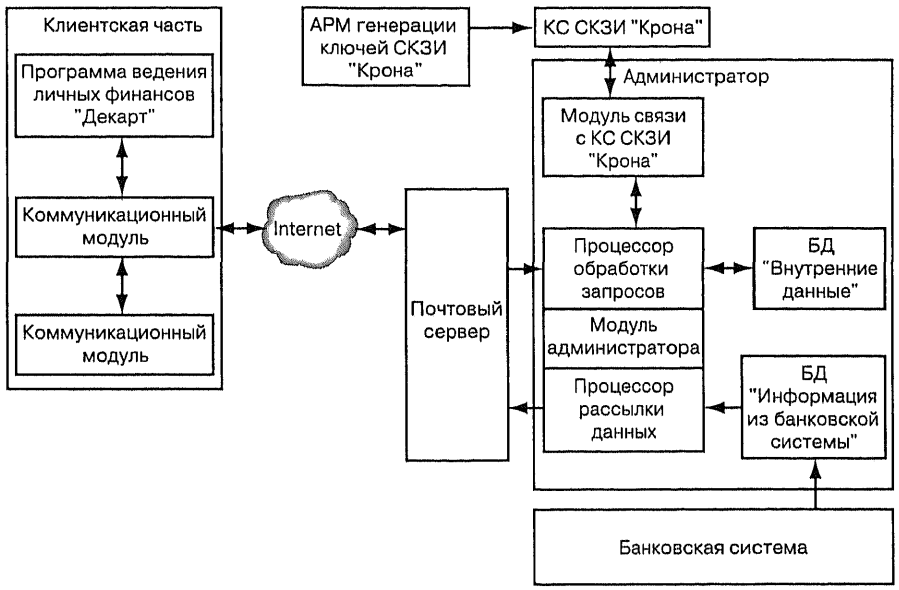

Рис. 3.37. Архитектура Decart Home Bank

- организация доступа к данным по счетам клиентов, хранящимся в банковской системе, и формирование данных для передачи по запросам клиентов. При этом клиенту передается следующая информация: выписки по счету клиента банка, курсы валют, текущие условия обслуживания по счетам, текущее состояние счета клиента и информация о подписках, уже оформленных клиентом и всех предоставляемых банком;
- поддержка механизма рассылки информации клиентам об операциях по их счетам и курсах валют;
- организация передачи и приема информации от клиента по сети Internet;
- ведение журнала безопасности системы;
- обеспечение диагностики системы и автоматического оповещения администратора о сбоях в работе и нарушениях системы защиты.

Клиентская часть системы предназначена для:

- ведения финансов;
- планирования бюджета;
- выполнения операций с пластиковыми картами;
- персонального учета расходов;
- проведения анализа банковских операций.

Перечисленные выше возможности системы основаны на следующих доступных пользователю операциях:

- ведении списка операций по мультивалютным счетам;
- проведении учета операций по пластиковым картам, открытым на корпоративных и личных счетах;
- ведении учета расходов и доходов по любым другим реальным банковским счетам или вне привязки к таковым;
- ведении пополняемых и настраиваемых справочников по счетам, картам, платежным схемам, категориям операций, назначениям операций, адресатам, валютам, курсам валют;
- расчете оперативных показателей текущего состояния счетов;
- формировании разнообразных текстовых и графических отчетов по счетам.

Для понимания принципов функционирования подобной системы приведем типовой сценарий ее работы:

- при запуске программы «Декарт» пользователь вводит пароль, что позволяет ограничить доступ к локально хранимой в программе финансовой информации о клиентах банка;
- пользователь программы вызывает модуль связи с банком (коммуникационный модуль) и выбирает одну из следующих операций: получение информации от банка (например, получение ежедневной рассылки курсов валют), изменение условий подписки на автоматическую рассылку информации или выполнение специального запроса (например, запрос выписки по счету за прошлый месяц);
- модуль связи зашифровывает запрос, затем соединяется с почтовым сервером банка и передает запрос. Шифрование запроса происходит автоматически с помощью клиентского модуля СКЗИ «Крона»; при этом используется ключ, который пользователь получает во время регистрации в банке;
- зашифрованный запрос пользователя попадает в процессор обработки запросов, установленный на рабочем месте администратора, который расшифровывает его с помощью криптосервера (КС) СКЗИ «Крона», при этом проверяется целостность сообщения и право пользователя на получение запрашиваемой информации. Затем запрос обрабатывается и подготавливается ответ;
- ответ на запрос передается процессору рассылки информации, который зашифровывает его для соответствующего абонента с помощью

КС СКЗИ «Крона» и передает на почтовый сервер, откуда тот запрашивается клиентом в удобное для него время;

- модуль связи соединяется с сервером банка и проверяет наличие информации для клиента. В случае обнаружения нужной информации модуль снимает ее с сервера, затем расшифровывает при помощи клиентского модуля СКЗИ «Крона», проверяя ее целостность и истинность отправителя. После чего информация (выписки по счетам, курсы валют) передается ПО «Декарт», где она представляется в удобном для пользователя виде. ПО «Декарт» автоматически вводит полученную информацию в свою базу данных.

Ориентированная на пользователей, не имеющих специальных знаний в области бухгалтерского учета и финансов, программа «Декарт» проста в освоении и удобна в работе. Ее основное рабочее поле представляет собой таблицу, где каждая строка – некая финансовая операция, что является наиболее доступным и интуитивно понятным способом фиксации каждой операции. Программа позволяет ввести любое количество счетов, операции по каждому из которых заносятся в отдельную таблицу.

Простота и наглядность работы программы с финансовой информацией, возможность оценки и анализа текущего состояния финансов сделали ее популярной среди владельцев домашних компьютеров, держателей пластиковых карт, менеджеров по финансовому планированию.

Важным положительным качеством системы Decart Home Bank является подсистема безопасности, созданная на основе отечественных СКЗИ.

Описание подсистемы безопасности системы Decart Home Bank

В этой системе реализован комплекс программно-технических и организационных решений, направленных на предотвращение возможных атак потенциальных нарушителей. Данный комплекс включает:

- защиту от несанкционированного доступа;
- обеспечение конфиденциальности и целостности передаваемой информации с использованием средств криптографической защиты;
- регистрацию и учет действий пользователей и событий в системе.

Основу подсистемы защиты составляет СКЗИ «Крона» (разработка МО ПНИЭИ), однако наряду с ней могут применяться следующие средства защиты информации:

- средства типа брандмауэров (рекомендуются к применению в качестве дополнительных блоков, поскольку они не входят в состав поставки системы), обеспечивающие защиту от удаленных атак нарушителей;
- система оповещения администратора безопасности о нарушениях защиты;
- ведение журналов безопасности, позволяющих администраторам производить анализ функционирования средств защиты информации и оперативно выявлять возникающие проблемы;
- встроенные средства контроля, фиксирующие ошибочные действия оператора;
- механизм повторной передачи информации по требованию клиента (специальный запрос), использующийся для предотвращения потери данных, передаваемых и хранящихся в системе;
- парольная защита при входе в ПО «Декарт» и при обращении к криптосерверу (КС).

СКЗИ «Крона» позволяет выполнить следующие задачи информационной безопасности:

- конфиденциальность и целостность передаваемой информации на основе ГОСТ 28147-89;
- двустороннюю аутентификацию в режиме online на основе ГОСТ Р 34.10-94, ГОСТ 28147-89 и рекомендаций X.509;
- обеспечение юридической значимости передаваемых электронных документов на основе ГОСТ Р 34.10-94;
- единое управление ключевой структурой из единого центра.

Симметричная ключевая структура построена по принципу «звезда» – в центре находится банковская часть системы, а на периферии – клиенты. Соответственно каждый клиент имеет симметричные ключи, предназначенные исключительно для связи с банковской частью. Это приводит к тому, что защищенное взаимодействие можно устанавливать только с банком и при компрометации ключей у какого-то единичного пользователя не потребуется менять ключ во всей системе в целом. В качестве ключевых носителей могут использоваться гибкие магнитные диски, смарт-карты и touch memory.

Другой положительный момент – встраивание СКЗИ «Крона» происходит с применением криптосервера (о достоинствах данного подхода говорилось выше). Таким образом, в банковскую часть системы встраиваются только процедуры удаленного вызова функций КС с использованием разработанного МО ПНИЭИ CryptoAPI.

3.9.2. Теоретические основы электронных денег

Совершенно уникальные возможности для реализации возможностей электронных платежных систем предоставляют так называемые *электронные деньги*. Фактически они выступают электронным эквивалентом бумажных денег, хотя при этом у них есть и своя специфика:

- электронные деньги могут пересылаться с использованием Internet или обычных телефонных каналов;
- для их хранения могут использоваться как физические устройства (например, смарт-карты), так и рабочая станция пользователя;
- электронные деньги (как, впрочем, и любую другую информацию, представленную в электронном виде) можно легко размножить или скопировать, поэтому при построении систем, имеющих дело с подобной формой платежей, необходимо обеспечить уникальность каждой электронной банкноты;
- передача по общедоступным каналам связи приводит к тому, что появляется потенциальная возможность отследить, где, когда и какие средства были потрачены тем или иным субъектом. Понятно, что это обстоятельство приводит к необходимости разрабатывать особые меры по обеспечению анонимности плательщика. Кстати, подобная проблема возникает и в существующих на сегодняшний день электронных платежных системах, в которых при проведении платежа между покупателем и плательщиком обязательно участвует финансовая организация. В результате появляется еще одна инстанция, которая обладает всей полнотой информации о проводимых сделках и их деталях.

Вывод очевиден – создание электронных денег невозможно без повсеместного использования как широко известных криптографических механизмов (симметричные алгоритмы шифрования, хэш-функции и ЭЦП), так и совершенно новых идей, например затемняющей подписи и др.

Примеры построения платежных систем с использованием электронных денег

Пример 1

Представляемая здесь система построена с участием третьей стороны – банка или другой финансовой организации, в функции которой входит как создание электронных монет, так и проверка их подлинности и уникальности при проведении платежа между покупателем и продавцом.

Перед тем как прибегнуть к услугам предлагаемой системы, пользователь должен пройти процедуру регистрации, которая заключается в следующем:

1. Пользователь создает пару ключей для подписи – KP (открытый) и KP^1 (секретный).
2. Банк создает сертификат для пользователя, содержащий имя банка (B), имя пользователя (P), срок действия сертификата (e), сумму денежного лимита пользователя (S), открытый ключ пользователя (KP), и подписывает его на своем секретном ключе подписи (KB^1). В итоге сертификат имеет следующий вид:

   $CertP = \{B, e, S, P, KP\}_{KB^1}$.

3. Банк создает симметричный ключ (K) и разделяет его знание с пользователем.
4. Банк создает электронные монеты, представляющие 64-битные последовательности, зашифрованные на симметричном ключе банк-пользователь и содержащие серийный номер монеты и имя банка, выпустившего ее. Для простоты изложения предположим, что все монеты имеют одну и ту же стоимость.
5. Монеты объединяются в «пачки» по k монет в каждой, и каждая монета имеет следующий вид:

   $C_i = \{i \mid j \mid B\}_K$,

   где i – номер монеты в «пачке», j – номер «пачки» и | – операция конкатенации.
6. Монеты объединяются в «пачки», и банк их подписывает:

   $CS = C_1, C_2, ..., C_K, \{B, t, C_1, ..., C_K\}_{KB^1}$, где t – время подписи.

7. При осуществлении оплаты покупатель выбирает хэш-функцию h и производит следующие вычисления:

   $h_K = h(C_K)$
   for i = k – 1 downto1 do
   $h_i = h(C_i \mid h_{i-1})$.

8. Покупатель посылает продавцу CertP и следующее сообщение:

   $S_P = \{h_1, P, B, M, d\}_{KP^1}$,

   где M – имя продавца, а d – время подписи.
9. Продавец, используя сертификат покупателя, выданный банком, проверяет подпись покупателя.

Далее происходит сам процесс оплаты:

1. Покупатель, желая потратить монету C_1, посылает продавцу значения C_1 и h_2.
2. Продавец проверяет равенство $h_1 = h(C_1 \mid h_2)$, и в случае его выполнения монета C_1 принимается;
3. Если покупатель хочет потратить другие монеты, производятся аналогичные операции.

В ходе проверки легитимности монеты покупателя продавец передает в банк B S_P, полученные монеты C_1, ..., C_m и соответствующие хэш-коды h_1, ..., h_{m+1}. Банк проверяет каждую монету в отдельности, используя при этом формулу из шага 2, убеждается, что эти монеты не были потрачены раньше, и в случае подтверждения производит перевод денег на счет продавца.

Если продавец обслуживается в другом банке, нежели покупатель, то используются традиционные схемы взаимодействия между банком, выдавшим монеты, и банком продавца. Продавец представляет в свой банк следующие данные: M, P, B, d, S_P, $C_1...C_m$, $h_1...h_{m+1}$, которые в дальнейшем передаются через системы электронных платежей, например через клиринговую инфраструктуру.

Приведенная выше схема организации платежей на практике может быть реализована с использованием смарт-карт или других физических устройств, обеспечивающих хранение ключей пользователей, сертификатов, а также проведение вычислений (шаги 4 и 5). При этом физический вычислитель может использоваться в качестве еще одного уровня защиты от повторного применения электронных монет.

Недостатком подобной системы является отсутствие анонимности действий покупателя, то есть в банке могут отслеживать проходящие платежи.

Платежные системы обычно имеют многоуровневую защиту, обеспечивающую устойчивость системы к различным атакам и несанкционированным действиям легальных пользователей. Основная задача обеспечения информационной безопасности решается при организации самих платежных систем (например, банкноты должны быть надежно защищены от подделки). В существующих платежных системах данные функции ложатся на подсистему, связанную с бухгалтерским учетом проводимых операций. Однако многие подобные системы имеют и второй уровень защиты – банкноты нумеруются, а на кредитные карты наносится голограмма.

Организовать на качественно новом уровне защиту от активных несанкционированных действий, направленных на нарушение выбранной политики безопасности, можно только с использованием смарт-карт.

Безопасность в электронных платежных системах строится на разных уровнях. Первый уровень защиты основан на стойкости используемых криптографических механизмов (хэш-функций, ЭЦП и т.д.). Вторым уровнем является безопасность смарт-карт или других физических устройств, осуществляющих хранение ключевой информации и других данных. Третий рубеж – строгая подотчетность используемых электронных монет в организациях, обслуживающих их обращение.

Подобная организация системы безопасности имеет несколько разновидностей, но принципиальные моменты, как правило, совпадают. Иной подход к организации электронных платежных систем, построенных на основе электронных денег, был предложен Шаумом.

Пример 2

Работа данной схемы начинается с того, что банком выбираются и публикуются величины $N = pq$ (p и q, см. RSA), $N_1 = p_1q_1$, а также однонаправленная функция f(n), где n фактически является номером электронной монеты. Электронной монетой в целом в этих условиях считается пара $(n, f(n)^{1/S} \bmod N)$, где S характеризует номинал электронной монеты. Для значений S устанавливается соответствие с номиналом монеты – 1 (S = 3), 2 (S = 5), 3 (S = 3 × 5), 4 (S = 7), 5 (S = 3 × 7) и т.д.

