Министерство образования и науки Российской Федерации

КАЗАНСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
им. А.Н. ТУПОЛЕВА

И.В. Аникин, В.И. Глова, А.Н. Нигматуллина

МЕТОДЫ И СРЕДСТВА ЗАЩИТЫ КОМПЬЮТЕРНОЙ ИНФОРМАЦИИ

Учебное пособие

Казань – 2008

Министерство образования и науки Российской Федерации

КАЗАНСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
им. А.Н. ТУПОЛЕВА

И.В. Аникин, В.И. Глова, А.Н. Нигматуллина

МЕТОДЫ И СРЕДСТВА ЗАЩИТЫ КОМПЬЮТЕРНОЙ ИНФОРМАЦИИ

Казань – 2008

УДК 681.391.825

И.В. Аникин, В.И. Глова, А.Н. Нигматуллина Методы и средства защиты компьютерной информации // Учебное пособие. Казань: Изд-во Казан. гос. техн. ун-та, 2008 с. 212

ISDN

Учебное пособие посвящено методам и средствам обеспечения информационной безопасности и защиты информационных ресурсов. Рассматриваются основные понятия и определения предмета защиты информации, модели политик безопасности, вопросы идентификации и аутентификации пользователей, криптографическая защита и вопросы контроля целостности информации, организация ключевых систем, защита программного обеспечения от несанкционированного использования, вопросы инженерно-технической защиты.

Предназначено для студентов очной формы обучения по направлению 230100 «Информатика и вычислительная техника». Содержание пособия разработано в соответствии с государственным образовательным стандартом по направлению – «Информатика и вычислительная техника» для дисциплины «Методы и средства защиты компьютерной информации». Разделы пособия можно рекомендовать студентам и магистрантам, обучающимся по специальностям направления «Информационная безопасность».

Содержание

АННОТАЦИЯ

Рассматриваются общие проблемы информационной безопасности, политики безопасности, идентификация и аутентификация объектов, математические положения теории чисел, основные симметричные и асимметричные алгоритмы шифрования, защита от разрушающих программных воздействий. Представлены проблемы и методы защиты в компьютерных системах и сетях. Отдельно выделяются вопросы инженерно – технической и правовой защиты.

Предлагается набор практических заданий в виде лабораторного практикума с большим разнообразием представленного материала.

В каждом разделе имеются вопросы для самоконтроля.

СПИСОК СОКРАЩЕНИЙ

АС – автоматизированная система

АСОИ – автоматизированная система обработки информации

БЛМ – модель Белла-ЛаПадулла

И/АУ – идентификация и аутентификация

ГПСЧ – генератор псевдослучайных чисел

ИБ – информационная безопасность

ИПС – изолированная программная среда

КС – компьютерная система

МЭ – межсетевой экран

НСД – несанкционированный доступ

ОК – открытый ключ

ПЗУ – постоянное запоминающее устройство

ПО – программное обеспечение

ПЭМИН – побочные электромагнитные излучения и наводки

РПВ – разрушающие программные воздействия

СВТ – средство вычислительной техники

СЗИ – система защиты информации

СК – секретный ключ

ЦП – центральный процессор

ЭЦП – электронно-цифровая подпись

DES – Data Encryption Standard, США

DoS-атаки - Denied of Service – отказ в обслуживании

HASP – электронные ключи

MLS- решетка - Multilevel Security

PKI (Public Key Infrastructure) - инфраструктуры системы открытых ключей

VPN – виртуальная частная сеть

ВВЕДЕНИЕ

Развивающиеся информационные технологии быстро внедряются во все сферы человеческого общества. Информация теперь официально определена идеальным объектом, имеющим ценность и стоимость как обычный продукт, причем зачастую, ее стоимость во много раз превосходит стоимость самой компьютерной системы, в которой она хранится и обрабатывается.

В связи с этим, собственникам, владельцам и пользователям информации необходимо принимать во внимание возможные злонамеренные воздействия со стороны злоумышленников по отношению к информационным системам. Например, нарушитель может пытаться выдать себя за другого пользователя, прослушать канал связи, перехватить и модифицировать информацию, которой обмениваются пользователи системы, расширить свои полномочия для получения доступа к информации, к которой ему представлен только частичный доступ, попытаться разрушить систему.

По мере развития и усложнения методов, средств и форм автоматизации процессов обработки информации повышается зависимость общества от степени безопасности используемых информационных технологий.

Интенсивное развитие открытых компьютерных сетей привлекает все большее внимание к ним пользователей. Большинство компаний и организаций подключают свои внутренние локальные сети к сети Internet, чтобы воспользоваться ее ресурсами и преимуществами. Однако при включении возникают серьезные проблемы с обеспечением информационной безопасности подключаемых локальной или корпоративной сети. Сеть предоставляет злоумышленникам множество возможностей для вторжения во внутренние сети компаний и организаций с целью хищения, искажения или разрушения конфиденциальной информации.

В связи с этим, в настоящее время все большую актуальность приобретает проблема обеспечения информационной безопасности (ИБ) и защиты информации от несанкционированного доступа, умышленного изменения,

кражи, уничтожения, негативного психологического воздействия на человека и других преступных действий. По этой причине проблемы защиты информации привлекают все большее внимание как специалистов в области компьютерных систем и сетей, так и многочисленных пользователей современных ПК.

В предлагаемом учебном пособии для изучения предлагаются вопросы защиты информации для комплексного начального изучения.

1. Основные понятия и определения предмета защиты информации

1.1. Санкционированный и несанкционированный доступ

Под *безопасностью автоматизированных систем обработки информации* (АСОИ) понимают их защищенность от случайного или преднамеренного вмешательства в нормальный процесс их функционирования, а также от попыток хищения, изменения или разрушения их компонентов [13].

Одним из основополагающих понятий в ИБ является понятие доступа к информации.

Под *доступом к информации* понимается ознакомление с ней, ее обработка, в частности копирование, модификация и уничтожение.

Понятие доступа к информации неразрывно связано с понятиями субъекта и объекта доступа (рис. 1.1).

Субъект доступа – это активный компонент системы, который может стать причиной потока информации от объекта к субъекту или изменения состояния системы (пользователь, процесс, прикладная программа и т.п.).

Объект доступа – это пассивный компонент системы, хранящий, принимающий или передающий информацию (файл, каталог и т.п.).

Зачастую, один и тот же компонент системы может являться и субъектом и объектом различных доступов. Например, программа PROGRAM.COM, запускаемая пользователем системы является объектом доступа для данного пользователя. Если та же самая программа

PROGRAM.COM читает с диска некоторый файл FILE.TXT, то при данном доступе она является уже субъектом.

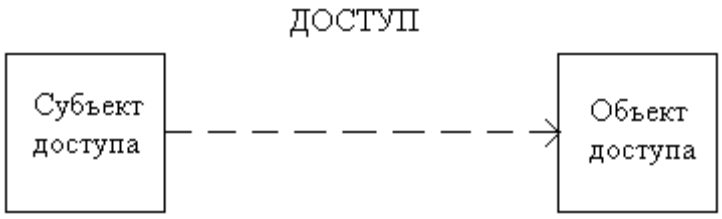


Рис. 1.1. Субъект и объект доступа

В информационной безопасности различают два типа доступа – санкционированный и несанкционированный.

Санкционированный доступ к информации – это доступ, не нарушающий установленные *правила разграничения доступа*, служащие для регламентации прав доступа субъектов к объектам доступа.

Несанкционированный доступ (НСД) к информации – доступ, нарушающий установленные правила разграничения доступа. Субъект, осуществляющий НСД, является нарушителем правил разграничения доступа. НСД является наиболее распространенным видом нарушений безопасности информации.

1.2. Базовые свойства безопасности информации. Угрозы безопасности и каналы реализации угроз

С точки зрения информационной безопасности выделяют следующие свойства информации: конфиденциальность, целостность и доступность.

Конфиденциальность информации – это ее свойство быть известной только допущенным и прошедшим проверку (*авторизованным*) субъектам системы. Для остальных субъектов системы эта информация должна быть неизвестной (рис 1.2).

Рис. 1.2. Авторизация субъекта

Проверка субъекта при допуске его к информации может осуществляться путем проверки знания им некого секретного ключа, пароля, идентификации его по фиксированных характеристикам и т.п.

Целостность информации – ее свойство быть неизменной в семантическом смысле при функционировании системы в условиях случайных или преднамеренных искажений или разрушающих воздействий.

Доступность информации – ее свойство быть доступной для авторизованных законных субъектов системы, готовность служб к обслуживанию запросов.

Целью злоумышленника является реализация какого-либо рода действий, приводящих к невыполнению (нарушению) одного или нескольких из свойств конфиденциальности, целостности или доступности информации.

Угроза безопасности АСОИ – потенциальная возможность определенным образом нарушить информационную безопасность (разрушение системы, кража паролей, денег).

Угрозы реализуются только при существовании *уязвимостей*.

Уязвимость АСОИ – узкое место, недостаток в системе безопасности, делающей возможной реализацию угрозы (плохие пароли, несвоевременное удаление учетных записей, ошибки ПО).

Атака на компьютерную систему – это непосредственная реализация злоумышленником угрозы безопасности.

Если уязвимостей нет, атаки, как правило, невозможны.

Цель системы защиты информации – противодействие угрозам безопасности в АСОИ.

По цели воздействия выделяют три основных типа угроз безопасности АСОИ [1]:

1. угрозы нарушения конфиденциальности информации;
2. угрозы нарушения целостности информации;
3. угрозы нарушения работоспособности системы (отказы в обслуживании).

Угрозы нарушения конфиденциальности информации направлены на перехват, ознакомление и разглашение секретной информации. При реализации этих угроз информация становится известной лицам, которые не должны иметь к ней доступ. Угроза нарушения конфиденциальности имеет место всякий раз, когда получен НСД к некоторой закрытой информации, хранящейся к компьютерной системе, или передаваемой от одной системы к другой. Большие возможности для реализации злоумышленником данного типа угроз существуют в открытых локальных сетях, интрасетях, сетях Internet в связи с незащищенностью протоколов передачи данных, и возможностью прослушивания канала передачи (сниффинга) путем перевода сетевой платы в «смешанный режим» (promiscuous mode).

Угрозы нарушения целостности информации, хранящейся в компьютерной системе или передаваемой по каналу связи, направлены на ее изменение или искажение, приводящее к нарушению ее качества или полному уничтожению.

Угрозы нарушения работоспособности (отказ в обслуживании) направлены на создание таких ситуаций, когда определенные преднамеренные действия либо снижают работоспособность АСОИ, либо блокируют доступ к некоторым ее ресурсам. Атаки, реализующие данный тип угроз, называются также DoS-атаками (Denied of Service – отказ в обслуживании). При реализации угроз нарушения работоспособности может преследоваться цель нанесения ущерба (вандализм), либо может являться промежуточной целью при реализации угроз нарушения конфиденциальности и целостности (нарушение работоспособности системы защиты информации).

Кроме этого, угрозы безопасности АСОИ можно поделить на *случайные и преднамеренные*. Причинами *случайных воздействий* могут служить аварийные ситуации из-за стихийных бедствий и отключений электропитания, отказы и сбои в аппаратуре, ошибки в программном обеспечении, ошибки в работе обслуживающего персонала и пользователей и т.д.

Преднамеренные угрозы связаны с целенаправленными действиями нарушителя и могут быть обусловлены разными мотивами: недовольством служащего карьерой, материальным интересом, любопытством, конкурентной борьбой и т.д.

При реализации угроз безопасности злоумышленник может воспользоваться самыми различными *каналами реализации угроз* – каналами НСД, каналами утечки.

Под *каналом утечки информации* понимают совокупность источника информации, материального носителя или среды распространения, несущего указанную информацию сигнала и средства выделения информации из сигнала или носителя. Средство выделения информации из сигнала или носителя может располагаться в пределах контролируемой зоны, охватывающей АСОИ или вне ее.

Применительно к АСОИ выделяют следующие основные каналы утечки информации [2].

1. *Электромагнитный канал*. Причиной его возникновения является электромагнитное поле, связанное с протеканием электрического тока в аппаратных компонентах АСОИ. В связи с этим, в близко расположенных проводных линиях возникают побочные электромагнитные излучения и наводки (ПЭМИН), анализ которых может позволить злоумышленнику получить доступ к АСОИ. Данный канал в свою очередь делится на следующие каналы: радиоканал, низкочастотный канал, наводки на сеть электропитания, наводки на провода заземления, наводки на линии связи между ПК.

2. *Виброакустический канал* - связан с возможностью анализа злоумышленником звуковых волн, распространяющихся в воздухе, возникающих при разговоре в закрытом помещении.

3. *Визуальный канал* - связан с возможностью визуального наблюдения злоумышленником за работой устройств отображения информации в АСОИ без проникновения в помещения, используя скрытые системы видеонаблюдения.

4. *Информационный канал* - связан с возможностью локального или удаленного доступа злоумышленника к элементам АСОИ, к носителям информации, к программному обеспечению, к линиям связи. Данный канал условно может быть разделен на следующие каналы: канал коммутируемых линий связи, канал выделенных линий связи, канал локальной сети, канал машинных носителей информации, канал терминальных и периферийных устройств.

Более подробно вопрос утечки информации по техническим каналам рассмотрен в главе 10.

1.3. Основные принципы обеспечения информационной безопасности

Основными принципами обеспечения информационной безопасности в АСОИ являются следующие [5].

1. Системности.
2. Комплексности.
3. Непрерывности защиты.
4. Разумной достаточности.
5. Гибкости управления и применения.
6. Открытости алгоритмов и механизмов защиты.
7. Простоты применения защитных мер и средств.

Принцип системности предполагает необходимость учета всех слабых и уязвимых мест АСОИ, возможных объектов и направлений атак, высокую квалификацию злоумышленника, текущих и возможных в будущем каналов реализации угроз.

Принцип комплексности. В распоряжении специалистов по информационной безопасности (ИБ) имеется широкий спектр мер, методов и средств защиты компьютерных систем. Принцип комплексности предполагает согласование работы разнородных СЗИ при построении целостной системы защиты, отсутствие слабых мест при стыковке различных СЗИ, покрытие ими всех существенных каналов реализации угроз.

Принцип непрерывности защиты. Защита информации – это не разовое мероприятие, а непрерывный целенаправленный процесс, предполагающий принятие соответствующих мер на всех этапах жизненного цикла АС. Например, большинству физических и технических средств защиты для эффективного выполнения своих функций необходима постоянная организационная поддержка (своевременная смена и обеспечение правильного хранения и применения имен, паролей, ключей шифрования, переопределение полномочий и т.п.). Перерывы в работе СЗИ могут быть использованы злоумышленником для анализа применяемых методов и средств защиты, внедрения специальных программных и аппаратных «закладок» и других средств преодоления системы защиты после восстановления ее функционирования.

Принцип разумной достаточности. Создать абсолютно защищенную систему защиты принципиально невозможно, взлом системы – это вопрос только времени и средств. Например, любые средства криптографической защиты не гарантируют абсолютной стойкости, а обеспечивают конфиденциальность информации в течение приемлемого для защищающейся стороны времени. В связи с этим, при проектировании СЗИ имеет смысл вести речь только о некотором приемлемом уровне безопасности. Важно выбрать золотую середину между стойкостью защиты и ее стоимостью, потреблением вычислительных ресурсов, удобством работы пользователей и другими характеристиками СЗИ.

Принцип гибкости управления и применения системы защиты предполагает возможность варьировать уровень ее защищенности. При определенных условиях функционирования АС, СЗИ, обеспечивающие ее защищен-

ность могут обеспечивать как чрезмерный, так и недостаточный уровень защиты. Гибкость управления и применения системы защиты спасает владельцев АС от необходимости принятия кардинальных мер по полной замене средств защиты на новые при смене условий функционирования АС.

Принцип открытости алгоритмов и механизмов защиты говорит о том, что защита не должна обеспечиваться только за счет секретности структурной организации СЗИ и алгоритмов функционирования ее подсистем. Знание алгоритма защиты не должно давать злоумышленнику возможности ее преодоления или снижать стойкость защиты.

Принцип простоты применения защитных мер и средств говорит о том, что механизмы защиты должны быть интуитивно понятны и просты в использовании.

1.4. Ценность информации

Информационные системы требуют защиты именно потому, что обрабатываемая информация бывает ценной не зависимо от происхождения. Реализация любой из угроз может привести к нарушению свойств конфиденциальности, целостности или доступности. При этом собственник информации несет определенные потери, связанные с нарушением этих свойств. Данные потери могут носить различный характер и могут быть выражены различным способом, и зачастую, в денежном эквиваленте. Для защиты информации затрачиваются определенные силы и средства, а для этого надо знать, какие потери мы понесем при реализации различных видов угроз (денежные, время на восстановление системы и т.п.). Ясно, что в денежном выражении затраты на защиту не должны превышать возможные потери.

Под *ценностью информации* понимается ее свойство, характеризующее потери собственника данной информации при реализации определенной угрозы, выраженные в стоимостном, временном либо ином эквиваленте.

Среди подходов к построению моделей защиты ИС, основанных на понятии ценности информации наиболее известными являются: оценка, анализ

и управление рисками ИБ [31], порядковые шкалы ценностей, модели решетки ценностей [3].

Подход, основанный на оценке, анализе и управлении рисками ИБ часто используется, когда требуется независимое рассмотрение большого количества угроз и уязвимостей, и существует возможность прямыми либо косвенными методами определить (хотя бы приблизительно) возможности реализации угроз и уязвимостей и стоимости возникающих при этом потерь.

Однако далеко не всегда возможно и нужно давать денежную оценку ценности информации [3]. Например, оценка личной информации, политической информации или военной информации не всегда разумна в денежном исчислении. В этом случае предпочтительнее использовать подход, связанный со сравнением ценности отдельных информационных элементов между собой и введением порядковой шкалы ценностей.

Пример 1.1. При оценке ценности информации в государственных структурах используется линейная порядковая шкала ценностей. Всю информацию сравнивают экспертным путем и относят к различным уровням ценности. В этом случае документам, отнесенным к некоторому уровню по шкале, присваиваются соответствующие грифы секретности. Сами грифы секретности образуют порядковую шкалу, например (принятую почти всеми государствами): НЕСЕКРЕТНО < КОНФИДЕНЦИАЛЬНО < СЕКРЕТНО < СОВЕРШЕННО СЕКРЕТНО. Более высокий класс имеет более высокую ценность и поэтому требования по его защите от несанкционированного доступа более высокие.

Рассматриваемая шкала хронологически была самой ранней и перестала удовлетворять требованиям информационных технологий, более детальной классификации. Разработка формализованных моделей информационных систем привело к разработке ценностной модели в виде решетки ценностей, которая является обобщением порядковой шкалы. Ее элементы представляют дискретную модель на базе введенной алгебры: с требованиями рефлек-

сивности, транзитивности, антисимметричности, а также верхней и нижней грани.

В основе государственных стандартов оценки ценности информации обычно используют MLS решетку (Multilevel Security).

1.5. Меры обеспечения безопасности компьютерных систем

По способам осуществления все меры обеспечения безопасности компьютерных систем подразделяют на:

- правовые (законодательные);
- морально-этические;
- организационно-административные;
- физические;
- аппаратно-программные.

К *правовым мерам* защиты информации относятся действующие в стране законы, указы, положения, инструкции и другие нормативные акты, регламентирующие правила обращения с информацией ограниченного использования и ответственности за их нарушения. Этим они препятствуют несанкционированному использованию информации и являются сдерживающим фактором для потенциальных нарушителей.

В РФ основы законодательного регулирования информационных отношений в обществе заложены в федеральных законах «О связи», «Об информации, информатизации и защите информации», «О государственной тайне», а также других законодательных и нормативных актах.

Российским законодателем определены составы преступлений в области компьютерной информации статьями уголовного кодекса 272, 273, 274 [27, 22] (более подробно данные вопросы освящены в разделе 13).

К *морально-этическим* мерам противодействия относятся всевозможные нормы поведения, которые традиционно сложились или складываются в обществе по мере распространения компьютеров в стране. Эти нормы быва-

ют как неписанными (общепризнанные нормы честности, патриотизма и т.д.), так и оформленными в некий свод правил или предписаний.

Организационно-административные меры защиты регламентируют процессы функционирования АСОИ; использование ресурсов АСОИ; деятельность персонала АСОИ; порядок взаимодействия пользователей с системой, с тем, чтобы в наибольшей степени затруднить или исключить возможность реализации угроз безопасности.

Организационно-административные меры включают в себя [13]:

– разработку правил обработки информации в АСОИ;

– совокупность действий при проектировании и оборудовании вычислительных центров и других объектов АСОИ (учет влияния стихии, пожаров, охрана помещений и т.п.);

– совокупность действий при подборе и подготовке персонала (проверка новых сотрудников, ознакомление их с порядком работы с конфиденциальной информацией, с мерами ответственности за нарушение правил ее обработки; создание условий, при которых персоналу было бы невыгодно допускать злоупотребления и т.д.);

– организацию надежного пропускного режима;

– организацию учета, хранения, использования и уничтожения документов и носителей с конфиденциальной информацией;

– распределение реквизитов разграничения доступа (паролей, полномочий и т.п.);

– организацию скрытого контроля за работой пользователей и персонала АСОИ;

– совокупность действий при проектировании, разработке, ремонте и модификации оборудования и программного обеспечения (сертификация используемых технических и программных средств, строгое санкционирование, рассмотрение и утверждение всех изменений, проверка на удовлетворение требованиям защиты, документальная фиксация изменений и т.п.).

К *физическим мерам* защиты относятся различные механические, электро- и электромеханические устройства или сооружения, специально предназначенные для создания физических препятствий на возможных путях проникновения и доступа нарушителей (турникеты, колючая проволока, кодовые замки, системы охранно-пожарной сигнализации и т.п.).

К *аппаратно-программным мерам* защиты относятся различные электронные устройства и специальные программы, которые реализуют самостоятельно или в комплексе с другими средствами следующие способы защиты:

- идентификацию и аутентификацию субъектов АСОИ;
- разграничение доступа к ресурсам АСОИ;
- контроль целостности данных;
- обеспечение конфиденциальности данных;
- аудит событий, происходящих в АСОИ;
- резервирование ресурсов и компонентов АСОИ.

1.6. Характеристика способов защиты компьютерной информации с помощью аппаратно-программных мер

Как показано выше, аппаратно-программные меры реализуют следующие основные способы защиты компьютерной информации:

- идентификацию и аутентификацию субъектов АСОИ;
- разграничение доступа к ресурсам АСОИ;
- контроль целостности данных;
- обеспечение конфиденциальности данных;
- аудит событий, происходящих в АСОИ;
- резервирование ресурсов и компонентов АСОИ.

Доступ к любой компьютерной информации в АСОИ, обладающей какой-либо ценностью, должен быть разрешен только определенному кругу лиц, предварительно прошедших регистрацию и подтвердивших свою подлинность на этапе идентификации и аутентификации (глава 3), который и

является первым краем обороны АСОИ. Основным требованием к его реализации является стойкость к взлому путем подбора или подмены информации, подтверждающей подлинность пользователя (пароля, ключа, и т.д.). Информация, подтверждающая подлинность пользователя должна храниться в секрете, лучше – на внешнем аппаратном устройстве, максимально защищенном от НСД.

Для разделения привилегий на доступ к информации и контроля прав доступа субъектов к этой информации используется подсистема разграничения доступа к ресурсам, которая функционирует руководствуясь сформированной администратором *политикой безопасности* (глава 2), определяющей набор норм, правил и практических рекомендаций, регламентирующих процесс обработки информации, выполнение которых обеспечивает защиту от заданного множества угроз. Одним из основных требований к реализации подсистемы разграничения доступа является разработка политики безопасности, адекватной защищаемой информации, и отсутствие возможностей у злоумышленника совершить НСД в обход подсистемы разграничения доступа.

Обеспечение конфиденциальности данных основано на применении, наряду с подсистемой разграничения доступа к ресурсам, различных *криптографических преобразований* защищаемой информации (глава 5).

Использование криптографических преобразований позволяет скрыть защищаемую информацию M путем перевода ее в нечитаемый вид C. При этом, чтение информации возможно только после дешифрования сообщения C на секретном ключе K, известном легальным пользователям и неизвестном злоумышленнику. *Стойкость криптографических преобразований основана только на секретности ключа дешифрования K.*

Существует два подхода к криптографической защите – симметричное шифрование и асимметричное шифрование (шифрование с открытым ключом). Симметричные криптосистемы используют для шифрования и дешифрования информации один и тот же ключ *K* (рис. 1.3). Асимметричные крип-

тосистемы шифруют информацию на общедоступном (открытом) ключе *ОК*, а дешифруют информацию на парном ему секретном ключе *СК*.

Стойкость асимметричных криптосистем основана на свойствах однонаправленных функций, целочисленной арифметике, свойствах операции сравнимости по модулю (глава 4).

Рис. 1.3. Схема симметричной криптосистемы

Обеспечение целостности обрабатываемой информации реализуется с помощью технологии электронно-цифровой подписи и функций хэширования (глава 6). Кроме этого, электронно-цифровая подпись позволяет реализовать подтверждения авторства получаемых сообщений. Реализация технологии электронно-цифровой подписи осуществляется в рамках использования асимметричных криптосистем.

Реализация функций защиты АСОИ, связанных с криптографией, контролем целостности информации, аутентификацией субъектов требует грамотной реализации подсистемы управления криптографическими ключами, которая включает в себя этапы генерации ключевой информации, ее хранения и распределения. Данным вопросам посвящена глава 7.

Под *аудитом безопасности* в АСОИ понимают постоянное отслеживание событий, связанных с нарушением безопасности, контроль, наблюдение за ними, в целях своевременного обнаружения нарушений политики безопасности, а также попыток взлома. Политика аудита безопасности должна формироваться обоснованно. Правильно построенный аудит позволяет не только выявлять нарушения безопасности, но также обнаруживать действия,

являющиеся начальным этапом взлома, с целью своевременной корректировки политики безопасности, принятия контрмер, что очень важно при обеспечении в АСОИ.

Резервирование ресурсов и компонентов АСОИ является одним из способов защиты от угрозы отказа доступа к информации. Один из наиболее действенных и эффективных методов, обеспечивающих восстановление системы при аварии, - резервное копирование данных и использование дисковых массивов.

1.7. Вопросы для самоконтроля

1. Что понимают под безопасностью автоматизированных систем обработки информации?
2. Дайте определение субъекта и объекта доступа. Приведите пример, когда некий элемент компьютерной системы в одном случае является субъектом, а в другом – объектом доступа.
3. Что понимают под санкционированным и несанкционированным доступом к информации?
4. Что понимают под конфиденциальностью, целостностью и доступностью информации? Приведите примеры нарушения данных свойств.
5. Охарактеризуйте понятия угрозы, уязвимости и атаки на компьютерную систему.
6. Перечислите основные классы угроз компьютерной системе. Кратко охарактеризуйте их.
7. Что понимают под каналом утечки информации?
8. Перечислите основные каналы утечки информации. Кратко охарактеризуйте их.
9. Перечислите основные принципы обеспечения ИБ в АСОИ. Кратко охарактеризуйте их.
10. Перечислите меры обеспечения безопасности компьютерных систем.

11. Охарактеризуйте три состава преступлений в области компьютерной информации, определенных Российским законодателем.

12. Что включают в себя организационно-административные меры защиты информации?

13. Какие основные способы защиты реализуют аппаратно-программные меры?

14. Что понимают под ценностью информации?

15. Перечислите и охарактеризуйте наиболее известные подходы к построению моделей защиты АСОИ, основанных на понятии ценности информации.

2. Разграничение доступа к ресурсам

2.1. Политики безопасности. Классификация политик безопасности

Под *политикой безопасности* (ПБ) понимается совокупность норм, правил и практических рекомендаций, регламентирующих процесс обработки информации, выполнение которых обеспечивает защиту от заданного множества угроз и составляет необходимое (а иногда и достаточное) условие безопасности компьютерной системы [1].

Основная цель создания ПБ информационной системы – определение условий, которым должно подчиняться поведение подсистемы безопасности.

Наиболее исследованными на практике моделями безопасности являются модели, защищающие информацию от нарушения свойств конфиденциальности и целостности. Они могут быть на верхнем уровне подразделены на два больших класса – формальных моделей и неформальных моделей. Возможная классификация моделей политик безопасности представлена на рис. 2.1.

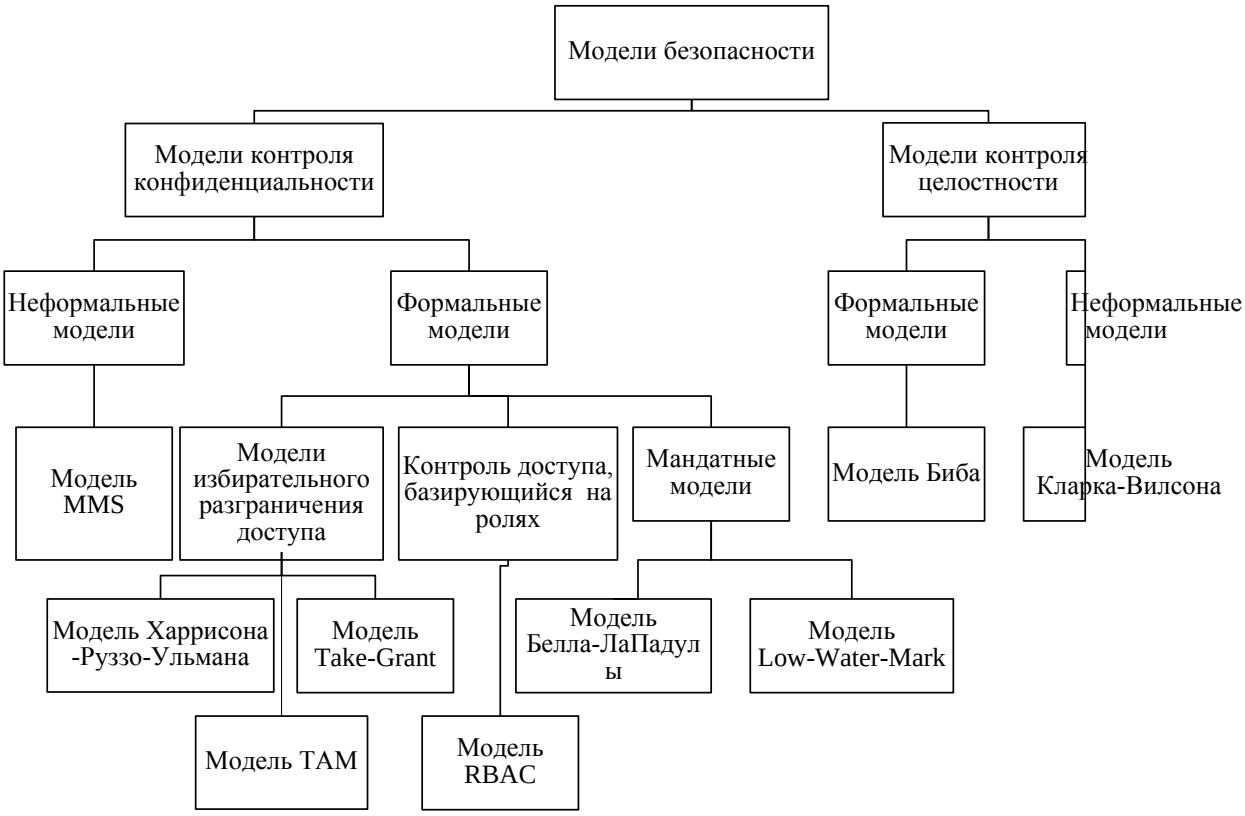


Рис. 2.1. Классификация моделей политик безопасности

Формальные модели политик безопасности позволяют описать поведение подсистемы безопасности в рамках строгих математических моделей, правил. С их помощью можно доказать безопасность системы, опираясь при этом на объективные и неопровержимые постулаты математической теории. Формирование данных политик предполагает выработку критерия безопасности системы и проведение формального доказательства соответствия системы этому критерию при соблюдении установленных правил и ограничений.

Неформальные модели политик безопасности предполагают описание поведения подсистемы безопасности в рамках вербальных (словесных) утверждений, не обладающих математической строгостью. Утверждения в неформальных моделях, как правило, формируют требования к поведению подсистемы безопасности на общем уровне без указания особенностей их реализации.

Достоинством формальных моделей является их математическая строгость и возможность формального доказательства того, что система, нахо-

дящаяся в безопасном состоянии, не может перейти в небезопасное состояние при соблюдении всех установленных правил и ограничений.

Недостатком формальных моделей является их большая абстрактность, что, зачастую, не позволяет использовать правила данных моделей ко всем субъектам и объектам компьютерной системы.

2.2. Политики избирательного разграничения доступа

Исходная политика избирательного разграничения доступа к информации (дискреционная модель) формируется путем задания администратором набора троек следующего вида $(S_i, O_j, T_k), i = \overline{1,N}, j = \overline{1,M}, k = \overline{1,K}$, где $S_i \in S$ - субъект доступа, $O_j \in O$ - объект доступа, $T_k \subset T$ - множество прав доступа, которыми наделен субъект S_i к объекту O_j (например, чтение, запись, исполнение и т.д.) [3].

При формировании дискреционной политики безопасности обычно формируют дискреционную матрицу доступов $M_{N \times M}$, строки которой соответствуют субъектам системы, столбцы – объектам, а в ячейках матрицы хранят множество типов доступов. Пример данной матрицы представлен в таблице 2.1.

Табл. 2.1. Дискреционная матрица доступов.

Объект / Субъект	**Файл_1**	**Файл_2**	**CD-RW**	**Флоппи-дисковод**
Администратор	Полные права	Полные права	Полные права	Полные права
Гость	Запрет	Чтение	Чтение	Запрет
Пользователь_1	Чтение, передача прав	Чтение, запись	Полные права	Полный запрет

Для матрицы доступа, представленной в таблице 2.1, Пользователь_1 имеет права на чтение и запись в Файл_2. Передавать эти права другому пользователю он не может.

Модель безопасности Харрисона-Руззо-Ульмана (HRU) [1] является классической дискреционной моделью реализующей произвольное управле-

ние доступом субъектов к объектам и контроль за распространением прав доступа.

Здесь поведение системы безопасности моделируется с помощью автоматной модели, путем перехода автомата из состояния в состояние. Состояние системы безопасности в некоторый момент характеризуется состоянием автомата и описывается тройкой $Q=(S,O,M)$, где S – множество субъектов системы, O – множество объектов системы, $M=M[s,o]$ – матрица доступов. Права доступа берутся из некоторого конечного множества T. Переход автомата из состояния в состояние осуществляется согласно запросам на изменение матрицы доступов.

Вводятся следующие операции op изменяющие матрицу доступов.

- enter t into (s,o) – внести право t в (s,o);
- delete t from (s,o) – удалить право t в (s,o);
- create subject s – создать субъект s;
- create object o – создать объект o;
- destroy subject s – уничтожить субъект s;
- destroy object o – уничтожить объект o;

В модели HRU запросы на изменение матрицы доступов осуществляются в следующей форме:

ЕСЛИ

t_1 in $M[s_1,o_1]$ and

t_2 in $M[s_2,o_2]$ and

…

t_m in $M[s_m,o_m]$

ТО

op_1

op_2

…

op_n

В начальное время система находится в начальном состоянии Q_0.

Имея начальное состояние Q_0 и право t, говорят, что Q_0 безопасна по отношению к t, если отсутствует последовательность запросов на изменение матрицы доступов, при которой t запишется в ячейку матрицы доступов, где она отсутствует в настоящий момент. В модели HRU исследуется вопрос, сможет ли некоторый субъект s приобрести право t для объекта o, если система стартует из состояния Q_0.

Для данной модели доказано 2 теоремы.

Теорема 2.1. Существует алгоритм для монооперациональных систем (систем, у которых в заключении запроса – одна операция), определяющий, является либо не является данная система безопасной в состоянии Q_0 относительно операции t.

Теорема 2.2. Проблема определения безопасности системы в состоянии Q_0 относительно t в общем виде неразрешима.

Доказано, что проблема определения безопасности может быть решена для систем, не имеющих в заключении операторов create, а также для систем, не имеющих в заключении операторов destroy либо delete.

2.3. Мандатные политики безопасности

Мандатные модели управления доступом были созданы по результатам анализа правил секретного документооборота, принятых в государственных и правительственных учреждениях многих стран.

Исходная мандатная политика безопасности строится на базе следующей совокупности аксиом, определяющих правило разграничения доступа субъектов к обрабатываемой информации:

1. Вводится множество атрибутов (уровней) безопасности A, элементы которого упорядочены с помощью установленного отношения доминирования. Например, для России характерно использование следующего множества уровней безопасности A={открыто (О), конфиденциально (К), секретно (С), совершенно секретно (СС), особая важность (ОВ)}.

2. Каждому объекту $O_j \in O$ компьютерной системы ставится в соответствие атрибут безопасности $x_{O_j} \in A$, который соответствует ценности объекта O_j и называется его *уровнем (грифом) конфиденциальности*.

3. Каждому субъекту $S_i \in S$ компьютерной системы ставится в соответствие атрибут безопасности $x_{S_i} \in A$, который называется *уровнем допуска* субъекта и равен максимальному из уровней конфиденциальности объектов, к которому субъект S_i будет иметь допуск.

4. Если субъект S_i имеет уровень допуска x_{S_i}, а объект O_j имеет уровень конфиденциальности x_{O_j}, то S_i будет иметь допуск к O_j тогда и только тогда, когда $x_{S_i} \geq x_{O_j}$.

Основным недостатком исходной мандатной политики безопасности является то, что в ней не различаются типы доступа вида «чтение» и «запись». Это создает потенциальную возможность утечки информации сверху вниз, например, путем запуска в компьютерной системе программной закладки с максимальным уровнем допуска, способной записывать информацию из объектов с верхних уровней конфиденциальности в объекты с более низкими уровнями, откуда она может быть прочитана субъектами с низким уровнем допуска.