Пользователь, желающий получить от банка электронную монету достоинством, например, 10 (S = 5 × 11) центов, начинает с того, что выбирает случайное значение n_1 и вычисляет $f(n_1)$. Далее пользователю нужно пройти в банке процедуру подписи электронной монеты, то есть вычислить корень степени S. Но пользователь не может послать $f(n_1)$, поскольку, произведя оплату конкретной монетой, банк в дальнейшем узнает ее и анонимность действий пользователя будет нарушена. Вот почему пользователь выбирает множитель r_1 (данное число называется затемняющим множителем), возводит его в степень S и отправляет банку значение $f(n_1)r_1^S \bmod N$. Получив эту величину, банк извлекает из нее корень степени S по модулю N и отправляет пользователю полученное значение, далее пользователь снимает затемняющий множитель и получает электронную монету $(n, f(n)^{1/S} \bmod N)$.

Процесс оплаты происходит по следующей схеме:

1. Покупатель должен заплатить продавцу 2 цента. Для этого он возводит электронную монету достоинством 10 центов в степень $1/11 \times \times (f(n_1)^{1/5} \bmod N)$, выбирает случайное число m и вычисляет значение $f(m)r_1^{11} \bmod N_1$.
2. Покупатель отправляет продавцу значения $f(n_1)^{1/S} \bmod N$ и $f(m)r_1^{11} \bmod N_1$.
3. Продавец передает данные значения банку.

4. Банк, используя список зарегистрированных монет, проверяет, действительно ли пара n_1, $f(n_1)^{1/5} \bmod N$ является монетой достоинством 2 цента и не была ли она потрачена раньше.
5. Банк возвращает покупателю сдачу, для чего возводит $f(m)r_1^{11}$ в степень 1/11 по модулю N_1.

Стойкость такой схемы основана на алгоритме RSA и связанной с ним математической задаче. Подобная схема обеспечивает анонимность покупателя тогда, когда в ходе платежа банку нет необходимости возвращать сдачу покупателю. В противном случае существует вероятность несколько раз подобно копилке использовать значение $f(m)r^S \bmod N_1$ (этот вариант в книге не рассматривается), и тогда у банка становится меньше шансов провести тотальное отслеживание действий покупателя.

3.9.3. Смарт-карты

Смарт-карты, или интеллектуальные карты, были изобретены и запатентованы французским инженером Роланом Мореном еще в середине 70-х годов. Хотя в финансовых институтах магнитные карты используются достаточно давно, в качестве носителей идентификационной информации или финансовых данных, активное внедрение смарт-карт в практику финансовых обменов началось только в конце 80-х годов. В настоящее время эти устройства широко применяются не только для оплаты телефонных разговоров, а также в системах электронных расчетов, но и для организации хранения криптографических ключей и другой идентификационной информации, что позволяет пользоваться ими в системах контроля и с целью разграничения доступа к объектам или информации. Широкий спектр и масштабы применения смарт-карт были обусловлены их удобством, надежностью и многофункциональностью.

Смарт-карта – пластиковая карта со встроенным микропроцессором, выполняющим функции контроля доступа к памяти смарт-карты и производящим также ряд специфических функций. Главная особенность смарт-карт в том, что в них осуществляется не только хранение, но и обработка содержащейся информации.

Архитектура смарт-карты представлена на рис. 3.38 и состоит из следующих компонентов:

- микропроцессор (CPU). На сегодняшний день смарт-карты имеют 8-битный CPU;
- ОЗУ (RAM). Память для временного хранения данных, например для результатов вычислений, произведенных CPU. Типовая смарт-карта имеет объем памяти 128–256 Кб;

- ПЗУ (ROM). Память для хранения ОС карты, а также данных, которые не изменяются в процессе работы с ней. Информация в ПЗУ записывается на этапе выпуска смарт-карты. Обычно ПЗУ имеет объем до 24 Кб;
- ППЗУ (EPROM). Эта память может быть прочитана много раз, но запись на нее осуществляется только единожды. Данные в ППЗУ при отключении питания не теряются;
- ЭСППЗУ (EEPROM). Информация в этот участок может быть многократно перезаписана и считана. Данные в ЭСППЗУ при отключении питания не теряются;
- система ввода/вывода (I/O). Система для обмена данными с внешними устройствами;

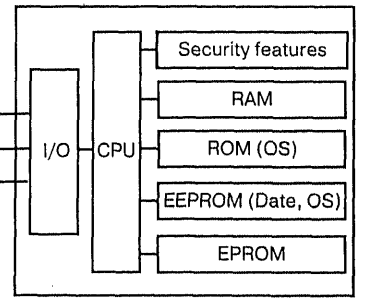

Рис. 3.38. Архитектура построения смарт-карты

- операционная система (ОС). ОС смарт-карты принципиально не отличается от ОС компьютера;
- система безопасности (Security features). Встроенная система безопасности для защиты данных, хранящихся и обрабатываемых в смарт-карте, может быть выполнена в виде криптографического сопроцессора, осуществляющего функции криптографического преобразования данных.

По методам сочетаемости с внешними устройствами смарт-карты могут быть разделены на следующие классы:

- контактные. На сегодняшний день получили наибольшее распространение. Взаимодействие с карт-ридером происходит на основе физического соприкосновения металлических контактов смарт-карты с контактами, имеющимися на карт-ридере;
- бесконтактные. Считывание и запись информации на карте осуществляется с помощью радиосигнала, передаваемого и принимаемого индуктивной катушкой смарт-карты. Расстояние между картой и считывающим устройством может колебаться от нескольких миллиметров до нескольких метров в зависимости от используемой конструкции;

Необходимо подчеркнуть, что основными областями применения смарт-карт являются:

- системы контроля доступа;
- хранение данных (в том числе и криптографических ключей);
- финансовые карты.

Поскольку второй и третий варианты применения смарт-карт рассматривались в предыдущих разделах, теперь остановимся на варианте, связанном с системами контроля доступа.

Карты контроля используются, чтобы обеспечить соответствующим лицам физический доступ в здания, комнаты, к автомобильным стоянкам, загражденным территориям и т.п., а также получить логический доступ к компьютеру или информации, содержащейся в нем.

Карты логического доступа позволяют управлять доступом к компьютерам и компьютерным сетям. Для получения разрешения пользователь должен вставить смарт-карту в ридер и ввести персональный идентификационный номер (PIN-код). Программное обеспечение позволяет установить несколько уровней безопасности, все они управляются системным администратором. Определенный уровень допуска может быть присвоен всем пользователям, другой – группе пользователей. Отдельным пользователям может устанавливаться индивидуальная форма допуска.

Смарт-карты в системах безопасности могут реализовать следующие функции:

- идентификацию и аутентификацию в системах безопасности;
- хранение ключей, сертификатов и профилей;
- выполнение операций шифрования/расшифрования;
- выполнение операций с ЭЦП (генерация и проверка).

Обеспечение безопасности при использовании смарт-карт

Очевидно, что безопасность смарт-карт во многом зависит от надежности их хранения. Злоумышленник, воспользовавшись чужой смарт-картой, сможет провести либо незаконные манипуляции с денежными средствами, хранящимися на карте, либо, если смарт-карта используется в качестве ключевого носителя, получить доступ к секретным данным.

Для обеспечения защиты от незаконных манипуляций со смарт-картами (при применении их в платежных системах) служат следующие механизмы:

- двусторонняя аутентификация смарт-карта/кард-ридер с использованием криптографических протоколов типа «запрос-ответ». Для реализации такого подхода необходимо, чтобы смарт-карта поддерживала криптографические механизмы;
- аутентификация пользователя смарт-карты. Реализуется за счет введения PIN-кода (кода персональной идентификации), хранящегося на карте пользователя. Очевидно, что PIN-код должен храниться пользователем в тайне. В некоторых типах смарт-карт используются

биометрические методы идентификации пользователей. Данный тип аутентификации необходим для авторизации проведения тех или иных операций со смарт-картой в платежных системах;

В связи с вышеизложенным подчеркнем, что возможность использования злоумышленником смарт-карты зарегистрированного пользователя зависит от надежности хранения PIN-кода и стойкости протоколов аутентификации. Правда, существует ряд методов, которые позволяют считать информацию, содержащуюся в смарт-карте, как получив к ней доступ, так и не имея непосредственного доступа к смарт-карте зарегистрированного пользователя.

Подробные описания конкретных атак на смарт-карты не входят в задачу книги; остановимся лишь на общих принципах их проведения:

- использование анализа сбоев в узлах смарт-карты, осуществляющих хранение и обработку криптографических данных. В этом случае нарушитель овладевает зашифрованными данными, полученными при возникновении ошибки, и теми же данными, зашифрованными на том же ключе, но при отсутствии сбоев в работе. Далее, используя математический аппарат с подзаголовоком «дифференциальный криптоанализ», вычисляет секретный ключ шифрования. С теоретической точки зрения атаке подобного типа подвержены все известные на сегодняшний блочные алгоритмы шифрования, но практическая сторона, или точнее – реальные возможности осуществления угрозы, на сегодняшний день не позволяют злоумышленнику впрямую воспользоваться этим методом. Очевидно, что сбой необходимо вызвать именно в той области, где находятся криптографические данные, что, в общем, маловероятно. С большей долей уверенности можно говорить, что сбой произойдет в узлах, где хранятся другие данные или ОС;
- атаки на смарт-карты, позволяющие получить данные, содержащиеся в их памяти. Атаки основаны на манипуляциях с тактовой частотой или напряжением питания; доказательством может служить увеличение тактовой частоты от 5 до 20 МГц или скачок напряжения. Данные атаки применяются по отношению к смарт-картам, использующимся в платном телевидении;
- анализ побочных сигналов, возникающих в ходе работы со смарт-картой, цель которых – выявление зависимости внешних сигналов или излучений от характера обрабатываемой информации. В качестве побочных каналов может выступать как электромагнитный канал, так и сигналы, возникающие в цепях питания смарт-карты;

- физическое вскрытие смарт-карты с целью получения доступа к хранящейся информацией. Для реализации данного подхода необходима высокочувствительная и дорогостоящая аппаратура.

Возможность подобных атак обусловлена тем, что в смарт-картах не реализованы средства обеспечения безопасности их работы, применяемые в сложных приборах, выполняющих функции шифрования, например устройствах тестирования работоспособности узлов смарт-карты или проверки выходных данных на предмет их корректности.

Очевидно, что уровень безопасности бесконтактных смарт-карт значительно ниже уровня безопасности контактных смарт-карт, поскольку в этом случае злоумышленнику может быть доступен протокол обмена с карт-ридером и данные, возникающие в ходе подобного обмена. Кроме того, злоумышленник сможет осуществлять навязывание ложной информации при использовании бесконтактных смарт-карт.

3.9.4. Средства обеспечения безопасности электронных платежных систем

При использовании электронных платежных систем, работающих с использованием Internet, кроме специфических вопросов, связанных с применением электронных денег (о них было сказано выше), возникают также проблемы обеспечения конфиденциальности, целостности и юридической значимости передаваемой через открытые каналы финансовой информации. Следует отметить, что конфиденциальность определенных финансовых сведений должна быть обеспечена и по отношению к определенным участникам электронной платежной системы. Например, информация о счете покупателя должна оставаться недоступной продавцу; доступ к ней должны иметь только финансовые организации, обслуживающие данного покупателя.

Обеспечение вышеперечисленных задач может осуществляться как с применением стандартных средств, описанных в разделе 3.4 «Защита информации при межсетевом взаимодействии», так и с помощью разработанных для этой цели криптографических средств защиты информации. Поскольку большинство существующих на сегодняшний день электронных платежных систем и систем электронной коммерции основываются на применении Web-технологий, то для защиты могут применяться такие средства, как SSL, PCT, Kerberos и др.

Обеспечение юридической значимости передаваемых финансовых данных и электронных документов осуществляется с использованием ЭЦП и развертывания системы, обеспечивающей разбор конфликтных ситуаций, связанных с применением ЭЦП.

Однако существуют специально разработанные средства, обеспечивающие защиту передаваемой финансовой информации. Примером может служить протокол безопасных электронных платежей – SET.

Система Secure Electronic Transaction

Secure Electronic Transaction (SET) – это не средство защиты финансовой информации, а прежде всего технология, обеспечивающая сохранение тайны и достоверности сделки между продавцом и покупателем с использованием кредитных карт (при этом в процесс обмена информацией вовлекаются финансовые учреждения).

SET используется для обеспечения:

- конфиденциальности передаваемой финансовой информации, то есть подобные данные должна быть доступны только указанному адресату;
- сохранности передаваемых данных. Участники электронных расчетов должны быть уверены в неизменности передаваемых платежных поручений;
- аутентификации пользователей электронной платежной системы; при этом используется как аутентификация плательщика, так и аутентификация номера счета;
- авторизации действий плательщика, то есть для подтверждения права проводить финансовые операции.

Для реализации перечисленных выше задач обеспечения информационной безопасности в электронных платежных системах в SET используются следующие механизмы:

- симметричное (используется для шифрования данных) и асимметричное (используется для шифрования сеансовых ключей) шифрование;
- ЭЦП;
- хэш-функции;
- двойные ЭЦП. Используются для связи сообщения с заказом, отправленным продавцу, с платежными инструкциями, содержащими информацию о счете, переданную финансовой организации. Продавец отправляет запрос на авторизацию финансовой организации, который включает платежные инструкции, посланные ему покупателем, и хэш-код сообщения с информацией о заказе;
- сертификаты открытых ключей подписи и шифрования. Для обеспечения надежного функционирования сертификатов открытых ключей пользователей применяется древовидная иерархия сертификации. В дополнительные параметры, присутствующие во всех известных типах сертификатов открытых ключей для данной системы, заносятся данные о номерах счетов пользователей и сроках действия сертификатов.

В общем случае протокол взаимодействия покупателя с продавцом выглядит следующим образом:

1. Продавец (обозначим его А) собирает информацию о предложении и вычисляет хэш-код на данную информацию (H(M)), этот хэш-код будет в дальнейшем использоваться для идентификации параметров сделки, и создается ЭЦП на данное предложение (S(H(M))). Продавец также генерирует симметричный сеансовый ключ К, на котором зашифровывается хэш-код, ЭЦП и сертификат открытого ключа подписи продавца ($Certs_A$). Ключ К зашифровывается с использованием открытого ключа шифрования покупателя (PK_D). Покупателю (В) отправляется следующее сообщение:

   $A \rightarrow B$: {H(M), S(H(M)), $Certs_A\}_K$, $\{K\}_{PKB}$.

2. Покупатель, получив сообщение, расшифровывает сеансовый ключ К посредством своего секретного ключа. Используя данный ключ, расшифровывает первую часть сообщения, проверяет ЭЦП с помощью открытого ключа продавца, расположенного в $Certs_A$, и на основе той же хэш-функции получает информацию о предложении. По результатам проведения проверок продавцу возвращается соответствующая информация.

Например, если покупатель хочет предложить продавцу купить товар и одновременно направить в свой банк информацию о проведении платежа в том случае, если продавец примет его предложение, но при этом не хочет, чтобы продавец получил информацию о его банковских реквизитах, а банк получил информацию о заказе. Покупатель может выполнить эти условия, используя механизм двойных подписей. То есть покупатель генерирует ЭЦП под двумя сообщениями (в банк и продавцу) с помощью одной операции генерации ЭЦП.