Пример 2.1

Пусть для компьютерной системы задано 4 субъекта доступа *S*={Administrator, User1, User2, Guest} и 5 объектов *O*={FILE1.DAT, FILE2.TXT, FILE3.TXT, CD-ROM, FDD}. Множество атрибутов безопасности определено как *A*={NONCONFIDENTIAL, CONFIDENTIAL, SECRET, TOP SECRET}.

Пусть уровни конфиденциальности объектов определены следующим образом:

FDD – NONCONFIDENTIAL.

CD-ROM – CONFIDENTIAL.

FILE1.DAT – SECRET.

FILE2.TXT – SECRET.

FILE3.TXT – TOP SECRET.

Пусть уровни допуска субъектов определены следующим образом:

Administrator – TOP SECRET.

User1 – SECRET.

User2 – CONFIDENTIAL.

Guest – NONCONFIDENTIAL.

Тогда, согласно правилам исходной мандатной модели:

субъект Administrator будет иметь допуск ко всем объектам;

субъект User1 будет иметь допуск к объектам FDD, CD-ROM, FILE1.DAT, FILE2.DAT;

субъект User2 будет иметь допуск к объектам FDD, CD-ROM;

субъект Guest будет иметь допуск только к объекту FDD.

Однако, злоумышленник, в качестве которого возьмем субъекта Guest, завербовав пользователя User1, сможет получить доступ к информации из объекта FILE1.DAT, если User1 запишет эту информацию в объект FDD, что будет ему разрешено.

Политика безопасности Белла-ЛаПадулы (БЛМ) устраняет данный недостаток исходной мандатной политики безопасности и осуществляет контроль доступа субъектов $S_i \in S$ к объектам $O_j \in O$ компьютерной системы в зависимости от уровня допуска субъекта S_i и уровня конфиденциальности объекта O_j на основании двух следующих правил:

1. **Правило NRU (нет чтения вверх).** Согласно данному правилу субъект S_i с уровнем допуска x_{S_i} может читать информацию из объекта O_j с уровнем безопасности x_{O_j} тогда и только тогда, когда $x_{S_i} \geq x_{O_j}$. Формально данное правило можно записать как $S_i \xrightarrow{read} O_j \Leftrightarrow x_{S_i} \geq x_{O_j}$ (рис. 2.2)

2. **Правило NWD (нет записи вниз).** Согласно данному правилу субъект S_i с уровнем допуска x_{S_i} может записывать информацию в объект O_j с

уровнем безопасности x_{O_j} тогда и только тогда, когда $x_{S_i} \le x_{O_j}$. Формально данное правило можно записать как $S_i \xrightarrow{write} O_j \Leftrightarrow x_{S_i} \le x_{O_j}$ (рис. 2.2).

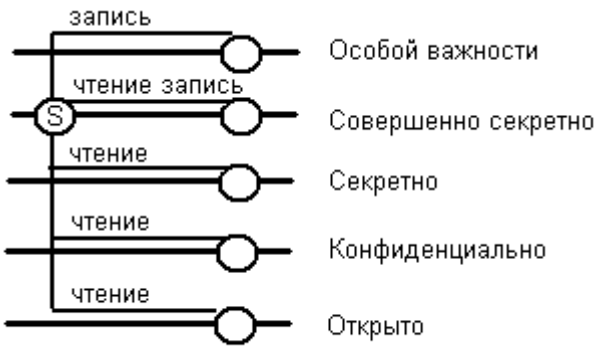


Рис. 2.2. Демонстрация правил политики безопасности Белла-ЛаПадулы

Введение свойства NWD разрешает проблему программных закладок, так как угроза записи информации на более низкий уровень, типичная для них, запрещена.

Пример 2.2.

Рассмотрим пример компьютерной системы, введенной в примере 2.1.

При ее реализации в рамках политики БЛМ возможно выполнение следующих операций:

1. субъект Administrator будет иметь допуск по чтению из всех объектов, и допуск по записи в объект FILE3.TXT;

2. субъект User1 будет иметь допуск по чтению из объектов FDD, CD-ROM, FILE1.DAT, FILE2.DAT и допуск по записи в объекты FILE1.DAT, FILE2.TXT, FILE3.TXT;

3. субъект User2 будет иметь допуск по чтению из объектов CD-ROM, FDD и допуск по записи в объекты FILE1.DAT, FILE2.TXT, FILE3.TXT, CD-ROM;

4. субъект Guest будет иметь допуск по чтению из объекта FDD и допуск по записи во все объекты.

2.4. Контроль доступа, базирующийся на ролях

Ролевую политику безопасности (контроль доступа, базирующийся на ролях – RBAC) нельзя отнести ни к дискреционным, ни к мандатным поли-

тикам, потому что управление доступом в ней осуществляется как на основе матрицы прав доступа для ролей, так и с помощью правил, регламентирующих назначение ролей пользователям и их активацию во время сеансов [28].

В ролевой модели классическое понятие «субъект» замещается понятиями *пользователь* и *роль*. Под *пользователем* понимается человек, работающий с системой и выполняющий определенные служебные обязанности. Под *ролью* понимается активно действующая в системе абстрактная сущность, с которой связан ограниченный, логически связанный набор полномочий, необходимых для осуществления определенной деятельности.

Ролевая политика безопасности очень близка к реальной жизни, так как работающие в компьютерной системе пользователи зачастую действуют не от своего личного имени, а исполняют определенные служебные обязанности, то есть выполняют некоторые роли, которые никак не связаны с их личностью. Использование ролевой политики безопасности позволяет учесть разделение обязанностей и полномочий между участниками прикладного информационного процесса, так как с точки зрения данной политики имеет значение не личность пользователя, осуществляющего доступ к информации, а то, какие полномочия ему необходимы для выполнения его служебных обязанностей.

При использовании ролевой политики управление доступом осуществляется в две стадии: во-первых, для каждой роли указывается набор полномочий, представляющий набор прав доступа к объектам, и, во-вторых, каждому пользователю назначается список доступных ему ролей. Полномочия назначаются ролям в соответствии с принципом наименьших привилегий, из которого следует, что каждый пользователь должен обладать только минимально необходимым для выполнения своей работы набором полномочий.

Формализация ролевой модели осуществляется в рамках следующих множеств:

U – множество пользователей компьютерной системы.

R – множество ролей.

P – множество полномочий на доступ к объектам, представленное, например, в виде матрицы прав доступа.

S – множество сеансов работы пользователей с компьютерной системой.

Для этих множеств определяются следующие бинарные отношения (рис. 2.3):

$PA \subseteq P \times R$ - отображение множества полномочий на множество ролей путем установления для каждой роли набора присвоенных ей полномочий.

$UA \subseteq U \times R$ - отображение множества пользователей на множество ролей путем определения для каждого пользователя набора доступных ему ролей.

Основными функциями в ролевой политике безопасности являются следующие:

$user : S \to U$ - для каждого сеанса s данная функция определяет пользователя, который осуществляет этот сеанс работы с компьютерной системой.

$roles : S \to \{R\}$ - для каждого сеанса s данная функция определяет набор ролей из множества R, которые могут быть одновременно доступны пользователю в этом сеансе.

$permissions : S \to P$ - для каждого сеанса s эта функция задает набор доступных в нем полномочий, который определяется как совокупность полномочий всех ролей, задействованных в этом сеансе.

Критерий безопасности ролевой модели: *компьютерная система считается безопасной, если любой пользователь системы, работающий в сеансе* s*, может осуществлять действия, требующие полномочия* p *только в том случае, если* $p \in permissions(s)$.

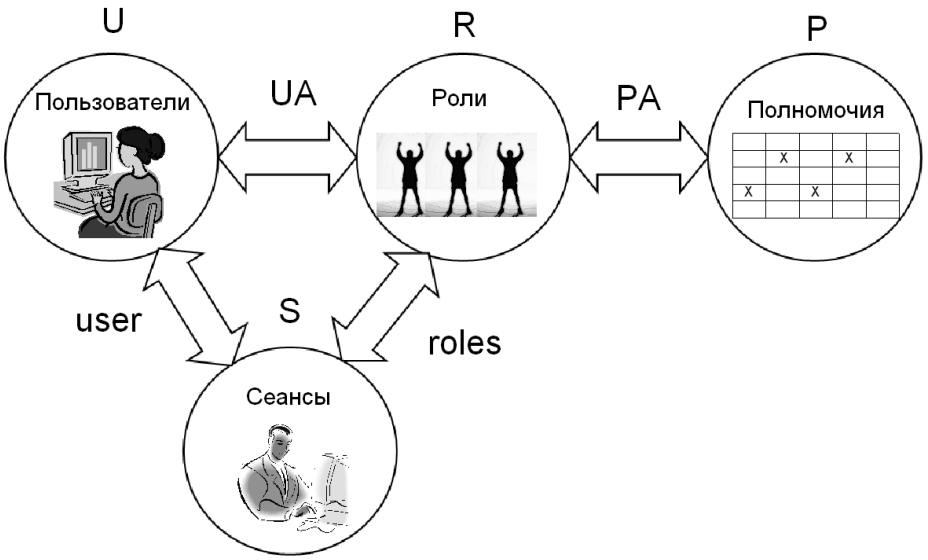


Рис. 2.3. Контроль доступа, базирующийся на ролях

В стандарте NIST 359 «Role Based Access Control» [9] представлены полные требования к функциональным возможностям ролевых политик безопасности.

2.5. Политики безопасности контроля целостности информационных ресурсов

При контроле доступа к информации, гарантирующем невозможность нарушения ее целостности, требуется пересмотр правил и требований ранее рассмотренных политик безопасности, гарантирующих невозможность нарушения конфиденциальности информации. В настоящее время наибольшее распространение получили две политики безопасности, контролирующих целостность информационных ресурсов – формальная модель политики безопасности Биба и неформальная модель политики безопасности Кларка-Вилсона [29].

Модель контроля целостности Биба является мандатной моделью, в которой вместо множества уровней безопасности *A* рассматривают множество уровней целостности информации, элементы которого также упорядочены с помощью установленного отношения доминирования. В качестве такого множества *A* можно, например, использовать следующее: *A*={не целостный, немного целостный, довольно целостный, совершенно целостный}.

Семантическая интерпретация этих уровней целостностей может быть выполнена следующим образом: *не целостный* – целостность объекта не контролируется никакими средствами, *немного целостный* – целостность объекта контролируется путем расчета нестойких контрольных сумм, *довольно целостный* – целостность объекта контролируется путем расчета стойких контрольных сумм (например, с помощью стойких функций хэширования), *совершенно целостный* – целостность объекта обеспечивается с помощью электронно-цифровых подписей.

Модель Биба выражается таким же способом, что и модель Белла-ЛаПадулы, за тем исключением, что правила данной модели являются полной противоположностью правилам БЛМ. При рассмотрении моделей контроля целостности запись наверх может представлять угрозу в том случае, если субъект с низким уровнем целостности искажает или уничтожает данные в объекте, лежащем на более высоком уровне целостности. Поэтому, исходя из задач обеспечения целостности обрабатываемой информации, нужно потребовать, чтобы такая запись была запрещена. Следуя подобным аргументам, можно рассматривать чтение снизу как поток информации, идущий из объекта нижнего уровня целостности и нарушающий целостность субъекта высокого уровня, поэтому и такое чтение необходимо запретить.

Наиболее распространены три вариации модели Биба: мандатная модель, модель понижения уровня субъекта и модель понижения уровня объекта.

Мандатную модель целостности Биба часто называют инверсией БЛМ. Основные правила этой модели переворачивают правила БЛМ. Контроль доступа субъектов $S_i \in S$ к объектам $O_j \in O$ компьютерной системы в зависимости от уровней целостности субъекта S_i и объекта O_j осуществляется на основании следующих правил:

1. Правило NRD (нет чтения снизу). Согласно данному правилу субъект S_i с уровнем целостности x_{S_i} может читать информацию из объекта O_j с

уровнем целостности x_{O_j} тогда и только тогда, когда $x_{S_i} \le x_{O_j}$. Формально данное правило можно записать как $S_i \xrightarrow{read} O_j \Leftrightarrow x_{S_i} \le x_{O_j}$.

2. Правило NWU (нет записи вверх). Согласно данному правилу субъект S_i с уровнем целостности x_{S_i} может записывать информацию в объект O_j с уровнем целостности x_{O_j} тогда и только тогда, когда $x_{S_i} \ge x_{O_j}$. Формально данное правило можно записать как $S_i \xrightarrow{write} O_j \Leftrightarrow x_{S_i} \ge x_{O_j}$.

Демонстрация работы данных правил представлена на рис. 2.4.

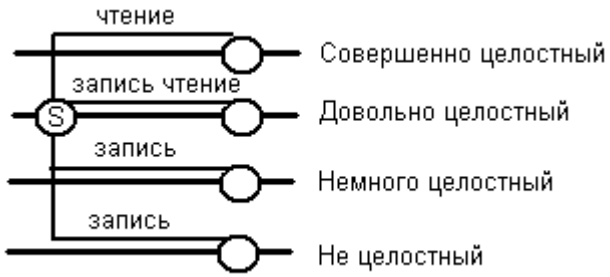


Рис. 2.4. Демонстрация правил мандатной политики безопасности Биба

В качестве практической демонстрации работы мандатной модели Биба можно привести следующий пример.

Пример 2.3

Любому печатному изданию (журналы, газеты и т.п.) можно присвоить негласный уровень контроля достоверности публикуемой информации, характеризующий то, насколько хорошо проверяется достоверность публикуемой в издании информация перед ее печатью. В данном случае множество уровней контроля достоверности может быть определено следующим образом: *A*={желтая пресса, нестрогий контроль, строгий контроль, совершенно строгий контроль}. Перепечатка информации из одного издания в другое может восприниматься как чтение и запись данной информации. В этом случае, перепечатка информации должна подчиняться правилам мандатной политики безопасности Биба – информация из желтой прессы не может быть напрямую перепечатана в печатном издании с совершенно строгим контролем публикуемой информации без прохождения дополнительных проверок на достоверность.

Модель понижения уровня субъекта заключается в небольшом ослаблении правила чтения снизу. В данной модели *субъекту разрешается осуществлять чтение снизу, но в результате такого чтения уровень целостности субъекта понижается до уровня целостности объекта.* Мотивом для введения такого правила может являться то, что субъекты с высокой целостностью рассматриваются как “чистые”. Когда к чистому субъекту попадает информация из менее чистого источника, субъект “портится”, и его уровень целостности должен быть соответственно изменен.

Модель понижения уровня объекта ослабляет правило для записи наверх. В данной модели субъекту разрешается записывать информацию наверх, но при этом *снижается уровень целостности объекта до уровня целостности субъекта, осуществлявшего запись*. Мотивы для такого правила те же, что и в модели понижения уровня субъекта.

Модель контроля целостности Кларка-Вилсона.

Модель контроля целостности Кларка-Вилсона (КВМ), в отличие от политики Биба является неформальной. Созданию этой модели способствовал анализ методов управления коммерческими организациями целостностью своих бумажных ресурсов в неавтоматизированном офисе, то есть был рассмотрен ряд хорошо известных методов учета целостности информационных ресурсов и сделана попытка распространения их на случай компьютерных приложений. Получившаяся модель целостности представляет собой руководство разработчикам и проектировщикам компьютерных систем для обеспечения целостности определенных вычислительных ресурсов.

Модель КВМ выражается в терминах конечного множества обрабатываемых данных *D*. Создатели модели разделили *D* на два непересекающиеся подмножества, которые называются *ограниченными элементами данных* (CDI) и *неограниченными элементами данных* (UDI), при этом $D = CDI \cup UDI$, $CDI \cap UDI = \varnothing$

Субъекты включены в КВМ как множество компонентов, которые могут инициировать так называемые *процедуры преобразования (ПП)*. Процедура преобразования определяется как любая ненулевая последовательность элементарных действий (чтение, запись и т.д.). Например, субъекты могут устранять элементы данных, изменять информацию в элементах данных, копировать их и т.д.

Модель КВМ можно рассматривать как набор, состоящий из следующих девяти правил.

Правило 1. В системе должны существовать особые процедуры преобразования IVP, утверждающие целостность любого CDI.

Можно представить себе IVP как некий тип процедуры проверки для утверждения целостности каждого CDI и подтверждения отсутствия целостности каждого UDI. Простейшим примером такой процедуры утверждения является проверка контрольной суммы. Различия в контрольных суммах сигнализируют о внесении изменений.

Правило 2. Применение любой ПП к любому CDI должно сохранять целостность этого CDI.

Это правило можно рассматривать как свойство скрытия применения ПП над CDI, то есть любое применение ПП над CDI не приведет к нарушению целостности CDI.

Правило 3. Только ПП может вносить изменения в CDI.

Другими словами, процедуры и действия, не являющиеся ПП, не могут изменить CDI. Это обеспечивает замкнутость преобразований в пределах набора CDI.

Правило 4. Субъекты могут инициировать только определенные ПП над определенными CDI.

Это правило предполагает, что система безопасности должна определять и поддерживать некоторые отношения между субъектами ПП и CDI - так называемые КВМ-тройки. Каждая такая тройка определяет возможность данного субъекта применить данную ПП к данному CDI.

Правило 5. КВМ - тройки должны проводить некоторую соответствующую политику разделения обязанностей субъектов.

Это правило предусматривает, что компьютерная система определяет такую политику, чтобы не позволять субъектам изменять CDI без соответствующего вовлечения других субъектов. Это предотвращает субъектов от возможности наносить ущерб целостности CDI. Некоторые системы управления конфигурацией предоставляют уровень разделения обязанностей. Например, в некоторых системах разработчики ПО должны представить свои модули на просмотр менеджеру по разработке ПО перед тем, как они смогут включить их в конфигурацию. Этот подход защищает целостность конфигурации ПО.

Правило 6. Некоторые специальные ПП могут превращать UDI в CDI.

Это правило позволяет определенным ПП получать на вход UDI и после соответствующего повышения целостности выдавать на выходе CDI.

Правило 7. Каждое применение ПП должно регистрироваться в специальном CDI, в который может производиться только добавление информации, регистрационная информация должна быть достаточной для восстановления картины о процессе работы этого CDI.

Это правило требует ведения специального регистрационного журнала, который хранится в определенном CDI.

Правило 8. Система должна распознавать субъекты, пытающиеся инициировать ПП.

Это правило определяет механизмы предотвращения атак, при которых один субъект пытается выдать себя за другого.

Правило 9. Система должна разрешать производить изменения в списках авторизации только специальным субъектам (например, офицерам безопасности).

Это правило гарантирует, что основная защита, определяемая КВМ-тройкой, не будет обойдена злоумышленником, пытающимся изменить содержание такого списка.

Основным преимуществом модели КВМ является то, что она основана на проверенных временем бизнес-методах обращения с бумажными ресурсами. Модель КВМ предоставляет исследователям методы работы с целостностью, отличные от традиционных уровне-ориентированных подходов, таких как модели БЛМ и Биба.

2.6. Вопросы для самоконтроля

1. Что понимают под политикой безопасности?
2. В чем заключается разница между формальными и неформальными политиками безопасности?
3. Как определяется исходная дискреционная модель политики безопасности?
4. Что из себя представляет дискреционная матрица доступов?
5. Как формально определяется модель безопасности Харрисона-Руззо-Ульмана? В чем отличие данной модели от исходной дискреционной?
6. Разрешима ли в общем случае проблема определения безопасности для модели Харрисона-Руззо-Ульмана? Приведите примеры случаев, когда данная проблема разрешима?
7. В чем заключается основное отличие мандатных политик безопасности от политик избирательного разграничения доступа?
8. Перечислите совокупность аксиом, определяющих исходную мандатную политику безопасности.
9. В чем заключается основной недостаток исходной мандатной политики безопасности?
10. Как формализуется модель безопасности Белла-ЛаПадулы. Как решается в этой модели проблема программных закладок, записывающих информацию в объекты с более низким уровнем конфиденциальности?
11. В чем заключаются основные отличия политик избирательного разграничения доступа от ролевых моделей?

12. Охарактеризуйте понятие «роль».

13. В чем заключается преимущество ролевой политики безопасности по сравнению с дискреционной?

14. В чем заключаются основные отличия политики безопасности Биба от политики безопасности Белла-ЛаПадулы?

15. Приведите возможные примеры уровней целостности информации. Проинтерпретируйте их.

16. Перечислите основные правила, разграничивающие доступ субъектов к объектам в модели Биба.

17. Перечислите девять правил, определяющих модель политки безопасности Кларка-Вилсона.

3. Идентификация и аутентификация субъектов

3.1. Классификация подсистем идентификации и аутентификации субъектов

Реализация никакой из политик безопасности не будет возможна в случае, если компьютерная система не сможет распознать (идентифицировать) субъекта, пытающегося получить доступ к объекту компьютерной системы. Поэтому защищенная КС обязательно должна включать в себя *подсистему идентификации*, позволяющую идентифицировать инициирующего доступ субъекта.

Под *идентификацией* понимают присвоение пользователю некоторого уникального *идентификатора*, который он должен предъявить СЗИ при осуществлении доступа к объекту, то есть назвать себя. Используя предъявленный пользователем идентификатор, СЗИ может проверить наличие данного пользователя в списке зарегистрированных и *авторизовать* его (то есть наделить полномочиями) для выполнения определенных задач.

В качестве идентификаторов могут использоваться, например, имя пользователя (логин), аппаратные устройства типа iButton (Touch Memory),

бесконтактные радиочастотные карты proximity, отдельные виды пластиковых карт и т.д.

Идентификаторы субъектов не являются секретной информацией и могут храниться в КС в открытом виде.

Для нейтрализации угроз, связанных с хищением идентификаторов и подменой злоумышленником легального пользователя необходимы дополнительные проверки субъекта, заключающиеся в подтверждении им владения предъявленным идентификатором. Данные проверки проводятся на этапе аутентификации пользователя.

Под *аутентификацией* понимают подтверждение пользователем предъявленного идентификатора, проверка его подлинности и принадлежности именно данному пользователю. Аутентификация выполняется для устранения фальсификации на этапе идентификации.

В качестве аутентифицирующей информации может использоваться, например, пароль, секретный код, пин-код и т.д. Информация, используемая субъектом для аутентификации, должна сохраняться им в секрете. Хищение данной информации злоумышленником ведет к тому, что злоумышленник сможет пройти этап идентификации и аутентификации без обнаружения фальсификации.

Этапы идентификации и аутентификации пользователя объединяются в единой подсистеме, называемой *подсистемой идентификации и аутентификации (И/АУ)*.

Атаки на подсистему идентификации и аутентификации пользователя являются одними из наиболее распространенных и привлекательных для злоумышленника, так как пройдя этап И/АУ злоумышленник получает все права легального пользователя, идентификатор которого был использован. В связи с этим, обеспечение стойкости ко взлому подсистемы И/АУ пользователя является очень важной задачей для безопасного функционирования компьютерной системы.

Стойкость к взлому подсистемы идентификации и аутентификации определяется гарантией того, что злоумышленник не сможет пройти аутентификацию, присвоив чужой идентификатор, либо украв его.

Наиболее распространенными методами идентификации и аутентификации пользователя являются:

– Парольные системы.

– Идентификация/аутентификация с использованием технических устройств.

– Идентификация/аутентификация с использованием индивидуальных биометрических характеристик пользователя.

3.2. Парольные системы идентификации и аутентификации пользователей

Парольные системы идентификации/аутентификации является одними из основных и наиболее распространенных в СЗИ методов пользовательской аутентификации. В данном случае информацией, аутентифицирующей пользователя, является некоторый секретный пароль, известный только легальному пользователю.

Совокупность идентификатора и пароля пользователя - основные составляющие его *учетной записи*. *База данных пользователей* парольной системы содержит учетные записи всех пользователей КС.

Парольные системы являются зачастую «передним краем обороны» всей системы безопасности. Отдельные ее элементы могут быть расположены в местах, открытых для доступа потенциальному злоумышленнику (в том числе и база данных учетных записей пользователей). В связи с этим, парольные системы становятся одним из наиболее привлекательных для злоумышленника объектов атаки. Основными типами угроз безопасности парольных систем являются следующие.

1. Перебор паролей в интерактивном режиме.
2. Подсмотр пароля.

3. Преднамеренная передача пароля его владельцем другому лицу.

4. Кража базы данных учетных записей с дальнейшим ее анализом, подбором пароля.

5. Перехват вводимого пароля путем внедрения в КС программных закладок (клавиатурных шпионов); перехват пароля, передаваемого по сети.

6. Социальная инженерия.

Многие недостатки парольных систем связаны с наличием человеческого фактора, который проявляется в том, что пользователь, зачастую, стремится выбрать пароль, который легко запомнить (а значит и подобрать), записать сложно запоминаемый пароль. Легальный пользователь способен ввести пароль так, что его могут увидеть посторонние, передать пароль другому лицу намеренно или под влиянием заблуждения.

Для уменьшения деструктивного влияния человеческого фактора необходимо реализовать ряд требований к выбору и использованию паролей [2].

1. Задание минимальной длины пароля для затруднения подбора пароля злоумышленником «в лоб» (полный перебор, brute-forcing) и подсмотра.

2. Использование в пароле различных групп символов для усложнения подбора злоумышленником пароля «в лоб».

3. Проверка и отбраковка пароля по словарю для затруднения подбора пароля злоумышленником с использованием словарей.

4. Установление максимального срока действия пароля для затруднения подбора пароля злоумышленником «в лоб», в том числе и в режиме «off-line» при взломе предварительно похищенной базы данных учетных записей пользователей.

5. Применение эвристического алгоритма, бракующего «плохие» пароли для усложнения подбора пароля злоумышленником «по словарю» или с использованием эвристического алгоритма.

6. Ограничение числа попыток ввода пароля для предотвращения интерактивного подбора пароля злоумышленником.

7. Использование задержки при вводе неправильного пароля для предотвращения интерактивного подбора пароля злоумышленником.

8. Поддержка режима принудительной смены пароля пользователя для эффективности реализации требования, ограничивающего максимальный срок действия пароля.

9. Запрет на выбор пароля самим пользователем и автоматическая генерация паролей для затруднения использования злоумышленником эвристического алгоритма подбора паролей.

Количественная оценка стойкости парольных систем может быть выполнена с помощью следующего подхода [2].

Пусть A – мощность алфавита паролей (количество символов, которые могут быть использованы при составлении пароля). Например, если при составлении пароля могут быть использованы только малые английские буквы, то A=26.

L – длина пароля.

$S = A^L$ - число всевозможных паролей длины L, которые можно составить из символов алфавита A. S также называют пространством атаки.

V – скорость перебора паролей злоумышленником.

T – максимальный срок действия пароля.

Тогда, вероятность P подбора пароля злоумышленником в течении срока его действия T определяется по следующей формуле.

$$P = \frac{V * T}{S} = \frac{V * T}{A^L}$$

Эту формулу можно обратить для решения следующей задачи:

ЗАДАЧА. Определить минимальные мощность алфавита паролей A и длину паролей L, обеспечивающих вероятность подбора пароля злоумышленником не более заданной P, при скорости подбора паролей V, максимальном сроке действия пароля T.

Данная задача имеет неоднозначное решение. При исходных данных V,T,P однозначно можно определить лишь нижнюю границу S* числа все-

возможных паролей. Целочисленное значение нижней границы вычисляется по формуле

$$S^* = \left\lceil \frac{V * T}{P} \right\rceil \qquad (3.1)$$

где ⌈ ⌉ - целая часть числа, взятая с округлением вверх.

После нахождения нижней границы S^* необходимо выбрать такие A и L, чтобы выполнялось неравенство (3.2).

$$S^* \le S = A^L \qquad (3.2)$$

При выборе S, удовлетворяющего неравенству (3.2), вероятность подбора пароля злоумышленником (при заданных V и T) будет меньше или равна P.

При вычислениях по формулам (3.1) и (3.2), величины должны быть приведены к одной размерности.

Пример 3.1

Исходные данные – P=10^{-6}, T=7 дней = 1 неделя, V=10 паролей / минуту = 10*60*24*7=100800 паролей в неделю.

Тогда, $S^* = \left\lceil \frac{100800 * 1}{10^{-6}} \right\rceil = 1008 * 10^8$.

Условию $S^* \le A^L$ удовлетворяют, например, такие пары величин A и L, как A=26, L=8 (пароли состоят из 8 малых символов английского алфавита), A=36, L=6 (пароли состоят из 6 символов, среди которых могут быть малые латинские буквы и цифры).

3.3. Вопросы для самоконтроля

1. Что понимают под идентификацией и аутентификацией? В чем заключается различие данных этапов и как они связаны между собой?

2. Приведите примеры различных идентификаторов и аутентификаторов пользователя.

3. Что понимают под авторизацией пользователя?

4. Чем определяется стойкость к взлому подсистемы идентификации и аутентификации?

5. Перечислите основные недостатки парольных подсистем идентификации и аутентификации.

6. Перечислите основные угрозы парольным подсистемам идентификации и аутентификации.

7. Перечислите требования к выбору и использованию паролей?

8. Как количественно оценить стойкость к взлому парольных подсистем идентификации и аутентификации?

9. Как изменится стойкость к взлому подсистемы парольной аутентификации при увеличении характеристик *A,L,V,T*? При их уменьшении?

4. Элементы теории чисел

При решении задач шифрования, дешифрования, построения ключевых систем в криптографии, используется представление обрабатываемого текста как целых чисел. Это дает возможность использовать математику целых чисел для создания стойких систем. Элементы теории целых чисел и рассматриваются в данной главе.

4.1. Модулярная арифметика

Пусть m – целое число. Тогда при делении любых целых чисел на m возможно получение ровно m остатков – $0,1,2,\ldots,m-1$.

Целые числа a и b называют *сравнимыми по модулю m*, если их разность $a-b$ делится без остатка на m, или, что то же самое, остатки, получаемые при делении чисел a и b на m, равны между собой. В этом случае число b называют *вычетом* числа a по модулю m.

Если a сравнимо с b по модулю m, то это записывают как $a \equiv b \pmod{m}$.

Пример 4.1

Целые числа 17 и 12 сравнимы между собой по модулю 5, то есть $17 \equiv 12 \pmod{5}$, кроме этого $17 \equiv 2 \equiv 7 \pmod{5}$.

Существует бесконечное количество чисел, сравнимых с числом a по модулю m, но только одно из них расположено в диапазоне от 0 до $m-1$.

Обычно, для целого числа $a>0$ предпочитают использовать вычеты $r \in \{0,1,...,m-1\}$. Набор целых чисел от 0 до (m-1) называют *полным набором вычетов по модулю m*.

Модулярная арифметика аналогична во многом обычной арифметике: она коммутативна, ассоциативна и дистрибутивна. Целые числа по модулю m по отношению к операциям сложения и умножения образуют коммутативное кольцо при соблюдении законов ассоциативности, коммутативности и дистрибутивности.

Основные свойства сравнений:

1. Рефлексивность: $a \equiv a \pmod m, \forall a$.
2. Симметричность: $a \equiv b \pmod m \Rightarrow b \equiv a \pmod m$.
3. Транзитивность: $a \equiv b \pmod m, b \equiv c \pmod m \Rightarrow a \equiv c \pmod m$.
4. Если $a \equiv b \pmod m$, k - произвольные целое число, то $k \cdot a \equiv k \cdot b \pmod m$.
5. Если $k \cdot a \equiv k \cdot b \pmod m$, наибольший общий делитель $НОД(k,m)=1$, то $a \equiv b \pmod m$.
6. Если $a \equiv b \pmod m$, $c \equiv d \pmod m$, то $a \pm c \equiv b \pm d \pmod m$.
7. Если $a \equiv b \pmod m$, $n \geq 0$, то $a^n \equiv b^n \pmod m$.
8. Если $a + b \equiv c \pmod m$, то $a \equiv c - b \pmod m$.
9. Если $a \equiv b \pmod m$, k - произвольные целое число, то $a \pm k \cdot m \equiv b \pmod m$.

При выполнении арифметических операций по модулю, можно либо сначала приводить операнды по модулю m, а затем выполнять операции, либо сначала выполнять операции, а затем приводить результат по модулю m.

В криптографии используется множество вычислений по модулю m, так как с вычислениями по модулю удобнее работать в связи с ограничением диапазона всех промежуточных величин и результата. Кроме того, решение задач вида вычисления дискретных логарифмов трудно в вычислительном плане.

Для модуля m длиной k бит промежуточные результаты любого сложения, вычитания или умножения будут не длиннее $2k$ бит. Поэтому такую операцию, как возведение в степень в модулярной арифметике можно выполнить без генерации очень больших промежуточных результатов.

Возведение числа a в степень x по модулю m, то есть нахождение $a^x \bmod m$ можно легко выполнить как ряд умножений. Особенно легко возводить в степень по модулю, если x - степень двойки.

Пример 4.2

Пусть, например, требуется вычислить $a^8 \bmod m$. В этом случае не следует выполнять серию умножений и одно приведение по модулю большого числа. Вместо этого выполняют три малых умножения и три малых приведения по модулю.

$$\left(\left(a^2 \bmod m\right)^2 \bmod m\right)^2 \bmod m$$

Например, $9^{16} \bmod 11 = \left(\left(\left(9^2 \bmod 11\right)^2 \bmod 11\right)^2 \bmod 11\right)^2 \bmod 11 =$

$$\left(\left(4^2 \bmod 11\right)^2 \bmod 11\right)^2 \bmod 11 = \left(5^2 \bmod 11\right)^2 \bmod 11 = \left(3 \bmod 11\right)^2 \bmod 11 = 9 \bmod 11$$

Вычисление $a^x \bmod m$, где x не является степенью двойки, немного сложнее. В этом случае степень x представляют в двоичной форме и представляют x как сумму степеней двойки.

Пример 4.3

Пусть $x=25_{(10)}=11001_{(2)}$, тогда $25=2^4+2^3+2^0$.

Тогда

$$a^{25} \bmod m = \left(a^{2^0+2^3+2^4} \bmod m\right) = \left(a \cdot a^8 \cdot a^{16}\right) \bmod m = a \cdot ((a^2)^2)^2 \cdot (((a^2)^2)^2)^2 \bmod m =$$

$$((((a^2 \cdot a)^2)^2)^2 \cdot a) \bmod m.$$

Поскольку многие алгоритмы шифрования основаны на возведении в большую степень больших чисел по большому модулю, целесообразно использовать рассмотренные выше алгоритмы быстрого возведения в степень.

4.2. Простые числа и их свойства

Натуральное число $n>1$ называется *простым*, если оно имеет в точности два различных натуральных делителя – 1 и n, в противном случае n называется *составным*.

Пример 4.4 Числа 2,3,7 являются простыми. Числа 4,6,8 – составными, так как их делителем является число 2.

Свойства простых чисел:

1. Если p_1 и p_2 – простые и $p_1 \neq p_2$, то p_1 не делится на p_2.
2. Пусть p – простое число, а n – натуральное, тогда n делится на p или наибольший общий делитель чисел n и p равен 1.
3. Если $m \cdot n$ делится на простое число p, то m делится на p или n делится на p.
4. Если $a_1 \cdot ... \cdot a_k$ делится на простое число p, то существует a_i, которое делится на p.

Известна следующая теорема:

Теорема 4.1. Любое натуральное число $n>1$ либо просто, либо раскладывается в произведение простых чисел и притом единственным образом с точностью до порядка следования сомножителей: $n = p_1^{\alpha_1} \cdot ... \cdot p_k^{\alpha_k}$, где $p_1 \neq p_2 \neq ... \neq p_k$. Данное разложение называют канонической формой числа n.

Задача представления числа n в канонической форме называется *задачей факторизации* числа n.

Существенный с точки зрения криптографии факт состоит в том, что в арифметике не известно эффективного алгоритма факторизации числа n. Никаких эффективных методов неизвестно даже в таком простом случае, когда необходимо найти два простых числа p и q, таких, что $n = p \cdot q$.

Известен ряд подходов, позволяющих выполнить проверку простоты целого числа n – решето Эратосфена, критерий Вильсона, тестирование на основе малой теоремы Ферма, тест Соловея-Штрассена, тест Рабина-Миллера и др.

Наибольшим общим делителем целых чисел *a* и *b*, обозначаемым как *НОД*(*a*,*b*) или просто (*a*,*b*), называют наибольшее целое, делящее одновременно числа *a* и *b*. Если (*a*,*b*)=1, то *a* и *b* называют *взаимно простыми*.

4.3. Числовые функции

В теории чисел и в криптографии большое значение имеют следующие числовые функции [23].

$\pi(n)$ - определяет количество простых чисел от 2 до *n*. Точной формулы для вычисления данной функции не известно. Грубой оценкой данной функции является следующая: $\pi(n) \approx \frac{n}{\ln n}$ [23].

$\Im(n)$ - определяет количество всех делителей числа *n*.

Пусть канонической формой числа *n* является $n = p_1^{\alpha_1} \cdot \ldots \cdot p_k^{\alpha_k}$. Тогда $\Im(n) = (\alpha_1 + 1) \cdot \ldots \cdot (\alpha_k + 1)$.

$\delta(n)$ - определяет сумму всех делителей числа *n*, $\delta(n) = \frac{p_1^{\alpha_1+1} - 1}{p_1 - 1} \cdot \ldots \cdot \frac{p_k^{\alpha_k+1} - 1}{p_k - 1}$

$\varphi(n)$ - функция Эйлера, определяет количество чисел меньших *n* и взаимнопростых с *n*,

$$\varphi(n) = p_1^{\alpha_1 - 1} \cdot (p_1 - 1) \cdot \ldots \cdot p_k^{\alpha_k - 1} \cdot (p_k - 1) \qquad (4.1)$$

Пример 4.5

Для числа *n*=720 найдем $\Im(n)$, $\delta(n)$, $\varphi(n)$.

Представим число 720 в канонической форме - $720 = 2^4 \cdot 3^2 \cdot 5$.

Тогда $\Im(720) = (4+1) \cdot (2+1) \cdot (1+1) = 30$

$\delta(720) = \frac{2^5 - 1}{2 - 1} \cdot \frac{3^3 - 1}{3 - 1} \cdot \frac{5^2 - 1}{5 - 1} = 31 \cdot 13 \cdot 6 = 2418$

$\varphi(720) = 2^3 \cdot (2-1) \cdot 3^1 \cdot (3-1) \cdot 5^0 \cdot (5-1) = 8 \cdot 3 \cdot 2 \cdot 4 = 192$

4.4. Вопросы для самоконтроля

1. Дайте определение сравнимости по модулю.
2. Приведите примеры чисел, сравнимых с 5 по модулю 7.
3. Что называют полным набором вычетов по модулю?
4. Перечислите основные свойства сравнений.
5. Дайте определение простого и составного числа. Приведите примеры.
6. Что называют канонической формой числа n.
7. В чем заключается задача факторизации числа n.
8. Факторизуйте следующие числа: 200, 143, 89.
9. Дайте определение наибольшего общего делителя чисел a и b.
10. Найдите наибольший общий делитель следующих чисел – 10 и 4, 20 и 21, 3 и 90.
11. Какие числа называют взаимно простыми? Приведите примеры взаимно простых чисел.
12. Найти $\Im(200)$, $\delta(200)$, $\varphi(200)$.

5. Методы и средства криптографической защиты

5.1. Принципы криптографической защиты информации

Криптография представляет собой совокупность методов преобразования данных (*шифрования*), направленных на то, чтобы сделать эти данные бесполезными для противника. Эти преобразования позволяют решить проблему обеспечения конфиденциальности данных. Для ознакомления с зашифрованной информацией применяется обратный процесс – *дешифрование*.

Для шифрования обычно используется некоторый алгоритм или устройство, реализующее заданный алгоритм, которые могут быть известны широкому кругу лиц. Управление процессом шифрования осуществляется с помощью периодически меняющегося *ключа шифрования*, обеспечивающего

каждый раз оригинальное представление информации при использовании одного и того же алгоритма или устройства. Знание ключа дешифрования позволяет просто и надежно расшифровать текст. Однако, без знания этого ключа процедура дешифрования может быть практически невыполнима даже при известном алгоритме. *Ключ шифрования K* - конкретное состояние некоторого параметра (параметров), обеспечивающее выбор одного преобразования из совокупности возможных для используемого метода шифрования.

Будем называть *открытым текстом M* исходное сообщение, которое шифруют для его сокрытия от посторонних лиц. Сообщение, формируемое в результате шифрования открытого текста, будем называть *закрытым текстом (шифротекстом) C*.

Обратной стороной криптографии является *криптоанализ*, который пытается решить обратную задачу, характерную для злоумышленника – раскрыть шифр, получив открытый текст, не имея подлинного ключа шифрования.

Существуют несколько основных типов криптоаналитических атак [13]. Реализация каждой из них предполагает, что злоумышленник знает применяемый алгоритм шифрования.

1. Криптоаналитическая атака при наличии только известного закрытого текста *C*.

2. Криптоаналитическая атака при наличии известного открытого текста (атака по открытому тексту). В этом случае, криптоаналитику известен открытый текст *M* и соответствующий ему закрытый текст *C*. Задача криптоаналитика состоит в нахождении ключа шифрования *K* для возможности прямой расшифровки последующих шифротекстов. Более мощным вариантом данного метода криптоанализа является криптоаналитическая атака при возможности выбора криптоаналитиком открытого текста.

3. Криптоаналитическая атака методом полного перебора всех возможных ключей. Данная атака предполагает использование криптоаналитиком

известного шифротекста и осуществляется посредством полного перебора всех возможных ключей с проверкой осмысленности получаемого открытого текста. Такой подход требует привлечения предельных вычислительных ресурсов и иногда называется силовой атакой, атакой «в лоб», или brute-forcing.

4. Криптоаналитическая атака методом анализа частотности закрытого текста. Реализация данной атаки предполагает использование криптоаналитиком информации о частоте встречаемости символов в закрытом тексте с целью получения информации о символах открытого текста.

Основной характеристикой шифра является его *криптостойкость*, которая определяет его стойкость к раскрытию с помощью методов криптоанализа. Обычно эта характеристика определяется интервалом времени, необходимым для раскрытия шифра.

К шифрам, используемым для криптографической защиты информации, предъявляется ряд требований.

1.Зашифрованный текст должен поддаваться чтению только при наличии секретного ключа шифрования.

2.Закон Керхоффа – знание алгоритма шифрования не должно влиять на надежность защиты, стойкость шифра должна определяться только секретностью ключа. Иными словами, данное требование предполагает, что весь алгоритм шифрования, кроме значения секретного ключа, известен криптоаналитику противника.

3. Единственно возможный метод раскрытия шифротекста должен заключаться в дешифровании его на секретном ключе. Единственно возможный способ нахождения ключа дешифрования должен заключаться в полном их переборе.

4. При знании криптоаналитиком шифротекста C и соответствующего ему открытого текста M, для нахождения ключа шифрования необходим полный перебор ключей (невозможность криптоаналитической атаки по открытому тексту).

5. Незначительное изменение ключа шифрования или открытого текста должно приводить к существенному изменению вида шифротекста.

6. Избыточность информации, вносимая в шифротекст за счет шифрования, должна быть незначительной.

7. Алгоритм шифрования должен допускать как программную, так и аппаратную реализацию.

5.2. Традиционные симметричные криптосистемы

В *симметричных криптосистемах* (*криптосистемах с секретным ключом*) шифрование и дешифрование информации осуществляется на одном ключе *K*, являющемся секретным. Рассекречивание ключа шифрования ведет к рассекречиванию всего защищенного обмена. Для того, чтобы подчеркнуть факт использования одного и того же ключа в шифраторе источника и дешифраторе получателя сообщений, криптосистемы с секретными ключами называют также *одноключевыми*.

Функциональная схема взаимодействия участников симметричного криптографического обмена приведена на рис. 5.1.

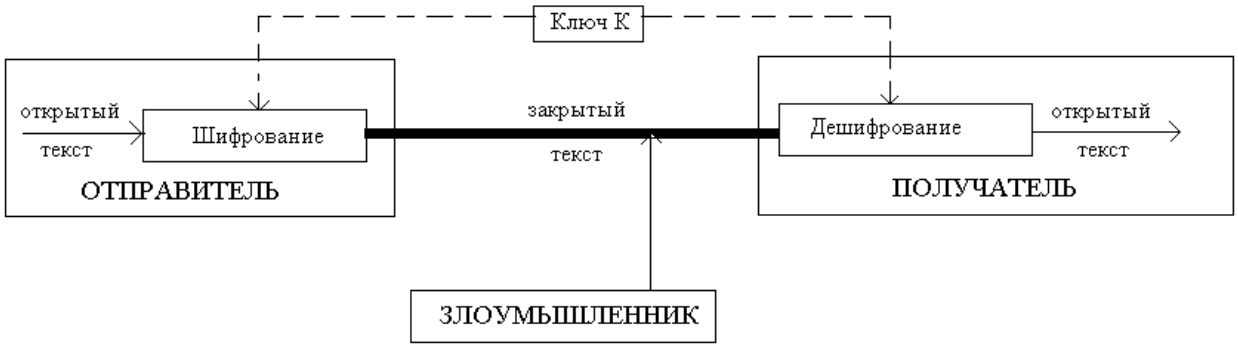


Рис. 5.1. Функциональная схема симметричной криптосистемы

В симметричной криптосистеме секретный ключ необходимо передать всем участникам криптографической сети по некоторому защищенному каналу.

Традиционные симметричные криптосистемы можно разделить на следующие основные виды [4,5].

1. Шифры замены.

2. Шифры перестановки.

3. Шифры гаммирования.

5.2.1. Шифрование методом замены

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

Данные шифры являются наиболее древними. Принято делить шифры замены на моноалфавитные и многоалфавитные.

При *моноалфавитной замене* каждой букве алфавита открытого текста ставится в соответствие одна и та же буква шифротекста из этого же алфавита одинаково на всем протяжении текста.

Рассмотрим наиболее известные шифры моноалфавитной замены.

Шифрование методом Цезаря

Свое название данный шифр получил по имени римского императора Гая Юлия Цезаря, который использовал этот шифр при переписке с Цицероном (около 50 г. до н.э).

При шифровании исходного текста по данному методу каждая буква заменяется на другую букву того же алфавита путем ее смещения в используемом алфавите на число позиций, равное K. При достижении конца алфавита выполняется циклический переход к его началу.

Общая формула шифра Цезаря имеет следующий вид:

$$C=P+K \ (mod\ M), \qquad (5.1)$$

где P – номер символа открытого текста, C – соответствующий ему номер символа шифротекста, K – ключ шифрования (коэффициент сдвига), M – размер алфавита (для русского языка M=32)

Для данного шифра замены можно задать фиксированную таблицу подстановок, содержащую соответствующие пары букв открытого текста и шифротекста.

Пример 5.1

Таблица подстановок для символов русского текста при ключе K=3 представлена в таблице 5.1. Данной таблице соответствует формула

$$C=P+3 \ (mod\ 32) \qquad (5.2)$$

Табл. 5.1. Табл. подстановок шифра Цезаря для ключа K=3

А	→	Г	Р	→	У
Б	→	Д	С	→	Ф
В	→	Е	Т	→	Х
Г	→	Ж	У	→	Ц
Д	→	З	Ф	→	Ч
Е	→	И	Х	→	Ш
Ж	→	Й	Ц	→	Щ
З	→	К	Ч	→	Ь
И	→	Л	Ш	→	Ы
Й	→	М	Щ	→	Ъ
К	→	Н	Ь	→	Э
Л	→	О	Ы	→	Ю
М	→	П	Ъ	→	Я
Н	→	Р	Э	→	А
О	→	С	Ю	→	Б
П	→	Т	Я	→	В

Согласно формуле (5.2) открытый текст «БАГАЖ» будет преобразован в шифротекст «ДГЖГЙ».

Дешифрование закрытого текста, зашифрованного методом Цезаря согласно (5.1), осуществляется по формуле

$$C=P-K \ (mod\ M), \qquad (5.3)$$

Простая моноалфавитная замена

Шифр простой моноалфавитной замены является обобщением шифра Цезаря и выполняет шифрование по следующей схеме:

$$C = a \cdot P + K \ (\mathrm{mod}\ M), \qquad (5.4)$$

где $0 \le a, K < M$ - ключ шифрования, $НОД(a, M) = 1$.

Преобразование согласно схеме (5.4) является взаимно однозначным отображением только в том случае, если a и M взаимно простые. В этом случае для дешифрования закрытого текста выполняют обратное преобразование по формуле (5.5)

$$P = a^{-1} \cdot (C - K) \pmod{M} \qquad (5.5)$$

Пример 5.2.

Пусть $M=26$, $a=3$, $K=6$, НОД(3,26) = 1. Тогда получаем следующую таблицу подстановок для шифра простой моноалфавитной замены (в таблице указаны коды букв русского алфавита).

	A	B	C	D	E	F	G	H	I	G	K	L	M	N	O	P	Q	R	S
P	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
C	6	9	12	15	18	21	24	1	4	7	10	13	16	19	22	25	2	5	8
	T	U	V	W	X	Y	Z												
P	19	20	21	22	23	24	25												
C	11	14	17	20	23	0	3												

Тогда открытый текст «HOME» будет преобразован в шифротекст «BWQS».

Шифрующие таблицы Трисемуса

Данный шифр был предложен в 1508 году аббатом из Германии Иоганом Трисемусом [4]. Для получения данного шифра замены им было предложено использовать таблицу для записи букв алфавита и ключевого слова или фразы. В таблицу сначала вписывалось по строкам ключевое слово, причем повторяющиеся буквы отбрасывались. Затем эта таблица дополнялась не вошедшими в нее буквами алфавита по порядку. При шифровании в построенной таблице находят очередную букву открытого текста и записывают в шифротекст букву, расположенную ниже ее в том же столбце. Если

буква открытого текста оказывается в нижней строке таблицы, тогда для шифротекста берут самую верхнюю букву из того же столбца.

Для русского алфавита шифрующая таблица может иметь размер 4x8.

Пример 5.3

Выберем в качестве ключа слово ПАМЯТНИК. Шифрующая таблица с данным ключом представлена в таблице 5.2.

Табл. 5.2. Шифрующая таблица Трисемуса

П	А	М	Я	Т	Н	И	К
Б	В	Г	Д	Е	Ж	З	Й
Л	О	Р	С	У	Ф	Х	Ц
Ч	Ш	Щ	Ь	Ы	Ъ	Э	Ю

Тогда открытый текст «НЕРУКОТВОРНЫЙ» будет преобразован в закрытый текст «ЖУЩЫЙШЕОШЩЖТЦ».

Достоинством методов моноалфавитной замены является простота шифрования и дешифрования.

Основным недостатком данных методов является то, что подстановки, выполняемые в соответствие с данными методами, не маскируют частоты появления различных букв закрытого текста. Это позволяет легко атаковать данные методы шифрования путем анализа частотности символов закрытого текста. Особенности реализации данного метода криптоанализа будут рассмотрены далее.

При *многоалфавитной замене* каждой букве алфавита открытого текста в различных ситуациях ставятся в соответствие различные буквы шифротекста в зависимости от соответствующего ей элемента ключа. В данном случае для шифрования каждого символа открытого текста применяют свой шифр моноалфавитной замены, причем смена алфавитов осуществляется последовательно и циклически, т.е. первый символ заменяется соответствующим символом первого алфавита, второй – символом второго алфавита и т. д. до тех пор, пока не будут использованы все выбранные алфавиты. После этого использование алфавитов повторяется.

Многоалфавитные шифры замены предложил и ввел в практику криптографии Леон Батист Альберти. Рассмотрим ряд примеров шифров многоалфавитной замены.

Шифр Гронсфельда

Данный шифр представляет собой модификацию шифра Цезаря с числовым ключом. При реализации данного шифра под буквами исходного сообщения записывают цифры числового ключа. Если ключ короче сообщения, то его запись циклически повторяют. Получение символа шифротекста осуществляют также, как это делается в шифре Цезаря, при этом смещение символа открытого текста производят на количество позиций, соответствующего цифре ключа, стоящей под ним.

Пример 5.4.

Пусть необходимо зашифровать исходное сообщение «НОЧЕВАЛА ТУЧКА ЗОЛОТАЯ», в качестве ключа возьмем *К*=193431.

Сообщение	Н	О	Ч	Е	В	А	Л	А	Т	У	Ч	К	А	З	О	Л	О	Т	А	Я
Ключ	1	9	3	4	3	1	1	9	3	4	3	1	1	9	3	4	3	1	1	9
Шифротекст	О	Ч	Ь	Й	Е	Б	М	Й	Х	Ч	Ь	Л	Б	Р	С	П	С	У	Б	И

Для того, чтобы зашифровать первую букву сообщения Н, необходимо сдвинуть ее в алфавите русских букв на число позиций 1, в результате чего получим букву О.

Дешифрование шифротеста предполагает сдвиг его символов на необходимое число позиций в обратную сторону.

Система шифрования Вижинера

Отличие системы Вижинера от шифра Гронсфельда заключается в том, что элементами ключа в данном случае могут быть не только цифры от 0 до 9, но и произвольные символы некоторого алфавита.

При шифровании исходного сообщения его, как и в шифре Гронсфельда, выписывают в строку, а под ним записывают ключевое слово или фразу.

Если ключ оказался короче сообщения, то его циклически повторяют. Все символы используемого алфавита пронумерованы от 0 до *M-1*, где *M* – размер алфавита. Преобразование символа открытого текста осуществляется по формуле

$$C_i=P_i+K_i \ (mod\ M), \qquad (5.6)$$

где P_i - номер символа открытого текста, K_i – номер расположенного под ним символа ключа, C_i – номер символа шифротекста.

Преобразование символа закрытого текста в символ открытого осуществляется по формуле

$$P_i=C_i-K_i \ (mod\ M)$$

Пример 5.6.

Рассмотрим пример шифрования сообщения ПРИЛЕТАЮ ДНЕМ по методу Вижинера с помощью ключевого слова СИСТЕМА

Сообщение	П	Р	И	Л	Е	Т	А		Ю	Д	Н	Е	М
Ключ	С	И	С	Т	Е	М	А		С	И	С	Т	Е
Шифротекст	А	Ш	В	Э	К	Ю	А		П	М	Ю	Ч	С

В данном случае буквы русского алфавита пронумерованы от 0 до 31: А-0, Б-1, В-2, ... , Я-31.

Шифрование методом Вернама

Система шифрования Вернама является частным случаем системы шифрования Вижинера при значении модуля *M*=2 [13].

При шифровании открытого текста, каждый его символ представляется в двоичном виде. Ключ шифрования также представляется в двоичной форме. Шифрование исходного текста осуществляется путем сложения по модулю 2 двоичных символов открытого текста с двоичными символами ключа согласно (5.7).

$$Y=P\oplus K \qquad (5.7)$$

Дешифрование состоит в сложении по модулю 2 символов шифротекста с ключом.

Общая схема системы шифрования Вернама представлена на рис. 5.2.

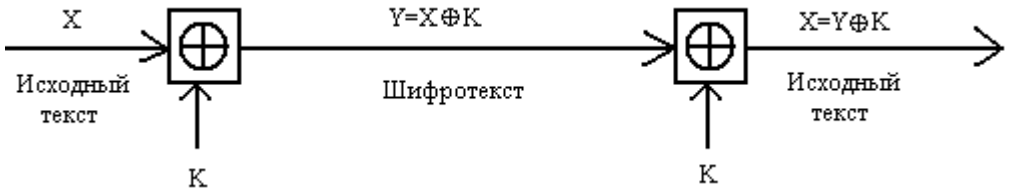


Рис. 5.2. Схема системы шифрования Вернама

Модификация системы шифрования Вернама используется для криптографической защиты информации в архиваторе ARJ. Формула (5.7) в этом случае преобразуется в следующую:

$$Y=P\oplus(K+\text{VALUE}), \qquad (5.8)$$

где VALUE – фиксированное значение.

Пример 5.7.

Зашифруем с помощью системы Вернама открытый текст «БЛАНК» с помощью ключа «ОХ».

Преобразуем открытый текст «БЛАНК» в ASCII коды: Б=193, Л=203, А=192, Н=205, К=202. В двоичном виде последовательность 193, 203, 192, 205, 202 представится в виде 11000001 11001011 11000000 11001101 11001010.

Преобразуем ключ «ОХ» в ASCII коды: О=206, Х=213. В двоичном виде последовательность 206, 213 представится в виде 11001110 11010101.

Подпишем циклически ключ под открытым текстом и выполним сложение по модулю 2 соответствующих битов.

Открытый текст	1	1	0	0	0	0	0	1	1	1	0	0	1	0	1	1	1	1	0	0
Ключ	1	1	0	0	1	1	1	0	1	1	0	1	0	1	0	1	1	1	0	0
Закрытый текст	0	0	0	0	1	1	1	1	0	0	0	1	1	1	1	0	0	0	0	0

Открытый текст	0	0	0	0	1	1	0	0	1	1	0	1	1	1	0	0	1	0	1	0

Ключ	1	1	1	0	1	1	0	1	0	1	0	1	1	1	0	0	1	1	1	0
Закрытый текст	1	1	1	0	0	0	0	1	1	0	0	0	0	0	0	0	0	1	0	0

G-контурная многоалфавитная замена

Данный метод шифрования предполагает многократное использование системы шифрования Вижинера при использовании различных ключей. *n*-контурная многоалфавитная замена предполагает наличие *n* различных ключей – K_1, K_2,..., K_n. Открытый текст *T* вначале шифруется с помощью ключа K_1, затем результат шифрования обрабатывается с помощью ключа K_2 и.т.д. до ключа K_n. Полученный в результате шифрования на ключе K_n текст и является искомым шифротекстом.

5.2.2. Шифрование методами перестановки

Шифрование перестановкой заключается в том, что символы открытого текста переставляются по определенному правилу в пределах некоторого блока этого текста. Данные преобразования приводят к изменению только порядка следования символов исходного сообщения.

При достаточной длине блока, в пределах которого осуществляется перестановка, и сложном неповторяющемся порядке перестановки можно достигнуть приемлемой для простых практических приложений стойкости шифра.

При шифровании *методом простой перестановки* производят деление открытого текста на блоки одинаковой длины, равной длине ключа. Ключ длины *n* представляет собой последовательность неповторяющихся чисел от 1 до *n*, в этом случае каждое из данных чисел встретится в ключе ровно один раз. Символы открытого текста внутри каждого из блоков переставляют в соответствие с символами ключа. Элемент ключа K_i в заданной позиции блока говорит о том, что на данное место будет помещен символ открытого текста с номером K_i из соответствующего блока.

Пример 5.8.

Зашифруем открытый текст «ПРИЕЗЖАЮДНЕМ» методом перестановки с ключом K=3142.

П	Р	И	Е	З	Ж	А	Ю	Д	Н	Е	М
3	1	4	2	3	1	4	2	3	1	4	2
И	П	К	Р	А	З	Ю	Ж	Е	Д	М	Н

Для дешифрования шифротекста необходимо символы шифротекста перемещать в позицию, указанную соответствующим им символом ключа K_i.

Весьма высокую стойкость шифрования можно обеспечить усложнением перестановок по *маршрутам типа гамильтоновских* [26]. При этом, для записи символов шифруемого текста используются вершины некоторого гиперкуба, а знаки зашифрованного текста считываются по маршрутам Гамильтона, причем используется восемь различных маршрутов. Размер ключа перестановки в данном случае равен восьми. Для примера, два из маршрутов Гамильтона представлено на рис. 5.3. Первому маршруту соответствует перестановка 4-0-2-3-1-5-7-6, второму 4-6-2-0-1-5-7-3 (нумерация символов в блоке осуществляется с нуля).

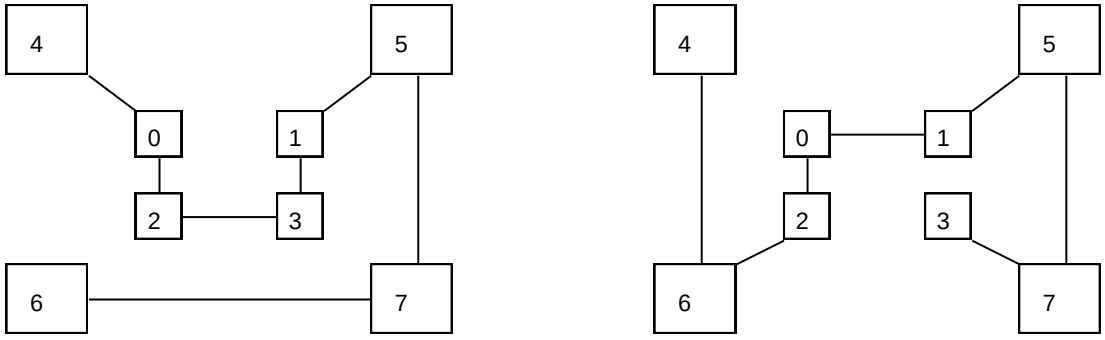


Рис. 5.3. Пример маршрутов Гамильтона

Пример 5.9.

Зашифруем открытый текст «ВОСЕМЬ МАРШРУТОВ» с помощью перестановок Гамильтона при использовании в качестве ключа двух перестановок, представленных на рис. 5.3.

0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
В	О	С	Е	М	Ь		М	А	Р	Ш	Р	У	Т	О	В
4	0	2	3	1	5	7	6	4	6	2	0	1	5	7	3

5.2.3. Шифрование методом гаммирования

Под *гаммированием* понимают наложение на открытые данные по определенному закону гаммы шифра [13].

Гамма шифра – псевдослучайная последовательность, вырабатываемая по определенному алгоритму, используемая для зашифровки открытых данных и дешифровки шифротекста.

Общая схема шифрования методом гаммирования представлена на рис. 5.4.

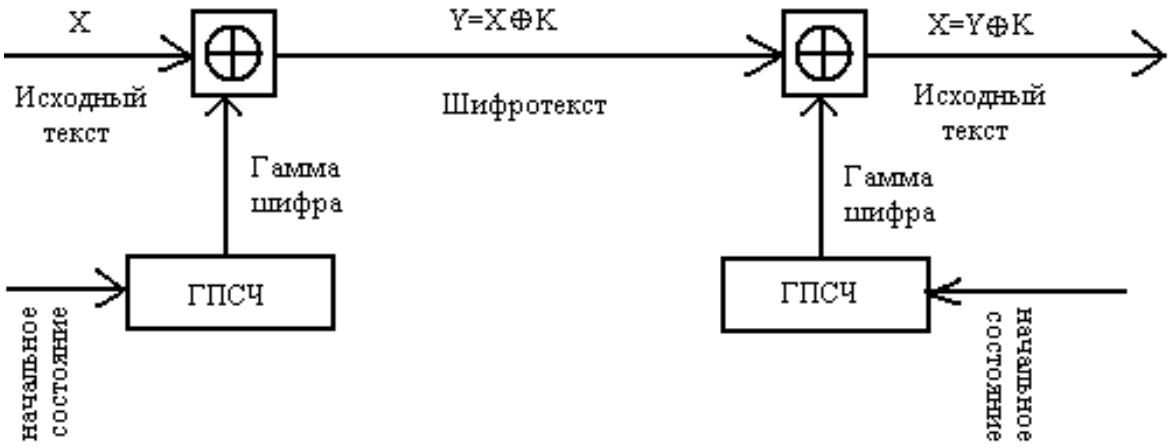


Рис. 5.4. Схема шифрования методом гаммирования

Принцип шифрования заключается в формировании генератором псевдослучайных чисел (ГПСЧ) гаммы шифра и наложении этой гаммы на открытые данные обратимым образом, например путем сложения по модулю два. Процесс дешифрования данных сводится к повторной генерации гаммы шифра и наложении гаммы на зашифрованные данные. Ключом шифрования в данном случае является начальное состояние генератора псевдослучайных чисел. При одном и том же начальном состоянии ГПСЧ будет формировать одни и те же псевдослучайные последовательности.

Перед шифрованием открытые данные обычно разбивают на блоки одинаковой длины, например по 64 бита. Гамма шифра также вырабатывается в

виде последовательности блоков той же длины. Схему шифрования можно записать в этом случае в виде

$$T_{Ш}^{(i)} = \Gamma_{Ш}^{(i)} \oplus T_{О}^{(i)}, i = \overline{1, N} \qquad (5.9)$$

где $T_{Ш}^{(i)}$ - i-ый блок шифротекста, $\Gamma_{Ш}^{(i)}$ - i-ый блок гаммы шифра, $T_{О}^{(i)}$ - i-ый блок открытого текста, N – количество блоков открытого текста.

Дешифрование в данном случае осуществляется по следующей формуле:

$$T_{О}^{(i)} = \Gamma_{Ш}^{(i)} \oplus T_{Ш}^{(i)}, i = \overline{1, N}$$

Стойкость шифрования методом гаммирования определяется главным образом свойствами гаммы – длиной периода и равномерностью статистических характеристик. Последнее свойство обеспечивает отсутствие закономерностей в появлении различных символов в пределах периода. Полученный зашифрованный текст является достаточно трудным для раскрытия в том случае, если гамма шифра не содержит повторяющихся битовых последовательностей. По сути дела гамма шифра должна изменяться случайным образом для каждого шифруемого блока.

Обычно разделяют две разновидности гаммирования – с конечной и бесконечной гаммами. При хороших статистических свойствах гаммы стойкость шифрования определяется только длиной периода гаммы. При этом, если длина периода гаммы превышает длину шифруемого текста, то такой шифр теоретически является абсолютно стойким, т.е. его нельзя вскрыть при помощи статистической обработки зашифрованного текста, а можно раскрыть только прямым перебором. Криптостойкость в этом случае определяется размером ключа.

В настоящее время разработано множество алгоритмов работы генераторов псевдослучайных чисел, которые обеспечивают удовлетворительные характеристики гаммы. Рассмотрим несколько примеров данных алгоритмов.

Метод фон Неймана

Суть данного метода состоит в том, что каждое последующее случайное число получается путем возведения в квадрат предыдущего числа с отбрасыванием цифр младших и старших разрядов.

Пусть A_0 – четырехзначное число - начальное состояние ГПСЧ. Тогда i – ое псевдослучайное число A_i получается из предыдущего числа A_{i-1} в результате следующих преобразований:

1. Возведение A_{i-1} в квадрат, то есть нахождение числа A_{i-1}^2.

2. В качестве A_i выбирают четыре средние цифры числа A_{i-1}^2.

Пример 5.10

Пусть A_0=1204, A_0^2=1449616. Тогда A_1=4496, A_1^2=20214016, A_2=2140 и т.д

Однако метод фон Неймана является очень ненадежным, обладает множеством недостатков, в связи с чем, используется достаточно редко.

Линейный конгруэнтный метод

Данный генератор вырабатывает последовательность псевдослучайных чисел $Y_1,Y_2,...,Y_{i-1},Y_i,...$, используя соотношение

$$Y_i=(a*Y_{i-1}+b) \bmod m, \qquad (5.10)$$

где Y_i – i-ое (текущее) число последовательности; Y_{i-1} – предыдущее число последовательности; a,b,m – константы; m – модуль; a – коэффициент; b – приращение; Y_0 – начальное состояние ГПСЧ.

Обычно значение модуля m берется равным 2^n, либо простому числу. Приращение b должно быть взаимно простым с m, коэффициент a должен быть нечетным числом.

Линейный конгруэнтный метод является одним из самых простейших методов генерации псевдослучайных последовательностей. Существует ряд методов, формирующих намного более криптографически стойкие псевдослучайные последовательности.

5.3. Элементы криптоанализа

Любая попытка со стороны злоумышленника расшифровать шифротекст *C* и получить открытый текст *M* не имея подлинного ключа, называется *криптоаналитической атакой*.

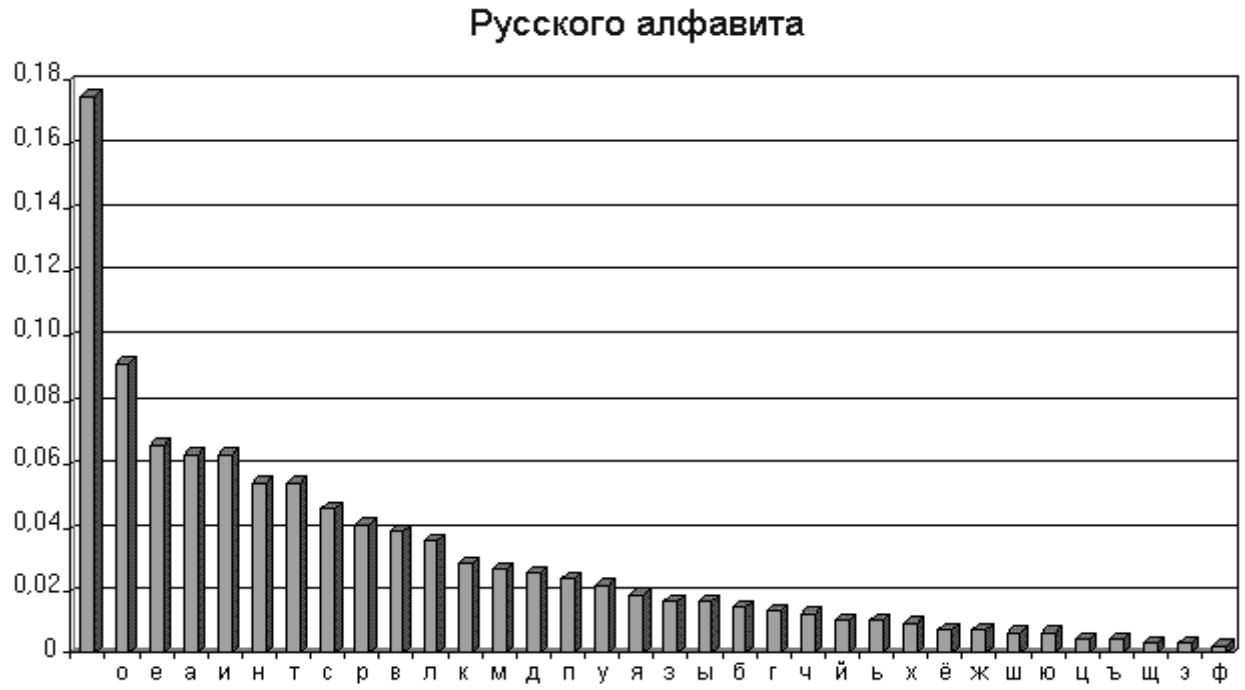
Криптоанализ – это наука о раскрытии исходного текста зашифрованного сообщения без доступа к ключу. Успешный криптоанализ может раскрыть исходный текст или ключ. Он позволяет также обнаружить слабые места в криптосистеме, что в конечном счете ведет к тем же результатам.

Один из широко используемых методов криптоанализа для недостаточно криптостойких алгоритмов заключается в анализе частотности символов, встречающихся в зашифрованном тексте. Такой криптоанализ называют криптоанализом, основанном на исследовании частотности символов закрытого текста.

Особенностью большинства искусственных языков (и всех естественных) является то, что они имеют характерное частотное распределение букв и других знаков.

При этом многие, недостаточно стойкие простейшие алгоритмы шифрования сохраняют частотность символов в тексте. В основном этот недостаток свойственен простейшим методам замены (например, шифру Цезаря и ему подобным). Это распределение частотности дает криптоаналитику путь к раскрытию шифра.

Частотное распределение букв русского и английского алфавита в художественных текстах представлено ниже



Исследовав шифротекст и обнаружив, что наиболее часто встречаемый в нем символ – это «Б», а второй по встречаемости - «К», криптоаналитик может сделать вывод, что символ «Б» это «Пробел», а «К» это буква «о».

Другим примером криптоаналитической атаки является атака по открытому тексту. Например, зная непрерывную часть открытого текста, зашифрованного методом Вернама, большую или равную длине ключа, можно вычислить ключ путем сложения по модулю 2 открытой и соответствующей ей закрытой части текста. Данный факт объясняет малую криптостойкость за-

крытых архивов ARJ, в которых система Вернама используется для криптографической защиты.

5.4. Современные симметричные системы шифрования

При построении стойких шифров необходимо использовать два основных принципа – рассеивание и перемешивание [13].

Рассеивание предполагает распространение влияния одного знака открытого текста на множество знаков шифротекста, что позволяет скрыть статистические свойства открытого текста.

Перемешивание предполагает использование таких шифрующих преобразований, которые усложняют восстановление взаимосвязи статистических свойств открытого и шифротекста.

Обычно для достижения эффектов рассеивания и перемешивания используют шифры, реализованные в виде последовательности простых традиционных шифров, каждый из которых вносит свой вклад в суммарное рассеивание и перемешивание. Наиболее часто при этом используют традиционные шифры перестановки и замены.

При многократном чередовании простых перестановок и замен, управляемых достаточно длинным секретным ключом, можно получить очень стойкий шифр с хорошим рассеиванием и перемешиванием. Большинство существующих стандартов шифрования построены в полном соответствии с данной методологией.

5.4.1. Стандарт шифрования DES (США)

Алгоритм, изложенный в стандарте DES (Data Encryption Standard), наиболее распространен и широко применяется для шифрования данных в США [7]. Он был разработан фирмой IBM для собственных целей, но после проверки Агентством Национальной Безопасности США был рекомендован к применению в качестве федерального стандарта шифрования. Алгоритм DES не является закрытым и был опубликован для широкого ознакомления.

Алгоритм предназначен для зашифровки и расшифровки блоков данных длиной по 64 бита под управлением 64-битового ключа, в котором значащими являются 56 бит. Дешифрование в DES выполняется путем повторения операций шифрования в обратной последовательности.

Обобщенная схема шифрования алгоритма DES представлена на рис. 5.5.

Рис. 5.5. Обобщенная схема шифрования алгоритма DES

Пусть из открытого текста взят очередной 64-битовый блок T. Этот блок T преобразуется с помощью *матрицы начальной перестановки* IP. Данная перестановка фиксирована и приведена в таблице 5.3.

Табл. 5.3. Начальная перестановка (IP)

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

Например, бит 58 входного блока T становится битом 1, бит 50 – битом 2 и т.д.

Далее, полученная в результате перестановки последовательность битов T_0 разделяется на 2 последовательности: L_0 – старшие 32 бита, R_0 – младшие 32 бита.

Затем выполняется итеративный процесс шифрования, состоящий из 16 циклов. Если T_i – последовательность битов, полученная на i – ой итерации, $T_i=L_iR_i$, то результат i-ой итерации описывается следующими формулами:

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

где f – функция шифрования.

По окончании шифрования осуществляется конечная перестановка позиций битов последовательности с помощью матрицы обратной перестановки IP^{-1} (табл. 5.4).

Табл. 5.4. Конечная перестановка(IP^{-1})

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

Схема вычисления функции шифрования $f(R_{i-1},K_i)$ представлена на рис. 5.6.

Для вычисления функции f используются:

– функция E, расширяющая 32-битовое значение до 48 бит;

– функции $S_1,...,S_8$, преобразующие 6-битовое число в 4-битовое;

– функция P, осуществляющая перестановку битов в 32-битовой последовательности.

Функция расширения E определяется таблицей 5.5 выборки битов.

После расширения R_{i-1} результат складывается по модулю два с текущим значением ключа K_i и затем разбивается на восемь 6-битовых блоков $B_1,B_2,...,B_8$. Далее каждый из этих блоков используется как номер элемента в функциях-матрицах $S_1,S_2,...,S_8$, содержащих 4-битовые значения.