Двойная подпись создается путем генерации хэш-кодов для двух сообщений (H(M1) и H(M2)), вычисления итогового хэш-кода (H(M3)) от первых двух хэш-кодов и генерации ЭЦП для трех хэш-кодов (S(H(M1)), S(H(M2)), S(H(M3))). После чего покупатель направляет по два сообщения в банк и продавцу.

Система защиты информации

Определенный интерес для защиты представляет разработка фирмы «АйТи», обеспечивающая разграничение доступа и организацию защищенного обмена информацией между пользовательскими приложениями.

Достоинством системы является легкая интегрируемость в прикладное ПО. Фактически данное средство защиты информации представляет собой набор динамических подключаемых библиотек (DLL) с реализованными в них функциями по организации защиты информации. Для подключения к прикладному ПО необходимо на этапе его разработки встроить в него обращение к функциям, реализованным в DLL. Полностью документированный интерфейс позволяет использовать библиотеки с ПО, написанным на Visual C++, Visual Basic, Delphi, Java 1.1 и др.

Система защиты информации работает по схеме «клиент-сервер». Хранение секретной информации осуществляется на серверной стороне в базе данных, а на клиентской – на смарт-картах пользователей.

Системе присущи следующие функциональные возможности:

- аутентификация пользователей;
- обеспечение конфиденциальности и целостности передаваемой информации;
- административные функции в отношении пользователей (добавление/удаление пользователей, изменение прав доступа пользователей, генерация ключей с записью на смарт-карты и базу данных, просмотр информации по пользователям и их смарт-картам).

Структура защищенного обмена информацией и этапы прохождения сообщения представлены на рис. 3.39.

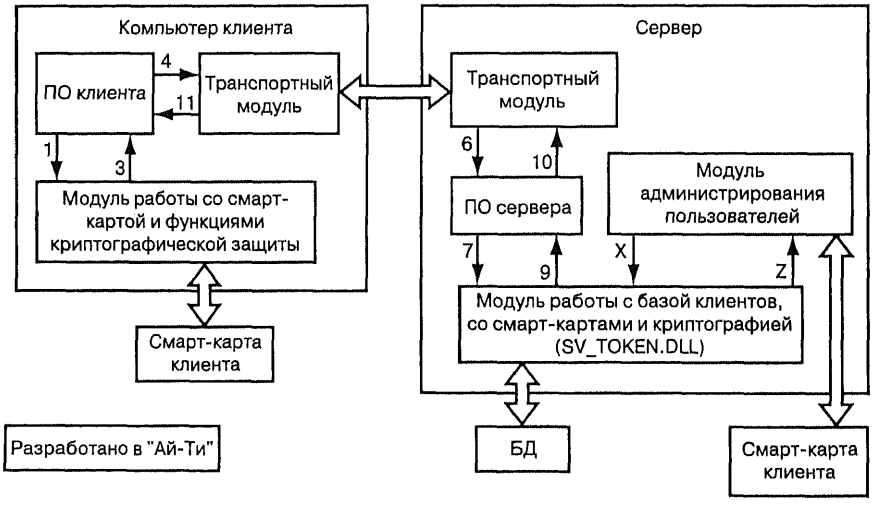

Рис. 3.39. Структура защищенного обмена информацией

Передача зашифрованного сообщения с ЭЦП от клиента к серверу производится следующим образом:

1. Клиентское ПО приглашает клиента идентифицировать себя (ввести PIN-код) и передает эту информацию модулю cl_token.dll, который работает со смарт-картами и функциями криптографической защиты.
2. Модуль идентифицирует клиента. Если введенный PIN-код и PIN-код, записанный на карте, совпадают, то модуль зашифровывает и подписывает сообщение с помощью ключей, хранящихся на карте.
3. Модуль возвращает в ПО зашифрованное сообщение с ЭЦП.
4. Клиентское ПО передает зашифрованное сообщение с ЭЦП транспортному модулю для передачи его на сервер.
5. Транспортный модуль клиента передает сообщение транспортному модулю сервера по каналу связи.
6. Серверное ПО получает зашифрованное сообщение с ЭЦП от транспортного модуля.
7. Сообщение передается модулю sv_token.dll, который работает с базой клиентов и функциями криптографической защиты.
8. Модуль расшифровывает сообщение и проверяет ЭЦП, используя информацию о клиенте.
9. Модуль возвращает серверному ПО расшифрованное сообщение и код результата проверки ЭЦП.

Передача зашифрованного сообщения с ЭЦП от сервера к клиенту выглядит следующим образом:

1. Серверное ПО подготавливает сообщение для передачи определенному клиенту. Затем передает эту информацию модулю sv_token.dll, который работает с базой клиентов и функциями криптографической защиты.
2. Модуль зашифровывает и подписывает сообщение с помощью ключей, хранящихся в базе данных.
3. Модуль возвращает приложению зашифрованное сообщение с ЭЦП.
4. Серверное ПО передает зашифрованное сообщение с ЭЦП транспортному модулю.
5. Транспортный модуль сервера передает сообщение транспортному модулю клиента.
6. Клиентское ПО получает зашифрованное сообщение с ЭЦП от транспортного модуля.
7. Сообщение передается модулю cl_token.dll для расшифрования и проверки ЭЦП.

8. Модуль расшифровывает сообщение и проверяет ЭЦП с помощью информации о сервере, хранящейся в карте клиента.
9. Модуль возвращает ПО клиента, расшифрованное сообщение и результат проверки ЭЦП.
10. Добавление нового клиента и выпуск карты с помощью утилиты администрирования:
11. Администратор системы с помощью модуля администрирования вводит информацию о новом клиенте и передает ее серверному модулю.
12. Модуль создает запись о новом клиенте, генерирует сертификат клиента (а также открытый ключ клиента) и сохраняет его в базе данных по картам и владельцам.
13. Кроме этого, модуль генерирует закрытый ключ клиента и записывает его вместе с сертификатом на карте клиента.
14. Модуль возвращает серверному ПО информацию о результатах операции.

В заключение хотелось бы сказать, что тематика, которой посвящена представленная работа, далеко не исчерпывается рассмотренными вопросами, да и рамки одной книги не позволяют детализировать многие из изложенных здесь тем. Для тех, кто интересуется проблемами компьютерной безопасности и криптографическими методами защиты информации, можно порекомендовать поискать нужные сведения в Internet (список наиболее интересных, по мнению автора, Internet-ресурсов приведен ниже) либо обратиться к дополнительной литературе (список рекомендованной литературы тоже прилагается).

Приложение 1

Сравнительные характеристики отечественных средств построения VPN

Для сравнительных испытаний были выбраны некоторые продукты из виртуальных приватных систем (VPN). В перечень вошли средства организации VPN на третьем уровне модели OSI (сетевом уровне), главную роль в котором в современных корпоративных сетях играет проткол IP. Это ФПСУ-IP от фирмы Амикон, «Застава» от ОАО «Элвис-плюс» и «Шип» от МО ПНИЭИ.

Продукты оценивались по их работе на различных платформах, полноте набора функций управления доступом и обеспечения защиты корпоративной системы; по величинам накладных расходов, удобству администрирования, производительности, возможностям обеспечения собственной безопасности и ценовым характеристикам. Сравнительные характеристики систем и результаты сравнения приведены в табл. П.1.1 и П.1.2.

1. «ФПСУ-IP». Производитель – ООО «АМИКОН», г. Москва

В настоящее время близки к завершению работы по созданию подсистемы централизованного удаленного управления и мониторинга. Повышенные требования к обслуживающему персоналу способствовали резкому снижению эффективности ФПСУ-IP. Этот недостаток отмечался как один из существенных, заметно ограничивающих сферу применения всей системы. Дело в том, что при отсутствии централизованного удаленного управления в подразделении, использующем ФПСУ-IP, должен находиться квалифицированный персонал для выполнения функций администратора системы, в полной мере владеющий технологиями и методами защиты

информации в IP-сетях. К тому же необходимо обеспечить постоянную возможность оперативного управления и мониторинга ФПСУ-IP. С учетом сказанного использование ФПСУ-IP для защиты территориально распределенных сетей представляется достаточно проблематичным, так как в удаленных подразделениях обычно не хватает квалифицированных специалистов. Кроме того, в настоящее время стандартом является централизованное управление сетевыми и информационными ресурсами с помощью платформ сетевого управления типа OpenView и протокола SNMP.

Документация на систему выполнена в объеме, достаточном для администрирования и сопровождения продукта, но не лишена недостатков, к которым можно причислить отсутствие примеров построения системы защиты на основе ФПСУ-IP. Также имеется ряд мелких недочетов, не влияющих на полноту документации.

При проверке ФПСУ-IP на устойчивость к имеющимся в настоящее время атакам на стек TCP/IP применялся программный продукт фирмы ISS Internet Security Scanner 5.6 для Windows NT. После проведения всех атак типа «отказ в обслуживании» (denial-of-service) работоспособность ФПСУ-IP сохранилась.

Комплекс ФПСУ-IP продемонстрировал великолепные скоростные характеристики при сжатии, шифровании трафика и построении VPN. При использовании в качестве аппаратной части компьютеров Pentium 200 с 32 Мб ОЗУ максимальная скорость передачи IP-трафика составила для режима шифрования 12 Мбит/с, а при включении сжатия – 10,7 Мбит/с.

Зависимость скорости передачи информации по протоколу TCP от MTU для WAN 64 Кбит/с (Frame Relay) в различных режимах (шифрование, сжатие, совместный режим) изображена на рис. П.1.1.

ФПСУ-IP осуществляет сжатие трафика при MTU 1500 байт, обеспечивая для каналов 19,6, 64 и 256 Кбит увеличение скорости передачи информации в 2,4 раза для текстовой передачи и в 1,17 раза для передачи заархивированных данных. При уменьшении MTU до 128 байт сжатие становится неэффективным и не дает никакого выигрыша в скорости обмена, но и не ухудшает скоростных характеристик канала. ФПСУ-IP обеспечивает шифрование трафика, добавляя накладные расходы в количестве 18–20 байт на несжимаемый пакет. Для каналов 19,6, 64 и 256 Кбит разница в скоростях обмена при использовании шифрования и без него для MTU 1500 составляет 2%, для MTU 128 – 28%.

Для ЛВС 100 Мбит применение ФПСУ-IP в режиме ретрансляции (без шифрования и сжатия) уменьшает скорость обмена в два раза при MTU 1500 и в 2,5 раза для MTU 128, что является классическим случаем использования дополнительного маршрутизатора. При включенном

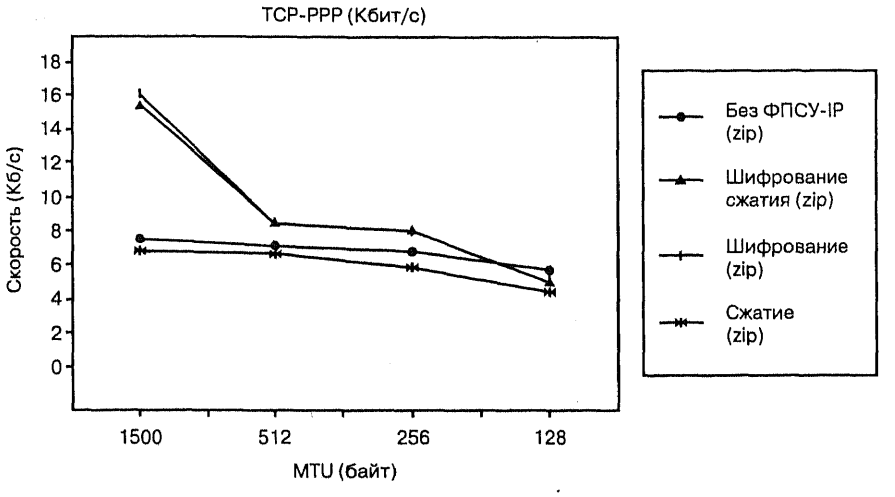

Рис. П.1.1. Зависимость скорости передачи данных по протоколу TCP от MTU

сжатии и шифровании скорость уменьшается в 12 раз для MTU 1500 и 128 байт.

Все сказанное позволяет сделать вывод о неэффективности применения ФПСУ-IP для разделения сегментов ЛВС 100 Мбит с использованием процессоров, задействованных во время испытаний.

То, что механизмы защиты ФПСУ-IP не удовлетворяют стандартам IPSec, делает невозможным совместное использование ФПСУ-IP и остальных шифраторов IP-протоколов и средств организации VPN других производителей, ориентирующихся на использование IPSec.

Из недостатков ФПСУ-IP необходимо отметить следующие: во время произведения настроек комплекс находится в нерабочем состоянии, то есть сетевой трафик через него не проходит, не реализована возможность использования DNS, что затрудняет анализ статистической информации.

Также необходима реализация подсистемы синхронизации времени между распределенными ФПСУ-IP. Ее отсутствие не позволяет четко контролировать выполнение правил фильтрации по времени.

Требуется исследовать вопрос о целесообразности использования ФПСУ-IP с более чем двумя сетевыми картами, что, возможно, расширит область применения ФПСУ-IP при организации виртуальных сетей в рамках корпоративных сетей.

Стабильность и корректность работы ФПСУ-IP зависит от конкретной сетевой платы и аппаратного обеспечения. При использовании некоторых драйверов сетевых карт наблюдались фатальные ошибки и сбои в работе ФПСУ-IP. Таким образом, для обеспечения стабильного и надежного

функционирования этой системы разработчику рекомендуется производить поставки ФПСУ-IP в виде законченного аппаратно-программного комплекса («под ключ»), включающего в себя аппаратное устройство с предустановленным ПО и сетевыми картами.

Для обеспечения возможности создания более гибких VPN представляется целесообразным разработка клиентской части ФПСУ-IP под Windows 95/98/NT для обеспечения возможности организации VPN внутри одного сегмента ЛВС и с удаленных станций. При реализации клиентской части необходимо предусмотреть работу пользователей по протоколу Mobile IP. Также полезной была бы реализация открытого криптоинтерфейса и описание требований к встраиваемому криптоядру. Из положительных качеств следует сказать о довольно низкой цене ФПСУ-IP.

В заключение заметим, ФПСУ-IP показал себя надежным, динамично развивающимся продуктом, который в ближайшем будущем станет хорошим средством организации корпоративных VPN.

2. «Застава». Производитель – ОАО «Элвис-Плюс», г. Зеленоград

По результатам тестовых испытаний продукт «Застава» версии 2.01 зарекомендовал себя как хорошее, вполне пригодное для создания VPN программное обеспечение. СЗИ «Криптон-Застава» надежна, достаточно проста, удобна в эксплуатации и обладает практически всеми функциональными возможностями, описанными в документации на комплекс программных компонентов. Средства управления дают возможность с достаточной гибкостью заниматься администрированием всего программного комплекса, в том числе и ключевой системы. Система регистрации обеспечивает сбор и анализ необходимой статистической информации.

Представленная документация на СЗИ «Криптон-Застава» в полном объеме описывает администрирование всего программного комплекса, однако в ней отсутствуют разделы, содержащие анализ аварийных ситуаций и рекомендации по восстановлению работоспособности системы, а также квалификационные требования к обслуживающему персоналу. Кроме того, разработчику необходимо реализовать обозначенные в документации некоторые возможности программных компонентов СЗИ «Застава», а именно: журналирование пакетов в ПС «Застава-Офис» и графическую оболочку администрирования МЭ «Застава».