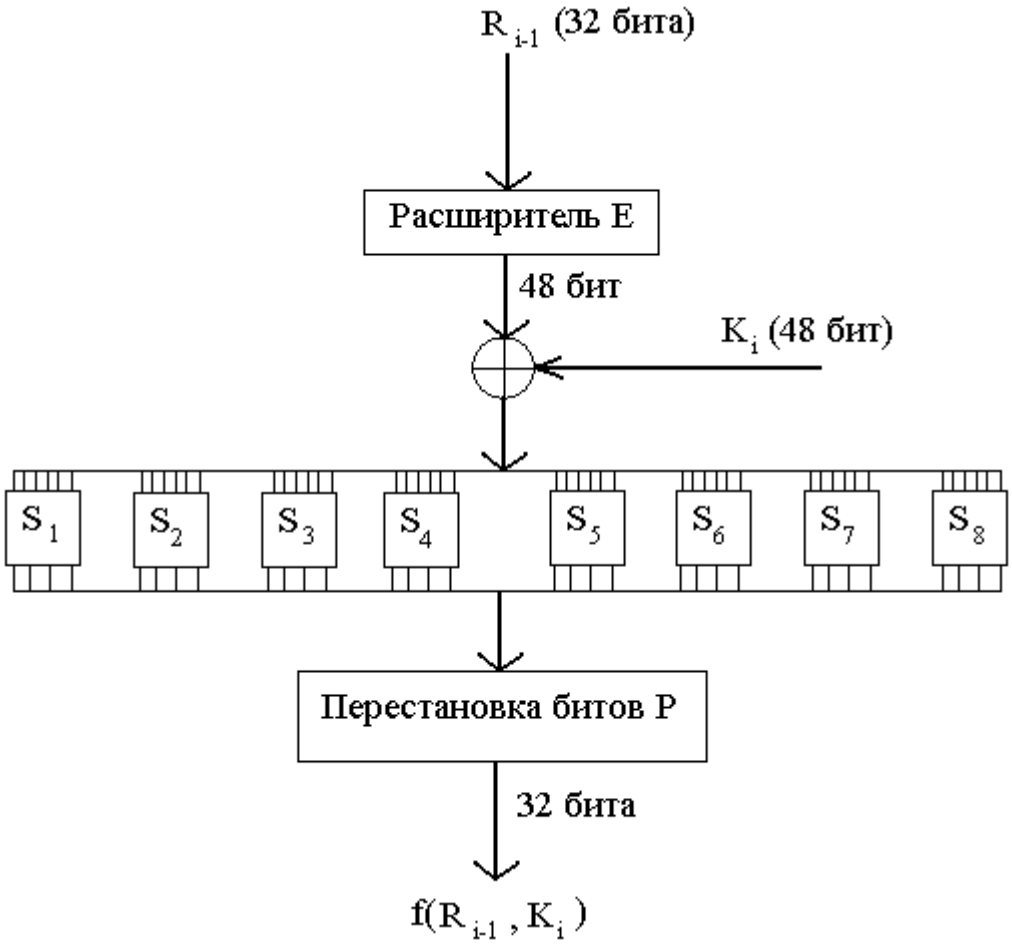


Рис. 5.6. Схема вычисления функции шифрования f

Табл. 5.5. Табл. функции расширения E

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

Выбор элемента в матрице S_j осуществляется следующим образом. Пусть на вход матрицы S_j поступает 6-битовый блок $B_1=b_1b_2b_3b_4b_5b_6$, тогда двухбитовое число b_1b_2 указывает номер строки, а четырехбитовое число $b_2b_3b_4b_5$ – номер столбца матрицы S_j, откуда и берется требуемое четырехбитовое значение (табл. 5.6). Совокупность 6-битовых блоков $B_1B_2,...,B_8$ обеспечивает выбор четырехбитового элемента в каждой из матриц $S_1,S_2,...,S_8$.

Табл. 5.6. Функции преобразования $S_1,...,S_8$

Блок замены 1 (S[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

Блок замены 2 (S[2])

15	1	8	14	6	11	3	4	9	7	2	14	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

Блок замены 3 (S[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

Блок замены 4 (S[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	5	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

Блок замены 5 (S[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	9	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

Блок замены 6 (S[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

Блок замены 7 (S[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

Блок замены 8 (S[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

В результате получаем значение $S_1(B_1)$ $S_2(B_2)$... $S_8(B_8)$ – 32 битовый блок, который преобразуется с помощью функции перестановки битов P (таблица 5.7.)

Табл. 5.7. Функция перестановки битов P

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

Необходимо отметить, что на каждой итерации в схеме на рис. 5.6 используется новое значение ключа K_i. Новое значение ключа K_i вычисляется из начального ключа K. Ключ K представляет собой 64-битовый блок с 8 битами контроля по четности, расположенными в позициях 8,16,24,32,40,48,56,64. Для удаления контрольных битов и подготовки ключа к работе используется функция G первоначальной подготовки ключа (табл. 5.8). Данная таблица имеет размер 7x8 и из нее исключены контрольные биты, то есть они не используются при шифровании.

Табл. 5.8. Функция G первоначальной подготовки 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

Полученную в результате перестановки битов ключа последовательность делят на 2 части – C_0 и D_0. После этого, рекурсивно вычисляются C_i,D_i, i=1,2,...,16. Для этого применяются операции независимого для C_i и D_i циклического сдвига влево на 1 или 2 бита в зависимости от номера шага итерации. Число сдвигов на итерацию приведено в таблице ниже.

Итерация № 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

Сдвиги влево 1 1 2 2 2 2 2 2 1 2 2 2 2 2 2 1

Ключ K_i, определяемый на каждом шаге итерации, есть результат перестановки конкатенации C_iD_i согласно функции H, представленной в таблице 5.9.

Табл. 5.9. Функция H завершающей обработки ключа

14	17	11	24	1	5
3	28	15	6	21	10
23	19	12	4	26	8
16	7	27	20	13	2
41	52	31	37	47	55
30	40	51	45	33	48
44	49	39	56	34	53
46	42	50	36	29	32

Для шифрования или дешифрования более чем 64 битного блока существуют официальные режимы. Один из них - вычисления для каждого блока в ряде. Он назван *режимом электронной кодовой книги*. Более сильный метод заключается в суммировании по модулю два блока открытого текста с блоком шифротекста, прежде зашифрованным. Данный метод называется *связыванием шифр-блоков* (Cifer Block Chaining, CBC).

Другие два режима - *режим с выходной обратной связью* (Output Feedback mode, OFB) и *шифрование с обратной связью* (Cifer Feedback mode, CFB) распространяют искажения в открытом тексте, что применяется для проверки целостности информации.

Число различных ключей DES-алгоритма равно $2^{56} = 7*10^{16}$. Недавние исследования показали, что современная технология позволяет создать вычислительное устройство стоимостью около 1 млн. долларов, способное вскрыть секретный ключ с помощью полного перебора в среднем за 3,5 часа.

В настоящее время криптостойкость алгоритма DES не удовлетворяет реальным потребностям, в связи с чем, данный алгоритм в настоящее время заменен в США на более стойкий алгоритм AES.

5.4.2. Отечественный стандарт симметричного шифрования

Российская Федерация имеет свой собственный стандарт симметричного шифрования. Этот стандарт закреплен ГОСТом №28147-89, принятом в 1989 году в СССР [10]. Данный стандарт обязателен для организаций, предприятий и учреждений, применяющих криптографическую защиту данных, относящихся к государственной тайне, хранимых и передаваемых в сетях ЭВМ и в отдельных вычислительных комплексах. Помимо нескольких тесно связанных между собой процедур шифрования, в стандарте описан алгоритм выработки *имитовставки*. Последняя является не чем иным, как криптографической контрольной комбинацией, то есть кодом, вырабатываемым из исходных данных с использованием секретного ключа с целью *имитозащиты*, или защиты данных от внесения в них несанкционированных изменений.

Алгоритм предусматривает четыре режима работы:

- шифрование данных в режиме простой замены;
- шифрование данных в режиме гаммирования;
- шифрование данных в режиме гаммирования с обратной связью;
- выработка имитовставки.

В ГОСТе 28147–89 содержится описание алгоритмов нескольких уровней. На самом верхнем уровне находятся практические алгоритмы, предназначенные для шифрования массивов данных и выработки для них имитовставки. Все они опираются на три алгоритма низшего уровня, называемые в ГОСТе *циклами*. Эти фундаментальные алгоритмы чаще называют *базовыми циклами*, чтобы отличать их от всех прочих циклов. В свою очередь, каждый из базовых циклов представляет собой многократное повторение одной единственной процедуры, называемой *основным шагом криптопреобразования*.

В ГОСТе ключевая информация состоит из двух структур данных - собственно ключа, необходимого для всех шифров, и таблицы замен. Ключ является массивом из восьми 32-битных элементов кода (всего 256 бит). Обозначим его символом $\boldsymbol{K}$: $K = \{K_i\}_{0 \le i \le 7}$. *Таблица замен* является матрицей

8×16, содержащей 4-битовые элементы, которые можно представить в виде целых чисел от 0 до 15. Строки таблицы замен называются *узлами замен*, они должны содержать различные значения, то есть каждый узел замен должен содержать 16 различных чисел от 0 до 15 в произвольном порядке. Таблицу замен обозначим символом $\boldsymbol{H}$: $H = \{H_{i,j}\}_{0 \le j \le 15}^{0 \le i \le 7}, 0 \le H_{i,j} \le 15$. Общий объем таблицы замен равен: 8 узлов × 16 элементов/узел × 4 бита/элемент = 512 бит или 64 байта.

Основной шаг криптопреобразования

Основной шаг криптопреобразования по своей сути является оператором, определяющим преобразование 64-битового блока данных. Дополнительным параметром этого оператора является 32-битовый блок, в качестве которого используется элемент ключа. Схема алгоритма основного шага приведена на 5.7.

Шаг 0. Определяет исходные данные для основного шага криптопреобразования:

– $\boldsymbol{N}$ – преобразуемый 64-битовый блок данных, в ходе выполнения шага его младшая ($\boldsymbol{N}_1$) и старшая ($\boldsymbol{N}_2$) части обрабатываются как отдельные 32-битовые целые числа без знака. $\boldsymbol{N}=(\boldsymbol{N}_1,\boldsymbol{N}_2)$.

– $\boldsymbol{X}$ – 32-битовый элемент ключа;

Шаг 1. Сложение с ключом. Младшая половина преобразуемого блока складывается по модулю 2^{32} с используемым на шаге элементом ключа.

Шаг 2. Поблочная замена. 32-битовое значение, полученное на предыдущем шаге, интерпретируется как массив из восьми 4-битовых блоков кода: $\boldsymbol{S}=(\boldsymbol{S}_0,\boldsymbol{S}_1,\boldsymbol{S}_2,\boldsymbol{S}_3,\boldsymbol{S}_4,\boldsymbol{S}_5,\boldsymbol{S}_6,\boldsymbol{S}_7)$. Далее значение каждого из восьми блоков заменяется на новое, которое выбирается по таблице замен следующим образом: значение блока $\boldsymbol{S}_i$ заменяется на $\boldsymbol{S}_i$-ый по порядку элемент (нумерация с нуля) $\boldsymbol{i}$-го узла замен (т.е. $\boldsymbol{i}$-ой строки таблицы замен, нумерация также с нуля). Другими словами, в качестве замены для значения блока выбирается элемент из таблицы замен с номером строки, равным номеру заменяемого бло-

ка, и номером столбца, равным значению заменяемого блока как 4-битового целого неотрицательного числа.

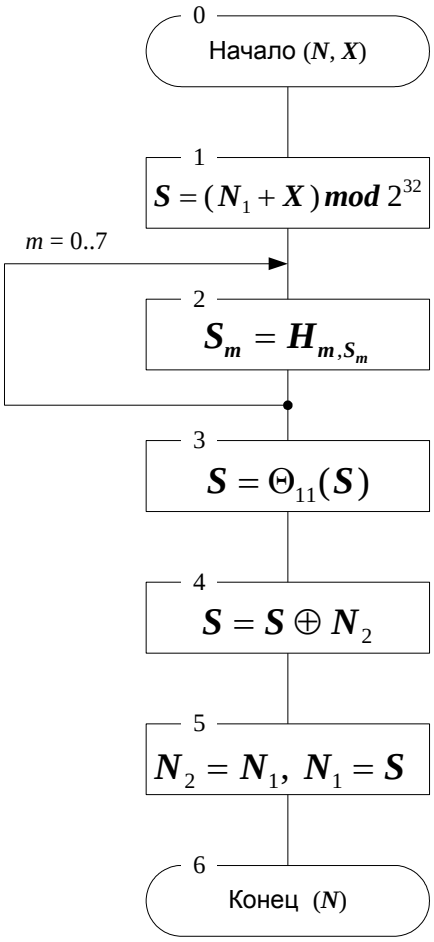


Рис. 5.7. Схема основного шага криптопреобразования алгоритма ГОСТ 28147-89

Шаг 3. Результат предыдущего шага сдвигается циклически на 11 бит в сторону старших разрядов и передается на следующий шаг. На схеме алгоритма символом Θ_{11} обозначена функция циклического сдвига своего аргумента на 11 бит в сторону старших разрядов.

Шаг 4. Побитовое сложение: значение, полученное на шаге 3, побитно складывается по модулю 2 со старшей половиной преобразуемого блока.

Шаг 5. Сдвиг по цепочке: младшая часть преобразуемого блока сдвигается на место старшей, а на ее место помещается результат выполнения предыдущего шага.

Шаг 6. Полученное значение преобразуемого блока возвращается как результат выполнения алгоритма основного шага криптопреобразования.

Базовые циклы криптографических операций

ГОСТ 28147-89 относится к классу блочных шифров, то есть единицей обработки информации в нем является блок данных. Таким образом, в нем определены алгоритмы для криптографических преобразований одного блока данных. Именно эти алгоритмы и называют *базовыми циклами* ГОСТа. Они заключаются в многократном выполнении *основного шага* с использованием разных элементов ключа и отличаются друг от друга только числом повторения шага и порядком использования ключевых элементов. Каждый из циклов имеет собственное буквенно-цифровое обозначение, соответствующее шаблону «*n-X*», где *n* задает число повторений основного шага в цикле, а *X*, буква, задает порядок зашифрования («З») или расшифрования («Р») в использовании ключевых элементов.

Приведем этот порядок использования ключевых элементов для различных циклов.

1. **Цикл зашифрования 32-З**:

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

2. **Цикл расшифрования 32-Р**:

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

3. **Цикл выработки имитовставки 16-З:**

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

Цикл расшифрования должен быть обратным циклу зашифрования, то есть последовательное применение этих двух циклов к произвольному блоку должно дать в итоге исходный блок:

$$Ц_{32\text{-}P}(Ц_{32\text{-}З}(T))=T,$$

где $\boldsymbol{T}$ – произвольный 64-битный блок данных, $\boldsymbol{Ц_X(T)}$ – результат выполнения цикла *X* над блоком данных $\boldsymbol{T}$. Для выполнения этого условия для алгоритмов, подобных ГОСТ 28147-89, необходимо и достаточно, чтобы порядок использования ключевых элементов соответствующими циклами был вза-

имно обратным. Из сказанного вытекает следствие: свойство цикла быть обратным другому циклу является взаимным, то есть цикл 32-З является обратным по отношению к циклу 32-Р. Другими словами, зашифрование блока данных теоретически может быть выполнено с помощью цикла расшифрования, в этом случае расшифрование блока данных должно быть выполнено циклом зашифрования. Из двух взаимно обратных циклов любой может быть использован для зашифрования, тогда второй должен быть использован для расшифрования данных, однако стандарт ГОСТ 28147-89 закрепляет роли за циклами и не предоставляет пользователю права выбора в этом вопросе.

Цикл выработки имитовставки вдвое короче циклов шифрования, но порядок использования ключевых элементов в нем такой же, как в первых 16 шагах цикла зашифрования, поэтому этот порядок в обозначении цикла кодируется той же самой буквой «З». Между циклами шифрования и вычисления имитовставки есть еще одно отличие: в конце базовых циклов шифрования старшая и младшая часть блока результата меняются местами, это необходимо для их взаимной обратимости.

В дальнейшем будем использовать следующие обозначения:

$T_{\text{о}}, T_{\text{ш}}$ – массивы соответственно открытых и зашифрованных данных;

$T_i^{\text{О}}, T_i^{\text{Ш}}$ – i-ые по порядку 64-битные блоки соответственно открытых и зашифрованных данных: $T_{\text{О}} = (T_1^{\text{О}}, T_2^{\text{О}}, ..., T_n^{\text{О}})$, $T_{\text{Ш}} = (T_1^{\text{Ш}}, T_2^{\text{Ш}}, ..., T_n^{\text{Ш}})$, $1 \le i \le n$, последний блок может быть неполным: $|T_i^{\text{О}}| = |T_i^{\text{Ш}}| = 64$ при $1 \le i < n, 1 \le |T_n^{\text{Ш}}| = |T_n^{\text{Ш}}| \le 64$;

n – число 64-битных блоков в массиве данных;

$Ц_{\text{Х}}$ – функция преобразования 64-битного блока данных по алгоритму базового цикла «Х»;

Рассмотрим основные режимы шифрования ГОСТ 28147-89.

Простая замена

Зашифрование в данном режиме заключается в применении цикла 32-З к блокам открытых данных, расшифрование – цикла 32-Р к блокам зашифрованных данных. Это наиболее простой из режимов, 64-битовые блоки данных обрабатываются в нем независимо друг от друга.

Размер массива открытых или зашифрованных данных, подвергающийся соответственно зашифрованию или расшифрованию, должен быть кратен 64 битам: $|T_o|=|T_ш|=64\cdot n$, размер полученного массива данных не изменяется.

Режим шифрования простой заменой имеет следующие особенности:

1) Так как блоки данных шифруются независимо друг от друга и от их позиции в массиве, при зашифровании двух одинаковых блоков открытого текста получаются одинаковые блоки шифротекста и наоборот. Это свойство позволит криптоаналитику сделать заключение о тождественности блоков исходных данных, если в массиве зашифрованных данных ему встретились идентичные блоки, что является недопустимым для серьезного шифра.

2) Если длина шифруемого массива данных не кратна 64 битам, возникает проблема, чем и как дополнять последний неполный блок данных массива до полных 64 бит.

На первый взгляд, перечисленные выше особенности делают практически невозможным использование режима простой замены, ведь он может применяться только для шифрования массивов данных с размером кратным 64 битам, не содержащим повторяющихся 64-битных блоков. Кажется, что для любых реальных данных гарантировать выполнение указанных условий невозможно, но есть одно очень важное исключение: размер ключа составляет 32 байта, а размер таблицы замен – 64 байта. Кроме того, наличие повторяющихся 8-байтовых блоков в ключе или таблице замен будет говорить об их весьма плохом качестве, поэтому в реальных ключевых элементах такого повторения быть не может. Таким образом, режим простой замены вполне подходит для шифрования ключевой информации, тем более, что прочие режимы для этой цели менее удобны, поскольку требуют наличия дополнительного синхронизирующего элемента данных – синхропосылки.

ГОСТ 28147-89 предписывает использовать режим простой замены исключительно для шифрования ключевых данных.

Гаммирование

Для того, чтобы избавиться от недостатков режима простой замены, необходимо сделать возможным шифрование блоков с размером менее 64 бит и обеспечить зависимость блока шифротекста от его номера - *рандомизировать* процесс шифрования. В ГОСТ 28147-89 это достигается двумя различными способами в двух режимах шифрования, предусматривающих *гаммирование* – наложение (снятие) на открытые (зашифрованные) данные криптографической гаммы для получения зашифрованных (открытых) данных. Для наложения гаммы используется операция побитного сложения по модулю 2. Гаммирование решает обе проблемы; во-первых, все элементы гаммы различны для реальных шифруемых массивов и, следовательно, результат зашифрования даже двух одинаковых блоков в одном массиве данных будет различным. Во-вторых, хотя элементы гаммы и вырабатываются одинаковыми порциями в 64 бита, использоваться может и часть такого блока с размером, равным размеру шифруемого блока.

Гамма вычисляется следующим образом: с помощью заданного алгоритмического рекуррентного генератора последовательности чисел (РГПЧ) вырабатываются 64-битные блоки данных, которые далее подвергаются преобразованию по циклу 32-З, то есть зашифрованию в режиме простой замены, в результате получаются блоки гаммы. Благодаря тому, что наложение и снятие гаммы осуществляется при помощи одной и той же операции побитового «исключающего или», алгоритмы зашифрования и расшифрования в режиме гаммирования идентичны. РГПЧ, используемый для выработки гаммы, является рекуррентной функцией $\Omega_{i+1}=f(\Omega_i)$, где Ω_i – элементы рекуррентной последовательности, f – функция преобразования. Неизбежно возникает вопрос об инициализации РГПЧ, то есть элементе Ω_0. Этот элемент данных является параметром алгоритма для режимов гаммирования, на схемах он обозначен как *S*, и называется в криптографии *синхропосылкой*, а в ГОСТ

28147-89 – *начальным заполнением* одного из регистров шифрователя. Разработчики ГОСТ 28147-89 используют для инициализации РГПЧ не саму синхропосылку, а результат ее преобразования по циклу 32-З: $\Omega_0 = Ц_{32\text{-}З}(S)$. Последовательность элементов, вырабатываемых РГПЧ, целиком зависит от его начального заполнения, т.е. ее элементы являются функцией своего номера и начального заполнения РГПЧ. С учетом преобразования по алгоритму простой замены, добавляется еще и зависимость от ключа:

$$\Gamma_i = Ц_{32\text{-}З}(\Omega_i) = Ц_{32\text{-}З}(f_i(\Omega_0)) = Ц_{32\text{-}З}(f_i(Ц_{32\text{-}З}(S))) = \Omega_i(S,K),$$

где Γ_i – i-тый элемент гаммы, K – ключ.

Таким образом, последовательность элементов гаммы для использования в режиме гаммирования однозначно определяется ключевыми данными и синхропосылкой. Естественно, для обратимости процедуры шифрования в процессах за- и расшифрования должна использоваться одна и та же синхропосылка. Из требования уникальности гаммы, невыполнение которого приводит к катастрофическому снижению стойкости шифра, следует, что для шифрования двух различных массивов данных на одном ключе необходимо обеспечить использование различных синхропосылок. Это приводит к необходимости хранить или передавать синхропосылку по каналам связи вместе с зашифрованными данными, хотя в отдельных особых случаях она может быть предопределена или вычисляться особым образом, если исключается шифрование двух массивов на одном ключе.

Схема алгоритма шифрования в режиме гаммирования приведена на рис. 5.8.

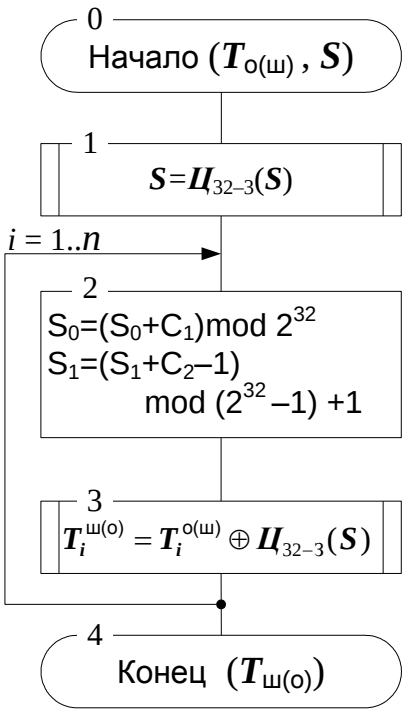


Рис. 5.8. Алгоритм зашифрования (расшифрования) данных в режиме гаммирования

Шаг 0. Определяет исходные данные для основного шага криптопреобразования:

$T_{о(ш)}$ – массив открытых (зашифрованных) данных произвольного размера, подвергаемый процедуре зашифрования (расшифрования), по ходу процедуры массив подвергается преобразованию порциями по 64 бита;

S – синхропосылка, 64-битный элемент данных, необходимый для инициализации генератора гаммы;

Шаг 1. Начальное преобразование синхропосылки, выполняемое для ее «рандомизации», то есть для устранения статистических закономерностей, присутствующих в ней, результат используется как начальное заполнение;

Шаг 2. Один шаг работы РГПЧ, реализующий его рекуррентный алгоритм. В ходе данного шага старшая (S_1) и младшая (S_0) части последовательности данных вырабатываются независимо друг от друга;

Шаг 3. Гаммирование. Очередной 64-битный элемент, выработанный РГПЧ, подвергается зашифрованию по циклу 32–З, результат используется как элемент гаммы для зашифрования (расшифрования) очередного блока открытых (зашифрованных) данных того же размера.

Шаг 4. Результат алгоритма – зашифрованный (расшифрованный) массив данных.

Рассмотрим РГПЧ, используемый в ГОСТ 28147-89 для генерации элементов гаммы. РГПЧ спроектирован разработчиками, исходя из необходимости выполнения следующих условий:

– период повторения последовательности чисел, вырабатываемой РГПЧ, не должен сильно отличаться от максимального при данном размере блока значения 2^{64};

– соседние значения, вырабатываемые РГПЧ, должны отличаться друг от друга в каждом байте, иначе задача криптоаналитика будет упрощена;

– РГПЧ должен быть достаточно просто реализуем как аппаратно, так и программно.

Исходя из перечисленных принципов, создатели ГОСТа спроектировали РГПЧ, имеющий следующие характеристики:

– в 64-битовом блоке старшая и младшая части обрабатываются независимо друг от друга: $\Omega_i = (\Omega_i^0, \Omega_i^1), |\Omega_i^0| = |\Omega_i^1| = 32, \Omega_{i+1}^0 = \hat{f}(\Omega_i^0), \Omega_{i+1}^1 = \tilde{f}(\Omega_i^1)$, фактически, существуют два независимых РГПЧ для старшей и младшей частей блока.

– рекуррентные соотношения для старшей и младшей частей следующие:

$\Omega_{i+1}^0 = (\Omega_i^0 + C_1) \bmod 2^{32}$, где $\boldsymbol{C}_1 = 1010101_{16}$;

$\Omega_{i+1}^1 = (\Omega_i^1 + C_2 - 1) \bmod (2^{32} - 1) + 1$, где $\boldsymbol{C}_2 = 1010104_{16}$;

Константы, используемые на данном шаге, записаны в 16-ричной системе счисления.

– период повторения последовательности для младшей части составляет 2^{32}, для старшей части 2^{32}–1, для всей последовательности период составляет $2^{32} \cdot (2^{32}–1)$.

Гаммирование с обратной связью

Данный режим очень похож на режим гаммирования и отличается от него только способом выработки элементов гаммы – очередной элемент гаммы вырабатывается как результат преобразования по циклу 32-З предыдущего блока зашифрованных данных, а для зашифрования первого блока массива данных элемент гаммы вырабатывается как результат преобразования по тому же циклу синхропосылки. Этим достигается зацепление блоков – каждый блок шифротекста в этом режиме зависит от соответствующего и всех предыдущих блоков открытого текста.

Режим выработки имитовставки

Для решения задачи обнаружения искажений в зашифрованном массиве данных с заданной вероятностью, в ГОСТе предусмотрен дополнительный режим криптографического преобразования – выработка имитовставки, контрольной комбинации, зависящей от открытых данных и секретной ключевой информации.

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

Имитозащита – это защита системы шифрованной связи от навязывания ложных сообщений, с целью обнаружения всех случайных или преднамеренных изменений в массиве информации.

Схема алгоритма выработки имитовставки в ГОСТ 28147-89 приведена на рис. 5.9. В качестве имитовставки берется часть блока, полученного на выходе, обычно 32 его младших бита.

При выборе размера имитовставки надо принимать во внимание, что вероятность успешного навязывания ложных данных равна величине 2^{-P} на одну попытку подбора. При использовании имитовставки размером 32 бита эта вероятность равна $2^{-32} \approx 0.23 \cdot 10^{-9}$.

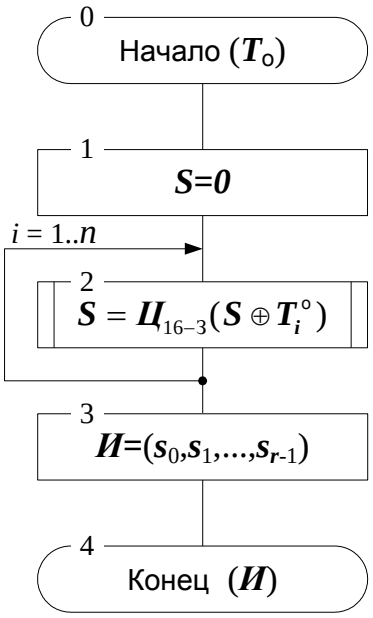


Рис.5.9. Алгоритм выработки имитовставки для массива данных

В заключении рассмотрим вопрос качества ключевой информации и источников ключей.

Ключ должен являться массивом статистически независимых битов, принимающих с равной вероятностью значения 0 и 1. При этом, некоторые конкретные значения ключа могут оказаться «слабыми», то есть шифр может не обеспечивать заданный уровень стойкости в случае их использования. Однако, доля таких значений в общей массе всех возможных ключей ничтожно мала. Поэтому ключи, выработанные с помощью некоторого датчика истинно случайных чисел, будут качественными с вероятностью, отличающейся от единицы на ничтожно малую величину.

Если же ключи вырабатываются с помощью генератора псевдослучайных чисел, то используемый генератор должен обеспечивать указанные выше статистические характеристики, и, кроме того, обладать высокой криптостойкостью, не меньшей, чем у самого ГОСТ 28147-89. Для отбраковки ключей с плохими статистическими характеристиками могут быть использованы различные статистические критерии. На практике обычно хватает двух критериев – для проверки равновероятного распределения битов ключа между значениями 0 и 1 обычно используется критерий «хи квадрат», а для проверки независимости битов ключа – критерий серий.

В аппаратном устройстве криптографической защиты информации «КРИПТОН-4», являющемся аппаратной реализацией ГОСТ 28147-89, для генерирования ключевой информации используется аппаратно реализованный датчик случайных чисел, основанный на шумящем диоде. На рис. 5.10 представлен внешний вид устройства «КРИПТОН-4/PCI».

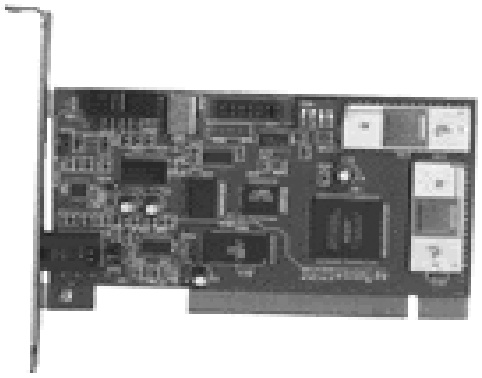


Рис. 5.10. Внешний вид устройства КРИПТОН-4

Таблица замен является долговременным ключевым элементом, то есть действует в течение гораздо более длительного срока, чем отдельный ключ. Предполагается, что она является общей для всех узлов шифрования в рамках одной системы криптографической защиты. Даже при нарушении конфиденциальности таблицы замен стойкость шифра остается чрезвычайно высокой и не снижается ниже допустимого предела.

5.5. Асимметричные криптосистемы

5.5.1. Недостатки симметричных криптосистем и принципы асимметричного шифрования

Наряду с вычислительной простотой и интуитивной понятностью симметричных криптосистем, они обладают рядом серьезных недостатков. К основным недостаткам симметричных криптосистем относят *проблему распространения симметричных ключей* и *проблему их хранения* [13].

При использовании симметричных криптосистем для шифрования информации между пользователями криптографической сети необходимо обеспечить безопасную передачу ключей шифрования между всеми доверенными пользователями (участниками криптографического обмена). При

этом передача ключа шифрования обязательно должна осуществляться по закрытому каналу, так как перехват злоумышленником данного ключа ведет к компрометации всей криптографической сети, и дальнейшее шифрование информации теряет смысл. Однако наличие закрытого канала связи позволяет передавать и сам открытый текст по данному каналу. Таким образом, необходимость шифрования как бы отпадает. Аргументы вида «ключ шифрования необходимо передавать достаточно редко по сравнению с передачей закрытых сообщений» хотя и приемлемы, но оставляют данную проблему нерешенной.

Проблема хранения симметричных ключей шифрования заключается в том, что все участники криптографической сети должны обладать ключом шифрования, то есть иметь к нему доступ. При большом количестве участников криптографического обмена данный факт значительно повышает вероятность компрометации ключей шифрования. В связи с этим, использование симметричных алгоритмов предполагает наличие взаимного доверия сторон. Недобросовестность отношения одного из тысячи участников криптографического обмена к вопросу хранения ключей может привести к утечке ключевой информации, из-за чего пострадают все участники, в том числе и добросовестно относящиеся к своим обязанностям по хранению ключей. Вероятность компрометации ключей тем выше, чем большее количество пользователей входит в криптографическую сеть. Это является большим недостатком симметричных криптосистем.

В отличие от симметричных криптосистем, *асимметричные криптосистемы* используют различные ключи для шифрования и дешифрования сообщений.

Ключи в асимметричных криптосистемах всегда генерируются парами и состоят из двух частей – открытого ключа (ОК) и секретного ключа (СК).

Ключевая пара	
СК	ОК

Открытый ключ используется для шифрования информации, является доступным для всех пользователей и может быть опубликован в общедоступном месте для использования всеми пользователями криптографической сети. Дешифрование информации с помощью открытого ключа невозможно.

Секретный ключ является закрытым и не может быть восстановлен злоумышленником из открытого ключа. Этот ключ используется для дешифрования информации и хранится только у одного пользователя – сгенерировавшего ключевую пару.

Функциональная схема взаимодействия участников асимметричного криптографического обмена представлена на рис. 5.11.

В данной схеме участвует получатель секретного сообщения А и отправитель секретного сообщения В. $ОК_А$ – открытый ключ пользователя А, $СК_А$ – секретный ключ пользователя А. Ключевая пара ($ОК_А$, $СК_А$) сгенерирована на стороне получателя А, после чего открытый ключ данной пары $ОК_А$ отправляется по открытому каналу пользователю В. Предполагается, что злоумышленнику также известен открытый ключ $ОК_А$.

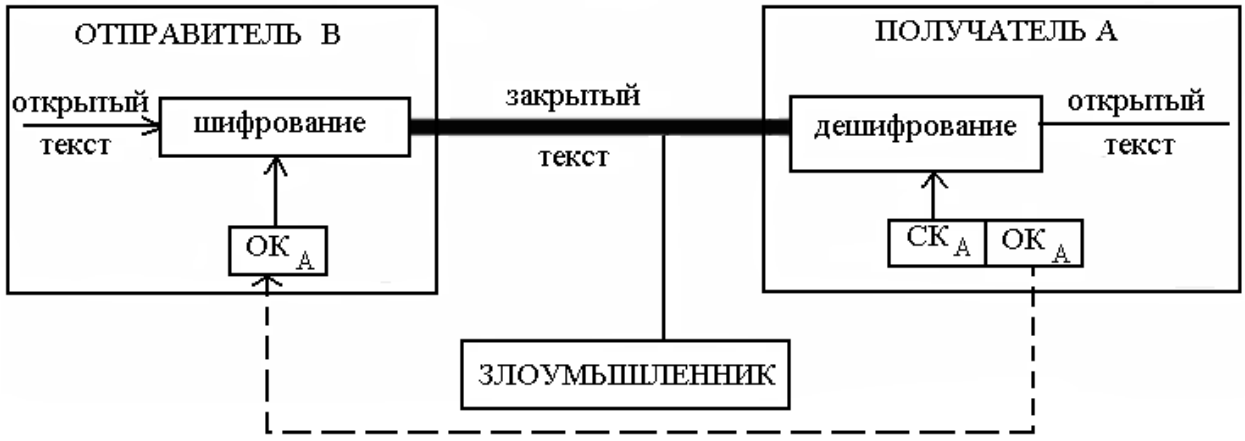


Рис. 5.11. Функциональная схема асимметричной криптосистемы

Отправитель В, зная открытый ключ получателя А, может зашифровать на данном ключе открытый текст и переслать его пользователю А. Пользователь А с помощью своего секретного ключа, соответствующего $ОК_А$, может дешифровать присланное пользователем В сообщение. Злоумышленник, зная $ОК_А$ и закрытый текст, не может получить доступ не к $СК_А$, не к открытому тексту.

Рис 5.11 отражает только одностороннюю схему взаимодействия в рамках асимметричных криптосистем. Для реализации двустороннего обмена необходима реализация следующих шагов:

1. Пользователь А генерирует ключевую пару ($ОК_A$,$СК_A$).

2. Пользователь В генерирует ключевую пару ($ОК_B$,$СК_B$).

3. Пользователи А и В должны обменяться своими открытыми ключами. Пользователь А передает свой открытый ключ $ОК_A$ пользователю В, пользователь В передает свой открытый ключ $ОК_B$ пользователю А.

4. Пользователь А шифрует информацию для пользователя В на ключе $ОК_B$, пользователь В шифрует информацию для пользователя А на ключе $ОК_A$.

5. Пользователь А дешифрует информацию, присланную ему от пользователя В, на ключе $СК_A$, пользователь В дешифрует информацию, присланную ему от пользователя А, на ключе $СК_B$.

Обмен открытыми ключами в современных криптографических сетях, насчитывающих десятки и даже сотни тысяч пользователей более удобно реализовывать, используя специально выделенные для этого *центры распределения ключей*. Пользователь А может выложить на центр распределения ключей свой открытый ключ и любой другой пользователь, желающий шифровать информацию для А, может обратиться в данный центр и забрать его открытый ключ. Схема распределения ключей в данном случае может выглядеть следующим образом (рис. 5.12).

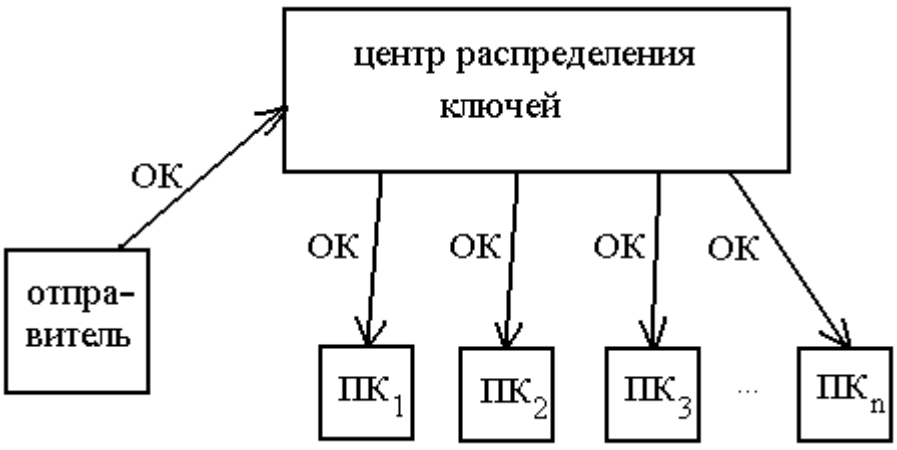


Рис. 5.12. Схема распределения ОК с использованием центра распределения ключей

В настоящее время все более распространенным подходом к распределению ключей становится подход, основанный на реализации инфраструктуры открытых ключей PKI и удостоверяющих центров (УЦ).

У. Диффи и М. Хеллман сформулировали требования, выполнение которых обеспечивает безопасность асимметричной криптосистемы [30]:

1. Вычисление ключевой пары (ОК, СК) должно быть достаточно простым.

2. Отправитель, зная открытый ключ получателя, может легко получить шифротекст.

3. Получатель, используя свой секретный ключ, может легко из шифротекста восстановить исходное сообщение.

4. Знание открытого ключа злоумышленником не должно влиять на криптостойкость системы. При попытке вычислить злоумышленником закрытый ключ по открытому, он должен наталкиваться на непреодолимую вычислительную проблему.

5. Злоумышленник, зная шифротекст и открытый ключ, на котором осуществлялось шифрование, при попытке восстановить исходный текст должен наталкиваться на непреодолимую вычислительную проблему.

5.5.2. Однонаправленные функции

Реализация асимметричных криптосистем основана на использовании однонаправленных функций.

Пусть X и Y – некоторые произвольные множества. Функция $f: X \to Y$ называется *однонаправленной функцией*, если для любого элемента $x \in X$ можно легко вычислить его образ $y = f(x)$, однако, зная элемент $y \in Y$, достаточно сложно получить его прообраз $x = f^{-1}(y)$, хотя такой элемент x однозначно существует хотя бы один.

Одним из основных критериев, по которому функцию f можно считать однонаправленной, является отсутствие эффективных алгоритмов обратного

преобразования $Y \to X$, что не позволяет обратить данную функцию за приемлемое время.

Рассмотрим несколько примеров однонаправленных функций, имеющих большое значение для криптографии.

Целочисленное умножение

Вычисление произведения двух очень больших целых чисел P и Q ($N=P*Q$) является несложной задачей для ЭВМ. Однако, решение обратной задачи, заключающейся в нахождении делителей P и Q большого числа N (в особенности, когда P и Q – большие простые числа), является практически неразрешимой задачей при больших N. Если $N \approx 2^{664}$ и $P \approx Q$, то задача факторизации не разрешима за приемлемое время на современных ЭВМ. Поэтому целочисленное умножение является однонаправленной функцией.

Модульная экспонента

Возведение очень большого числа A в очень большую степень x по любому модулю M ($0 \le A, x < M$), то есть вычисление $y = A^x \pmod{M}$ является несложной задачей для ЭВМ. Однако решение обратной задачи – нахождения степени x по известным y,A,M такой, что $A^x \pmod{M} = y$ (задача дискретного логарифмирования, $x = \log_A y$), практически не разрешима за приемлемое время на современных ЭВМ (эффективного алгоритма вычисления дискретного логарифма пока не найдено). Поэтому модульная экспонента является однонаправленной функцией.

Кроме однонаправленных функций важное значение для криптографии с открытым ключом имеют *однонаправленные функции с «потайным входом»*, эффективное обращение которых возможно, если известен секретный «потайной ход» (секретное число или другая информация, ассоциируемая с функцией).

5.5.3. Алгоритм шифрования RSA

Алгоритм RSA был предложен в 1978 году Р.Райвестом, А. Шамиром, А. Адлеманом и был назван по первым буквам фамилий его авторов. Данный алгоритм стал первым алгоритмом шифрования с открытым ключом. Надежность данного алгоритма основывается на трудности факторизации больших чисел и вычисления дискретных логарифмов [8,13].

В криптосистеме RSA открытый ключ ОК, секретный ключ СК, исходное сообщение M и шифротекст C являются целыми числами от 0 до N-1, где N – модуль.

Пусть пользователь А является получателем сообщения, которое ему должен переслать отправитель В.

Пользователь А должен вначале сгенерировать ключевую пару RSA, это он делает следующим образом.

Алгоритм формирования ключевой пары пользователем А

1. Выбираем случайные большие простые числа P и Q. Для обеспечения максимальной безопасности P и Q выбирают примерно равной длины и хранят в секрете.

2. Вычисляем модуль $N = P \cdot Q$. Согласно формуле (4.1) $\varphi(N) = (P-1)\cdot(Q-1)$, где $\varphi(N)$ - функция Эйлера.

3. Открытый ключ $ОК_A$ выбирается случайно таким образом, чтобы выполнялись следующие условия:

$$1<ОК_A<\varphi(N),\ НОД(ОК_A,\ \varphi(N))=1 \qquad (5.11)$$

4. Секретный ключ $СК_A$ находится по сформированному открытому ключу так, что

$$СК_A \cdot ОК_A \equiv 1 \pmod{\varphi(N)}$$

или (5.12)

$$СК_A = ОК_A^{-1} \pmod{(P-1)\cdot(Q-1)}$$

Пользователь А может легко найти $СК_A$, используя расширенный алгоритм Евклида, зная числа P и Q, а значит и $\varphi(N)$.

Любой другой пользователь не может, зная открытый ключ $ОК_A$ вычислить $СК_A$, так как ему не известны числа P и Q. Для их нахождения ему потребуется факторизовать известное ему число N, что является вычислительно сложной задачей.

Шифрование и дешифрование сообщений в криптосистеме RSA

Для того, чтобы зашифровать открытое сообщение *M*, отправитель B должен возвести его в степень открытого ключа пользователя A по модулю *N*. То есть шифрование выполняется в соответствие с формулой (5.13).

$$C = M^{OK_A} \pmod{N} \qquad (5.13)$$

Обращение данной функции, то есть определение значения *M* по известным значениям *C*, $ОК_A$, *N* практически не осуществимо при больших *N* ($N \approx 2^{512}$).

Однако знание секретного ключа $СК_A$ позволяет обратить данную функцию, то есть решить задачу дешифровки криптограммы *C*. Для дешифровки криптограммы *C* необходимо возвести ее в степень секретного ключа пользователя A по модулю *N*. Таким образом, дешифрование сообщения выполняется в соответствие с формулой (5.14).

$$M = C^{CK_A} \pmod{N} \qquad (5.14)$$

Действительно,

$$C^{CK_A} \pmod{N} = (M^{OK_A})^{CK_A} \pmod{N} = M^{OK_A \cdot CK_A} \pmod{N}$$

В теории чисел известна теорема Эйлера, утверждающая, что если НОД(*x*,*N*)=1, то $x^{\varphi(N)} \equiv 1 (\text{mod } N)$.

Согласно 5.12, $СК_A \cdot ОК_A \equiv 1 \pmod{\varphi(N)}$, то есть $СК_A \cdot ОК_A = k \cdot \varphi(N) + 1$. Таким образом,

$$M^{OK_A \cdot CK_A} = M^{k \cdot \varphi(N)} \cdot M \pmod{N} = M \pmod{N}$$

Таким образом, показано, что $C^{CK_A} \pmod{N} = M \pmod{N}$.

Получатель A, который создает ключевую пару ($ОК_A$,$СК_A$) защищает два параметра: 1) секретный ключ $СК_A$; 2) пару чисел *P* и *Q*. Рассекречива-

ние данных чисел приводит к тому, что злоумышленник сможет вычислить $\varphi(N)$, а значит и вычислить секретный ключ $СК_А$ согласно (5.12).

Открытыми в криптосистеме RSA являются только значения $ОК_А$ и N.

В настоящее время разработчики криптоалгоритмов с открытым ключом на базе RSA предлагают применять в качестве чисел P,Q,N – числа длиной не менее 200-300 десятичных разрядов.

Пример 5.11

Зашифруем сообщение DAC по алгоритму RSA. Для простоты вычислений будем оперировать с небольшими числами P и Q.

Действия получателя А

1. Выберем $P = 5$ и $Q = 13$

2. Модуль $N = P \cdot Q = 5 \cdot 13 = 65$

3. $\varphi(N) = \varphi(65) = (5-1) \cdot (13-1) = 4 \cdot 12 = 48$

4. В качестве OK_A необходимо выбрать значение, удовлетворяющее условиям $1 < OK_A < 48$, $НОД(OK_A, 48) = 1$. Пусть $OK_A = 5$.

5. Необходимо найти CK_A, такой что $CK_A \cdot OK_A = CK_A \cdot 5 \equiv 1 \pmod{48}$. Это CK_A=29. Действительно, $29 \cdot 5 = 145 \equiv 1 \pmod{48}$.

6. Отправляем пользователю В пару чисел (N=65, OK_A=5)

Действия отправителя В

1. Представим отправляемое сообщение в виде последовательности целых чисел от 0 до 63. Присвоим букве А номер 1, букве В – 2, С – 3, D – 4 и т.д. Тогда открытый текст DAC запишется в виде последовательности чисел 413, то есть M_1=4, M_2=1, M_3=3.

2. Сформируем шифротекст по формуле 5.13:

$C_1 = M_1^{OK_A} \pmod{N} = 4^5 \pmod{65} = 1024 \pmod{65} = 49$,

$C_2 = 1^5 \pmod{65} = 1$,

$C_3 = 3^5 \pmod{65} = 243 \pmod{65} = 48$.

3. Пользователь В отправляет А криптограмму C_1, C_2, C_3=49, 1, 48.

Действия пользователя А

1. Раскрываем шифротекст по формуле 5.14:

$M_1 = C_1^{CK_A} \pmod{N} = 49^{29} \pmod{65} = 4,$

$M_2 = 1^{29} \pmod{65} = 1,$

$M_3 = 48^{29} \pmod{65} = 3.$

Таким образом, восстановлено исходное сообщение M_1=4=D, M_2=1=A, M_3=3=C. Исходное сообщение – DAC.

5.6. Вопросы для самоконтроля

1. Что понимают под криптографией?
2. Дайте определение ключа шифрования.
3. Что понимают под криптоанализом?
4. Приведите примеры криптоаналитических атак. Кратко охарактеризуйте их.
5. Какие требования предъявляются к стойким шифрам, используемым для криптографической защиты информации?
6. Сформулируйте закон Керхоффа.
7. Охарактеризуйте подход к криптографической защите, используемый в симметричных криптосистемах.
8. Перечислите недостатки симметричных криптосистем.
9. Охарактеризуйте шифры замены.
10. В чем отличие методов моноалфавитной замены от методов многоалфавитной замены? Приведите примеры шифров каждого из этих классов.
11. Опишите подход к шифрованию, используемый в шифре Цезаря.
12. В чем заключается разница между шифром Цезаря и простой моноалфавитной заменой?
13. В чем заключаются сходство и различие шифров Цезаря, Гронсфельда и Вижинера. Попарно сравните данные шифры.
14. Опишите подход к криптографической защите, используемый в шифре Вернама? В чем его недостатки?

15. В чем заключается шифрование методами перестановки?

16. Опишите подход к шифрованию методами перестановки, основанный на маршрутах Гамильтона.

17. В чем заключается подход к шифрованию методом гаммирования?

18. Дайте определение гаммы шифра.

19. Что является ключом шифрования в шифрах гаммирования?

20. Опишите линейный конгруэнтный метод формирования псевдослучайных последовательностей.

21. Как выполняется криптоанализ, основанный на исследовании частотности символов в тексте?

22. Приведите примеры симметричных алгоритмов шифрования.

23. Опишите схему шифрования информации в алгоритме DES. Какие основные этапы включает данный алгоритм?

24. Каков размер ключа шифрования алгоритма DES?

25. Охарактеризуйте основные режимы шифрования алгоритма DES.

26. Перечислите режимы работы Российского стандарта симметричного шифрования ГОСТ 28147-89.

27. Что понимают под таблицей замен и узлом замены в ГОСТ 28147-89?

28. В чем заключается отличие асимметричных криптосистем от симметричных?

29. Что собой представляет ключевая пара?

30. Охарактеризуйте подход к криптографической защите, используемый в асимметричных криптосистемах.

31. На каком из ключей происходит шифрование информации в асимметричных криптосистемах, а на каком дешифрование?

32. Перечислите требования Диффи-Хеллмана к реализации асимметричных криптосистем.

33. Дайте определение однонаправленной функции и однонаправленной функции с «потайным входом».

34. Приведите примеры однонаправленных функций.

35. Приведите примеры асимметричных алгоритмов шифрования.

36. Опишите схему формирования открытого и закрытого ключей в алгоритме шифрования RSA.

37. Опишите схему шифрования и дешифрования информации в алгоритме RSA.

38. Приведите несколько примеров ключевых пар RSA.

6. Контроль целостности информации. Электронно-цифровая подпись

6.1. Проблема обеспечения целостности информации

В настоящее время повсеместное внедрение информационных технологий отразилось и на технологии документооборота внутри организаций и между ними, между отдельными пользователями. Все большее значение в данной сфере приобретает электронный документооборот, позволяющий отказаться от бумажных носителей (или снизить их долю в общем потоке) и осуществлять обмен документами между субъектами в электронном виде. Преимущества данного подхода очевидны: снижение затрат на обработку и хранение документов, более быстрый их поиск и т.д. В эпоху «информационного бума» данный подход является единственным выходом из затруднительного положения, связанным с экспоненциальным ростом объемов обрабатываемой информации.

Однако переход от бумажного документооборота к электронному ставит ряд проблем, связанных *с обеспечением целостности* (подлинности) передаваемого документа и *аутентификации подлинности его автора*.

Как для отправителя, так и для получателя электронного сообщения необходима гарантия того, что данное сообщение не было изменено в процессе его передачи. Необходима реализация технологии документооборота, затрудняющей злоумышленнику вносить преднамеренные искажения в передаваемый документ. Если же искажения в документ были внесены, то его

получатель должен иметь возможность с вероятностью близкой к 100% распознать этот факт.

Проблема аутентификации подлинности автора сообщения заключается в том, чтобы обеспечить гарантию того, что никакой субъект не сможет подписаться под сообщением ни чьим другим именем, кроме своего. Если же он подписался чужим именем, то опять же получатель должен иметь возможность с вероятностью близкой к 100% распознать этот факт.

В обычном, бумажном документообороте, эти проблемы решаются за счет того, что информация в документе и рукописная подпись автора жестко связаны с физическим носителем (бумагой). Элементами, обеспечивающими целостность передаваемых сообщений и подлинность авторства, в этом случае являются: рукописные подписи, печати, водяные знаки на бумаге, голограммы и т.д. Для электронного же документооборота жесткая связь информации с физическим носителем отсутствует, в связи с чем, требуется разработка иных подходов для решения перечисленных выше проблем.

Приведем несколько практических примеров, связанных с необходимостью обеспечения целостности и подлинности авторства электронных документов.

Пример 6.1. Подача налоговой и бухгалтерской отчетности в электронном виде по телекоммуникационным каналам.

Пример 6.2. Передача распоряжений, указов руководством компании своим отделениям по электронной почте.

В данном случае, у получателя и отправителя должна быть гарантия того, что отправленное сообщение не осело, например, где-либо на почтовом сервере, где его мог изменить другой пользователь и отправить по назначению далее, исходное письмо в этом случае до адресата не доходит.

В отдельных случаях, при пересылке электронных документов по открытым каналам связи сама информация может быть и открыта, однако любое незначительное ее изменение может привести к катастрофическим последствиям.

Рассмотрим возможности злоумышленника при реализации угроз, направленных на нарушение целостности передаваемых сообщений и подлинность их авторства [13].

1. Активный перехват. Нарушитель, имеющий доступ к каналу связи перехватывает передаваемые сообщения и изменяет их.

2. Маскарад. Нарушитель посылает документ абоненту В, подписавшись именем абонента А.

3. Ренегатство. Абонент А заявляет, что не посылал сообщения абоненту В, хотя на самом деле посылал. В этом случае, абонент А является злоумышленником и использует теоретическую возможность маскарада для того, чтобы «облагородить» себя.

4. Подмена. Абонент В изменяет или формирует новый документ и заявляет, что получил его от абонента А. В этом случае, в качестве недобросовестного пользователя выступает получатель сообщения.

5. Повтор. Злоумышленник повторяет ранее переданный документ, который абонент А посылал абоненту В.

Следует отметить, что известные в теории информации методы защиты сообщений, передаваемых по каналам связи, от случайных помех не работают в том случае, когда злоумышленник преднамеренно реализует угрозу нарушения целостности информации. Например, контрольные суммы, используемые для этой цели передатчиком и приемником, могут быть пересчитаны злоумышленником так, что приемником изменение сообщения не будет обнаружено. Таким образом, контрольные суммы могут быть скомпрометированы злоумышленником, они не защищают от активных изменений. Для обеспечения целостности электронных документов и установления подлинности авторства необходимо использовать иные методы, отличные от контрольных сумм. Для решения данных задач используют технологию электронно-цифровой подписи.

6.2. Функции хэширования и электронно-цифровая подпись

Электронно-цифровая подпись (ЭЦП) сообщения является уникальной последовательностью, связываемой с сообщением, подлежащей проверке на принимающей стороне с целью обеспечения целостности передаваемого сообщения и подтверждения его авторства.

Электронно-цифровая подпись (ЭЦП) используется для аутентификации текстов, передаваемых по открытым каналам связи. Ее использование позволяет гарантировать выполнение следующих условий.

1. Лицо или процесс, идентифицируемый как отправитель электронного документа, действительно является инициатором отправления.
2. Целостность передаваемой информации не нарушена.
3. Не дает отказаться лицу, идентифицируемого как отправителя электронного документа, от обязательств, связанных с подписанным текстом.

ЭЦП представляет собой относительно небольшое количество цифровой информации, дополняющей электронный документ и передаваемой вместе с ним.

Использование ЭЦП предполагает введение асимметричной системы шифрования и, следовательно, ключевой пары (ОК,СК), а также двух процедур: 1. Процедуру установки ЭЦП (подписывание документа); и 2. процедуру проверки ЭЦП (аутентификация документа).

Процедура установки ЭЦП использует секретный ключ отправителя сообщения, а процедура проверки ЭЦП – открытый ключ отправителя сообщения (рис. 6.1). Здесь *M* – электронный документ, *E* – электронно-цифровая подпись.

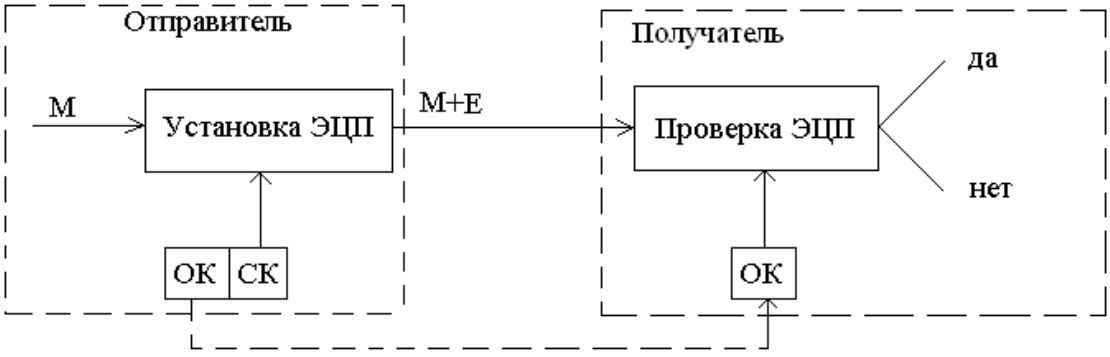


Рис. 6.1. Схема использования ЭЦП

В технологии ЭЦП ведущее значение имеют однонаправленные функции хэширования. Использование *функций хэширования* позволяет формировать криптографически стойкие контрольные суммы передаваемых сообщений.

Функцией хэширования H называют функцию, сжимающую сообщение произвольной длины *M*, в значение фиксированной длины *H(M)* (несколько десятков или сотен бит), и обладающую свойствами необратимости, рассеивания и чувствительности к изменениям. Значение *H(M)* обычно называют *дайджестом* сообщения *M*.

Свойство *необратимости* подразумевает вычислительную трудоемкость создания документа *M* с заданным хэш-образом *H(M)*. Этот хэш-образ сложным образом зависит от документа *M* и не позволяет его восстановить.

Свойство *рассеивания* подразумевает то, что вероятность совпадения значений хешей двух различных документов M_1 и M_2 должна быть чрезмерно мала.

Свойство *чувствительности к изменениям* подразумевает то, что хэш-функция должна быть очень чувствительна к всевозможным изменениям в документе *M*, таким, как вставки, выбросы, перестановки и т.д.

Наиболее известными алгоритмами хэширования являются MD4, MD5, SHA.

Электронно-цифровая подпись формируется как результат шифрования дайджеста сообщения с помощью секретного ключа, ставящего подпись. Схемы процедур установки и проверки ЭЦП представлены на рис. 6.2.

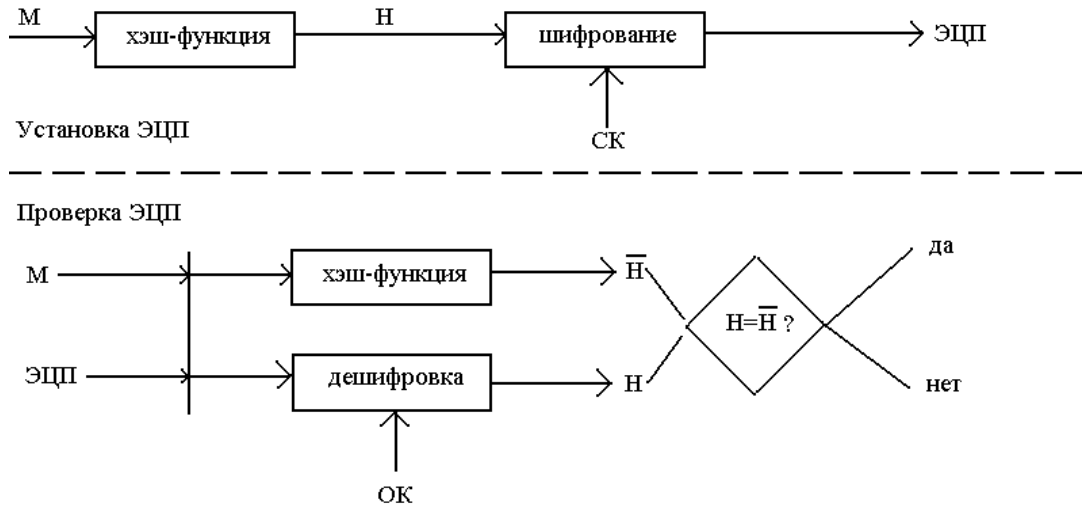


Рис. 6.2. Схема процедур установки и проверки ЭЦП

Таким образом, схемы установки и проверки ЭЦП выглядят следующим образом.

Схема установки ЭЦП

1. Для документа *M* формируется дайджест *H* с помощью заданного алгоритма хэширования.

2. Сформированный дайджест *H* шифруют на секретном ключе отправителя сообщения. Полученная в результате шифрования последовательность и есть ЭЦП.

3. Сообщение *M* и его ЭЦП передаются получателю сообщения.

Схема проверки ЭЦП

1. Получатель для проверки ЭЦП должен иметь доступ к самому сообщению *M* и его ЭЦП.

2. Зная алгоритм хэширования, который был использован при установке ЭЦП, получатель получает хэш $\overline{H}$ присланного сообщения *M*.

3. Зная открытый ключ отправителя, получатель дешифрует ЭЦП, в результате чего получает хэш *H*, сформированный на этапе установки ЭЦП.

4. Критерием целостности присланного сообщения *M* и подтверждения его автора является совпадение хэшей *H* и $\overline{H}$. Если это равенство не выполнено, то принимается решение о некорректности ЭЦП со всеми вытекающими отсюда последствиями.

Целостность передаваемого сообщения гарантируется свойствами функции хэширования. Подлинность авторства сообщения гарантируется используемой технологией асимметричного шифрования. Злоумышленник не сможет подписаться другим пользователем, так как не имеет доступа к его секретному ключу, владелец же секретного ключа может ставить подпись на данном ключе.

Следует отметить, что использование секретного ключа на этапе установки ЭЦП защищает сообщение от активных изменений. Злоумышленник уже не способен скомпрометировать контрольную сумму, в качестве которой здесь выступает дайджест сообщения.

Наиболее известными алгоритмами ЭЦП являются RSA, Эль-Гамаля, DSA. Отечественным стандартом ЭЦП является ГОСТ 34.10-94 [11].

6.3. Инфраструктура открытых ключей PKI

Вступивший в силу с 22 января 2002 года Федеральный закон «Об электронно-цифровой подписи (ЭЦП)» явился базовым законом, в рамках которого возможна организация защищенного документооборота на федеральном уровне. При этом ЭЦП стала иметь доказательную силу при возникновении конфликтных ситуаций.

Одной из наиболее актуальных задач при реализации защищенного документооборота, в том числе и положений, устанавливаемых Федеральным законом об ЭЦП, является реализация сервиса безопасности, отвечающего за распределение криптографических ключей. Реализация угроз, нарушающих безопасное функционирование данного сервиса, может иметь катастрофическое значение для безопасности электронного документооборота. Наиболее безопасным способом реализации данного сервиса является способ, основанный на управлении открытыми ключами третьей стороной. Систематическим, расширяемым, унифицированным и легко управляемым подходом к распределению открытых ключей стало введение *сертификатов открытых ключей*. Технология PKI (*инфраструктура открытых ключей*) является продуманной инфраструктурой безопасности, предназначенной для распро-

странения ОК, управления цифровыми сертификатами и ключами пользователей.

Использование инфраструктуры открытых ключей позволяет обеспечить выполнение следующих условий [12].

1. Лицо или процесс, идентифицируемый как отправитель электронного документа, действительно является инициатором отправления.

2. Лицо или процесс, выступающий получателем электронного документа, действительно является тем получателем, которого имел в виду отправитель.

3. Целостность и конфиденциальность передаваемой информации не нарушена.

Реализация PKI связана с решением ряда проблем. Приведем некоторые из них.

1. Инструментальные системы поддержки инфраструктуры открытых ключей должны отвечать требованиям международных и Российских стандартов. Достижение этого возможно только при использовании специальных сертифицированных программно-аппаратных компьютерных систем.

2. Распространение и хранение ключей должно производиться в юридически точно (де-юре) определенной системе на базе международных криптографических стандартов.

3. Администраторы и пользователи электронного документооборота с ЭЦП должны пройти обучение и получить соответствующие права и сертификаты.

Структура, сервисы и архитектура PKI

Основной информационной единицей, используемой при распространении ОК, является его цифровой сертификат.

Под *цифровым сертификатом* понимается цифровой документ, подтверждающий соответствие открытого ключа информации, идентифицирующей владельца ключа [12].

Цифровой сертификат позволяет защитить открытый ключ от его подделки злоумышленником. Он содержит подписанную информацию о владельце ОК, сведения об ОК, его назначении, области применения и т.д.

В настоящее время количество приложений, использующих криптографические функции с открытым ключом, все больше возрастает. Вместе с этим возрастает и количество разнородных сертификатов. Задачу единообразной организации сервиса управления сертификатами и решает инфраструктура открытых ключей.

PKI представляет собой комплексную систему, обеспечивающую все необходимые сервисы для использования цифровых сертификатов, нацеленную на поддержку надежного, доверенного взаимодействия между пользователями. PKI позволяет реализовывать сервисы шифрования и выработки ЭЦП согласованно с широким кругом приложений, функционирующих в среде ОК.

Основными компонентами технологии PKI являются следующие [5].

1. Удостоверяющий центр.
2. Регистрационный центр.
3. Реестр сертификатов.
4. Архив сертификатов.
5. Конечные субъекты.

Основная функция *удостоверяющего центра* (*УЦ*) - заверение цифрового сертификата ОК субъекта своей подписью, поставленной на своем секретном ключе. УЦ является как бы нотариальной конторой, подтверждающей подлинность сторон, участвующих в обмене информацией. Любой субъект может верифицировать сертификат партнера, проверив подпись УЦ под его сертификатом. Это гарантирует то, что злоумышленник не сможет выдать себя за отправителя подписанных данных, заменив значение ОК своим.

Другими функциями УЦ являются:

1. формирование собственного СК и самоподписанного сертификата;

2. выпуск (создание и подписывание) сертификатов подчиненных УЦ;

3. ведение базы всех изданных сертификатов и формирование списка аннулированных сертификатов.

Регистрационный центр является необязательным компонентом PKI. Он может брать на себя часть функций УЦ – регистрацию пользователей, обеспечение их взаимодействия с УЦ, и сбор и передачу УЦ информации от заявителя, вносимой в сертификат.

Реестр сертификатов – специальный объект PKI, представляющий собой БД, хранящей сертификаты и списки аннулированных сертификатов.

Большинство серверов каталогов сертификатов и прикладное ПО субъектов доступа поддерживают протокол LDAP облегченного доступа к каталогам сертификатов.

Архив сертификатов выполняет функцию долговременного хранения информации обо всех изданных сертификатах.

Конечные субъекты – пользователи PKI, делящиеся на две категории: владельцев сертификатов и доверяющие стороны. Владельцами сертификата может быть доверенное физическое или юридическое лицо, приложение, сервер и т.д.

Реализация технологии PKI требует поддержки следующих сервисов:

1. *криптографические сервисы*, включающие в себя сервисы генерации пар ключей, выработки и верификации ЭЦП.

2. *сервисы управления сертификатами*, включающие в себя сервисы выпуска сертификатов, управления жизненным циклом сертификата и ключей, поддержки реестра сертификатов, хранения действительных и отозванных сертификатов в архиве.

3. сервисы регистрации, нотариальной аутентификации, создания резервных копий и восстановления ключей, неотказуемости, авторизации, корректировки ключей и управления историями ключей.

Существуют несколько архитектур построения PKI. В настоящее время наиболее распространенной архитектурой PKI является *гибридная* архитек-

тура, включающая в себя достоинства иерархической и сетевой архитектур (рис. 6.3).

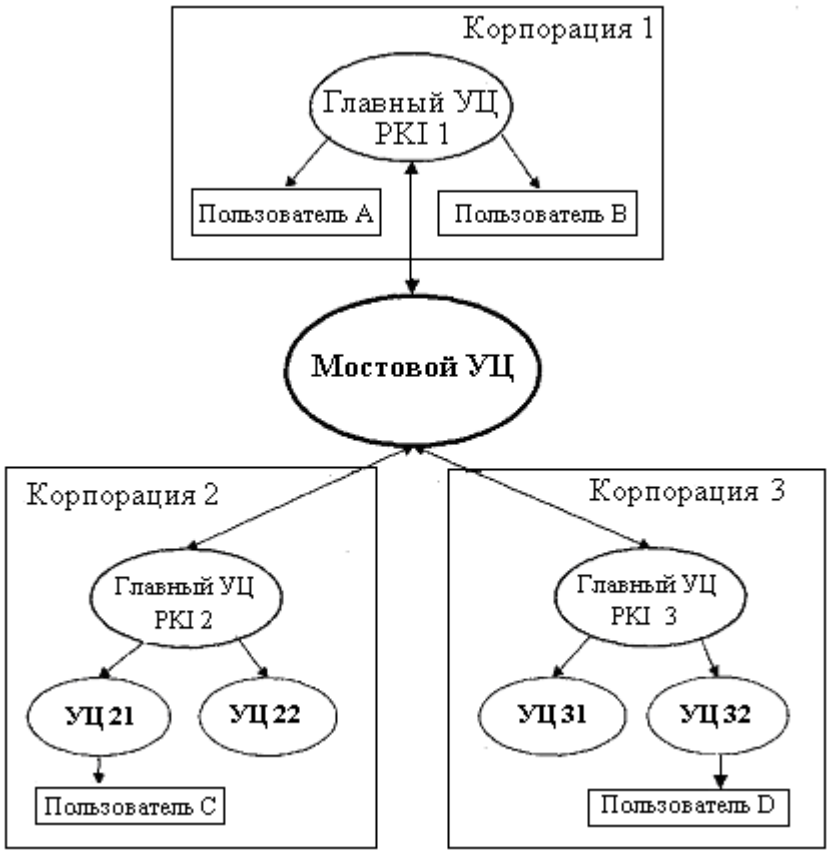


Рис. 6.3. Гибридная архитектура PKI

Внутри корпораций 1,2,3 архитектура УЦ является иерархической, управляемой главным УЦ соответствующей корпорации. Главный УЦ выпускает самоподписанный сертификат и сертификаты для подчиненных УЦ. Подчиненные УЦ могут выпускать сертификаты для УЦ нижних уровней, или для пользователей. Соединение корпоративных PKI независимо от их архитектуры достигается введением нового - *мостового* УЦ, единственным назначением которого является установление связей между ними.

Система PKI должна взаимодействовать с множеством различных приложений: программное обеспечение групповой работы, электронной почты, сетей VPN и т.д. Наиболее общая функциональная схема взаимодействия компонентов PKI представлена на рис. 6.4.

Наиболее часто используемым подходом к реализации PKI является подход, основанный на сертификатах формата X.509.

Формат сертификата открытого ключа X.509.V3 определен в документе RFC 3280 Certificate & CRL Profile. Он представляет собой структурированную двоичную запись, содержащую ряд полей с элементами данных, сопровождаемые цифровой подписью издателя сертификата. Структура сертификата X.509.V3 представлена в таблице 6.1.

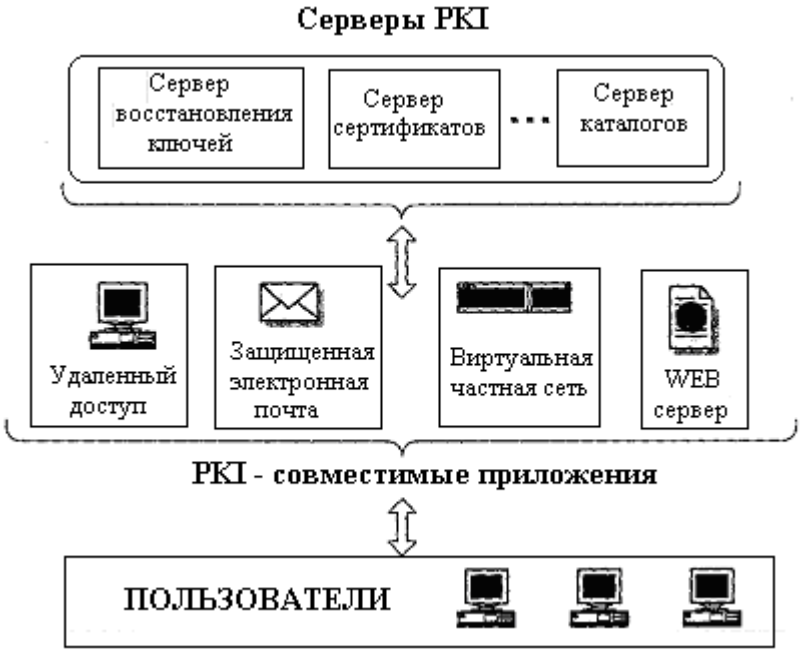


Рис. 6.4. Взаимодействие компонентов PKI

Табл. 6.1. Структура сертификата X.509.V3

Номер поля	Имя поля
1	Номер версии сертификата (в рассматриваемом случае – 3)
2	Уникальный серийный номер сертификата
3	Идентификатор алгоритма ЭЦП, используемого для защиты сертификата от подделки
4	Имя издателя, выпустившего данный сертификат
5	Период действия сертификата (дата начала/дата конца действия)
6	Имя владельца секретного ключа, соответствующего ОК
7	Открытый ключ субъекта
8	Уникальный идентификатор издателя
9	Уникальный идентификатор субъекта
10	Расширения – содержат дополнительную информацию, определяющую наличие у владельца сертификата прав доступа к той или иной системе и др.
11	ЭЦП сертификата

Каждый раз, при использовании сертификата необходимо верифицировать его подпись, а также то, что сертификат является действующим. Серти-

фикаты, срок действия которых истек, должны аннулироваться УЦ. Сертификат может также аннулироваться до истечения срока своего действия, например, при компрометации секретного ключа, увольнении служащего организации и т.д.

Политика и регламент PKI

Фундаментом инфраструктуры открытых ключей является «практика доверия», поэтому при развертывании данной инфраструктуры компании должны придти к общему соглашению о том, что она из себя представляет. Компании должны создать собственные уникальные соглашения о доверии, регулирующие обязательства и ответственности каждой из сторон, то есть создать *политику инфраструктуры открытых ключей*.

Политика большинства систем PKI направлена на решение технических, административных, юридических и кадровых проблем. К основным требованиям к политике PKI относятся: ее соответствие общей корпоративной политике безопасности, четкость и однозначность формулировок, доступность изложения, разграничение ответственности между субъектами PKI, адекватность ограничений и пределов ответственности требованиям сферы применения сертификатов.

Основным стандартом, регламентирующим разработку политики PKI, является RFC 2527 «Certificate Policy and Certification Practices Framework». Согласно данному стандарту основными документами, описывающими политику PKI, являются *документ о политике применения сертификатов* и *регламент*.

Под *политикой применения сертификатов* понимается установленный набор правил, характеризующих возможность применения сертификата определенным сообществом и/или классом приложений с определенными требованиями безопасности. Данная политика позволяет доверяющей стороне оценить надежность использования сертификата для определенного приложения.

Регламент удостоверяющего центра есть документ, в котором в четкой, детальной форме изложена та система и практика, которых придерживается УЦ при работе с сертификатами. В нем точно формулируются обязанности УЦ перед доверяющей стороной, а также все то, что потенциально необходимо для понимания и принятия во внимание доверяющими сторонами.

Политика и регламент применения сертификатов взаимно дополняют друг друга, формат их публикации устанавливается стандартом RFC 2527 - «Политика применения сертификатов и структура регламента». Перечень разделов, рекомендуемых к включению в описание политики PKI согласно данному стандарту, представлен на рис. 6.5.