На испытаниях было обнаружено, что особенности протокола SKIP, в частности требование передачи пакетного ключа вместе с каждым

IP-пакетом, а это 112 байт служебной информации для ГОСТа, приводят к значительному возрастанию служебного трафика во время использования некоторых клиент-серверных технологий. При превалировании в сети пакетов длиной равной или близкой MTU (Maximum Transfer Unit, для Ethernet это 1500 байт) подобная нагрузка может и не считаться избыточной – примерно 10% от общего объема трафика, однако для большинства клиент-серверных технологий средняя длина IP-пакета в сети намного меньше и служебный трафик SKIP может составлять больше половины от всего сетевого IP-трафика. Это сказывается на производительности вычислительных систем, работающих в распределенной среде с использованием низкоскоростных (меньше 64 Кбит) каналов связи, связывающих подсети корпоративной сети. Зависимость скорости передачи данных по протоколу FTP от MTU для WAN 64 Кбит/с (Frame Relay) с использованием и без использования криптографической защиты канала с помощью «Заставы» приведена на рис. П.1.2. Для опытов с WAN были задействованы рабочие станции SunUltra 5 277 МГц с 64 Мб ОЗУ и установленной ОС Solaris 2.6.

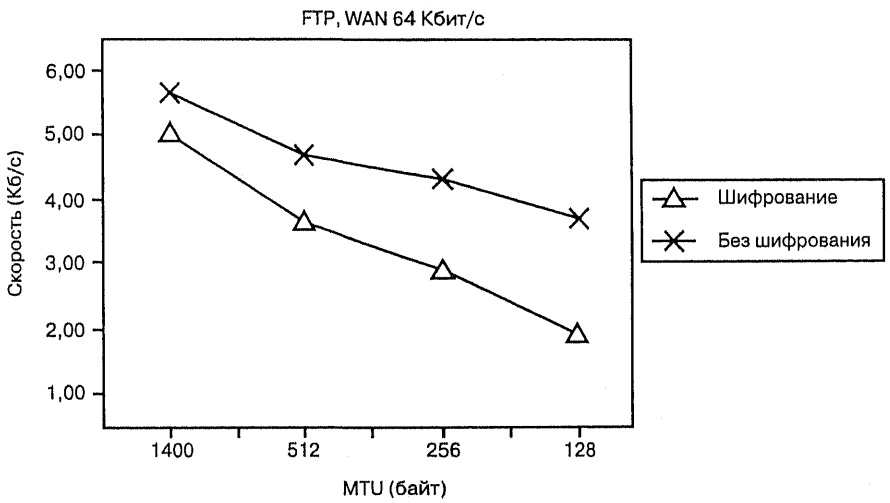

Рис. П.1.2. Влияние «Заставы-Офис» на производительность WAN

Из этого рисунка видно, что в канале WAN при уменьшении размера IP-пакета скорость передачи данных падает более чем в два раза, что вызвано ростом служебного трафика при сохранении полосы пропускания. Максимальная скорость шифрования, декларируемая разработчиком, составляет 8 Мбит/с.

К недостаткам также можно отнести невозможность «Заставы-Офис» версии 2.01 работать с фрагментированными IP-пакетами (с установленным флагом IP-заголовка DF). Такие пакеты системой просто игнорируются. Это может вызвать проблемы при взаимодействии различных телекоммуникационных сред, характеризующихся разными MTU. В настоящее время эти проблемы решены в версии 2.5. Кроме того, по сравнению с другими средствами защиты информации, располагающимися в том же сегменте рынка, что и «Застава», данный продукт имеет более высокую цену. Производителем декларируется совместимость SKIP-продуктов серии «Застава» с продуктами других компаний (среди них SUN и CheckPoint). Однако SKIP-продукты «Застава» различных версий, например 1.62 и 2.01, несовместимы между собой при использовании ГОСТа в качестве алгоритма шифрования по причине того, что в версии 2.01 были исправлены отдельные ошибки реализации SKIP. Отметим, что обновление версии в этом случае производится бесплатно.

Разработчику также необходимо предусмотреть дополнительную защиту (возможно, криптографическую) от несанкционированного доступа ключевых материалов системы, защита которых в настоящих версиях возложена на штатные средства Solaris.

Следует отметить надежность и управляемость «Заставы». В течение года непрерывной эксплуатации сбои программного обеспечения наблюдались всего дважды, и они не приводили к фатальному краху межсетевого взаимодействия. Так как смена пакетных ключей комплекса производится по умолчанию через равные промежутки времени (120 с) либо при прохождении 102400 байт по соединению, необходимо обеспечить синхронизацию таймеров взаимодействующих станций. Это можно сделать с использованием NTP (Network Time Protocol). Межсетевой экран «Застава» поддерживает управление с помощью протокола SNMPv1 и v2, что позволяет интегрировать его в корпоративную систему сетевого управления типа HP OpenView.

Пакет «Застава-Офис» имеет графические средства администрирования, реализованные с помощью Java RMI. Они довольно удобны в работе и позволяют полностью выполнить настройку системы. Однако Java RMI весьма чувствителен к пропускной способности сети, и более пригодным может оказаться использование утилит командной строки системы защиты «Застава».

Наличие сервера сертификатов делает управление ключевой системой достаточно гибким, особенно по сравнению со схемой таблиц ключей, которая используется в ФПСУ-IP. Так, в частности, допускается наличие

нескольких центров генерации и сертификации ключевой информации. Однако плановую смену ключей все равно придется производить вручную, причем на каждом межсетевом экране. Этот процесс может осуществляться централизованно с рабочего места администратора по защищенным каналам с помощью удаленного терминала и утилит командной строки «Заставы».

В заключение следует сказать, что комплекс «Застава» постоянно развивается, и уже в следующих версиях в нем планируется полностью реализовать вырабатываемые в настоящее время стандарты IPSec, что позволит строить и управлять VPN более гибко. Высокой оценки заслуживает надежность и управляемость комплексом, на основе которого строятся корпоративные и межкорпоративные VPN.

3. «Шип». Производитель – МО ПНИЭИ, г. Москва

Результаты испытаний системы «Шип» показали, что в целом этот комплекс реализует характеристики, отраженные в документации, однако у него есть и существенные недостатки. Например, при плановой смене ключей наблюдались разрывы связи, доходившие до нескольких минут, что, конечно, может привести к нарушению работы сетевых приложений.

Отсутствие поддержки протоколов динамической маршрутизации (например, RIP), в случае отказа АПК «Шип», не позволяет перейти на резервную конфигурацию связи по открытому каналу без перекоммутации сетевых кабелей. Вариант обхода неисправного АПК «Шип» с использованием резервного маршрутизатора LAN-LAN требует наличия в холодном резерве дополнительного сетевого оборудования в каждом филиале. Заметим также, что через «Шип» не может транслироваться незашифрованный трафик, поэтому с его помощью трудно построить многочисленные гибкие VPN смешанного назначения и обеспечить из внутренней сети одновременный доступ к защищенным комплексам «Шип» и общедоступным ресурсам сети. Кроме того, обнаружилось, что затруднительно управлять оборудованием региональной сети из локальной сети организации, так как сетевое оборудование находится за комплексом «Шип». Для того чтобы справиться с этой проблемой, придется резервировать дополнительные порты сетевого оборудования, из-за чего увеличивается общая стоимость сети. «Застава» и ФПСУ-IP имеют возможность транслировать незашифрованный трафик и тем самым поддерживать отдельную VPN, служащую только для защиты системы управления корпоративной сетью.

Необходимость постоянной связи с центром управления ключевой системой (ЦУКС) также нельзя отнести к достоинствам этого комплекса, потому что «Шип» осуществляет периодический (15 мин) опрос ЦУКС. Разрыв этого канала может привести к нарушению функционирования всей сети. Продукты же типа «Застава», имея загруженные в локальные базы данных сертификаты, могут обходиться без сервера сертификатов. ФПСУ-IP пока не предусматривает наличия централизованного управления ключами. Отметим, что ЦУКС «Шип» распространяет таблицы соответствия ключей и абонентов, а не сами ключи. Так что при использовании всей ключевой матрицы новую таблицу все равно придется ставить на каждый «Шип» вручную.

Максимальная скорость шифрования, декларируемая разработчиком, составляет, как и у «Заставы», 8 Мбит/с.

Документация на «Шип» является самой полной по сравнению с описанными выше продуктами, хотя и не лишена мелких недочетов.

Так как «Шип» использует SKIP, то недостатки, которые описаны при анализе «Заставы», в равной мере относятся и к «Шипу».

По стоимости «Шип» занимает промежуточное положение между ФПСУ-IP и «Заставой». В отличие от ФПСУ-IP в составе комплекса «Шип», по заявлению разработчиков, имеется клиентское программное обеспечение, разработанное для Windows NT, хотя, скорее всего, реализация под Windows 95 тоже могла бы быть полезной.

После доработки и устранения недостатков, в частности увеличения надежности и управляемости, «Шип» может стать хорошим средством защиты корпоративных сетей.

4. Заключение

По результатам проведенных стендовых испытаний и опытной эксплуатации в региональных сетях наиболее пригодным для создания внутрикорпоративной VPN считается ФПСУ-IP. Вместе с тем лучшим для реализации системы управления и распределения сертификатов, а также наиболее соответствующим стандартам IPSec для обеспечения безопасности межкорпоративной территориально распределенной сети, на взгляд авторов приложения, является комплекс «Застава». АПК «Шип» может использоваться в корпоративных сетях среднего размера с устойчивыми каналами связи между узлами системы и центром управления ключевыми системами.

Следует отметить, что общий уровень отечественных продуктов организации VPN является довольно высоким, и есть надежда, что развитие рынка таких систем сделает возможным реализацию надежной защиты корпоративных сетей и обеспечит безопасное межкорпоративное взаимодействие через сети общего пользования, в том числе и через Internet.

Таблица П.1.1. Сравнительные характеристики Firewall/VPN-систем

	ШИП	Застава	ФПСУ-IP
Общие сведения			
Производитель, поставщик	МО ПНИЭИ	ЭЛВИС-плюс	АМИКОН
Сертификация Гостехкоммисией при Президенте РФ или ФАПСИ	Сертификат ФАПСИ	класс 3 № 145 от 14.01.98 г. (под Solaris) ЭЛВИС-плюс. Сертифицировано серийное производство	класс 3 № 233 от 14.04.99 г. Сертифицировано серийное производство
Используемые ОС	FreeBSD	Windows NT/95/98, Sparc/Intel Solaris	Собственная ОС
Использование отечественных криптографических стандартов для организации VPN-сервисов	ГОСТ 28147-89	ГОСТ 28147-89	ГОСТ 28147-89
Максимальное количество сетевых интерфейсов	5	5	2
Функции управления доступом (фильтрация данных и трансляция адресов)			
Фильтрация пакетов служебных протоколов, служащих для диагностики и управления работой сетевых устройств	Да	Да	Да
Фильтрация с учетом входного и выходного интерфейса для проверки подлинности адресов	Да	Да	Да
Фильтрация с учетом любых значимых полей сетевых пакетов	Да	Да	Да
Фильтрация на транспортном уровне запросов на установление виртуальных соединений с учетом транспортных адресов	Да (TCP/ UDP)	Да (TCP/ UDP)	Да (TCP/ UDP)
Фильтрация на прикладном уровне запросов к прикладным сервисам	Нет	Нет	Нет
Фильтрация с учетом даты/времени	Нет	Нет	Да
Трансляция номеров портов/сетевых адресов	-/+	-/+	+/+
Возможность сокрытия субъектов (объектов) и/или прикладных функций защищаемой сети	Да	Да	Да
Возможность скрытия межсетевого экрана	Нет	Нет	Да

Таблица П.1.1. Сравнительные характеристики Firewall/VPN-систем (продолжение)

	ШИП	Застава	ФПСУ-IP
Идентификация и аутентификация сетевого трафика			
Возможность аутентификации трафика	Да (хэш-функции ГОСТ Р34.11-94)	Да (хэш-функции по алгоритму MD5)	Да (хэш-функции ГОСТ Р34.11-94)
Администрирование			
Проведение идентификации и аутентификации администратора МЭ по идентификатору (коду)	Средствами FreeBSD	Средствами Solaris	ТМ Аккорд
Проведение идентификации и аутентификации запросов на доступ удаленных администраторов	Средствами FreeBSD + конфигурацией ПО	Средствами Solaris + конфигурацией ПО	Строгая двустороняя аутентификация при защите запросов
Возможность централизованного управления	Да (telnet)	Да (telnet + графическая оболочка)	Да (АРМ ЦУБ)
Регистрация действий администратора МЭ по изменению правил фильтрации	Да	Да	Да
Графический интерфейс удаленного управления (GUI)	Нет (только текстовый)	Есть (Java)	Есть АРМ ЦУБ (управление безопасностью), АРМ ЦКС (контроль связи)
ОС, поддерживающие GUI	-	Любой Java browser	DOS
Сигнализация в центре управления о событиях на удаленных МЭ	Да	Да	Да
Протоколирование событий/формирование отчетов			
Учет использования	Да	Да	Да
Сортировка/фильтрация сообщений	+/+	+/+	+/+
Формат журнала регистрации	Syslog	Syslog	хранилище, подобное MIB
Форматы экспорта журнала регистрации	Текст	Текст	DBF-формат
Параметры VPN			
Применение отечественных алгоритмов (ГОСТ) для построения VPN	Да	Да	Да
Базовый протокол	SKIP	SKIP	Собственный
Накладные расходы на поддержку туннелей	112 на IP-пакет	112 байт на IP-пакет	18–20 байт на IP-пакет
Возможность сжатия трафика	Да	Нет	Да
Наличие клиентских частей	Да (Windows NT)	Да (Solaris Windows 95/98/NT)	Нет

Таблица П.1.1. Сравнительные характеристики Firewall/VPN-систем (окончание)

	ШИП	Застава	ФПСУ-IP
Количество одновременно поддерживаемых независимых туннелей	Н/д	1400	До 1024 на каждом сетевом интерфейсе
Возможность вложенности VPN-туннелей друг в друга (каскадирование)	Да	Да	Да
Возможность VPN-поддержки каналов управления пограничными маршрутизаторами	Нет	Да	Да
Возможность VPN-взаимодействия с VPN других организаций	Да (при использовании ими SKIP)	Да (при использовании ими SKIP)	Нет (если не ФПСУ-IP)
Собственная безопасность			
Защита целостности среды	Нет	Нет	Да (по классу 1а)
Защита целостности ПО	Да (контрольные суммы)	Да (контрольные суммы)	Да (по классу 1а)
Разграничение полномочий обслуживающего персонала	Да (средства FreeBSD)	Нет Нет	Да (с помощью ТМ)
Защита используемой ключевой информации	Да (главные ключи грузятся с дискеты)	Да (средствами ОС Solaris)	Да (собственными средствами)
Аутентификация программных модулей/дополнений	Нет	Нет	Да (хэш-функции)
Характеристики производительности (пропускная способность)			
В режиме шифрования	8 Мбит/с (Pentium 200)	8 Мбит/с (Pentium 200)	11 Мбит/с (Pentium 200)
Дополнительные возможности			
Open CryptoAPI™	Нет	Да	Да
Стоимостные характеристики			
Цена	Н/д	$2500–3000	$1000–1500

Таблица П.1.2. Результаты сравнения

Параметр	Относительный вес, %	ШИП	Застава	ФПСУ-IP
Сертификация в ГТК или ФАПСИ и общие характеристики	15	3	4	4
Функции управления доступом	20	4	4	4
Администрирование	10	3	5	4
Протоколирование событий/ формирование отчетов	5	3	4	4
VPN-параметры	30	4	4	4
Собственная безопасность	5	3	2	5
Производительность	15	4	4	5
Итого	100	3,65	4	4,2

Приложение 2

Система санкционированного доступа к ресурсам корпоративной информационной системы

Здесь рассматривается класс информационных систем повышенной сложности, которые обычно называют корпоративными информационными системами или системами масштаба предприятия. Как правило, любая корпоративная информационная система состоит из ряда подсистем (сбора данных, технологической обработки данных, управления предприятием, поддержки принятия решения, информационно-аналитическая и т.д.).

При построении систем масштаба предприятия для сокращения общих затрат, связанных с их установкой и эксплуатацией, желательно объединять все входящие в этот класс подсистемы в один комплекс и придерживаться следующих основополагающих правил:

- поддержка открытых стандартов, подразумевающая соответствие общепринятым стандартам;
- масштабируемость, означающая, что программное обеспечение должно работать с приемлемой производительностью без внесения в него существенных изменений при увеличении мощности и количества используемого оборудования;
- многозвенность; принцип многозвенности означает, что каждый уровень системы (клиент, Web-сервер, сервер приложений, сервер баз данных) отвечает и реализует функции, наиболее присущие ему;
- аппаратно-платформенная независимость программного обеспечения, используемого при разработке системы;
- коммуникативность. Этот принцип означает, что различные уровни системы могут взаимодействовать между собой как по данным, так и по приложениям.

Подход к построению системы санкционированного доступа к ресурсам корпоративных информационных систем основан на архитектуре «клиент-сервер» и Web-технологии.

В настоящее время наиболее развивающейся технологией для построения корпоративных ИС является intranet, которая предусматривает специфические решения реализации приложений архитектуры «клиент-сервер».

Под термином «intranet» подразумевается многообразие технологий и протоколов, разработанных для глобальной сети Internet, в закрытой корпоративной сети, что предусматривает:

- применение в качестве транспортного протокола TCP/IP;
- применение встроенных средств защиты и аутентификации (IPSec, SSL, SHTTP, S/MIME SESAME, Kerberos 5 и т.п.);
- использование при разработке приложений технологии WWW в архитектуре «клиент-Web-сервер приложений-сервер баз данных»;

Вместе с тем Web-технологии при всех своих заметных преимуществах создают и новые технические проблемы, такие как масштабируемость, управление сеансами и состоянием, трудности с защитой потока данных и возможными изменениями стандартов:

- масштабируемость – прикладные программы Web в период пиковых нагрузок могут вести себя непредсказуемым образом. Большие загрузки, создаваемые запросами пользователей, требуют высокоэффективной архитектуры аппаратной и программной платформы, которые должны допускать масштабируемость ресурсов;
- управление сеансом и состоянием. В WWW-среде клиентское и серверное ПО, к сожалению, является слабосвязанным. Прикладные программы сервера должны хранить информацию о состоянии сеанса при переходе от одной страницы к другой, например, если необходимо избежать требования повторного ввода пользователем имени и пароля для доступа к новой странице;
- централизованное управление ресурсами и разграничение доступа. Как правило, управление ресурсами и разграничение доступа ориентировано на отдельный WWW-сервер и не охватывает все информационные ресурсы корпорации;
- защита. Проблемы защиты становятся первостепенными, когда компании делают внутренние базы данных доступными для внешних пользователей. Установление подлинности пользователя и безопасность передачи данных становится большой проблемой в среде Web из-за огромного количества потенциально анонимных пользователей;

- стандарты. Реализации технологии WWW все еще изменяются, и стандарты окончательно не устоялись. Так, например, в настоящее время ожидается расширение HTML языком описания Web-документов XML.

При реализации корпоративных информационных систем на базе технологий Internet/intranet важнейшими вопросами являются: организация защиты информации, централизованное управление информационными ресурсами, разграничение доступа к ресурсам. Особенно это важно при организации доступа пользователей из внешних сетей к ресурсам корпоративной ИС, так называемая extranet-технология (рис. П.2.1).

Общепринятый подход к решению вопросов защиты – использование в корпоративных сетях, имеющих выход в публичные сети Internet, систем и устройств, объединенных под общим названием Firewall (брандмауэр, межсетевой экран). Firewall – это система или группа систем, которая предписывает определенную стратегию управления доступом между двумя сетями. Она обладает следующими свойствами:

- весь трафик, как из внутренней сети во внешний мир, так и в обратном направлении, должен контролироваться системой;
- пройти через систему может только авторизованный трафик, который определяется стратегией защиты.

Другими словами, Firewall – это механизм, используемый для защиты доверенной сети от сети, доверия не имеющей. Обычно в качестве двух таких объектов рассматриваются внутренняя сеть организации (доверенная сеть) и Internet (недоверенная сеть), хотя в определении Firewall нет

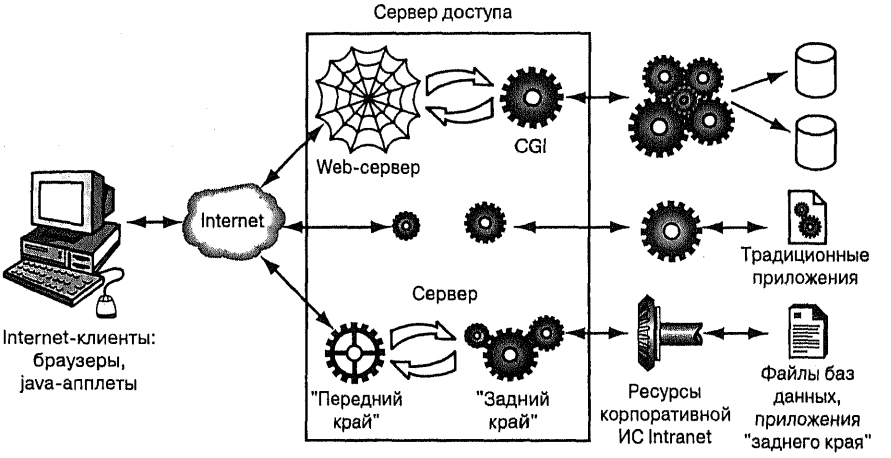

Рис. П.2.1. Extranet-технология

ничего, что жестко привязывало бы ее именно к Internet. Несмотря на то что большинство брандмауэров в настоящее время развернуто между Internet и внутренними сетями (intranet), имеет смысл использовать их в любой сети, базирующейся на технологии Internet, например в распределенной корпоративной сети.

Не детализируя функции систем Firewall, рассмотрим возможность их использования внутри корпоративной сети для организации защиты и управления доступом к ресурсам. При этом следует отметить, что основными объектами, с которыми оперирует типовая система firewall, являются:

- пакеты IP, TCP, UDP и других протоколов;
- сервисы, реализуемые прикладными протоколами Telnet, FTP, SMTP, POP3, HTTP, NNTP, Gopher и т.д.;
- пользователи, которым предоставляется (или запрещается) доступ к тому или иному ресурсу или VPN (Virtual Private Networks).

С пакетами работают системы Firewall, относящиеся к типу пакетных фильтров, пропускающих или не пропускающих через себя пакеты в зависимости от содержимого заголовков этих пакетов.

Понятие сервиса и пользователей используется системами Firewall типа серверов прикладного уровня или уровня соединения для предоставления пользователям, прошедшим процедуры аутентификации, тех или иных типов сервисов.

Оценивая возможность использования систем Firewall (PIX, ФПСУ-IP, «Застава», «Застава-Джет», Net-pro и др.) для управления и разграничения доступа к внутренним информационным ресурсам, следует отметить, что для данного случая наиболее подходят Firewall уровня соединения и уровня приложений. Это связано с тем, что контроль в данном случае осуществляется на уровне «сервисов» (например, WWW (HTTP), SMTP, FTP) и в предельном случае имен host-машин. Одновременно следует отметить, что контроль в данных Firewall не распространяется на другие атрибуты ресурсов (каталоги, файлы, программы, типы доступа, допустимые значения параметров и т.п.). Исходя из этого, управление и разграничение доступа к ресурсам отнесено непосредственно к серверам этих ресурсов (HTTP, FTP и т.п.), которые реализуют только локальные (по отношению к собственным ресурсам) стратегии управления и разграничения доступа. При этом недоступными являются следующие возможности:

- создания и управления централизованным каталогом ресурсов предприятия;
- поддержки каталога прав доступа пользователей к ресурсам;
- поддержки единой политики разграничения доступа к ресурсам;

- протоколирования сеансов работы пользователей (по отношению к ресурсам) и централизованного получения статистической информации о работе пользователей с ресурсами корпоративной информационной системы.

1. Система санкционированного доступа к ресурсам

Наличие большого числа информационных и вычислительных ресурсов (баз данных и приложений), используемых на предприятии и функционирующих на различных аппаратных и программных платформах, делает особо актуальной задачу создания и внедрения системы санкционированного доступа к ресурсам, цель которой – построение единого информационного пространства предприятия.

Кроме перечисленных ранее основных критериев построения корпоративных информационных систем, при создании системы санкционированного доступа к ресурсам необходимо учитывать следующие требования:

- обеспечение единого механизма доступа к ресурсам;
- обеспечение единой политики безопасности и защиты информации;
- централизованное и непрерывное управление, администрирование и контроль за использованием ресурсов;
- наличие большого числа наследуемых приложений, используемых на предприятии.

Исходя из этих требований и существующих ограничений, при проектировании программного обеспечения с использованием технологий Internet/intranet предлагается использовать следующую схему доступа к данным (информационным и вычислительным ресурсам). Она представляет собой многозвенную архитектуру, включающую в себя:

- клиентский уровень. В него входит терминальный компьютер пользователя под управлением ОС Windows 98/NT, использующий один из известных браузеров – Microsoft Internet Explorer или Netscape Navigator;
- уровень VPN, обеспечивающий отделение терминальной подсети пользователей от уровня серверов доступа и приложений;
- уровень серверов доступа (Access Server). В него входит специализированный сервер приложений, реализующий функции аутентификации пользователей, управления правами доступа к информационным

и вычислительным ресурсам корпоративной ИС (распределенный каталог), контроля и протоколирования сеансов;
- уровень Web-серверов типа MS IIS, Apache и т.п.;
- уровень серверов приложений – масштабируемая структура серверов, обеспечивающих унифицированные средства представления информации и функционирования подсистем КИС предприятия;
- уровень серверов баз данных.

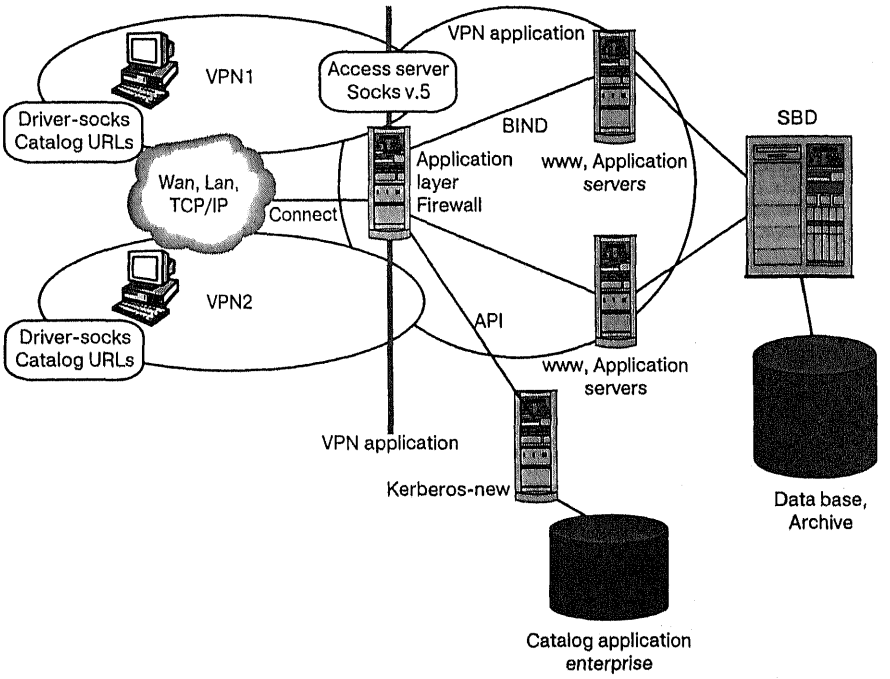

Рис. П.2.2. Построение системы санкционированного доступа к ресурсам

Данная архитектура позволяет скрыть и тем самым обезопасить от конечного пользователя сложность внутренней структуры системы и максимально оптимизировать загрузку вычислительных средств.

Функции, реализуемые уровнем сервера доступа (access server):

- аутентификация пользователей системы;
- контроль версий используемого клиентом ПО (активных объектов);
- управление состоянием и сеансами;
- предоставление санкционированного доступа к ресурсам корпоративной ИС;

- управление авторизацией и разграничением доступа;
- протоколирование сеансов работы абонентов;
- регистрация попыток несанкционированного доступа;
- общее централизованное администрирование системы;
- получение статистической информации о работе абонентов.

2. Аутентификация пользователей системы

Аутентификация пользователей предполагает определение подлинности пользователя перед началом работы с системой. Для аутентификации могут использоваться различные атрибуты и их комбинации: имя, пароль, IP-адрес, сетевое имя компьютера, время и место подключения и т.п. Применяемая схема паролей должна исключать возможность перехвата пароля путем активного сканирования сети и последующего его незаконного использования, а также возможность накопления статистического материала об используемых паролях. Такая схема может быть реализована посредством случайных сеансовых паролей, передаваемых по сети в зашифрованном виде. Все алгоритмы шифрования и соответствующее программное обеспечение должны быть сертифицированы и разрешены к применению. Для данной реализации наиболее подходят такие продукты, как:

- ФОРТ (фирма «Анкей» – www.ankey.ru);
- Net-Pro (фирма Signal-com – www.signal-com.ru);
- IP-LIR (фирма Infotecs);
- SOCKS5 Border Control Framework (фирма NEC Inc. – www.socks.nec.com); данный пакет поставляется с исходными текстами;
- пакет исходных текстов SSLeay Eric Young – eay@mincom.oz.au; www.psy.uq.oz.au/~ftp/Crypto/.

3. Контроль версий, используемых клиентом ПО

Возможности Web-технологии должны быть использованы для обновления клиентских активных объектов.

4. Управление состоянием и сеансами

Протоколы, используемые при обмене данными между Web-сервером и клиентом, являются сеансонезависимыми, то есть Web-сервер не может соотнести запрос клиента с его предыдущими запросами. Реализация этого соответствия должна быть произведена сервером доступа для подтверждения возможности выполнения запроса и отслеживания сеансового

контекста пользователя. Кроме того, функция управления состоянием и сеансами должна обеспечивать динамическое отслеживание состояния системы, подключенных пользователей, выполняемых действий.