Программные средства поддержки PKI

Процесс развертывания PKI осуществляется на выбранных программных и программно-аппаратных средствах. Наиболее известными продуктами, на базе которых разворачивается инфраструктура открытых ключей, являются:

1. Entrust/PKI фирмы Entrust Technologies.
2. Baltimore UniCERT фирмы Baltimore Technologies LTD.
3. BT TrustWise Onsite фирмы VeriSign Inc.
4. IBM Trust Authority.
5. RSA Keon Certification Authority фирмы RSA Security Inc.
6. VCERT PKI компании ЗАО «МО ПНИЭИ».
7. Семейство продуктов «КриптоПро».

Для российских условий наиболее адаптированным и полнофункциональным продуктом, на базе которого можно развернуть инфраструктуру открытых ключей, является «КриптоПро».

Рис. 6.5. Перечень разделов политики PKI

Программный комплекс «Удостоверяющий центр» – «КриптоПро УЦ» позволяет в полном объеме реализовать инфраструктуру открытых ключей. В состав КриптоПро УЦ входят следующие компоненты:

1. Центр сертификации, функционирующий на платформе Windows 200 Server.

2. Центр регистрации, функционирующий на платформе Windows 2000 Server и использующий для решения своих задач базу данных Microsoft SQL 2000, Microsoft IIS 5.0, CRYPTO API 2.0.

3. АРМ администратора ЦР, функционирующий в ОС Windows 2000 Professional в рамках Microsoft Management Console и предназначенный для выполнения организационно-технических мероприятий, связанных с регистрацией пользователей, генерацией ключей и сертификатов.

4. АРМ пользователя, представляющий собой web-приложение, размещенное на сервере ЦР. Функционирует в ОС Windows 95 и выше. Данный АРМ обеспечивает шифрование информации, передаваемой ЦР с использованием протокола TLS с двусторонней аутентификацией.

5. Программный интерфейс взаимодействия с УЦ.

Архитектура УЦ КриптоПро представлена на рис. 6.6.

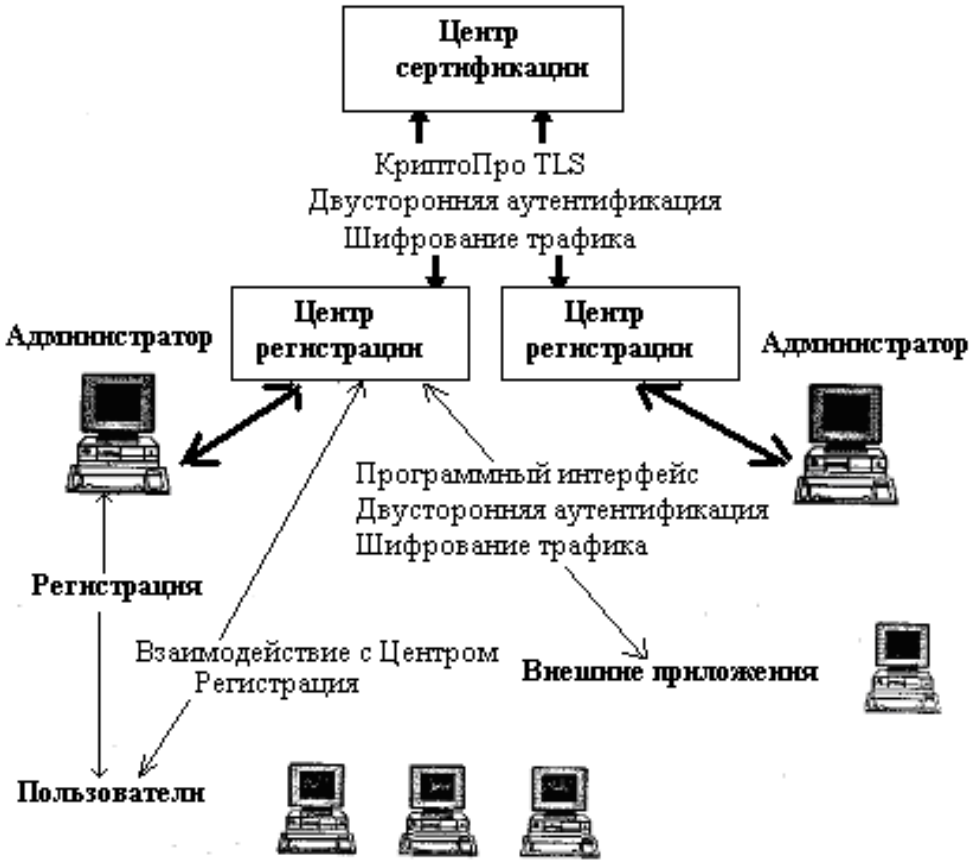


Рис. 6.6. Архитектура удостоверяющего центра КриптоПро

6.4. Вопросы для самоконтроля

1. В чем заключается проблема обеспечения целостности и аутентификации подлинности авторства электронных документов?

2. Как решается проблема обеспечения целостности и аутентификации подлинности авторства для бумажных документов? Почему этот подход нельзя использовать для электронных документов?

3. Перечислите возможности злоумышленника при реализации угроз, направленных на нарушение целостности передаваемых сообщений и подлинность их авторства.

4. Что понимают под функцией хэширования?

5. Перечислите требования к функциям хэширования. Кратко охарактеризуйте их.

6. Что понимают под ЭЦП?

7. Перечислите условия, реализацию которых позволяет гарантировать использование ЭЦП.

8. На каком из ключей выполняется процедура установки ЭЦП? Проверка ЭЦП?

9. Опишите схемы процедур установки и проверки ЭЦП.

10. Приведите примеры алгоритмов хэширования.

11. Приведите примеры алгоритмов ЭЦП.

12. Что понимают под инфраструктурой открытых ключей?

13. Дайте определение цифрового сертификата. В чем заключаются его функции?

14. Перечислите основные компоненты технологии PKI.

15. В чем заключаются функции удостоверяющего центра?

16. Перечислите основные сервисы технологии PKI.

17. Что понимают под политикой и регламентом инфраструктуры открытых ключей?

7. Хранение и распределение ключевой информации. Протоколы безопасной аутентификации пользователей

7.1. Типовые схемы хранения ключевой информации

Рассмотрим типовые схемы хранения ключевой информации в открытых компьютерных системах на примере хранения информации для аутентификации пользователей.

Предположим, что *i*-й аутентифицируемый субъект содержит два информационных поля: ID_i - неизменяемый идентификатор *i*-го пользователя, который является аналогом имени и используется для идентификации пользователя, и K_i - аутентифицирующая информация пользователя, которая может изменяться и служит для аутентификации.

Пара (ID_i, K_i) составляет базовую информацию, относящуюся к учетной записи пользователя, которая хранится в базе данных аутентификации компьютерной системы.

Базу данных аутентификации в открытых компьютерных системах (не использующих специализированных аппаратных средств) приходится хранить в некотором объекте файловой системы ПК. Это приводит к потенциальной возможности реализации угроз, направленных на кражу базы данных аутентификации злоумышленником и ее дальнейшего исследования.

Базу данных аутентификации в КС необходимо защищать от двух основных видов угроз.

1. Угрозы прямого доступа к базе данных аутентификации с целью ее копирования, исследования, модификации.

Реализация защиты от данного вида угроз в ОС типа Windows NT, 2000, XP подразумевает контроль доступа к базе данных аутентификации на уровне операционной системы и запрет любого доступа к этим базам за исключением привилегированных системных процессов.

В ОС типа UNIX защита от подобных угроз реализуется путем соответствующего определения дискреционной политики безопасности.

Однако следует отметить, что реализации данных защит практически никогда не работают корректно. Например, базу данных аутентификации ОС, построенных на технологии NT, злоумышленник может получить с помощью специализированных утилит из реестра, куда она копируется при загрузке ОС, либо загрузившись с другого носителя. В связи с этим, при защите баз данных аутентификации большее внимание уделяется защите от второго вида угроз. При этом предполагается, что злоумышленник смог получить доступ к содержимому базы данных аутентификации.

2. Угрозы исследования содержимого базы данных аутентификации.

Пароли доступа не могут храниться в прямом виде в базе данных аутентификации, так как злоумышленник может получить доступ к этой базе и раскрыть все пароли. При хранении в данной базе пароли должны закрываться. Такой метод закрытия паролей, как шифрование, не обладает необходимой стойкостью, так как шифрование должно производиться на некотором ключе, который также необходимо где-то хранить, следовательно, существует потенциальная возможность раскрытия ключа шифрования злоумышленником. Кроме этого желательно, чтобы подсистема аутентификации пользователя не осуществляла сравнение введенного пользователем пароля с реальным паролем непосредственно в оперативной памяти, так как существующие средства отладки, типа SoftIce, позволяют отладить в пошаговом режиме процедуру аутентификации и получить доступ к реальным паролям (узнать, что хочет видеть компьютерная система на этапе аутентификации).

Таким образом, закрытие паролей в базах данных аутентификации должно осуществляться методами, отличными от шифрования, и так, чтобы эталонные пароли не были известны даже самой подсистеме аутентификации. С другой стороны, подсистема аутентификации должна однозначно определять корректность введенного пароля, не зная эталонного.

Существует две типовые схемы хранения ключевой информации в базах данных аутентификации, позволяющие решить эти задачи [14].

Схема 1. В компьютерной системе выделяется объект-эталон для идентификации и аутентификации. Структура объекта-эталона может быть представлена в виде таблицы 7.1.

Табл. 7.1. Первая типовая схема хранения ключевой информации

Номер пользователя	Информация для идентификации	Информация для аутентификации
1	ID_1	E_1
2	ID_2	E_2
…	…	…
N	ID_N	E_N

$E_i=F(ID_i,K_i)$, где F – некоторая функция хэширования. При этом, зная E_i и ID_i вычислительно невозможно восстановить K_i.

Таким образом, в базе данных аутентификации вместо эталонных паролей K_i хранится результат их одностороннего преобразования. В качестве односторонней функции для хэша NTLM в Windows NT используется алгоритм хэширования MD4.

Алгоритм идентификации и аутентификации для схемы 1

1. Пользователь предъявляет свой идентификатор ID.

2. Если ID не совпадает ни с одним ID_i, зарегистрированным в компьютерной системе, то идентификация отвергается - пользователь не допущен к работе, иначе (существует $ID_i = ID$) устанавливается факт «пользователь, назвавшийся пользователем i, прошел идентификацию».

3. Субъект аутентификации запрашивает у пользователя аутентификатор K и вычисляет значение $Y=F(ID_i, K)$.

4. Субъект аутентификации производит сравнение E_i и Y. При совпадении фиксируется событие «пользователь успешно аутентифицирован в системе», в противном случае аутентификация отвергается и пользователь не допускается к работе.

Впервые данная методика была предложена Роджером Ниджемом и Майком Гаем (проект «Титан», 1967 год, Кембридж).

Вторая типовая схема хранения ключевой информации несколько модифицирует схему 1.

Схема 2. В компьютерной системе выделяется объект-эталон, структура которого показана в таблице 7.2.

Табл. 7.2. Вторая типовая схема хранения ключевой информации

Номер пользователя	Информация для идентификации	Информация для аутентификации
1	ID_1, S_1	E_1
2	ID_2, S_2	E_2
…	…	…
N	ID_N,S_N	E_N

В данной таблице $E_i=F(S_i, K)$, где S_i - случайный вектор, формируемый при создании пользователя с номером i; F – необратимая функция, для которой невозможно восстановить K по E_i и S_i.

Алгоритм идентификации и аутентификации для схемы 2

1. Пользователь предъявляет свой идентификатор ID.

2. Если ID не совпадает ни с одним ID_i, зарегистрированным в компьютерной системе, то идентификация отвергается - пользователь не допущен к работе, иначе (существует $ID_i=ID$) устанавливается факт «пользователь, назвавшийся пользователем i, прошел идентификацию».

3. По идентификатору ID_i из базы данных аутентификации выделяется информация S_i.

4. Субъект аутентификации запрашивает у пользователя аутентифицирующую информацию K. и вычисляет значение $Y=F(S_i, K)$.

5. Субъект аутентификации сравнивает E_i и Y. При совпадении фиксируется событие «пользователь успешно аутентифицирован в КС», в противном случае аутентификация отвергается и пользователь не допускается к работе.

Достоинством второй схемы является то, даже в случае выбора пользователями одинаковых паролей, информация E_i для них будет различаться. В

рамках первой же схемы значение $E_i=F(ID_i,K_i)$, как правило, вычисляют в виде $E_i=F(K_i)$, что не позволяет достичь такого результата. Вторая схема хранения ключевой информации используется для защиты базы данных аутентификации в ОС UNIX.

Если для защиты паролей используются криптографически стойкие функции F, то единственно возможным способом взлома ключевой системы является полный перебор ключей. В этом случае злоумышленник должен последовательно перебирать ключи K, для каждого из ключей формировать информацию E, закрывая его по известному алгоритму, и сравнивать полученную информацию E с информацией для аутентификации E_i.

Покажем, к чему может привести использование криптографически нестойких алгоритмов хэширования в качестве функции F.

Пример 7.1

Для защиты книг Microsoft Excel используется подход к защите пароля, аналогичный схеме 1. В документе Excel хранится хэш-образ пароля, с которым производится сравнение хэша пароля, вводимого пользователем при снятии данной защиты. Длина хэша составляет 16 бит. Используемая функция хэширования не обладает хорошими свойствами рассеивания. Это приводит к тому, что многим паролям соответствует один и тот же хэш-образ. Например, если попытаться защитить книгу Excel на пароле «test», то в качестве верного будет принят и пароль «zzyw». Данный факт является очень плохим свойством, в особенности для баз данных аутентификации ОС.

При защите хранилищ ключевой информации в рамках схем 1 и 2, в том числе и баз данных аутентификации, необходимо принимать во внимание следующее утверждение.

Утверждение (о подмене эталона). Если пользователь имеет возможность записи объекта хранения эталона, то он может быть идентифицирован и аутентифицирован (в рамках рассмотренных схем), как любой пользователь.

Доказательство. Пусть имеется пользователь i. Покажем, что он может выдать себя за любого пользователя j. Возможность записи в объект, содержащий эталоны, означает возможность замены любой записи на произвольную. Пользователь i меняет в j-ой записи информацию E_j на свою E_i (или дополнительно еще S_j на S_i). При следующей процедуре идентификации, введя идентификатор ID_j и свой пароль, он будет опознан как пользователь j. Утверждение доказано.

Смысл данного утверждения состоит в том, что возможность записи объекта хранения эталонов должны иметь только субъекты со специально наделенными привилегиями, отвечающие за управление безопасностью.

Пример 7.2

При защите документов Microsoft Word на изменение, а также книг и листов Microsoft Excel используется подход к защите паролей, аналогичный схеме 1. В документе хранится хэш эталонного пароля, с которым производится сравнение хэша пароля, вводимого пользователем при снятии данной защиты. Для взлома данной защиты достаточно просто записать на место хранения хэш-образа эталонного пароля заранее вычисленный хэш-образ известного пароля, либо хэш-образ, соответствующий беспарольному варианту. Так и поступают многочисленные взломщики защит документов Word и Excel.

7.2. Защита баз данных аутентификации в ОС Windows NT и UNIX

Защита баз данных аутентификации в ОС, построенных на технологии Windows NT

База данных аутентификации в ОС, построенных на технологии NT, имеет название SAM (Security Accounts Manager) и располагается в каталоге Winnt\System32\Config\ [18].

Информация в этой базе данных хранится в служебном формате, а доступ к ней ограничен со стороны ОС. Любое обращение к этой базе со сторо-

ны пользователя (копирование, чтение, запись и т.д.) блокируется. Кроме этого, данная база данных при загрузке ОС копируется в реестр.

Существующие средства в Windows NT, ограничивающие доступ к базе данных SAM, не работают корректно, и злоумышленник обходными путями может получить доступ к этой базе данных, в том числе и скопировать ее для последующего анализа.

Рассмотрим реализованный Microsoft способ защиты баз данных аутентификации SAM он несанкционированного изучения [18,25].

В базе данных аутентификации SAM для каждой учетной записи пользователя хранится два вида хэшей пароля – хэш LANMAN, используемый для аутентификации сетевых служб и совместимости с ранее разработанными ОС Windows 9x, и хэш NTLM, используемый при локальной аутентификации пользователя.

Алгоритм хэширования LANMAN

Схема данного алгоритма представлена на рис . 7.1.

Рис. 7.1.Схема алгоритма хэширования LANMAN

Шаг 1. Пользовательский пароль преобразуется путем замены всех малых символов, входящих в него, большими.

Шаг 2. Результат преобразуется в 14-символьную цепочку. Если пароль длиннее 14 символов, то лишние символы урезаются; если короче, то недостающие позиции заполняются нулями.

Шаг 3. Полученная цепочка из 14 символов делится на два блока по 7 символов, каждый из которых в дальнейшем обрабатывается независимо.

Шаг 4. Каждый из сформированных блоков используется в качестве ключа шифрования алгоритма DES известной 64-битовой последовательности (4B, 47, 53, 21, 40, 23, 24, 25). На выходе формируются два блока по 8 байт.

Шаг 5. Конкатенация двух 8-байтных блока является хэшем LANMAN (16 байт).

В алгоритме LANMAN используется свойство стойкости к атакам по открытому тексту алгоритма DES для формирования закрытых паролей. Даже зная 8-байтную последовательность, которая шифруется по данному алгоритму, восстановление ключа шифрования возможно только полным перебором.

Алгоритм LANMAN обладает рядом недостатков, которые значительно снижают его криптостойкость. Перечислим их.

1. Преобразование всех символов в заглавные значительно снижает объем ключевого пространства, которое нужно перебрать злоумышленнику. Данное преобразование уменьшает энтропию паролей.

2. Разбивка пароля на два фрагмента, обрабатываемых независимо, также приводит к значительному снижению объема ключевого пространства.

Предположим, что пароли составляются только из малых и больших английских букв (мощность алфавита равна 52). Так как все символы пароля преобразуются в большие, то мощность алфавита уменьшается до 26. Кроме этого, независимость обработки блоков из 7 символов приводит к тому, что для взлома хэша LANMAN приходится осуществлять перебор не 26^{14} вариантов ключей, а $2*26^{7}$, так как левая и правая части хэша LANMAN подбираются независимо. Таким образом, в силу двух приведенных недостатков,

для рассмотренного примера пространство перебора ключей содержит не 52^{14} паролей, а $2*26^{7}$. Перебрать такое количество ключей не трудно для современной вычислительной техники.

Алгоритм хэширования NTLM

Алгоритм хэширования NTLM свободен от недостатков, свойственных хэшу LANMAN. Схема данного алгоритма представлена на рис. 7.2.

В NTLM символы не преобразуются к верхнему регистру и могут быть любыми. Разбивка на два блока здесь также не используется. В качестве алгоритма хэширования использован MD4.

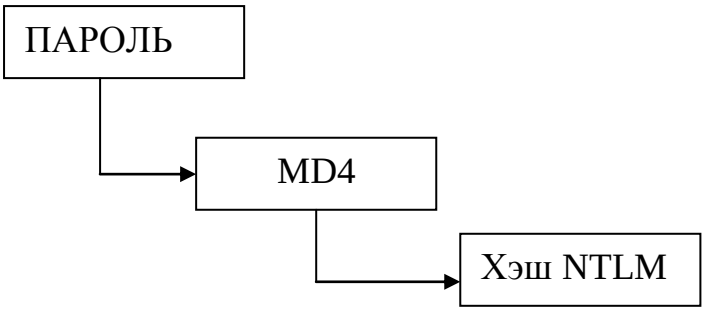


Рис. 7.2. Схема алгоритма хэширования NTLM

Следует отметить, что для совмещения с прошлыми версиями Windows, в базе данных SAM хранятся оба хэша – LANMAN и NTLM (за исключением паролей длины, большей 14). Поэтому, наличие хэша NTLM в SAM никак не усиливает защиту, взломать ее злоумышленник может так же быстро, подобрав вначале хэш LANMAN и определив пароль с приближением к верхнему регистру, затем найти истинный пароль, подобрав хэш NTLM путем перекомбинации больших и малых букв.

7.3. Иерархия ключевой информации

Другой подход, достаточно часто используемый для хранения ключевой информации с участием отчуждаемых носителей ключей, состоит в шифровании ключей и хранении их в зашифрованном виде. Кроме этого, данный подход часто используют для распределения ключевой информации в криптографических сетях.

Необходимость в хранении и передаче ключевой информации, зашифрованной с помощью других ключей, привел к развитию концепции *иерархии ключей*.

Иерархия ключевой информации может включать множество уровней, однако, наиболее часто выделяют главные ключи (мастер-ключи), ключи шифрования ключей и рабочие ключи (сеансовые).

Сеансовые ключи находятся на самом нижнем уровне и используются для шифрования данных. Когда эти ключи необходимо безопасным образом передать между узлами сети или безопасно хранить, их шифруют с помощью ключей следующего уровня – *ключей шифрования ключей*.

На верхнем уровне иерархии ключей располагается мастер-ключ. Этот ключ применяют для шифрования ключей шифрования, когда требуется безопасно хранить их на диске. Обычно в каждом компьютере используется только один мастер ключ, который отчуждается на внешнем носителе, как правило, защищенном от несанкционированного доступа, чтобы раскрыть значение этого ключа было невозможно (смарт-карта, e-Token и т.п.). Значение мастер-ключа фиксируется на длительное время (до нескольких недель или месяцев). Сеансовые ключи меняются намного чаще, например, при построении криптозащищенных туннелей их можно менять каждые 10-15 минут, либо по результатам шифрования заданного объема трафика (например, 1 Мб).

7.4. Распределение ключей

Распределение ключей является очень ответственным процессом в управлении ключами. Одним из основных требований к реализации этого процесса является сокрытие распределяемой ключевой информации.

Задача распределения ключей сводится к построению протокола распределения ключей, обеспечивающего:

1. взаимное подтверждение подлинности участников сеанса;

2. подтверждение достоверности сеанса для защиты от атак методов повторов;

3. использование минимального числа сообщений при обмене ключами;

Вообще говоря, выделяют два подхода к распределению ключевой информации в компьютерной сети [13].

1. Распределение ключевой информацией с использованием одного либо нескольких центров распределения ключей.

2. Прямой обмен сеансовыми ключами между пользователями.

Распределение ключевой информации с использованием центров распределения ключей

Данный подход предполагает, что центру распределения ключей известны распределяемые ключи, в связи с чем, все получатели ключевой информации должны доверять центру распределения ключей.

Достоинством данного подхода является возможность централизованного управления распределением ключевой информацией и даже политикой разграничения доступа удаленных субъектов друг к другу.

Данный подход реализован в протоколе Нидхема-Шредера и базирующемся на нем протоколе аутентификации Kerberos. Распределение ключевой информацией и разграничение доступа основывается в данных протоколах на выдаче мандатов центром распределения ключей. Использование данных протоколов позволяет безопасно распределить сеансовые ключи даже в случае взаимного недоверия двух взаимодействующих сторон.

Прямой обмен сеансовыми ключами между пользователями

Для возможности использования при защищенном информационном обмене между противоположными сторонами криптосистемы с секретным ключом, взаимодействующим сторонам необходима выработка общего секрета, на базе которого они смогут безопасно шифровать информацию или безопасным образом вырабатывать и обмениваться сеансовыми ключами. В первом случае общий секрет представляет собой сеансовый ключ, во втором

случае – мастер-ключ. В любом случае, злоумышленник не должен быть способен, прослушивая канал связи, получить данный секрет.

Для решения проблемы выработки общего секрета без раскрытия его злоумышленником существует два основных способа:

1. использование криптосистемы с открытым ключом для шифрования и передачи секретного ключа симметричной криптосистемы;

2. использование протокола открытого распространения ключей Диффи-Хеллмана.

Реализация первого способа не должна вызывать вопросов. Рассмотрим более подробно реализацию второго способа.

Протокол Диффи-Хеллмана

Протокол Диффи-Хеллмана был первым алгоритмов работы с открытыми ключами (1976 г.). Безопасность данного протокола основана на трудности вычисления дискретных логарифмов в конечном поле [13].

Пусть пользователи А и В хотят выработать общий секрет. Для этого они выполняют следующие шаги.

1. Стороны А и В договариваются об используемом модуле N, а также о примитивном элементе g, $1 \le g \le N$, степени которого образуют числа от 1 до N-1, то есть во множестве $\{g, g^2, ..., g^{N-1} = 1\}$ присутствуют все числа от 1 до N-1. Числа N и g являются открытыми элементами протокола.

2. Пользователи А и В независимо друг от друга выбирают собственные секретные ключи $СК_A$ и $СК_B$ (случайные большие целые числа, меньшие N, хранящиеся в секрете).

3. Пользователи А и В вычисляют открытые ключи $ОК_A$ и $ОК_B$ на основании соответствующих секретных ключей по следующим формулам:

$$OK_A = g^{CK_A} \pmod N; OK_B = g^{CK_B} \pmod N$$

4. Стороны А и В обмениваются между собой значениями открытых ключей по незащищенному каналу.

5. Пользователи А и В формируют общий секрет K по формулам:

Пользователь А: $K = (OK_B)^{CK_A} = \left(g^{CK_B}\right)^{CK_A} = g^{CK_B \cdot CK_A} \pmod N$

Пользователь В: $K = (OK_A)^{CK_B} = \left(g^{CK_A}\right)^{CK_B} = g^{CK_A \cdot CK_B} \pmod N$

Ключ K может использоваться в качестве общего секретного ключа (мастер-ключа) в симметричной криптосистеме.

Пример 7.3

Возьмем модуль N=47 и примитивный элемент g=23. Пусть пользователи А и В выбрали свои секретные ключи $СК_A$=12, $СК_B$=33. Тогда,

$$OK_A = g^{CK_A} \pmod{47} = 23^{12} \pmod{47} = 27$$

$$OK_B = g^{CK_B} \pmod{47} = 23^{33} \pmod{47} = 33$$

В данном случае общий секрет $K = (OK_B)^{CK_A} = 33^{12} \pmod{47} = 25$.

Алгоритм открытого распределения ключей Диффи-Хеллмана позволяет обойтись без защищенного канала для передачи ключей. Однако, необходима гарантия того, что получатель получил открытый ключ именно от того отправителя, от которого он его ждет. Данная проблема решается с помощью цифровых сертификатов и технологии ЭЦП.

Протокол Диффи-Хеллмана нашел эффективное применение в протоколе SKIP управления ключами. Данный протокол используется при построении криптозащищенных туннелей в семействе продуктов ЗАСТАВА.

7.5. Протоколы безопасной удаленной аутентификации пользователей

Одной из важнейших задач при удаленной аутентификации пользователей является обеспечение подлинности канала связи. Решение этой задачи путем передачи по каналу связи секретного ключа в закрытом виде (в зашифрованном, либо в виде хэш-образа) не является стойким к атакам, так как злоумышленник, слушая канал связи, может реализовать атаку методом повторов. Для обеспечения подлинности канала связи, и защиты от атак повторами обычно используют метод запрос-ответ, либо механизм отметки времени.

Механизм запрос-ответ заключается в том, что пользователь А при необходимости аутентификации пользователя В посылает ему запрос, в который включает непредсказуемый элемент (например, случайное число). Пользователь В должен ответить на этот запрос, предварительно выполнив некую обработку этого элемента. При этом злоумышленник не способен подделать ответ, так как в механизм обработки запроса включается секретная информация. После проверки результата пользователем А, присланным пользователем В, выполняется подтверждение или не подтверждение подлинности сеанса работы.

Механизм отметки времени заключается в том, что для каждого пересылаемого сообщения фиксируется время. Это позволяет каждому субъекту сети определить, насколько старо пришедшее сообщение, и отвергнуть его, если появится сомнение в его подлинности.

Рассмотрим ряд протоколов удаленной аутентификации пользователей.

Протокол CHAP (Challenge Handshaking Authentication Protocol)

Предполагается, что аутентифицируемая сторона (клиент) и аутентифицирующая (сервер) уже обладают общим секретом (например, паролем доступа к серверу). Задача состоит в безопасной удаленной аутентификации клиента, проверке его подлинности путем проверки знания общего секрета [6].

Схема протокола CHAP представлена на рис. 7.3

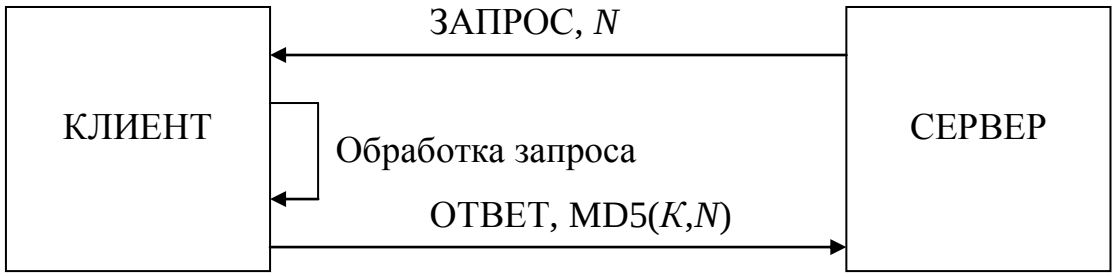


Рис. 7.3. Схема протокола CHAP

1. При необходимости прохождения аутентификации сервер посылает сообщение «ЗАПРОС» клиенту, в которое включает случайный, уникальный и непредсказуемый номер *N*.

2. Клиент обрабатывает запрос сервера и формирует ответную последовательность, хэшируя пароль и случайный номер N с помощью алгоритма MD5, то есть вычисляет значение MD5(K,N).

3. Клиент отправляет серверу для аутентификации пакет «ОТВЕТ», в которую включает вычисленное значение MD5(K,N).

4. Сервер, зная эталонный пароль клиента и посланное ему значение N, также вычисляет значение MD5(K,N) и сравнивает его с присланным клиентом. По результатам сравнения сервер принимает решение о прохождении либо не прохождении этапа аутентификации клиентом.

Использование в протоколе случайного числа N практически исключает возможность пересылки от клиента к серверу одинаковых последовательностей в течение длительного времени. Злоумышленник же, зная число N, не сможет восстановить ответ клиента, так как не знает секретного ключа K. В силу высокой криптостойкости функции хэширования MD5, злоумышленник, зная число N и значение MD5(K,N), не сможет восстановить ключ K.

Протокол одноразовых ключей S/KEY

Протокол одноразовых ключей S/KEY основан на независимом формировании клиентом и сервером последовательности одноразовых паролей, основанной на общем секрете K. При этом знание злоумышленником очередного пароля, пересылаемого на фазе аутентификации, не дает ему возможности выяснить следующий пароль [6].

Пусть K – пароль аутентификации, известный как подлинному клиенту, так и серверу. Клиент и сервер на основании ключа K могут вычислить последовательность из M одноразовых ключей $S_1,...,S_M$ следующим образом:

$$S_1 = MD4(K),$$

$$S_2 = MD4(S_1) = MD4(MD4(K)) = MD4^2(K),$$

...

$$S_M = MD4(S_{M-1}) = MD4^M(K)$$

Если клиент будет пересылать серверу на этапе аутентификации одноразовые пароли в обратной последовательности: при первой аутентифика-

ции S_M, затем S_{M-1},..., S_1, то знание злоумышленником очередного пароля S_i не позволит восстановить ему пароль S_{i-1}, который будет ожидаться сервером при следующей аутентификации, так как для этого ему потребуется обратить функцию хэширования MD4, что является вычислительно трудоемкой задачей. Поэтому описанный подход может быть использован для решения задачи безопасной удаленной аутентификации пользователя.

Недостатком описанной выше схемы является то, что после исчерпания всех одноразовых паролей (после M последовательных аутентификаций) необходимо менять общий секрет K, так как если пароли начнут передаваться заново, начиная с S_M, то злоумышленник, слушая канал связи, будет уже знать всю предысторию передаваемых паролей, и сможет пройти аутентификацию. Для устранения данного недостатка используют подход, основанный на передаче случайного числа N от клиента к серверу в момент формирования списка одноразовых паролей, и использование данного числа как второго аргумента функции хэширования MD4. Схема аутентификации клиента с помощью протокола S/KEY будет выглядеть в данном случае следующим образом.

1. Сервер высылает клиенту число M одноразовых паролей, список которых необходимо проинициализировать и случайное число N, используемое для генерирования уникального и непредсказуемого списка.

2. Клиент и сервер генерируют последовательность из M одноразовых паролей следующим образом:

$S_1 = MD4(K, N)$,

$S_2 = MD4(S_1, N) = MD4(MD4(K, N)) = MD4^2(K, N)$,

...

$S_M = MD4(S_{M-1}, N) = MD4^M(K, N)$

3. При необходимости аутентификации сервер посылает клиенту число t, в ответ клиент посылает серверу одноразовый пароль S_t. Сервер, анализи-

руя принятую информацию, принимает решение о принятии либо отвержении аутентификации.

4. В следующий раз сервер требует на этапе аутентификации пароль S_{t-1} ... пока не дойдет до S_1.

5. Если список одноразовых паролей исчерпан (переслали S_1), то клиентом и сервером выполняется повторная инициализация списка одноразовых паролей (при другом *N*).

Реализация метода «запрос-ответ» в ОС Windows при сетевой аутентификации

Метод «запрос-ответ» используется в ОС Windows при удаленной аутентификации пользователя, подключающегося к сетевым ресурсам общего пользования, с более старых ОС. При этом используется аутентификация с помощью хэша LANMAN [25]. Схема данного метода представлена на рис. 7.4.

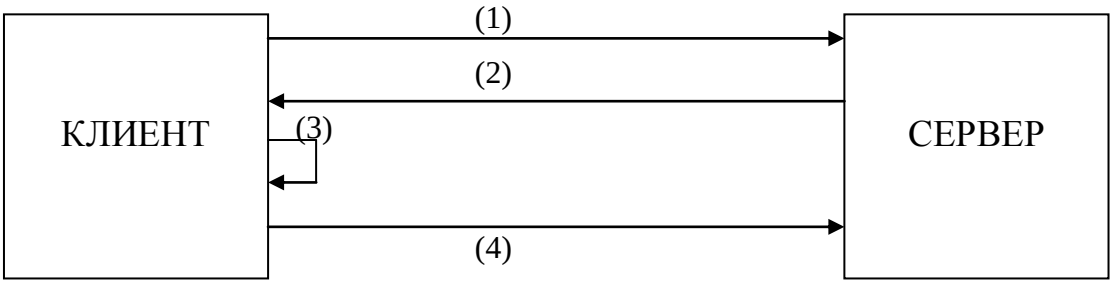


Рис. 7.4. Реализация метода «запрос-ответ» в ОС Windows

Шаг 1. Клиент запрашивает разрешение у сервера на подключение к сетевому ресурсу общего пользования.

Шаг 2. Сервер отвечает случайным восьмибайтовым числом.

Шаг 3. У клиента открывается окно для ввода идентификатора и пароля.

Шаг 4. Клиент формирует 24-байтный ответ серверу на основе следующего алгоритма:

Алгоритм формирования ответа

1. Пароль, введенный пользователем, хэшируется на стороне клиента с помощью алгоритма хэширования LANMAN. В результате этого формируется 16-байтовая свертка пароля.

2. Полученный 16-байтовый хэш разбивается на 3 блока по 56 бит. Последний блок до 56 бит дополняется нулями (рис. 7.5).

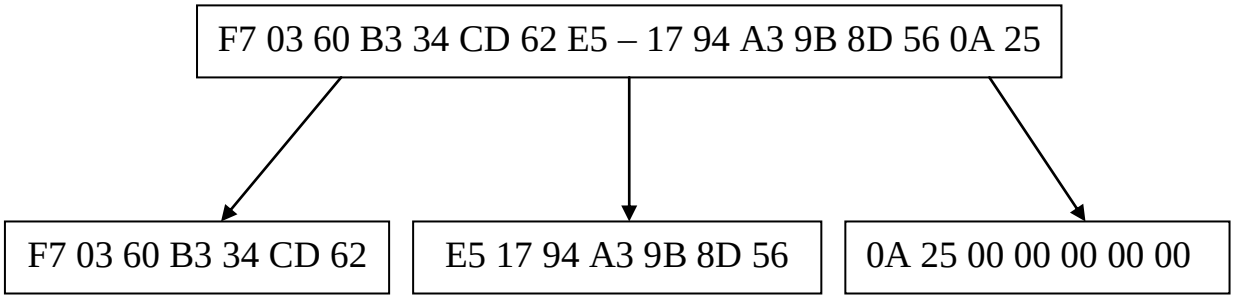


Рис. 7.5. Разбивка хэша LANMAN на три блока

3. Пришедший от сервера 8-байтовый ответ шифруется 3 раза с помощью трех ключей шифрования (представляющих собой три полученных на шаге 2 блока хэша LANMAN) по алгоритму DES. В результате этого формируется 24-байтный ответ, отправляемый серверу (рис. 7.6).

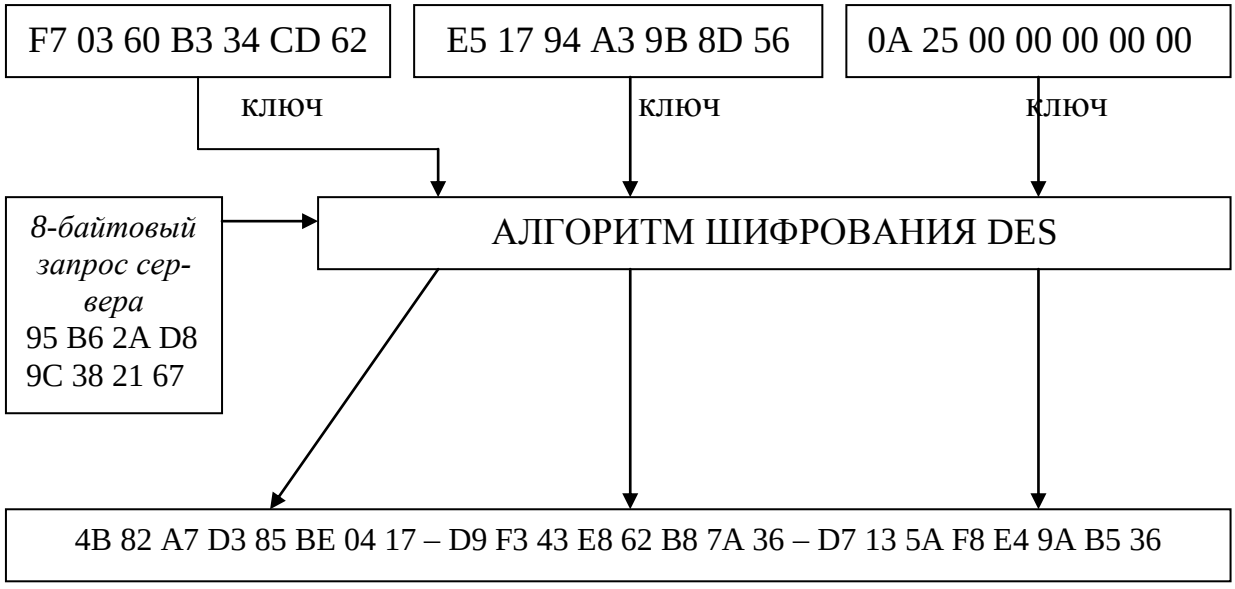


Рис. 7.6. Алгоритм формирования ответа серверу

Шаг 5. Сервер, получив ответ от клиента, может проверить его корректность, а по результатам проверки подтвердить либо отклонить аутентификацию.

Кроме рассмотренных выше протоколов безопасной удаленной аутентификации пользователей, широкое распространение получил также протокол аутентификации Kerberos.

7.6. Вопросы для самоконтроля

1. Перечислите основные угрозы базам данных аутентификации в компьютерных системах.
2. Опишите типовые схемы хранения ключевой информации в компьютерных системах и алгоритмы идентификации и аутентификации пользователей в рамках данных схем.
3. Сформулируйте и докажите утверждение о подмене эталона.
4. Опишите алгоритм хэширования LANMAN. Укажите на уязвимые места данного алгоритма.
5. Опишите алгоритм хэширования NTLM.
6. В чем заключаются функции сеансовых ключей и мастер-ключей?
7. Какие задачи должны решаться на этапе распределения ключей?
8. Опишите протокол Диффи-Хеллмана.
9. Что понимают под атаками методом повторов при удаленной аутентификации?
10. В чем заключается механизм запрос-ответ и механизм отметки времени при безопасной удаленной аутентификации?
11. Опишите схему протокола безопасной удаленной аутентификации CHAP. Почему злоумышленник не может восстановить ответ клиента?
12. Опишите протокол одноразовых ключей S/KEY. Почему злоумышленник не может восстановить следующий пароль в последовательности?

8. Защита информации в компьютерных сетях

8.1. Основные угрозы и причины уязвимости сети INTERNET

Интенсивное развитие INTERNET и INTRANET технологий, привлечение их для создания новых технологий хранения, поиска и обработки информации влечет за собой необходимость построения эффективных систем защиты информации в корпоративных сетях.

В настоящее время глобальные сети часто используются для передачи информации, содержащей сведения различного уровня конфиденциальности, например, для связи между головным и удаленными офисами организации, для доступа к WEB сайтам организации и т.д. Многие организации принимают решение об интеграции своих локальных и корпоративных сетей в глобальную сеть INTERNET, предоставляют различные услуги через данную сеть (организация электронных магазинов, системы дистанционного образования и т.д.).

Такой подход дает, несомненно, множество преимуществ, связанных с большими потенциальными возможностями коллективной работы в INTRANET и INTERNET, более эффективному интегрированию различных информационных технологий, связанных с хранением, поиском и обработкой информации.

Однако развитие глобальных сетей привело к многократному увеличению количества пользователей и атак на ПК, подключенных к сети INTERNET и внутренним сетям INTRANET организаций. Ежегодные потери, обусловленные недостаточным уровнем защищенности таких ПК, оцениваются десятками миллионов долларов. При подключении к Internet локальной или корпоративной сети необходимо позаботиться об обеспечении ИБ данной сети, подключенных к ней ПК [13].

Изначальная разработка сети INTERNET как открытой сети создает большие возможности для злоумышленника по воздействию на локальные и корпоративные сети организаций, имеющие выход в INTERNET. Через INTERNET злоумышленник может вторгнуться во внутреннюю сеть пред-

приятия и получить НСД к конфиденциальной информации, получить пароли доступа к серверам, а подчас и их содержимое.

Наиболее распространенные угрозы ИБ в INTERNET и INTRANET представлены ниже [35]:

1. несанкционированный (неавторизованный) доступ внешних пользователей к какому-либо виду сервисного обслуживания, предоставляемого легальным пользователям;

2. несанкционированный доступ к информации и базам данных организаций без идентификации и аутентификации внешнего пользователя в сети;

3. внедрение в системы и сети организаций разрушающих программных воздействий – вирусов, программных закладок, троянских коней и т.д., используя различные уязвимости удаленных систем (например, внедрение вирусов через электронную почту, используя уязвимости IIS);

4. нарушение целостности ПО систем и сетей организаций с целью модификации выполняемых ими функций;

5. нарушение конфиденциальности информационного обмена, осуществляемого по каналам связи абонентов систем и сетей организаций, с помощью их «прослушивания»; данный вид угроз для компьютерных сетей получил более конкретное название – **сниффинг (sniffing)**, а программы, реализующие эту угрозу, называют снифферами;

6. нарушение работоспособности программных компонентов удаленных систем с целью дезорганизации их работы – атаки вида отказа в обслуживании (DoS – Denied of Service); защита от данного вида атак очень актуальна в настоящее время для компаний, предоставляющих различные услуги посредством INTERNET;

7. получение прав доступа к удаленной системе, использующей нестойкие алгоритмы аутентификации пользователя (**маскарад, spoofing**);

8. доступ к информации о топологии сетей и используемых в них механизмах защиты, что облегчает злоумышленникам проникновение в сети;

достаточно часто информацию такого рода злоумышленник может получить путем удаленного сканирования системы.

Результаты воздействия угроз могут выражаться в появлении сбоев в работе информационных систем организаций, искажении либо разрушении циркулирующей или хранящейся в них информации, нарушении защитных механизмов систем, что позволяет осуществить злоумышленнику НСД к информации, контролировать работу информационных систем.

Основными причинами уязвимости сети INTERNET являются следующие [35]:

1. проектирование сети INTERNET как открытой и децентрализованной сети с изначальным отсутствием политики безопасности;
2. уязвимости служб протокола TCP/IP;
3. большая протяженность каналов связи;
4. множество уязвимостей программного и аппаратного обеспечения, работающего на ПК, подключенных в INTERNET (уязвимости ОС, WEB-серверов, почтовых клиентов и пр.); каталоги известных уязвимостей пополняются буквально каждую неделю (наиболее полный каталог известных уязвимостей доступен на сервере http://icat.nist.gov);
5. кажущаяся анонимность при работе в INTERNET, возможность скрытия о себе информации злоумышленником, использования анонимных proxy-серверов, ремэйлеров для электронной почты и пр.;
6. доступность информации о средствах и протоколах защиты, используемых в INTERNET;
7. работа в INTERNET обслуживается большим числом сервисов, информационных служб и сетевых протоколов, знание тонкостей и правильности конфигурирования которых одному человеку в лице администратора не всегда реально;
8. сложность конфигурирования средств защиты;
9. человеческий фактор.

Все эти факторы требуют разработки, внедрения и использования средств защиты локальных сетей организации, отдельных компьютеров локальных сетей, имеющих выход в INTERNET, либо непосредственно подключенных к нему.

8.2. Классификация типовых удаленных атак на интрасети. Подходы к защите от типовых удаленных атак

Принципиальным отличием атак, осуществляемых злоумышленниками в компьютерных сетях, является фактор расстояния злоумышленника от ПК, выбранного в качестве жертвы. В связи с этим, такие атаки принято называть **удаленными атаками (УА)** [35].

В настоящее время выделяют следующие классы типовых удаленных атак, осуществляемых на компьютерные сети.

Анализ сетевого трафика

Анализ сетевого трафика путем его перехвата (сниффинга) является внутрисегментной атакой и направлен на перехват и анализ информации, предназначенной для любого ПК, расположенного в том же сегменте сети, что и злоумышленник. Злоумышленник может захватить все проходящие через себя пакеты путем перевода своей сетевой платы в смешанный режим (promiscuous mode).

Реализация данной атаки позволяет злоумышленнику изучить логику работы сети (для получения информации, помогающей ему осуществить последующий взлом) либо перехватить конфиденциальную информацию, которой обмениваются узлы компьютерной сети. Многие протоколы (например, POP3, FTP и пр.) передают информацию об используемых паролях доступа по каналу связи в открытом виде. Анализ трафика позволяет злоумышленнику перехватить эти пароли доступа (например, к электронной почте, к FTP серверу) и использовать их в дальнейшем для выполнения несанкционированных действий.

Для защиты от анализа сетевого трафика с использованием снифферов известны следующие подходы:

1. диагностика перевода сетевой платы удаленного ПК в смешанный режим путем установки различных средств мониторинга; данный подход к защите достаточно трудоемок, и не является универсальным, поэтому используется недостаточно часто;

2. сегментация сетей – чем больше сегментов, тем меньше вероятность и последствия реализации внутрисегментной атаки;

3. шифрование сетевого трафика и использование безопасных протоколов удаленной аутентификации пользователей (S/KEY, CHAP и т.д.);

Подмена доверенного субъекта

Подмена доверенного субъекта и передача сообщений по каналам связи от его имени позволяет получить злоумышленнику доступ к удаленной системе от имени этого доверенного субъекта. Подобные атаки эффективно реализуются в системах с нестойкими алгоритмами идентификации и аутентификации хостов и пользователей. Например, подобные атаки эффективны для систем, использующих аутентификацию источника по его IP адресу, для злоумышленника в этом случае нетрудно формировать пакеты с IP адресами, которым «доверяет» удаленный узел.

Для защиты от подобных атак необходимо применение стойких алгоритмов идентификации и аутентификации хостов и пользователей. Нельзя допускать в компьютерную сеть организации пакеты, посланные с внешних ПК, но имеющих внутренний сетевой адрес.

Введение ложного объекта компьютерной сети

Реализация данной атаки позволяет навязать ложный маршрут потока информации так, чтобы этот маршрут лежал через компьютер злоумышленника, позволяет «заманить» легального пользователя на ПК злоумышленника (например, подменив WEB-сайт) с целью получения конфиденциальной информации.

Для защиты от данных атак необходимо использовать более стойкие протоколы идентификации и аутентификации хостов и устройств. Подобные протоколы рассмотрены в главе 7.

Отказ в обслуживании (DoS)

Реализация данной атаки направлена на нарушение работоспособности некоторой службы удаленного хоста, либо всей системы. Как правило, реализация предполагает посылку направленного «шторма запросов», переполнение очереди запросов, в силу чего удаленный ПК либо перезагружается, либо неспособен заниматься ничем, кроме обработки запросов. Примерами данных атак является SYN-Flooding, Ping of Death и пр.

Для защиты от данных атак необходимо использовать стойкие протоколы аутентификации, ограничивать доступа в сеть с использованием межсетевых экранов, применять системы обнаружения вторжений, разрабатывать адекватные политики безопасности, использовать для поддержки сервисов программные продукты, в которых устранены уязвимости, позволяющие выполнить подобные атаки.

В настоящее время большую актуальность представляет защита от распределенных DoS атак (DDoS), реализуемых путем заражения («зомбирования») множества ничего не подозревающих ПК, которые в заданный момент времени начинают посылать «шторм запросов» на объект атаки. В 2003 году таким образом был атакован сайт SCO Group.

Сканирование компьютерных сетей

Сетевое сканирование осуществляется злоумышленником на предварительной стадии атаки. Сканирование компьютерной сети позволяет получить злоумышленнику такую информацию, необходимую для дальнейшего взлома, как типы установленных ОС, открытые порты и связанные с ними сервисы, существующие уязвимости. Сам факт сетевого сканирования лишь говорит о реализации стадии, предваряющей атаку, и является важной информацией для сетевого администратора.

Для защиты от сетевого сканирования необходимо применять подходы, позволяющие скрыть внутреннюю структуру сети и идентифицировать факт сканирования, например, использовать межсетевые экраны, системы обнаружения вторжений.

Таким образом, для защиты от рассмотренных выше атак используют межсетевые экраны, виртуальные частные сети, стойкие протоколы аутентификации, системы обнаружения вторжений, а также анализ журналов безопасности (аудита) компьютерных систем. Ниже данные средства рассмотрены более подробно.

8.3. Ограничение доступа в сеть. Межсетевые экраны

Одна из важнейших задач, решаемая при защите компьютерных сетей, – ограничение доступа внешних пользователей к ресурсам внутренней сети организации, а также обеспечение безопасного доступа внутренних пользователей сети к ресурсам внешней. Это ограничение должно выполняться в соответствие с правилами, определяющими политику безопасности в сети организации.

Межсетевой экран (МЭ, firewall) – это система межсетевой защиты, позволяющая разделить общую сеть на две части или более и реализовать набор правил, определяющих условия прохождения сетевых пакетов через границу из одной части сети в другую [13].

МЭ пропускает через себя весь трафик, принимая для каждого из проходящих пакетов решение – пропускать его дальше или отбросить. Для этого на межсетевом экране задают набор правил фильтрации трафика.

Обычно межсетевые экраны защищают внутреннюю сеть организации от несанкционированного доступа из открытой сети INTERNET (рис. 10.1), однако, они могут использоваться и для ограничения доступа внутренних пользователей к различным подсетям внутри корпоративной сети предприятия. Таким образом, МЭ регламентирует использование ресурсов одних се-

тей пользователями других, для него, как правило, определены понятия «внутри» и «снаружи».

Решение о том, каким образом фильтровать пакеты, зависит от принятой в защищаемой сети политики безопасности, МЭ ее реализует. Как правило, с помощью МЭ ограничивается доступ к различным сетевым сервисам для различных сетевых адресов. Например, МЭ может запретить доступ по протоколам POP3 и SMTP для всех пользователей внутренней сети организации кроме почтового сервера, так чтобы пользователи были вынуждены забирать свою почту только с выделенного почтового сервера организации, на котором она проходит необходимые проверки.

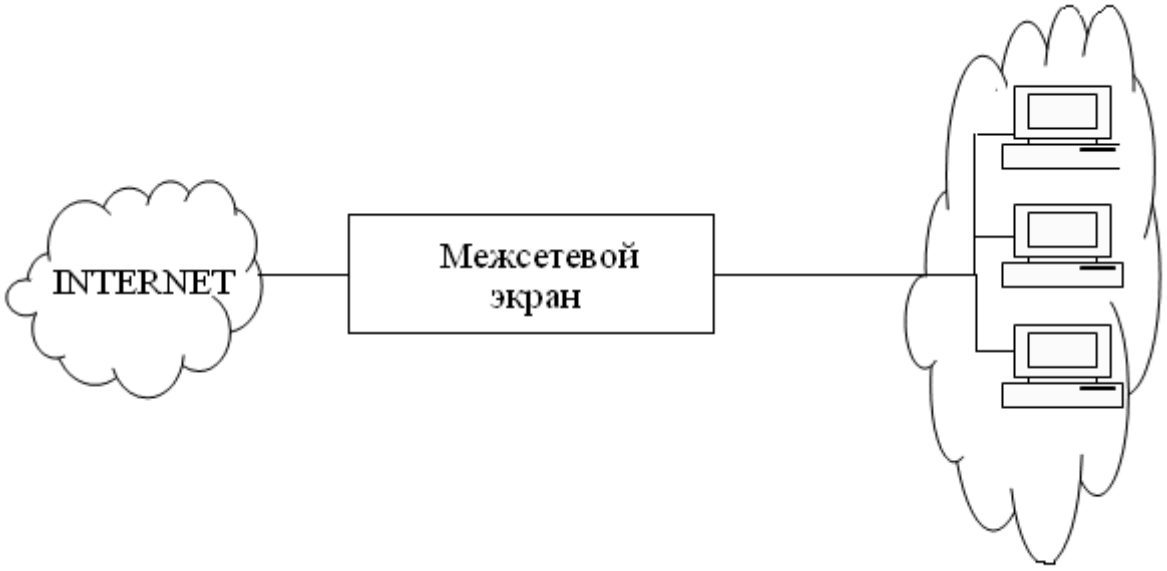


Рис. 10.1. Защита внутренней сети организации от несанкционированного доступа из сети INTERNET

Правила доступа к сетевым ресурсам, в соответствие с которыми конфигурируется МЭ, могут базироваться на одном из следующих принципов:

1. запрещать все, что не разрешено в явной форме;
2. разрешать все, что не запрещено в явной форме.

Реализация МЭ на основе первого принципа позволяет обеспечить более хорошую защищенность, но требует больших затрат и доставляет больше неудобств пользователям.

Различают следующие виды МЭ [13,35]:

1. фильтрующие маршрутизаторы (пакетные фильтры);
2. шлюзы сетевого уровня;
3. шлюзы прикладного уровня.

Фильтрующие маршрутизаторы (пакетные фильтры)

Данные МЭ осуществляют фильтрацию входящих в сеть и исходящих из сети пакетов на основе информации, содержащихся в их TCP и IP заголовках. Как правило, фильтрация осуществляется на основе следующих основных полей:

- IP адреса отправителя;
- IP адреса получателя;
- порта отправителя;
- порта получателя

Порты отправителя и получателя используются для идентификации сетевой службы, к которой производится обращение, например, FTP (21), TELNET (23) и т.д.

Пример набора правил фильтрации для такого МЭ представлен в таблице 10.1.

Табл. 10.1 Пример правил фильтрации

Тип	**Адрес отправителя**	**Адрес получателя**	**Порт отправителя**	**Порт получателя**	**Действие**
TCP	*	129.1.2.3	>1023	21	Разрешить
TCP	123.6.49.234	123.1.2.9	>1023	119	Разрешить

Основными достоинствами МЭ данного типа является невысокая их стоимость и скорость фильтрации.

Основные недостатки МЭ данного вида – не скрывают структуру внутренней сети, нестойкая процедура аутентификации по IP адресу, которую можно обмануть путем подмены IP адреса злоумышленником.

Шлюзы сетевого уровня

Использование подобных МЭ позволяет исключить прямое взаимодействие между хостами. Данные шлюзы принимают запросы доверенных клиентов, и после проверки допустимости сеанса связи устанавливают соединение с требуемым хостом. Такой МЭ выполняет роль посредника между соединяемыми хостами, не давая им взаимодействовать напрямую.

Данные МЭ выполняют также функцию трансляции адресов (NAT), скрывая внутреннюю структуру сети от внешних пользователей. Они выполняют преобразование внутренних IP-адресов сети в один «надежный» IP-адрес, ассоциируемый с МЭ. Внешние пользователи открытой сети «видят» только внешний IP-адрес шлюза.

Недостатком шлюзов сетевого уровня является невозможность фильтрации трафика «внутри службы».

Шлюз прикладного уровня

Данные МЭ позволяют не только пропускать либо не пропускать определенные службы, но и осуществлять фильтрацию трафика «внутри» таких служб, как TELNET, FTP, HTTP и т.д. Например, пользователю внутри FTP соединения может быть запрещено использовать команду put. Данные МЭ используют стойкие протоколы аутентификации пользователей, не позволяющие осуществить подмену доверенного источника, позволяют снизить вероятность взлома систем с использованием уязвимостей ПО.

Отметим, что в организации часто возникает потребность в создании в составе корпоративной сети нескольких сегментов с различными уровнями защищенности, например, свободных сегментов, сегментов с ограниченным доступом, закрытых сегментов. В этом случае могут понадобиться различные варианты установки МЭ. Рассмотрим основные схемы расстановки МЭ и реализуемые при этом функции по защите.

В простейших случаях, при необходимости защитить внутреннюю сеть организации от несанкционированного доступа внешних пользователей

INTERNET, используют схему, представленную на рис. 10.1, где МЭ используется как фильтрующий маршрутизатор.

Схема, представленная на рис. 10.2, позволяет организовать из видимых снаружи серверов отдельную сеть, с правилами доступа, отличающимися от правил доступа к ПК остальной части интрасети. Возможно ограничение доступа от пользователей INTERNET к серверам организации, пользователей интрасети к серверам организации.

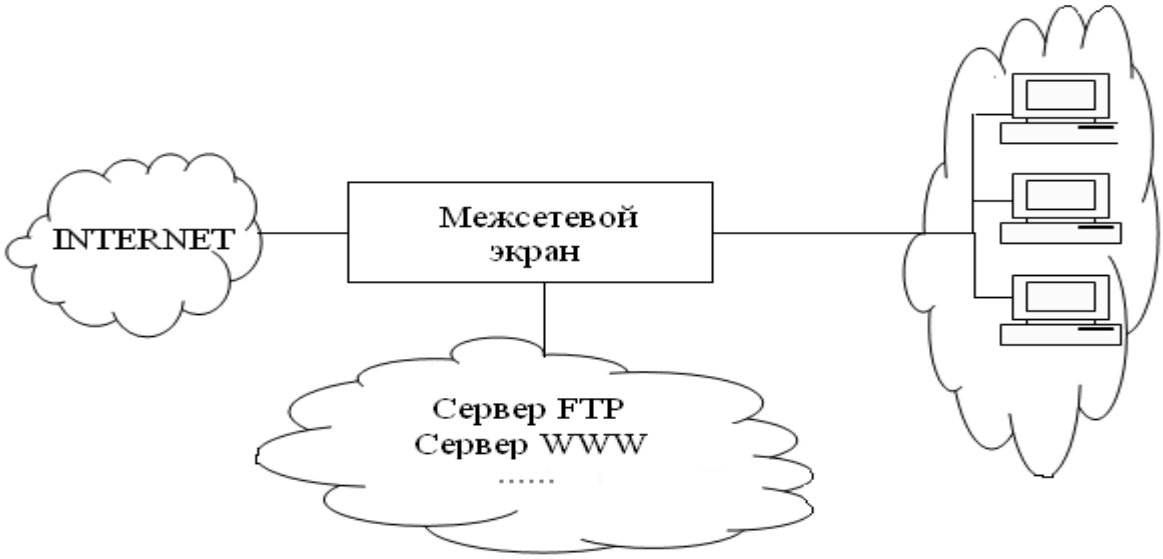


Рис. 10.2. Защита бастиона серверов

На рис. 10.3 представлен еще один вариант подключения МЭ – с выделением демилитаризованной зоны (DMZ). Организация DMZ предназначена для защиты хостов данной зоны от атак из INTERNET, а также от внутренних пользователей организации. В DMZ могут выноситься WEB, FTP SMTP, DNS серверы и пр.

Рис. 10.3. Схема подключения двух МЭ с введением демилитаризованной зоны

8.4. Виртуальные частные сети (VPN)

В настоящее время значительное число организаций имеют множество отделений, офисов, распределенных по различным городам внутри одной страны и даже по разным странам мира. Поэтому для организаций возникает насущная необходимость интеграции локальных сетей данных отделений в единую корпоративную сеть компании, в рамках которой сотрудники могли бы использовать все привычные для себя функции локальных сетей, не чувствовать себя отдаленными от сотрудников другого офиса, расположенного, быть может, на другом конце земного шара. Мобильные сотрудники данных организаций, перемещающиеся из страны в страну, должны иметь возможность доступа из любой точки земного шара к внутренней сети организации с помощью переносимых ПК.

Естественный вариант реализации такого объединения локальных сетей, мобильных пользователей в единую корпоративную сеть, видится с привлечением каналов открытой сети INTERNET. Основными задачами, которые должны быть решены при этом, являются [36]:

1. Аутентификация взаимодействующих сторон.
2. Криптографическая защита передаваемой информации.
3. Подтверждение подлинности и целостности доставленной информации.
4. Защита от повтора, задержки и удаления сообщений.
5. Защита от отрицания фактов отправления и приема сообщений.

Данные проблемы позволяют эффективно решить виртуальные частные сети (Virtual Private Network), к использованию которых все больше склоняются многие крупные компании.

Виртуальной частной сетью (VPN) называют объединение локальных сетей и отдельных компьютеров через открытую внешнюю среду передачи

информации в единую виртуальную сеть, обеспечивающую безопасность циркулирующих данных.

Защита информации при ее передаче по открытому каналу основана на построении *криптозащищенных туннелей* (*туннелей VPN*). Каждый из таких туннелей представляет собой виртуальное соединение, созданное в открытой сети, по которому передаются криптографически защищенные сообщения виртуальной сети. Пример возможной организации виртуальной частной сети представлен на рис. 11.4

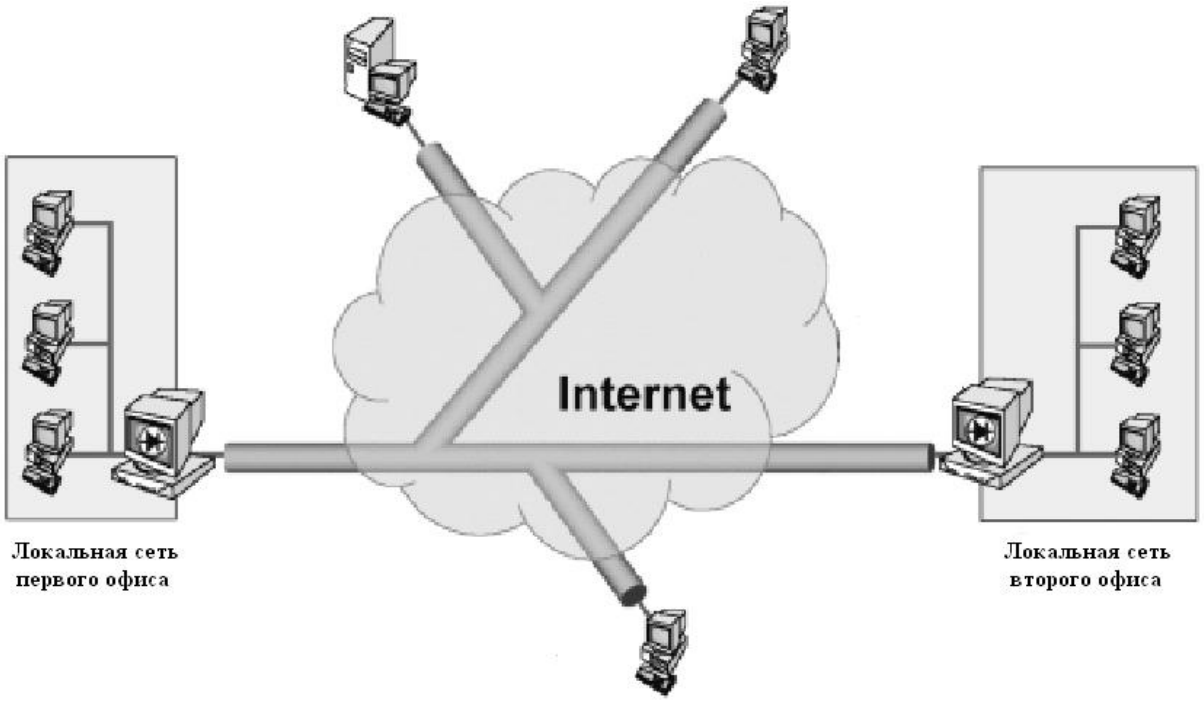


Рис. 10.4. Пример организации виртуальной частной сети

Известно несколько наиболее часто используемых способов образования защищенных виртуальных каналов [36].

1. Конечные точки защищенного туннеля совпадают с конечными точками защищаемого потока сообщений. Данный вариант является наилучшим с точки зрения безопасности. В этом случае обеспечивается полная защищенность канала вдоль всего пути следования пакетов сообщений. Однако, такой вариант ведет к децентрализации управления и избыточности ресурсных затрат.

2. Конечные точки защищенного туннеля совпадают с МЭ или пограничным маршрутизатором локальной сети. В данном случае поток сообще-

ний внутри локальной сети оказывается незащищенным, а все сообщения, выходящие из локальной сети, передаются по криптозащищенному туннелю.

3. Конечные точки – провайдеры INTERNET. В этом случае не защищаются каналы локальной сети и выделенные каналы связи, защищаются только каналы INTERNET.

Протоколы поддержки виртуальных частных сетей создаются на одном из трех уровней модели OSI – канальном, сетевом или сеансовом.

Канальному уровню соответствуют такие протоколы реализации VPN, как PPTP, L2F, L2TP. Сетевому уровню соответствуют протоколы IPSec, SKIP. Сеансовому уровню – SSL, SOCKS. Чем ниже уровень эталонной модели OSI, на котором реализуется защита, тем она прозрачнее для приложений и незаметнее для пользователей. Однако при снижении этого уровня уменьшается набор реализуемых услуг безопасности и становится труднее организация управления. Оптимальное соотношение между прозрачностью и качеством защиты достигается при формировании защищенных виртуальных каналов на сетевом уровне модели OSI.

Протокол SKIP

Протокол SKIP (Simple Key management for Internet Protocol) управляет ключами шифрования и обеспечивает прозрачную для приложения криптозащиту IP-пакетов на сетевом уровне модели OSI.

SKIP предусматривает самостоятельное формирование противоположными сторонами общего секретного ключа на основе ранее распределенных или переданных друг другу открытых ключей сторон. Выработка общего секретного ключа K_{AB} осуществляется в рамках протокола Диффи-Хеллмана.

Общий секретный ключ K_{AB} не используется непосредственно для шифрования трафика между узлами A и B. Вместо этого, для шифрования конкретного пакета или их небольшой группы передающая сторона вырабатывает случайный временный пакетный ключ K_p. Далее выполняются следующие действия:

1. Исходный IP пакет шифруется на пакетном ключе K_p и инкапсулируется в защищенный SKIP пакет.

2. Пакетный ключ K_p шифруется на общем секретном ключе K_{AB} и помещается в SKIP заголовок.

3. Полученный SKIP-пакет инкапсулируется в результирующий IP-пакет.

4. Для результирующего IP-пакета с помощью некой криптографической функции хэширования рассчитывается на пакетном ключе K_p имитовставка (для контроля целостности сообщения) и вставляется в зарезервированное поле SKIP-заголовка.

Применение для криптозащиты трафика не общего секретного ключа K_{AB}, а случайного пакетного ключа K_p, повышает безопасность защищенного туннеля. Это связано с тем, что долговременный секретный ключ K_{AB} не сможет быть скомпрометирован на основании анализа трафика, так как вероятный противник не будет иметь достаточного материала для проведения быстрого криптоанализа с целью раскрытия этого ключа. Защищенность обмена повышает также частая смена ключей шифрования, так как если пакетный ключ и будет скомпрометирован, то ущерб затронет лишь небольшую группу пакетов, зашифрованных по этому временному ключу.

Организация виртуальных частных сетей, основанных на протоколе SKIP, реализована в семействе продуктов VPN «ЗАСТАВА» компании ЭЛВИС+. Кроме этого, широко используемыми продуктами построения VPN являются «ТРОПА» компании Застава-JET, F-Secure VPN+ компании F-Secure Corporation, Check Point VPN-1/Firewall-1 и др.

8.5. Доменная архитектура в Windows NT. Служба Active Directory

Серьезной проблемой для организаций, содержащих сети больших масштабов, тысячи пользователей, множество серверов, является необходимость разработки и поддержки корпоративной политики безопасности в се-

ти. Для сетей такого масштаба поддержка отдельных независимых баз данных аутентификации становится практически неосуществимой. Для управления сетями Windows NT больших масштабов фирма Microsoft предлагает использовать многодоменную структуру и доверительные отношения между доменами.

Домен Windows NT представляет собой группу компьютеров сети, использующих общую модель обеспечения безопасности, а также имеющих единую базу данных SAM, содержащую информацию о пользователях и их группах. Использование доменных архитектур и служб каталогов позволяет осуществить централизованное хранение информации обо всей корпоративной сети. Администраторы создают для каждого пользователя одну учетную запись на контроллере домена и затем могут использовать эту запись для предоставления пользователю прав доступа к ресурсам, расположенным в сети.

Базы данных доменов Windows NT не поддерживают иерархической структуры и не могут быть распределены между несколькими серверами Windows NT. Эти обстоятельства существенно ограничивают максимальное количество объектов в домене. Однодоменная структура в Windows NT может быть реализована, если число пользователей измеряется одной - двумя сотнями. Для управления сетями Windows NT больших масштабов фирма Microsoft предлагает использовать многодоменную структуру и доверительные отношения между доменами. Доверительные отношения между доменами обеспечивают междоменное администрирование, позволяя предоставлять пользователям из одного домена доступ к ресурсам другого домена.

Реализация доменной архитектуры в Windows NT имеет ряд серьезных недостатков, которые Microsoft попыталась преодолеть в Windows 2000 за счет введения службы Active Directory.

Active Directory (AD) – это объектно-ориентированная, иерархическая, распределенная система базы данных службы каталогов, которая обеспечивает централизованное хранение информации об оборудовании, программ-

ном обеспечении и человеческих ресурсах всей корпоративной сети [37].

Active Directory включает в себя следующие компоненты:

1. Объекты.
2. Домены.
3. Организационные единицы.
4. Деревья.
5. Леса.

Под объектом в AD понимают определяющие ресурс набор атрибутов (имя, собственные права доступа, права доступа к объекту, дополнительная информация об объекте и т.д.). Пользователи сети представлены объектами в службе каталогов. Администраторы могут применять эти объекты для предоставления пользователям доступа к ресурсам корпоративной сети. Ресурсы также представляются в виде объектов. Группы доменов могут быть объединены в дерево, группы деревьев – в лес. В Active Directory права и разрешения объектов распространяются вниз по дереву.

Организационные единицы (контейнеры) являются группировкой других объектов. Контейнеры вводят для упрощения администрирования корпоративной политикой безопасности. Назначение контейнеру некоторого права или разрешения автоматически распространяет его на все объекты контейнера. Для группировки пользователей, компьютеров используют особые контейнеры – группы (Group).

Дерево в AD - это иерархия нескольких доменов (рис. 10.5), лес – множество деревьев.

В AD используется система аутентификации Kerberos, основанная на протоколе Нидхема-Шредера выдачи мандатов (билетов), предъявление которых позволяет получить доступ субъекта к некому сетевому ресурсу.

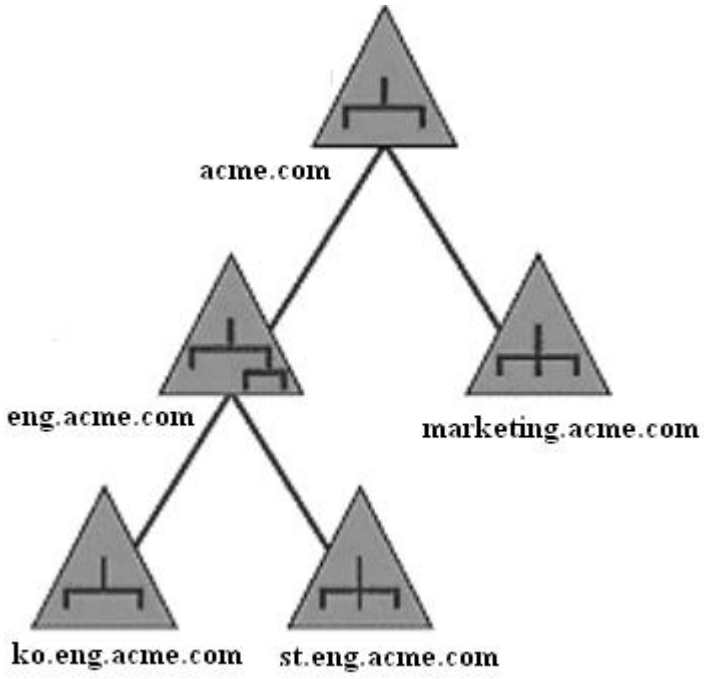


Рис. 10.5. Дерево AD

8.6. Централизованный контроль удаленного доступа. Серверы аутентификации

В части 7 настоящего учебного пособия рассмотрен ряд существующих протоколов безопасной удаленной аутентификации пользователей в компьютерных сетях.

В случае, когда локальная сеть является небольшой, для управления удаленными соединениями с этой сетью, как правило, бывает достаточно одного сервера удаленного доступа. Однако если локальная сеть объединяет достаточно большие сегменты и число удаленных пользователей существенно увеличивается, то одного сервера удаленного доступа становится недостаточно. В этом случае, как правило, вводят единый *сервер аутентификации*, для централизованного контроля удаленного доступа. В его функции входят проверка подлинности удаленных пользователей, определение их полномочий, а также фиксация и накопление регистрационной информации, связанной с удаленным доступом (рис. 11.6).