5. Предоставление санкционированного доступа к ресурсам информационной системы

Функция санкционированного доступа к ресурсам системы предполагает использование однотипных протоколов и интерфейсов для доступа к различным приложениям, будь это наследуемые DOS- или Windows-приложения, приложения, поддерживающие HTTP(XML)-протокол или предоставляющие входные и выходные данные в виде статических и/или динамических HTML-страниц. Такой унифицированный механизм доступа может быть реализован одним из следующих способов:

- путем инициирования сессии с сервером доступа, который, в свою очередь, взаимодействуя с некоторым промежуточным программным слоем (типа Citrix MetaFrame), предоставляет пользователю доступ к традиционным приложениям, преобразуя экранные формы в формат HTML, корректно интерпретируемый ICA-клиентом(piCasso) через браузер;
- путем обращения от WWW-клиента к серверу доступа за необходимыми сервисами (доступ к данным СУБД, обработка данных и т.п.), причем загрузка объектного сервиса на клиентское место может проводиться заранее или осуществляться с Web-сервера (Java-классы, объекты ActiveX и т.д.) в процессе работы.

6. Управление аутентификацией и разграничением доступа

Управление аутентификацией и разграничением доступа является важнейшей функцией, реализуемой сервером доступа, что позволяет говорить об однозначном соответствии между конкретным пользователем и доступным ему множеством ресурсов из состава корпоративной ИС. Такая реализация предполагает наличие единого списка именованных (каким-либо образом) ресурсов и единого списка пользователей. Она может использовать один из следующих методов:

- вся сеть делится на терминальную и серверную подсети;
- для каждого пользователя (и/или группы пользователей) существует список доступных ресурсов (или групп ресурсов) – прямое соответствие;

- для каждого ресурса (группы ресурсов) определен набор привилегий, обеспечивающих доступ; для каждого пользователя (группа пользователей) также определен присущий им набор привилегий; доступ к ресурсу может быть разрешен только при наличии у пользователя всех привилегий ресурса – косвенное соответствие.

Кроме того, могут быть дополнительно определены различные типы доступа к ресурсам (чтение, запись, удаление, запуск и т.п.), права на использование которых могут заметно различаться. Следует отметить, что в качестве прототипа данной реализации разрешается использовать аутентификационные системы Kerberos 5 или SESAME (www.esat.kuleuven.ac.be/cosic/sesame3.html) с модификацией, учитывающей разделение терминальной и серверной подсетей. В этом случае такие недостатки Kerberos 5, как необходимость модификации каждого сетевого ресурса (в него должен быть вставлен код аутентификации), могут быть реализованы на сервере доступа, находящемся на границе терминальной и серверной подсетей.

7. Протоколирование сеансов работы абонентов

Функция протоколирования и регистрации предусматривает запись информации о любых действиях пользователей по доступу к ресурсам в статические и динамические журналы, с указанием идентификаторов пользователей, ресурсов, типов доступа, времени, условий, результатов выполнения операций и т.д.

8. Общее централизованное администрирование системы

Функция администрирования предполагает наличие внешнего по отношению к серверу доступа программного компонента, обеспечивающего создание и изменение управляющей информации сервера доступа: списка пользователей, списка ресурсов, каталогов прав доступа и т.д. Каждая функция подсистемы администрирования рассмотрена отдельно.

9. Получение статистической информации о работе пользователей

Функция предусматривает возможность получения на основе данных протоколирования и регистрации статистических отчетов о работе пользователей и системы полностью в форме, удобной для восприятия и последующего анализа.

10. Информационные объекты и общий алгоритм работы

Основными информационными объектами, используемыми при работе сервера доступа, являются:

- каталог пользователей;
- каталог приложений (вычислительных ресурсов);
- каталог внутренних информационных ресурсов;
- каталог прав доступа;
- каталог активных сеансов;
- журнал регистрации;
- объекты, реализующие концепцию информационных ресурсов (HTML-страницы).

Вопросы конкретной реализации этих объектов здесь не рассматриваются. Наиболее естественным является их исполнение в виде таблиц соответствующей базы данных, размещенной на одном из доступных серверов БД. Доступ к этим ресурсам единого каталога может быть организован по протоколу LDAP.

11. Каталог пользователей

Каталог пользователей содержит информацию, идентифицирующую абонентов системы: код, имя, сведения, используемые в процессе аутентификации, принадлежность к организации или подразделению; данные, определяющие возможные режимы работы, время, место подключения; допустимые имена машин; IP-адреса и т.п. Кроме того, при реализации функции управления правами доступа к ресурсам на основе привилегий каталог пользователей должен содержать определенные для каждого абонента системы привилегии доступа. Пользователи могут быть объединены в группы по различным признакам.

12. Каталог приложений

Каталог приложений (вычислительных ресурсов) содержит информацию, определяющую структуру и содержание существующих в системе приложений. Информация, размещенная в каталоге приложений для каждого объекта, должна однозначно определять условия запуска (адреса, имена программ или иную информацию), возможность использования статических или динамических параметров, время запуска и т.п. Кроме того, при

реализации функции управления правами доступа к ресурсам на основе привилегий каталог приложений должен включать определенные для каждого приложения системы привилегии доступа.

13. Каталог внутренних информационных ресурсов сервера доступа

Под понятием внутреннего информационного ресурса, применительно к использованию Web-технологий, предлагается понимать логически связанную совокупность HTML-страниц (статических или динамических) с едиными правами доступа. Проверка полномочий в этом случае осуществляется только при попытке доступа к первой странице ресурса, в дальнейшем, при навигации внутри ресурса, контролируются только внешние по отношению к нему ссылки. Содержимое HTML-страниц может быть статическим или динамическим – как результат выполнения соответствующих приложений или прямых действий пользователей. Концепция внутренних информационных ресурсов позволяет реализовать средствами сервера доступа широкие возможности по предоставлению пользователям нормативно-справочной и другой информации с гарантированным разграничением доступа. Каталог информационных ресурсов содержит информацию, определяющую наименование ресурсов, их структуру, содержание (множество HTML-страниц), последовательность перехода и т.п. Кроме того, при реализации функции управления правами доступа к ресурсам на основе привилегий каталог информационных ресурсов должен иметь определенные для каждого ресурса (группы ресурсов) системы привилегии доступа.

14. Каталог прав доступа

Каталог прав доступа включает информацию, определяющую соответствие пользователей и ресурсов при реализации прямого соответствия прав. Для каждого пользователя (и/или группы пользователей) существует список доступных ресурсов (или групп ресурсов) с указанием типов доступа к ресурсу (чтение, запись, удаление, запуск и т.п.). При каждой попытке доступа пользователя к ресурсу производится проверка полномочий, и доступ либо разрешается, либо запрещается с регистрацией данного события в журнале попыток несанкционированного доступа. Данный каталог может быть реализован по принципу Kerberos 5 или SESAME.

15. Каталог активных сеансов и журнал регистрации

Каталог сеансов и журнал регистрации представляют собой регистрационные журналы (динамический и статический соответственно), где хранится информация о состоянии системы, фиксируемых событиях регистрации пользователей, доступа к ресурсам, попыток несанкционированного доступа. Каталог сеансов, кроме этого, содержит динамически обновляемую информацию об активных в настоящий момент пользователях и процессах, выполняющихся в системе. Данные каталога сеансов и журнала регистрации используются для динамического мониторинга системы и получения статистических отчетов о ее работе, активности пользователей, доступе к тем или иным ресурсам (рис. П.2.3).

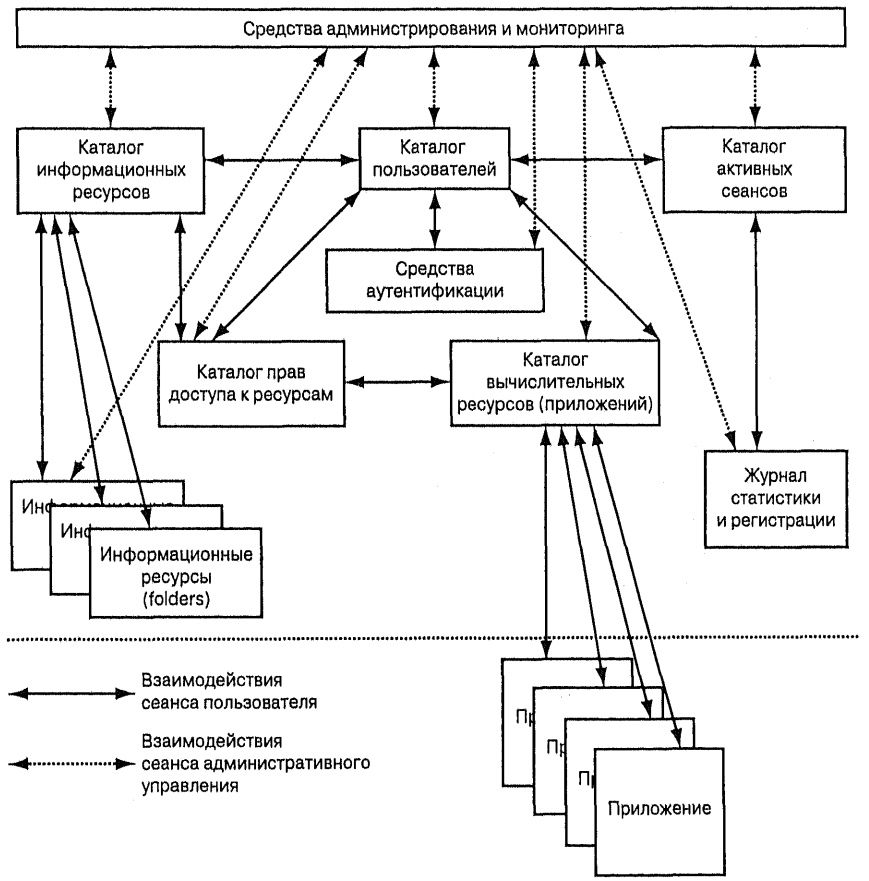

Рис. П.2.3. Взаимодействие компонентов в рамках каталога активных сеансов

16. Общий алгоритм работы сервера доступа к ресурсам корпоративной ИС

Общий алгоритм работы системы можно разделить на несколько этапов:

1. Аутентификация.
2. Доступ к приложению.
3. Доступ к информационному ресурсу.
4. Завершение работы.

Фаза регистрации начинается с момента подключения абонента к ресурсам Web-сервера и получения приглашения на ввод аутентифицирующей информации. Наряду с приглашением на ввод данных аутентификации на WWW-клиент может быть скопировано обновленное ПО аутентификации. После введения пользователем требуемых данных производится их прием сервером доступа, проверка подлинности, целостности данных и корректности подключения. В случае правильности принятых данных (соответствия хранящимся в каталоге пользователей данным аутентификации) пользователь регистрируется в системе, о чем производится запись в журнал регистрации и формируется соответствующая запись в каталоге сеансов. Пользователю передается динамически сформированная HTML-страница с перечнем доступных ему информационных и вычислительных ресурсов (приложений) в соответствии с его правами доступа.

Доступ к нужному приложению осуществляется путем запроса соответствующего указателя ресурса (URL). После проверки полномочий пользователя управление передается на запрашиваемый ресурс. Доступ к ресурсу фиксируется в журнале регистрации и каталоге сеансов. Дальнейшая работа с приложением может происходить без участия средств сервера доступа, однако моменты запуска и окончания приложения должны отслеживаться сервером доступа и фиксироваться в журнале. Результатом работы приложения могут быть HTML-документы, которые передаются через Web-сервер пользователю непосредственно для интерпретации WWW-клиентом или же записываются в соответствующий информационный ресурс с проверкой прав доступа. В дальнейшем эта информация может быть получена пользователем (необязательно тем, которым она была занесена в информационный ресурс) из соответствующего информационного ресурса.

Доступ к выбранному пользователем информационному ресурсу, как и к приложению, осуществляется путем запроса указателя ресурса (URL). После проверки полномочий управление передается на корневую HTML-страницу ресурса и производится фиксация доступа в журнале регистрации.

Дальнейшая навигация внутри ресурса осуществляется средствами WWW-клиента пользователя. Запрос любой внешней по отношению к ресурсу ссылки воспринимается как запрос к новому ресурсу, и весь процесс повторяется сначала. Следует отметить, что среди HTML-страниц информационного ресурса могут быть как статические или динамически формируемые HTML-страницы, так и страницы с элементами «форм» и активными элементами интерфейса. В любом случае технология доступа остается прежней.

В фазе завершения работы формируются соответствующие обобщенные записи журнала регистрации, удаляется запись каталога сеансов и абонент отключается от системы.

17. Функции административной подсистемы сервера доступа

В состав программного обеспечения системы санкционированного доступа должны входить программные средства администрирования и мониторинга.

Основными функциями программы администрирования являются: наполнение и поддержание в актуальном состоянии служебной базы данных информационных объектов сервера доступа, рассмотренных ранее.

К подобным функциям относятся:

- занесение информации новых абонентов в каталог пользователей системы;
- аутентификация пользователей системы;
- контроль версий используемого клиентом ПО (активных объектов);
- управление состоянием и сеансами;
- ведение каталогов унифицированного доступа к ресурсам информационной системы;
- управление авторизацией и разграничением доступа;
- распечатка журнала протоколирования сеансов работы абонентов;
- распечатка журнала регистраций попыток несанкционированного доступа;
- централизованное администрирование системы;
- получение статистической информации о работе абонентов;
- первая версия данной системы под названием СЕЗАМ была реализована в информационно-аналитической подсистеме МЦИ при ЦБ РФ (рис. П.2.4).

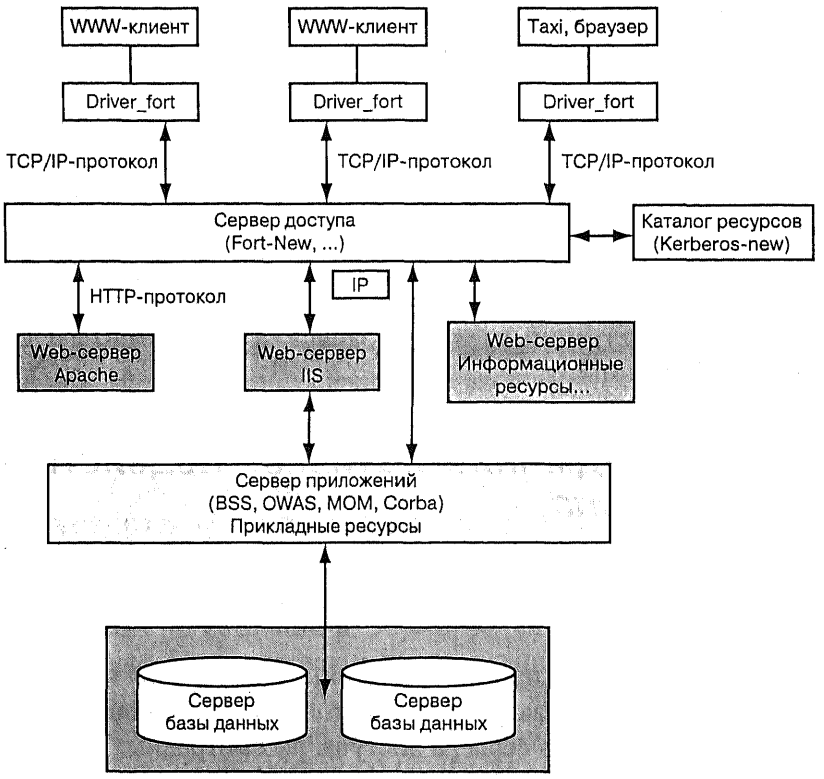

Рис. П.2.4. Архитектура системы СЕЗАМ

Приложение 3

Ресурсы в Internet, посвященные вопросам компьютерной безопасности

http://www.security.ru/ – Московское отделение Пензенского научно-исследовательского института (МО ПНИЭИ)

http://www.ankey.ru/ – Анкей/Информационные системы

http://www.infosec.ru/ – НИП Информзащита

http://www.ancud.ru/ – Анкад

http://www.signal-com.ru/ – Сигнал-КОМ

http://www.lancrypto.com/ – Лан-Крипто

http://www.elvis.ru/ – АО ЭЛВИС+

http://www.fssr.ru/ – Институт криптографии, связи и информатики (ИКСИ)

http://www.fzi.rsuh.ru/ – кафедра защиты информации РГГУ

http://www.confident.ru/ – журнал «Защита информации. Конфидент»

http://www.ssl.stu.neva.ru/ – Санкт-Петербургский специальный центр защиты информации

http://www.cert.org/ – сайт Computer Emergency Response Center (CERT). Координационный центр CERT изучает угрозы безопасности в Internet, публикует предупреждения об угрозах безопасности, изучает безопасность в глобальных сетях и предоставляет информацию о безопасности. Сайт содержит архив предупреждений (alerts) и советов (advisories) CERT, рассылаемых по электронной почте, а также множество полезной информации, касающейся информационной безопасности в Internet

http://www.nsa.gov/ – Агентство национальной безопасности (NSA) США

http://digitalid.verisign.com/ – VeriSign (сертификация открытых ключей)
http://www.eurosign.com/ – EuroSign (сертификация открытых ключей)

http://www.digicash.com/digicash/ – информация о проекте DigiCash
http://www.commerce.net/ – CommerceNet
http://www.cybercash.com – CyberCash
http://www.mondex.com/mondex/ – Mondex
http://www.fstc.org/ – Financial Services Technology Consortium
http://www.visa.com/ – Visa International
http://support.microsoft.com/support/ – Microsoft Knowledge Base. Содержит статьи по разным вопросам, связанным с продуктами Microsoft, в частности Windows NT
http://www.Ntsecurity.net/ – сайт посвящен вопросам безопасности Windows NT. Содержит описания известных на сегодняшний день методов нарушения безопасности Windows NT, NT Security FAQ и многие другие ресурсы, в частности программное обеспечение
http://ntbugtraq.ntadvice.com/ – NTBugTraq. Содержит архив списка рассылки ntbugtraq и ntsecurity, а также документацию, FAQ и прочее
http://www.avian.org/, ftp://ftp.avian.org/ – Avian Research. Самый ценный ресурс на этом сайте – статья hobbit@avian.org CIFS: Common Insecurities Fail Scrutiny, посвященная вопросам безопасности сетевой файловой системы, основанной на протоколе SMB
http://www.asmodeus.com/NT/ – содержит статьи по различным вопросам безопасности Windows NT
http://www.nmrc.org/ – Unofficial NT Hack FAQ
http://www.ntsecurity.com/ – Midwestern Commerce, Inc. Сайт фирмы, выпускающей программные продукты, связанные с информационной безопасностью
http://www.radium.ncsc.mil/ – описание результатов сертификации Windows NT 3.5 по уровню C2
http://www.incog.com/, http://skip.incog.com/ – Internet Commerce Group. Это подразделения Sun Microsystems занимается разработкой спецификации SKIP и SKIP-продуктов
http://www.ietf.cnri.reston.va.us/ – сервер Internet Security Group IETF
http://www.mit.edu/ – информация о Kerberos, а также много других сведений о криптографических протоколах
http://www.cryptosoft.com/ – исходник, информация, ссылки и многое другое по криптографии
http://www.cryptsoft.com/ – исходники, патчи и информация о SSL

http://www.quardralay.com/www/Crypt/ – информация по криптографии
http://www.rsa.com/ – RSA. Большое количество информации по криптографии
http://www.iss.net/ – Internet Security Systems, Inc. Системы обнаружения уязвимостей в информационно-телекоммуникационных системах
http://www.esat.kuleuven.ac.be/cosic/sesame.html – информация о проекте Sesame
http://www.geocities.com/SoHo/Studios/1059/ – PGP
http://www.ipv6.nasa.gov/ – IPSec
http://www.dec.com/ipv6/ – IPSec
http://www.europe.datafellows.com/f-secure/ – IPSec
http://www.internet2.edu/html/ – IPSec
http://www.ipsec.com/ – IPSec
http://www.ngi.gov/ – IPSec

Хакерские сайты
http://www.outerlimits.net/lordsome/
http://www.2600.com/
http://www.spartz.com/pecos/
http://www.hightop.nrlnavy.mil/raindow/
http://www.hackzone.ru/
http://www.infowar.co.uk/
http://www.hackersclub.com/
http://www.wored.com/
http://underground.org/
http://www.ba.com/
http://www.bst.bls.com/
http://www.cert.dfn.de/
http://www.eff.org/
http://www.first.org/first/
http://www.spy.org/

FTP-архивы
kampi.hut.fi:alo/des-dist.tar.Z
ftp.uu.net:bsd-sources/usr.bin/des/
ftp.uu.net:usenet/comp.sources.unix/volume10/cbw/
soda.berkeley.edu:/pub/cypherpunks
ftp.funet.fi:pub/unix/security/destoo.tar.Z
rsa.com:pub/faq/
ftp.psy.uq.oz.au:pub/DES/
rsa.com:?

ripem.msu.edu:pub/crypt/newdes.tar.Z
csrc.nist.gov:/bbs/nistpubs
ftp.3com.com:Orange-Book
prep.ai.mit.edu:pub/lpf/
ucsd.edu:hamradio/packet/tcpip/crypto/des.tar.Z
ripem.msu.edu:pub/crypt/other/tran-and-prngxor.shar
nic.merit.edu:documents/rfc/
beta.xerox.com:pub/hash/
chalmers.se:pub/unix/des/des-2.2.tar.Z
ripem.msu.edu:pub/crypt/other/tran-and-prngxor.shar
ftp.uu.net:usenet/comp.sources.unix/volume28/ufc-crypt/
garbo.uwasa.fi:pc/util/wppass2.zip
ftp://ftp.microsoft.com/bussys/winnt/winnt-public/fixes/usa/nt40

Список рекомендуемой литературы

1. Балакирский В. Б. Безопасность электронных платежей // Конфидент, № 5, 1996.
2. Вайнер П. С. Применение машин с ассоциативным поиском для шифрования и атаки на DES // Конфидент, № 6, 1996.
3. Гайкович В., Першин А. Безопасность электронных банковских систем. – М.: Единая Европа, 1994.
4. Зегжда Д. П. Как построить защищенную информационную систему // Под ред. Д. П. Зегжды и В. В. Платонова. – СПб: Мир и семья, 1995.
5. Мафтик С. Механизмы защиты в сетях ЭВМ: Пер. с англ. – М.: Мир, 1993.
6. Медведовский И. Д., Семьянов П. В., Платонов В. В. Атака через Internet // Под ред. П. Д. Зегжды. – СПб: Мир и семья, 1995.
7. Мельников Ю. Н. Защита информации в компьютерных системах. – М.: Финансы и статистика; Электроинформ, 1997.
8. Романец Ю. В., Тимофеев П. А., Шаньгин В. Ф. Защита информации в компьютерных системах и сетях // Под ред. В. Ф. Шаньгина. – М.: Радио и связь, 1997.
9. American National Standards Institute. Working Draft: American National Standard X9.30-199X: Public Key Cryptography Using Irreversible Algorithms for the Financial Services Industry: Part 1: The Digital Signature Algorithm (DSA). American Bankers Association, Washington, D.C., March 4, 1993.
10. Andelman A., Reeds J. On the cryptanalysis of rotor and substitution-permutation networks. IEEE Trans. on Inform. Theory, 28(4), 1982.
11. Angluin D., Lichtenstein D. Provable Security in Crypto-systems: a survey. Yale University, Department of Computer Science, #288, 1983.
12. ANSI X3.106, American National Standard for Information Systems-Data Link Encryption, American National Standards Institute, 1983.
13. Balenson D. Privacy Enhancement for Internet Electronic Mail: Part III: Algorithms, Modes, and Identifiers. RFC 1423, February 1993.
14. Bamford. The Puzzle Palace. Penguin Books, 1982.
15. Barlow J. P. Decrypting the puzzle palace. Communications of the ACM, 35(7), July 1992.

16. Bayer D., Haber S., Stornetta W.S. Improving the efficiency and reliability of digital time-stamping. In R.M. Capocelli, editor, Sequences '91: Methods in Communication, Security, and Computer Science, Springer-Verlag, Berlin, 1992.
17. P. Beauchemin P., Brassard G., Crepeau C., Goutier C., Pomerance C. The generation of random numbers that are probably prime. J. of Cryptology, 1, 1988.
18. Beker H., Piper F. Cipher Systems. Wiley, 1982.
19. Bennett J. Analysis of the Encryption Algorithm Used in the WordPerfect Word Processing Program. Cryptologia 11(4), 1987.
20. Bergen H. A., Caelli W. J. File Security in WordPerfect 5.0. Cryptologia 15(1), January 1991.
21. Beth T. Algorithm engineering for public key algorithms. IEEE Selected Areas of Communication, 1(4), 1990.
22. Biham E., Shamir A. Differential cryptanalysis of DES-like cryptosystems. Journal of Cryptology, vol. 4, #1, 1991.
23. Biham E., Shamir A. Differential cryptanalysis of Snefru, Khafre, REDOC-II, LOKI and LUCIFER. In Proceedings of CRYPTO '91, ed. by J. Feigenbaum, 1992.
24. Biham E. Shamir A. Differential Cryptanalysis of the Data Encryption Standard. Springer-Verlag, New York, 1993.
25. Biham E., Shamir A. Differential cryptanalysis of the full 16-round DES. In Advances in Cryptology – Crypto '92, Springer-Verlag, New York, 1993.
26. Blum M., Goldwasser S. An efficient probabilistic public-key encryption scheme which hides all partial information. In Advances in Cryptology – Crypto '84, Springer-Verlag, New York, 1985.
27. den Boer B., Bosselaers A. An attack on the last two rounds of MD4. In Advances in Cryptology – Crypto '91, Springer-Verlag, New York, 1992.
28. den Boer B., Bosselaers A. Collisions for the compression function of MD5. In Advances in Cryptology – Eurocrypt '93, 1993. Preprint.
29. Boyar J. Inferring Sequences Produced by Pseudo-Random Number Generators. Journal of the ACM, 1989.
30. Brandt J., Damgard I. On generation of probable primes by incremental search. In Advances in Cryptology – Crypto '92, Springer-Verlag, New York, 1993.
31. Brassard G. Modern Cryptology: a tutorial. Spinger-Verlag, 1988.
32. Brassard G. Modern Cryptology. Volume 325 of Lecture Notes in Computer Science, Springer-Verlag, Berlin, 1988.
33. Bressoud D. M. Factorization and Primality Testing. Undergraduate Texts in Mathematics, Springer-Verlag, New York, 1989.

34. Brickell E., Moore J., Purtill M. Structure in the S-boxes of DES. In Proceedings of CRYPTO '86, A. M. Odlyzko ed., 1987.
35. Brickell E.F., Denning D. E., Kent S. T., Maher D. P., Tuchman W. Skipjack Review, Interim Report: The Skipjack Algorithm. July 28, 1993.
36. Brown L. A proposed design for an extended DES, Computer Security in the Computer Age. Elsevier Science Publishers B.V. (North Holland), IFIP, W. J. Caelli ed., 1989.
37. Brown L., Pieprzyk J., Seberry J. LOKI – a cryptographic primitive for authentication and secrecy applications. In Proceedings of AUSCRYPT 90, 1990.
38. Campbell K. W., Wiener M.J. Proof the DES is Not a Group. In Proceedings of CRYPTO '92, 1993.
39. Carrol J., Martin S. The Automated Cryptanalysis of Substitution Ciphers. Cryptologia 10(4), 1986.
40. Carrol J., Robbins L. Automated Cryptanalysis of Polyalphabetic Ciphers. Cryptologia 11(4), 1987.
41. CCITT. Recommendation X.509: The Directory – Authentication Framework, 1988.
42. Chaum D. The dining cryptographers problem: Unconditional sender and recipient untraceability. Available from the author.
43. Chaum D. Privacy protected payments: Unconditional payer and/or payee untraceability. Available from the author.
44. Chaum, D. Snowing credentials without identification: Transferring signatures between unconditionally unlinkable pseudonyms. Available from the author.
45. Contact RSA Data Security, Inc.
46. Data Encryption Standard. National Bureau of Standards, FIPS PUB 46, Washington, DC, January 1977.
47. Davio M., Goethals J. Elements of cryptology in Secure Digital Communications, G. Longo ed.,1983.
48. Denning Dorothy E. The Clipper encryption system. American Scientist, 81(4), July–August 1993.
49. Diffie W. The first ten years of public-key cryptography. Proceedings of the IEEE, 76, 1988.
50. Diffie W., Hellman M. E. Exhaustive cryptanalysis of the NBS Data Encryption Standard. Computer, 10, 1977.
51. Diffie W. The first ten years of public key cryptography. IEEE proceedings, 76(5), 1988.
52. Diffie W., Hellman M. New Directions in Cryptography, IEEE Transactions on Information Theory, V. IT-22, n. 6, Jun 1977.

53. Diffie, W., Hellman, M.E. New directions in cryptography. IEEE Trans. Inf: Theory, IT-22, (November 1976).
54. W. Diffie and M.E. Hellman. New directions in cryptography. IEEE Transactions on Information Theory, IT-22, 1976.
55. Diffie W., Hellman M. Privacy and Authentication: An introduction to cryptography. IEEE proceedings, 1979.
56. ElGamal T. A public-key cryptosystem and a signature scheme based on discrete logarithms. IEEE Transactions on Information Theory, IT-31, 1985.
57. Ellison C. M. Solution of the Hebern Messages. Cryptologia, vol. XII, #3, Jul 1988.
58. Even S. Goldreich O. DES-like functions can generate the alternating group. IEEE Trans. on Inform. Theory, vol. 29, #6, 1983.
59. Feistel H. Cryptography and Computer Privacy. Scientific American, 228(5), 1973.
60. Feistel H., Notz W., Lynn Smith J. Some cryptographic techniques for machine-to-machine data communications, IEEE IEEE proceedings, 63(11), 1975.
61. Fiat A., Shamir A. How to prove yourself: Practical solutions to identification and signature problems. In Advances in Cryptology – Crypto '86, Springer-Verlag, New York, 1987.
62. Friedman W. F., Solving German Codes in World War I. Aegean Park Press.
63. Gaines H. Cryptanalysis, a study of ciphers and their solution. Dover Publications, 1944.
64. Garon J., Outerbridge R. DES watch: an examination of the sufficiency of the Data Encryption Standard for financial institutions in the 1990's. Cryptologia, vol. XV, #3, 1991.
65. Gillogly. Cryptologia 4(2), 1980.
66. Goldwasser S., Micali S. Probabilistic Encryption and How To Play Mental Poker Keeping Secret All Partial Information. Proceedings of the Fourteenth Annual ACM Symposium on Theory of Computing, 1982.
67. Goldwasser S., Micali S. Probabilistic encryption. J. of Computer and System Sciences, 28, 1984.
68. Good I. J. Good Thinking: the foundations of probability and its applications. University of Minnesota Press, 1983.
69. Gordon D. M. Discrete logarithms using the number field sieve. March 28, 1991.
70. Gordon D.M., McCurley K.S. Massively parallel computation of discrete logarithms. In Advances in Cryptology – Crypto '92, Springer-Verlag, New York, 1993.

71. Gustafson H., Dawson E., Caelli W. Comparison of block ciphers. In Proceedings of AUSCRYPT '90, J. Seberry and J. Piepryzk eds., 1990.
72. Hastad J. Solving simultaneous modular equations of low degree. SIAM J. Computing,17, 1988.
73. Hellman M. A cryptanalytic time-memory trade off. IEEE Transactions on Information Theory, IT-26, 1980.
74. Hellman M. The mathematics of public key cryptography. Scientific American, 1979.
75. Hinsley F. H. et al. British Intelligence in the Second World War. Cambridge University Press. (vol's 1, 2, 3a, 3b & 4, so far). XXX Years and authors, fix XXX
76. Hodges A., Turing A. The Enigma. Burnett Books Ltd., 1983
77. Hunter D. G. N., Kam G., Davida G. A structured design of substitution-permutation encryption networks. IEEE Trans. Information Theory, 28(10), 747–753, 1978.
78. Hunter D. G. N., McKenzie A. R. Experiments with Relaxation Algorithms for Breaking Simple Substitution Ciphers. Computer Journal 26 (1), 1983.
79. ISI for DARPA, RFC 793: Transport Control Protocol, September 1981.
80. Kahn D. The Codebreakers. Macmillan Co., New York, 1967.
81. Kahn D. Seizing the Enigma. Houghton Mifflin, 1991.
82. Kaliski B., Rivest R., Sherman A. Is the Data Encryption Standard a Group. Journal of Cryptology, vol. 1, #1, 1988.
83. Kaliski B. S. A survey of encryption standards. RSA Data Security, Inc., September 2, 1993.
84. Kalliski B. Privacy Enhancement for Internet Electronic Mail: Part IV: Key Certification and Related Services. RFC 1424, February 1993.
85. Kaliski B. S. Jr., Rivest R. L., Sherman A. T. Is the data encryption standard a group? J. of Cryptology, 1, 1988.
86. Kam J., Davida G. A structured design of substitution-permutation encryption networks. IEEE Trans. Information Theory, 28(10), 1978.
87. Kent S. RFC 1422: Privacy Enhancement for Internet Electronic Mail, Part II: Certificate-Based Key Management. Internet Activities Board, February 1993.
88. King and Bahler, An Algorithmic Solution of Sequential Homophonic Ciphers. Cryptologia 17(2), in press.
89. King and Bahler, Probabilistic Relaxation in the Cryptanalysis of Simple Substitution Ciphers. Cryptologia 16(3), 215–225, 1992.
90. Kinnucan P. Data encryption gurus: Tuchman and Meyer.Cryptologia, vol. II #4, 371-XXX, 1978.

91. Knuth D. E. The Art of Computer Programming, volume 2: Seminumerical Algorithms. Addison-Wesley, 1981.
92. Knuth D. E. The Art of Computer Programming. Volume 2, Addison-Wesley, Reading, Mass., 2nd edition, 1981.
93. Koblitz N. A course in number theory and cryptography. Springer-Verlag, 1987.
94. Kochanski M. Another Data Insecurity Package. Cryptologia 12(3), 165-177, 1988.
95. Kochanski M. Another Data Insecurity Package. Cryptologia 12(3), 1988.
96. Konheim A. Cryptography: a primer. Wiley, 1981.
97. Kozaczuk W. Enigma. University Publications of America, 1984.
98. Krawczyk H. IETF Draft: Keyed-MD5 for Message Authentication, November 1995.
99. Kruh, Cryptologia 12(4), 1988.
100. Kruh D., Kruh L. Machine Cryptography and Modern Cryptanalysis. Artech House, 1985.
101. Kullback S. Information Theory and Statistics. Dover, 1968.
102. Kullback S. Statistical Methods in Cryptanalysis. Aegean Park Press, 1976.
103. Lai X. On the Design and Security of Block Ciphers, ETH Series in Information Processing, v. 1, Konstanz: Hartung-Gorre Verlag, 1992.
104. Lai X., Massey J. A proposal for a new block encryption standard. EUROCRYPT 90, 1990.
105. Lakshmivarahan S. Algorithms for public key cryptosystems. In Advances in Computers, M. Yovtis ed., 22, Academic Press, 1983.
106. Lambros D. C., Friedman W. F. Military Cryptanalytics. Aegean Park Press.
107. Lempell A. Cryptology in transition, Computing Surveys, 11(4), 1979.
108. Lenstra A. K., Lenstra H. W. Jr. Algorithms in number theory. In J. van Leeuwen, editor, Handbook of Theoretical Computer Science, MIT Press/ Elsevier, Amsterdam, 1990.
109. A.K. Lenstra A. K., Lenstra H.W. Jr., Manasse M. S., Pollard J.M. The factorization of the ninth Fermat number. 1991.
110. Linn J. Privacy Enhancement for Internet Electronic Mail: Part I: Message Encryption and Authentication Procedures.
111. Lucks M. A Constraint Satisfaction Algorithm for the Automated Decryption of Simple Substitution Ciphers. In CRYPTO '88.
112. Massey J. An introduction to contemporary cryptology, IEEE proceedings, 76(5), 1988.
113. Merkle R. Fast software encryption functions. In Proceedings of CRYPTO '90, Menezes and Vanstone ed., 1991.

114. Meyer C. Ciphertext/plaintext and ciphertext/key dependence vs. number of rounds for the Data Encryption Standard. AFIPS Conference proceedings, 47, 1978.
115. Meyer C., Matyas S. Cryptography: A new dimension in computer security. Wiley, 1982.
116. National Institute of Standards and Technology (NIST). The Digital Signature Standard, proposal and discussion. Communications of the ACM, 35(7), July 1992.
117. National Institute of Standards and Technology (NIST). FIPS Publication 180: Secure Hash Standard (SHS). May 11, 1993.
118. National Institute of Standards and Technology (NIST). FIPS Publication 46-1: Data Encryption Standard. January 22, 1988. Originally issued by National Bureau of Standards.
119. National Institute of Standards and Technology (NIST). FIPS Publication 81: DES Modes of Operation. December 2, 1980. Originally issued by National Bureau of Standards.
120. National Institute of Standards and Technology (NIST). Notice of proposal for grant of exclusive patent license. Federal Register, 58(108), June 8, 1993.
121. National Institute of Standards and Technology (NIST). A proposed Federal Information Processing Standard for an Encryption Standard (EES). Federal Register, 58(145), July 30, 1993.
122. National Institute of Standards and Technology (NIST). Publication XX: Announcement and Specifications for a Digital Signature Standard (DSS). August 19, 1992.
123. NIST FIPS PUB 180-1. Secure Hash Standard, National Institute of Standards and Technology, U.S. Department of Commerce, DRAFT, 31 May 1994.
124. NIST FIPS PUB 186. Digital Signature Standard, National Institute of Standards and Technology, U.S. Department of Commerce, 18 May 1994.
125. NSA X22, Document # PD4002103-1.01, Fortezza: Application Implementors Guide, April 6, 1995.
126. Patterson W. Mathematical Cryptology for Computer Scientists and Mathematicians. Rowman & Littlefield, 1987.
127. Peleg S., Rosenfeld A. Breaking Substitution Ciphers Using a Relaxation Algorithm. CACM 22(11), 1979.
128. Pfleeger A. Security in Computing. Prentice-Hall, 1989.
129. Postel J., Reynolds J. RFC 959: File Transfer Protocol, October 1985.

130. Postel J., Reynolds J. RFC 854/5, May, 1993.
131. Price W., Davies D. Security for computer networks. Wiley, 1984.
132. Rakoff C., Luby M. How to construct psuedorandom permutations from psuedorandom functions. SIAM Journal of Computing, vol. 17, #2, 1988.
133. Reeds J. A., Weinberger P. J. File Security and the UNIX Crypt Command. AT&T Bell Laboratories Technical Journal, Vol. 63 #8, part 2, 1673–1684, October, 1984.
134. Reeds J. «Cracking» a Random Number Generator. Cryptologia 1(1), 20–26, 1977.
135. Rivest R. RFC 1319: The MD2 Message Digest Algorithm, April, 1992.
136. Rivest R., Shamir A., Adleman L. A method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM, 21, 2, (February, 1978).
137. Rivest R. RFC 1321: The MD5 Message Digest Algorithm, April 1992.
138. Rivest R., Shamir A., Adleman L. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. / Communications of the ACM, v. 21, n. 2, Feb 1978.
139. RSA Laboratories, PKCS #1: RSA Encryption Standard, version 1.5, November 1993.
140. RSA Laboratories, PKCS #6: RSA Extended Certificate Syntax Standard, version 1.5, November 1993.
141. RSA Laboratories, PKCS #7: RSA Cryptographic Message Syntax Standard, version 1.5, November 1993.
142. Rueppel R. Design and Analysis of Stream Ciphers. Springer-Verlag, 1986.
143. Saloma A. Public-key cryptography. Springer-Verlag, 1990.
144. Shannon C. Communication Theory of Secrecy Systems. Bell System Technical Journal 28(4), 1949.
145. Shimizu A., Miyaguchi S. Fast data encipherment algorithm FEAL. EUROCRYPT '87, 267–278, 1988.
146. Shirriff K., Welch C., Kinsman A. Decoding a VCR Controller Code. Cryptologia 16(3), 1992.
147. Schneier B. Applied Cryptography: Protocols, Algorithms, and Source Code in C. John Wiley & Sons, Inc. 1994.
148. Simmons G. (ed.), Contemporary Cryptology: the Science of Information Integrity. IEEE press, 1991.
149. Sinkov A. Elementary Cryptanalysis. Math. Assoc. Am. 1966.
150. Spillman R. et al., Use of Genetic Algorithms in Cryptanalysis of Simple Substitution Ciphers. Cryptologia 17(1), 1993.
151. Sorkin A. LUCIFER: a cryptographic algorithm. Cryptologia, 8(1), 1984.

152. Srinivansan R. Sun Microsystems, RFC-1832: XDR: External Data Representation Standard, August 1995.
153. Tuchman W. Hellman Presents No Shortcut Solutions To DES, IEEE Spectrum, v. 16, n. 7, July 1979.
154. Welchman G. The Hut Six Story. McGraw-Hill, 1982.
155. Welsh D. Codes and Cryptography. Claredon Press, 1988.
156. Yao A. Computational Information Theory. In Complexity in Information Theory, ed. by Abu-Mostafa, 1988.
157. Yardley H. O. The American Black Chamber. Aegean Park Press.

Уважаемый читатель!

Предлагаем Вам принять участие в работе над новыми книгами серии «Защита информации». Ваши замечания и предложения помогут сделать наши издания качественнее и актуальнее. Кроме того, читатели, обнаружившие серьезные ошибки в тексте, бесплатно получат экземпляр любой книги серии «Защита информации». О ходе конкурса и победителях можно узнать на нашем сайте в Internet www.dmk.ru. Заполненную анкету, координаты для связи с Вами, а также любые предложения по улучшению наших новых книг и исправлению ошибок в уже выпущенных изданиях Вы можете выслать по адресу: **105023, Москва, пл. Журавлева, 2/8, оф. 400.**

1. Где Вы приобрели эту книгу? _____

(в магазине (адрес), на рынке, у знакомых)

2. Вы приобрели эту книгу за _____ руб. Это очень дорого ☐ приемлемо ☐ дешево ☐

3. Какую книгу Вы охотнее купите?

а) отпечатанную на газетной бумаге по приемлемой цене ☐ **или** отпечатанную на бумаге высокого качества, но на 30% дороже ☐

б) в мягком переплете по приемлемой цене ☐ **или** в твердом переплете, но на 20% дороже ☐

4. Какого типа книгу Вы охотнее купите?

дешевый краткий справочник ☐ дорогое подробное руководство ☐ что-то среднее ☐

5. С какой целью Вы приобрели эту книгу?

книга необходима Вам по работе ☐ для самостоятельного изучения предмета ☐ почитать для общего развития ☐

6. Оцените по 5-балльной системе:

	1	2	3	4	5
а) качество выполнения иллюстраций	☐	☐	☐	☐	☐
б) качество изложения материала	☐	☐	☐	☐	☐
в) актуальность рассмотренных тем	☐	☐	☐	☐	☐
г) общее впечатление от книги	☐	☐	☐	☐	☐

7. Оцените удобство работы с книгой

а) шрифт хотелось бы крупнее ☐ нормально ☐ можно и помельче ☐

б) иллюстрации хотелось бы крупнее ☐ нормально ☐ можно и помельче ☐

в) формат хотелось бы крупнее ☐ нормально ☐ можно и помельче ☐

8. Чему, по Вашему мнению, следует уделить больше внимания в книге?

☐ описанию принципов работы с программами;
☐ описанию дополнительных и недокументированных возможностей программ;
☐ описанию конкретных примеров;
☐ _____

(впишите сюда то, что Вас интересует)

9.Книги по каким программным продуктам Вы хотели бы приобрести?

Название, производитель

☐ операционные системы _____
☐ офисные пакеты _____
☐ средства программирования _____
☐ издательские системы _____
☐ дизайнерские системы _____
☐ CAD-системы _____
☐ другое _____

10. Ошибки, обнаруженные в книге _____

_____

11. Сведения о себе

возраст ____ образование (среднее, неоконченное высшее, высшее)_____

профессия _____ адрес _____

Благодарим Вас за помощь!

Компьютерная безопасность

Петров Алексей Андреевич

Компьютерная безопасность
Криптографические методы защиты

Главный редактор *Захаров И. М.*
Литературный редактор *Ишков М. Н.*
Технический редактор *Прока С. В.*
Верстка *Татаринов А. Ю.*
Графика *Бахарев А. А.*
Дизайн обложки *Антонов А. И.*

ЛР № 065625 от 15.01.98

Подписано в печать 1.04.2000. Формат 70×100$^{1}/_{16}$.
Гарнитура «Петербург». Печать офсетная.
Усл. печ. л. 28. Тираж 3000. Зак. № 3

Издательство «ЛАЙТ Лтд.», 113093, Москва, Б. Серпуховская, 8/7, стр. 2

Отпечатано в полном соответствии
с качеством предоставленных диапозитивов
в ППП «Типография «Наука»
121099, Москва, Шубинский пер., 6.