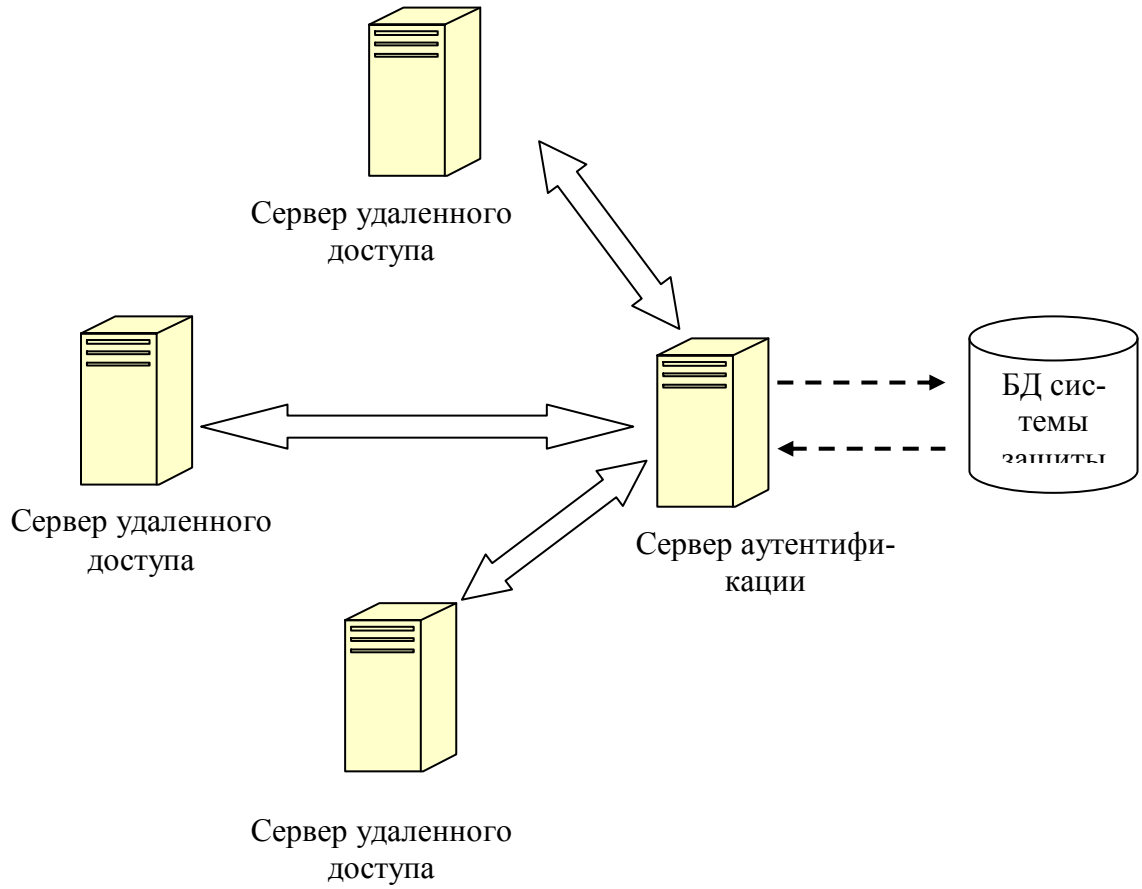


Рис. 10.6. Схема централизованного контроля удаленного доступа

Сервер аутентификации исполняет роль посредника во взаимодействии между серверами удаленного доступа и центральной базой данных системы защиты. Централизованный контроль удаленного доступа к ресурсам сети с помощью сервера аутентификации выполняют на основе специализированных протоколов. Эти протоколы позволяют объединить серверы удаленного доступа и сервер аутентификации в единую подсистему, выполняющую все функции контроля удаленных соединений на основе взаимодействия с центральной базой данных системы защиты. Сервер аутентификации в данном случае создает единую точку наблюдения и проверки всех удаленных пользователей и контролирует доступ к компьютерным ресурсам в соответствие с установленными правилами. Наиболее известными протоколами централизованного контроля удаленного доступа являются протоколы TACACS (Terminal Access Controller Access Control System) и RADIUS (Remote Authentication Dial-In User Service) [36].

В системах, основанных на протоколах TACACS и RADIUS, администратор может управлять базой данных идентификаторов и паролей пользователей, предоставлять им привилегии доступа и вести учет обращений к системным ресурсам. В рамках данных протоколов сервер аутентификации может иметь как собственную базу данных системы защиты, так использовать и службы каталогов – Novell Directory Services (NDS), Active Directory (AD).

8.7. Вопросы для самоконтроля

1. Перечислите наиболее распространенные угрозы безопасности в INTERNET и INTRANET. Дайте их характеристику.
2. Перечислите основные причины уязвимости сети INTERNET.
3. Перечислите основные классы удаленных атак. Дайте их характеристику.
4. Дайте понятие межсетевого экрана. В чем заключаются его функции?
5. Перечислите основные виды межсетевых экранов. Охарактеризуйте функции МЭ каждого вида, их возможности и недостатки.
6. Дайте понятие виртуальной частной сети. Какие основные задачи решают виртуальные частные сети?
7. Перечислите наиболее часто используемые способы образования защищенных виртуальных каналов. Охарактеризуйте достоинства и недостатки каждого из способов.
8. Перечислите несколько протоколов реализации VPN.
9. Охарактеризуйте суть доменной архитектуры Windows NT и службы каталогов Active Directory.

9. Защита программного обеспечения с помощью электронных ключей HASP

9.1 Электронные ключи серии HASP 4

Недостатком таких характеристик среды, как серийный номер, ключевой файл, ключевой носитель, конфигурация аппаратуры, информация в секретном секторе диска и т.д. является возможность сравнительно легкого раскрытия их злоумышленником, взлом посредством эмуляции характеристик среды. Для устранения этого недостатка данные характеристики должны быть вынесены во внешнее, максимально защищенное от несанкционированного доступа аппаратное устройство, затрудняющее свою эмуляцию и дублирование. Данную возможность предоставляют специализированные аппаратные средства, называемые электронными ключами. Одним из наиболее распространенных типов электронных ключей являются электронные ключи HASP.

Электронные ключи HASP являются разработкой фирмы Aladdin и представляют собой современное аппаратное средство защиты ПО от несанкционированного использования, позволяющее предотвратить несанкционированный доступ к защищаемым программам и их исполнение. Внешний вид электронных ключей HASP представлен на рисунке 9.1.

Рис. 9.1. Внешний вид электронных ключей HASP

Базовой основой ключей HASP является специализированная заказная микросхема (ASIC – Application Specific Integrated Circuit), имеющая уникальный для каждого ключа алгоритм работы. В процессе своего исполнения защищённая программа опрашивает подключенный к ПК HASP. Если HASP возвращает правильные ответы, работает по требуемому алгоритму и обла-

дает требуемыми эталонными характеристиками, то программа выполняется нормально. В противном случае реализуется определенная реакция на несанкционированное использование: запуск в демонстрационном режиме, блокировка отдельных функций, отказ в запуске и т.д.

Семейство ключей HASP включает в себя следующие модели:

1. HASP4 Standard.
2. MemoHASP.
3. TimeHASP.
4. NetHASP.

HASP4 Standard

Данный тип ключей является простейшей модификацией электронных ключей HASP. Основным элементом их защиты является аппаратно реализованная в них на ASIC микросхеме функция шифрования и связанная с ней функция отклика f(x), принимающая на вход 32-битный аргумент и формирующая на выходе четыре 32-битных значения.

Для предотвращения несанкционированного использования программного обеспечения система защиты может осуществлять:

- проверку наличия HASP Standard;
- проверку соответствия выходов, формируемых функцией отклика *f(x)* для различных значений *x*, эталонным значениям;
- использовать функцию шифрования электронного ключа для шифрования и дешифрования своего исполняемого кода, используемых данных и т.д.

С любым электронным ключом HASP связана его серия, идентифицирующая защищаемое программное обеспечение (например, IXGGR, RAOMG). Каждая из серий обладает своей уникальной функцией шифрования данных, что вносит уникальность в алгоритм защиты каждого защищаемого продукта.

Данный тип ключей HASP является наиболее хорошим решением для защиты недорогих программ. Стоимость данных ключей составляет порядка 14$.

MemoHASP

Ключи MemoHASP имеют в своём составе все компоненты HASP Standard. Базовым отличием данного типа ключей от HASP4 Standard является наличие встроенной в них энергонезависимой памяти (EEPROM), доступной для чтения и записи во время выполнения защищенной программы. Каждому из данных типов ключей также присваивается свой уникальный 32-битовый идентификационный номер ID, который позволяет идентифицировать конкретного пользователя продукта.

Модификации данных ключей связаны с объемом доступной энергонезависимой памяти.

HASP4 M1 – 112 байт EEPROM, возможность одновременной защиты до 16 программ, ориентировочная стоимость 20$.

HASP4 M4 – 496 байт EEPROM, возможность одновременной защиты до 112 программ, ориентировочная стоимость 29$.

Кроме подходов к защите ПО, свойственных HASP Standard, с помощью MemoHASP могут быть реализованы, например, следующие подходы:

1. Хранение в энергонезависимой памяти MemoHASP конфиденциальной информации – ключей шифрования, части исполняемого кода и т.д.
2. Хранение в энергонезависимой памяти информации о модулях защищённого программного обеспечения, к которым пользователь имеет доступ и о тех, к которым не имеет (в зависимости от заплаченной суммы за приобретение программы).
3. Хранение в энергонезависимой памяти информации о количестве запусков программы, либо об оставшемся количестве запусков. Данный подход актуален при создании демонстрационных версий программ, работа с которыми ограничена количеством запусков.

TimeHASP

Кроме функций MemoHASP, данные ключи обладают встроенными часами реального времени с автономным питанием от литиевой батарейки (отражающие время и дату). Используя часы реального времени, производитель может защищать свое программное обеспечение по времени использования и на основании этого строить гибкую маркетинговую политику – сдачу программ в аренду, лизинг ПО и периодический сбор платы за его использование и т.д. Стоимость данных ключей составляет порядка 33$.

NetHASP

Данные ключи имеют в своем составе все компоненты MemoHASP и предназначены для защиты ПО в сетевых средах. Один ключ, установленный на любом компьютере сети, способен защитить ПО от тиражирования, а также ограничить количество рабочих мест (лицензий), на которых ПО используется одновременно. Ключ может работать на выделенном либо невыделенном сервере, либо любой станции. Он поддерживает различные протоколы – IPX/SPX, NetBIOS, NetBEUI, TCP/IP.

Существует несколько моделей ключей HASP4 Net, которые позволяют лицензировать программы для 5, 10, 20, 50, 100 и для неограниченного количества пользователей. Все они могут защищать до 112 различных программ (или модулей программы). Номер модели определяет максимальное число рабочих мест для любой из этих программ. Например, HASP4 Net-5 защищает до 112 программ, и каждая программа может лицензироваться на количество рабочих мест от нуля до пяти (например, три).

Работа с NetHASP осуществляется через менеджер лицензий.

Менеджер лицензий HASP4 Net – это программа-посредник, обеспечивающая связь защищённых приложений с сетевым ключом. При этом сам ключ может стоять на любом компьютере в сети – выделенном или невыделенном файл-сервере или на любой станции. Чтобы использовать в сети сервис NetHASP, нужно выбрать рабочую станцию, присоединить к ней ключ NetHASP и загрузить на ней менеджер лицензий NetHASP.

Менеджер лицензий NetHASP способен обслуживать до 250 защищенных программ, работающих в сети, одновременно обслуживая их связь с несколькими ключами NetHASP.

Когда защищённое приложение стартует на компьютере, подключенном к сети, оно обращается к менеджеру лицензий и запрашивает разрешение выполняться дальше (NetHASP LOGIN), для отключения от сервера используется процедура (NetHASP LOGOUT). При этом менеджер лицензий проверяет соблюдение ряда условий:

- наличие необходимого ключа HASP4 Net на машине, где он загружен;
- наличие лицензии на выполнение данной программы;
- лимит рабочих мест для этой программы на данный момент не исчерпан.

Если результаты всех проверок положительны, менеджер лицензий даёт запросившей его программе разрешение на выполнение и заносит данные о ней в журнал доступа.

Менеджер лицензий ведёт журнал доступа, в котором отмечаются все подключившиеся приложения, выполнившие LOGIN. В журнале содержатся сведения о том, какая программа и на какой рабочей станции была запущена. Эти данные сохраняются в журнале до тех пор, пока программа не выполнит отключение (LOGOUT). При помощи журнала доступа менеджер лицензий отслеживает количество машин, на которых одновременно выполняется защищённая программа, и не допускает превышения максимального их числа, заданного разработчиком программы.

Цена на ключи NetHASP в зависимости от модели 37,5$ – 260$.

Общая схема взаимодействия с электронным ключом HASP

Рассмотрим общую схему функционирования HASP (см. рис. 9.2):

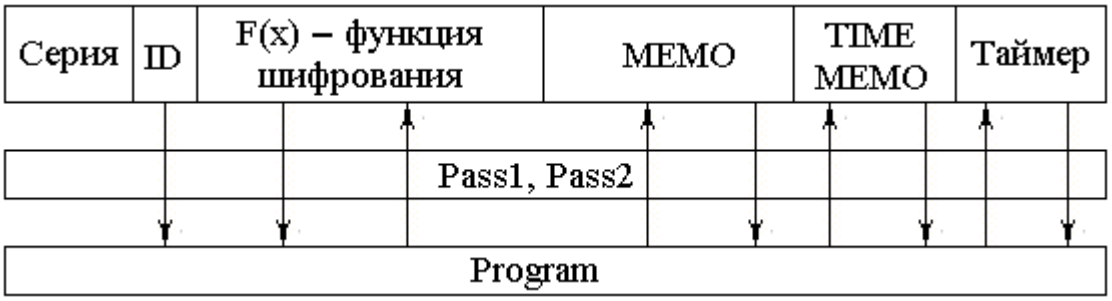


Рис. 9.2. Общая схема функционирования HASP

На данной схеме представлены основные элементы электронных ключей HASP и особенности взаимодействия с ними. Направление стрелок указывает направление потоков информации от защищённой программы к элементам HASP.

Доступ к функциям электронного ключа HASP возможен только при указании кодов доступа. Коды доступа представляют собой два целых 16-битовых числа. Они уникальны для каждой из серий HASP. Внутри серии данные коды определены однозначно. Не указав код, пользователь не сможет обеспечить себе доступ к функциям HASP. Таким образом, выполняется противодействие эмуляции и копированию HASP.

В таблице 9.1 представлены основные элементы электронных ключей HASP и типы HASP, в которых данные элементы присутствуют.

Таб. 9.1. Элементы электронных ключей HASP

Элемент	Описание	Типы ключей
Серия	Серия, присваиваемая каждому из производителей защищаемого продукта, либо каждому из продуктов. Различные серии ключей обладают различными функциями шифрования и различными кодами доступа. Серия не может быть программным путем прочитана, либо записана в HASP	HASP4 Standard, MemoHASP, TimeHASP, NetHASP
Идентификационный номер ID	Номер, уникально идентифицирующий каждый из выпущенных ключей HASP. Прошивается единожды в заводских условиях. Может быть прочитан программным путём	MemoHASP, TimeHASP, NetHASP
Функция шифрования	Аппаратно реализованная функция в HASP, позволяющая шифровать и дешифровывать информацию	HASP4 Standard, MemoHASP, TimeHASP, NetHASP

Энергонезависимая память MEMO	Защищенная по доступу память для долговременного хранения конфиденциальной информации. В нее могут быть программным путем записана информация, либо прочитана из неё	MemoHASP, TimeHASP, NetHASP
Энергонезависимая память TIME MEMO	Защищённая по доступу память для долговременного хранения информации, используемой TimeHASP при защите программного обеспечения по времени работы	TimeHASP
Таймер	Таймер, используемый для защиты программ по времени своей работы. Ограничение по доступу к таймеру с помощью кодов доступа HASP не позволяет злоумышленнику несанкционированно использовать программу путём корректировки таймера	TimeHASP

Существует два способа внедрения защитных механизмов в программное обеспечение с помощью электронных ключей HASP.

1. HASP API (с помощью API функций).
2. Пакетный режим (HASP Envelope).

Первый способ защиты используется для встраивания защитных механизмов в исходные тексты. Фирма Aladdin предлагает набор функций для взаимодействия с HASP практически для всех платформ. Сам код API функций защищён и зашифрован. С помощью данных функций можно обратиться к HASP из любой точки программы и на основании проведённых проверок предпринять необходимые шаги. Использование API функций позволяет программировать разработчику любую реакцию на несанкционированный запуск.

Второй способ служит непосредственно для защиты исполняемых файлов. Исполняемый файл заключается в защитную программную оболочку, кодирующую файл, и обладающую такими свойствами, как распознавание ключа и антиотладка.

При защите ПО более предпочтительно одновременное использование как первого, так и второго способа.

Система полного управления доступом в HASP (FAS)

Система полного управления доступом (Full Authorization System) в HASP позволяет производителю защитить несколько программ одним и тем

же ключом HASP, определив условия, при которых может работать конкретная программа. Возможно ограничение использования программ путём:

1. Задания количества запусков программы (MemoHASP, NetHASP). Эта возможность полезна при разработке демо-версий защищённых программ.
2. Задания допустимого срока работы программы (TimeHASP). Эта возможность полезна при лизинге и аренде программ.
3. Задания числа станций, на которых программа может работать одновременно (NetHASP).

В случае защиты с помощью подсистемы FAS, защищенная программа осуществляет несколько проверок.

1. В первую очередь, проверяется, присоединён ли к компьютеру соответствующий ключ.
2. Если ответ положителен, память HASP проверяется на предмет того, занесена ли программа в список разрешённых к работе программ.
3. При положительном ответе выполняется серия проверок в зависимости от используемой модели HASP.

В случае MemoHASP, память ключа опрашивается на предмет превышения допустимого количества запусков программ. С каждым новым запуском число допустимых запусков уменьшается на единицу. Как только это число станет равным нулю, работа программ прекращается, и выдаётся сообщение об ошибке.

В случае TimeHASP, в памяти ключа опрашивается список допустимых сроков, и результат сравнивается с реальным временем на таймере ключа.

HASP API

Все функции API HASP (кроме NetHASP) вызываются через единую функцию hasp(), которая имеет следующий синтаксис

Hasp(Service, SeedCode, LptNum, Pass1, Pass2, Par1, Par2, Par3, Par4)

Для NetHASP та же функция имеет следующий формат

Hasp(Service, SeedCode, ProgNum, Pass1, Pass2, Par1, Par2, Par3, Par4)

Здесь

Service – номер вызываемой функции (и соответствующей ей операции).

LptNum – номер параллельного порта, к которому подключен HASP (если 0, то драйвер его ищет автоматически, 1 – LPT1, 2 – LPT2, 3 – LPT3, 201-255 – определённый ключ HASP для порта USB).

SeedCode – значение, посылаемое в функцию отклика f(x).

Pass, Pass2 – пароль для HASP. Он должен быть указан при доступе ко всем функциям, кроме функции проверки наличия HASP.

Par1, Par2, Par3, Par4 – параметры, через которые передаются значения в HASP и через которые значения возвращаются.

Система удаленного обновления (RUS)

Система удаленного обновления (Remote Update System – RUS) представляет собой утилиту, позволяющую безопасным образом удаленно обновить содержимое ключей HASP у пользователя без необходимости раскрытия паролей. Эта возможность позволяет отказаться от необходимости отправки разработчиком нового ключа HASP, когда покупатель желает обновить его содержимое.

Использование RUS позволяет, например, принять разработчиком ПО оплату от покупателя, после чего удаленно разблокировать запрет использования определенного модуля. Это позволяет использовать так называемую концепцию многоуровневого лицензирования.

Технология применения утилиты RUS включает в себя 2 этапа:

1. Создание утилиты RUS.
2. Обновление памяти ключей у пользователя.

С помощью утилиты RUS продавец создает две утилиты – утилиту продавца и утилиту пользователя.

Утилита продавца остаётся у разработчика, а утилита пользователя передаётся клиенту. Для обновления содержимого ключей HASP разработчик и пользователь каждый раз используют соответствующие утилиты.

Чтобы обновить память ключа HASP выполняются следующие действия.

1. Покупатель использует утилиту Пользователя для нахождения идентификатора своего ключа, нуждающегося в обновлении, а затем эту информацию передает разработчику.
2. Разработчик вводит идентификатор и обновляемые данные в утилиту Продавца.
3. Разработчик генерирует пароли RUS в утилите Продавца и передаёт их пользователю.
4. Пользователь вводит эти пароли в утилиту Пользователя и обновляет память своего ключа HASP.

Процесс обновления памяти ключей пользователя полностью защищен. Все данные обмена шифруются случайным образом.

Модель защиты структурным кодом (PCS)

Защита структурным кодом (Pattern Code Security – PCS) является средством, значительно повышающим защищенность приложения, защищаемого с помощью электронных ключей HASP.

Защита структурным кодом реализуется в процессе защиты с помощью HASP API. Использование PCS возможно лишь при наличии доступа к исходным текстам защищаемого приложения.

Защита структурным кодом осуществляет последовательность скрытых вызовов процедуры hasp(), не включая эти вызовы в исходный код явно. После каждого вызова процедуры hasp() происходит переключение на следующий скрытый вызов. Если вызов hasp() вдруг удален из защищенной программы, скрытые вызовы не выполняются, а это означает, что кто-то вмешался в работу программы. Тем самым PCS не дает удалить либо «заклеить» обращения к процедуре hasp().

Можно определить до 25 шаблонов в исходном коде. Каждый раз, когда вызывается процедура hasp(), переключается столько скрытых вызовов, сколько определено шаблонов.

Шаблон – специальная статическая структура данных, определенная в приложении. Шаблон включает в себя сигнатуру, номер функции hasp() и параметры, необходимые для работы данной функции.

Каждый вызов процедуры hasp() автоматически обновляет все шаблоны в коде программы, их после выхода из функции можно проверить. Если динамически изменять в шаблонах значения входных переменных, то автоматически будут меняться и выходные. Аналогично, можно изменять и значения сервисов.

Преимущества использования PCS:

1. Скрывание обращения к HASP.
2. Трассировка вызовов hasp() для шаблонов практически невозможна, так как их нет в исходном коде.
3. Легче обнаруживается вмешательство извне (если процедуру отключили). Если вызов hasp() будет удалён, то шаблоны не обновятся, а это значит – кто-то вмешался в работу приложения.
4. Препятствует эмуляции процедуре hasp().

Рекомендации по наиболее надежной защите с помощью HASP

Электронные ключи HASP являются достаточно надёжным аппаратным средством защиты программ. Тем не менее, система защиты HASP надежна лишь настолько, насколько разработчик сделает её таковой.

Так как аппаратную часть HASP подделать в принципе невозможно, то все атаки будут осуществляться на программную часть – на трассировку кода и отключение защитных механизмов. Существует два способа атаки на защищенное приложение:

- заклеивание вызовов к защищающей приложение процедуре;
- заклеивание программы производителя ключа.

При попытке заклеить вызовы к защищающей приложение процедуре, взломщику придётся так изменить защищённое приложение, чтобы оно не посылало вызовов ключу, не проверяло бы результата вызовов, не реагировало бы указанными в приложении способами на результаты проверки. Этот

способ применим, если взломщик увидит, что защита реализована не очень хитрым способом.

При попытке заклеить программы производителя ключа, взломщику придётся изменить процедуры, отвечающие за связь с ключом (сами API). А когда эти процедуры будут изменены, они будут возвращать ожидаемый результат вызова, даже если надлежащий ключ и не подсоединён.

Для более надежной защиты ПО от несанкционированного использования с помощью HASP рекомендуется использовать следующие приёмы.

1. Использовать одновременно методы защиты с помощью оболочки и с помощью API. Они дополняют и усиливают друг друга.
2. Использовать больше вызовов hasp() и шаблонов PCS. Это создаст большие проблемы для взломщика в понимании схемы защиты и атаках на нее. Необходимо как можно больше рассеивать данные вызовы по всему приложению, чтобы затруднить анализ.
3. Шифровать внутренние и внешние данные защищаемого приложения. Дешифровку проводить на ключе HASP. В данном случае взломщику нужно будет не только взломать приложение, но и дешифровать данные. Нет необходимости шифровать все используемые приложением данные, но некоторые ключевые данные можно зашифровать. Объектом шифрования может быть все то, что оказывает влияние на основные функции приложения.
4. Избегать повторяющихся схем. Схему, которая повторяется в защищаемом коде легко обнаружить и трассировать. Как только взломщик поймет схему защиты, для него станет ясно, на что обратить внимание, что облегчит ему работу по снятию защиты.
5. Разделять в коде программы шаги вызова процедуры hasp(), анализа ответных значений, возвращенных данной процедурой, и реакцию программы на результат анализа. В данном случае их хуже трассировать, нежели последовательные шаги.

Например, можно проверять наличие ключа, когда пользователь щелкает мышью по определенной опции меню, после этого при сбое дать ему немного поработать, а затем выдать сообщение об ошибке. Это позволит скрыть действительное место проверки HASP.

1. Использование функционирования программы в качестве ответа на отсутствие HASP. Можно использовать целый ряд реакций на неприсоединение нужного ключа. Наиболее простое – вывод сообщения «HASP not FOUND», однако это подсказывает, что делалась проверка ключа. Лучше запрограммировать другую реакцию, например, отключить клавиатуру. После подсоединения ключа клавиатура включается.
2. Использовать HASP-зависимые данные. Если осуществляется проверка значений, внесённых в HASP, путём их сравнения с эталонными значениями, то это даёт дополнительную информацию для атаки. Необходимо считывать и использовать эти данные в приложении, не проверяя их корректность напрямую. Например, можно хранить в энергонезависимой памяти HASP метку перехода, считывать её и переходить по этой метке.

9.2 Электронные ключи серии HASP HL

Программное обеспечение является интеллектуальной собственностью. Чтобы сохранить и приумножить доходы, оно должно быть защищено от незаконного использования.

Война против нелегального использования программного обеспечения идёт ровно столько, сколько существует производство ПО. Объёмы продаж пиратских копий продолжают увеличиваться, постоянно сокращая доходы разработчиков. Последствия такого «теневого» бизнеса становятся всё более разрушительными. Уменьшается количество новых программ, меньше инвестиций вкладывается в маркетинг и в развитие каналов распространения. Единственный способ остановить пиратов – защитить свое ПО от несанкционированного использования.

Главный вопрос заключается в том, как можно получить реальную прибыль в мире, где пираты ежегодно крадут около 29 млрд. долларов в год (рис. 9.3)?

Рис. 9.3. Уровень пиратства в мире

HASP HL - решение для защиты ПО

HASP HL – новое поколение аппаратно-программных средств класса Software Digital Rights Management (система управления электронными правами на ПО).

Инструментарий HASP HL предназначен для разработчиков и издателей программного обеспечения, электронного контента и предоставляет им богатые возможности по защите программ и данных от несанкционированного использования и нелегального распространения, а также по реализации гибкой ценовой политики для различных моделей продаж ПО.

HASP HL позволяет:

- Повысить уровень продаж и увеличить доходы от реализации ПО
- Защитить свою интеллектуальную собственность

- Защитить разработанное и распространяемое программное обеспечение
- Управлять лицензированием ПО
- Реализовать различные модели продаж защищенного ПО

HASP HL включает линейку аппаратных средств (USB-ключей) и набор программного обеспечения (утилит, API) для построения защиты программ и данных, а также организации простой и удобной . дистрибуции защищенного ПО.

HASP HL – это не просто защита. Система лицензирования HASP HL позволяет совершенно по-другому взглянуть на продажи программного обеспечения, выйти на принципиально новые рынки сбыта, организовать продажи через дилеров, получить гораздо большую прибыль и выиграть в конкурентной борьбе.

Не прикладывая особых усилий, можно устанавливать и менять политику лицензирования буквально на лету, идя в ногу с динамически развивающимся рынком. Причем изменения моделей лицензирования производятся не силами разработчиков, которые, как правило, далеки от бизнеса и его потребностей, а с помощью менеджеров по продукту.

Многие разработчики программного обеспечения даже не задумываются о некоторых схемах продаж, считая их трудно реализуемыми. HASP HL позволяет быстро внедрить такие схемы.

Защита и лицензирование с помощью HASP HL

В системе HASP HL процессы построения защиты и установки лицензионных ограничений разделены.

Разработчики устанавливают защиту, а затем менеджеры независимо от них могут определять для каждого модуля, компоненты или функции ограничения по количеству запусков, времени использования или количеству одновременно работающих пользователей.

Таким образом, менеджеры становятся независимыми от разработчиков, у них появляется возможность оперативно реагировать на изменения рынка, меняя модели продаж без внесения изменений в защиту.

При этом все изменения в политике лицензирования распространяемого ПО могут производиться удаленно, например, через Интернет. Для этого достаточно лишь обновить память ключа HASP HL. Все обновления памяти подписываются с помощью ЭЦП (RSA/1024), что исключает возможность внесения несанкционированных изменений.

Построение защиты с помощью HASP HL

HASP HL предоставляет разработчикам два метода защиты:

- **HASP HL Envelope** – быстрый и простой метод автоматической защиты уже готовых приложений.
- **HASP HL API** – набор функций, встраиваемых в приложение, для создания мощной и гибкой системы защиты, разработки собственной системы лицензирования и использования дополнительных методов и схем защиты.

Автоматическая защита HASP HL Envelope

Для защиты готовых приложений без вмешательства в исходный код используется специальная утилита HASP HL Envelope. Утилита обрабатывает исполняемый файл (.exe или .dll), шифрует его, встраивает обращения к ключу HASP HL и мощные антиотладочные и антитрассировочные механизмы.

Данный метод позволяет за считанные минуты построить мощную и надежную защиту приложения. Для построения защиты при помощи HASP HL Envelope не требуется каких-либо специальных знаний в области информационных технологий. Использование утилиты настолько просто, что с этим справится любой пользователь (см. рис. 9.4).

Рис. 9.4. Иллюстрация работы HASP HL Envelope

Защита с использованием функций API

HASP API представляет собой мощнейший механизм защиты. Принцип работы заключается в том, что программа вызывает API функции (единые для всех моделей HASP) обращения к ключу, маскируя результаты в теле программы или в файлах .dll. Чем более сложным являются вызовы API, тем надежней будет обеспечиваемая HASP защита.

В любой момент работы программы с помощью API можно:

- Проверить наличие нужного ключа и предпринять определённые шаги при его отсутствии (перейти в демонстрационный режим, либо прекратить работу).
- Считать данные из памяти ключа или записать их в память.
- Зашифровать или расшифровать данные.
- Проверить уникальный номер ключа и другие параметры.

Примеры использования HASP HL API для большинства популярных языков программирования (C, C++, C#, Java, Delphi, VB и т. д.) поставляются в составе Комплекта разработчика HASP HL.

Кроме того, для упрощения и ускорения процесса построения защиты с помощью HASP HL API используется специальная утилита – генератор исходных кодов HASP HL ToolBox.

С помощью ToolBox разработчик может сгенерировать обращения к ключам HASP HL и быстро, через буфер обмена, вставить их в исходный код защищаемого приложения.

HASP HL ToolBox поддерживает все основные интерфейсы языков программирования: C, C++, C#, VisualBasic.

HASP HL Net. Защита и система лицензирования ПО в сетях

Для работы в сетях выпускаются специальные сетевые ключи HASP HL Net. Единственный ключ HASP HL Net, подключенный к любому компьютеру в сети, обеспечивает гибкую защиту ПО и управление лицензиями (рис. 9.5):

- Защищает ПО от несанкционированного использования
- Ограничивает число пользователей, одновременно работающих с защищенной сетевой программой
- Обеспечивает надежное хранение многочисленных лицензий, позволяя организации внедрять новые модели продаж и ценообразования

Рис. 9.5. Система лицензирования ПО в сетях

HASP HL Net содержит Менеджер Лицензий и инструментарий мониторинга для ОС Windows, Linux и Mac. Это дает системным администраторам необходимую гибкость для отслеживания использования защищенного приложения в сетевой среде.

9.3. Вопросы для самоконтроля

1. С помощью какого из типов электронных ключей семейств HASP возможно ограничить общее запусков ПО (например, ПО будет запускаться только 50 раз)?

2. Что такое ключи HASP HL и какие возможности они предоставляют?

3. С помощью какого типа из электронных ключей семейств HASP возможно ограничить работу ПО сроком на 1 год?

4. Назовите преимущества HASP HL.

5. Какие из следующих типов электронных ключей семейства HASP обладают энергонезависимой памятью?

6. Какие из следующих типов электронных ключей семейства HASP обладают функцией шифрования?

7. Назовите возможности HASP HL.

10. Руководящие документы России

В 1992 году Государственной технической комиссией России (ФСТЭК – Федеральной службой по техническому экспорту и контролю) были разработаны и опубликованы пять руководящих документов, посвященных вопросам защиты информации в системах ее обработки.

1. Защита от несанкционированного доступа к информации. Термины и определения.

2. Концепция защиты СВТ и АС от НСД к информации.

3. Автоматизированные системы. Защита от несанкционированного доступа к информации. Классификация автоматизированных систем и требования по защите информации.

4. Средства вычислительной техники. Защита от несанкционированного доступа к информации. Показатели защищенности от НСД к информации.

5. Временное положение по организации разработки, изготовления и эксплуатации программных и технических средств защиты информации от НСД в автоматизированных системах и средствах вычислительной техники.

Наибольший интерес представляют вторая, третья и четвертая части.

Концепция защиты СВТ и АС от НСД к информации

Данная концепция предусматривает существование двух относительно самостоятельных направлений в проблеме защиты информации от НСД: направления, связанного с защитой средств вычислительной техники (СВТ), и направления, связанного с защитой автоматизированных систем (АС). Различие этих направлений порождено тем, что СВТ разрабатываются и поставляются на рынок лишь как элементы, из которых в дальнейшем строятся функционально ориентированные АС, и поэтому, не решая прикладных задач, СВТ не содержат пользовательской информации. В случае СВТ можно говорить лишь о защищенности (защите) СВТ от НСД к информации, для обработки, хранения и передачи которой СВТ предназначено. Примером СВТ можно считать специализированную плату расширения с соответствующим аппаратным и программным интерфейсом, реализующую функции аутентификации пользователя по его биометрическим характеристикам. Или к СВТ можно отнести программу прозрачного шифрования данных, сохраняемых на жестком диске.

При создании АС появляются такие, отсутствующие при разработке СВТ характеристики, как полномочия пользователей, модель нарушителя, технология обработки информации. Типичным примером АС является многопользовательская и многозадачная ОС.

Для определения принципов защиты информации в руководящих документах ГТК дается понятие НСД к информации: *НСД-доступ к информации, нарушающий установленные правила разграничения доступа, с использованием штатных средств, предоставляемых СВТ или АС.* В данном определении под штатными средствами понимается совокупность программного, микропрограммного и технического обеспечения СВТ или АС.

Понятие НСД определяет, от чего сертифицированные по руководящим документам ГТК системы защиты АС и СВТ должны защищать информацию. Например, к НСД не отнесены разрушительные последствия стихийных бедствий, хотя они и представляют угрозу информации, в частности ее целостности и доступности

К основным способам НСД относятся:

– непосредственное обращение к объектам доступа (например, через получение программой, управляемой пользователем, доступа на чтение или запись в файл);

– создание программных и технических средств, выполняющих обращение к объектам доступа в обход средств защиты (например, используя люки, оставленные разработчиками системы защиты);

– модификация средств защиты, позволяющая осуществить НСД (на пример, путем внедрения в систему защиты программных закладок или модулей, выполняющих функции "троянского коня");

– внедрение в технические средства СВТ или АС программных или технических механизмов, нарушающих предполагаемую структуру и функции СВТ или АС и позволяющих осуществить НСД (например, путем загрузки на компьютер в обход штатной ОС иной ОС, не имеющей функций защиты).

В руководящих документах ГТК представлены семь принципов защиты информации:

– защита СВТ и АС основывается на положениях и требованиях существующих законов, стандартов и нормативно-методических документов по защите от НСД к информации;

– защита СВТ обеспечивается комплексом программно-технических средств;

– защита АС обеспечивается комплексом программно-технических средств и поддерживающих их организационных мер;

– защита АС должна обеспечиваться на всех технологических этапах обработки информации и во всех режимах функционирования, в том числе при проведении ремонтных и регламентных работ;

– программно-технические средства защиты не должны существенно ухудшать основные функциональные характеристики АС (надежность, быстродействие, возможность изменения конфигурации АС);

– неотъемлемой частью работ по защите является оценка эффективности средств защиты, осуществляемая по методике, учитывающей всю совокупность технических характеристик оцениваемого объекта, включая технические решения и практическую реализацию средств защиты;

– защита АС должна предусматривать контроль эффективности средств защиты от НСД, который либо может быть периодическим, либо инициироваться по мере необходимости пользователем АС или контролирующими органами.

В качестве нарушителя в руководящих документах Гостехкомиссии рассматривается субъект, имеющий доступ к работе со штатными средствами АС и СВТ, и являющийся специалистом высшей квалификации, знающим все о АС и, в частности, о системе и средствах ее защиты. В руководящих документах дается классификация нарушителя по уровню возможностей, предоставляемых ему штатными средствами АС и СВТ. Классификация является иерархической, т.е. каждый следующий уровень включает в себя функциональные возможности предыдущего. Выделяется четыре уровня этих возможностей.

– Первый уровень определяет самый низкий уровень возможностей ведения диалога в АС - запуск задач (программ) из фиксированного набора, реализующих заранее предусмотренные функции по обработке информации.

– Второй уровень определяется возможностью создания и запуска собственных программ с новыми функциями по обработке информации.

– Третий уровень определяется возможностью управления функционированием АС, т.е. воздействием на базовое программное обеспечение системы и на состав и конфигурацию ее оборудования.

– Четвертый уровень определяется всем объемом возможностей лиц, осуществляющих проектирование, реализацию и ремонт технических средств АС, вплоть до включения в состав СВТ собственных технических средств с новыми функциями по обработке информации.

Показатели защищенности средств вычислительной техники от НСД

В ч.2. руководящих документов Гостехкомиссии установлено 7 классов защищенности СВТ от НСД к информации. Самый низкий класс - седьмой, самый высокий – первый. Показатели защищенности и требования к классам защиты приведены в таблице 10.1.

Табл. 10.1. Показатели защищенности и требования к классам защиты СВТ от НСД

Наименование показателя	**Класс защищенности**					
	6	**5**	**4**	**3**	**2**	**1**
Дискреционный принцип контроля доступа	+	+	+	=	+	=
Мандатный принцип контроля доступа	-	-	+	=	=	=
Очистка памяти	-	+	+	+	=	=
Изоляция модулей	-	-	+	=	+	=
Маркировка документов	-	-	+	=	=	=
Защита ввода и вывода на отчужденный носитель информации	-	-	+	=	=	=
Сопоставление пользователя с устройством	-	-	+	=	=	=
Идентификация и аутентификация	+	=	+	=	=	=
Гарантии проектирования	-	+	+	+	+	+
Регистрация	-	+	+	+	=	=
Взаимодействие пользователя с КСЗ	-	-	-	+	=	=
Надежное восстановление	-	-	-	+	=	=
Целостность КСЗ	-	+	+	+	=	=
Контроль модификации	-	-	-	-	+	=
Контроль дистрибуции	-	-	-	-	+	=
Гарантии архитектуры	-	-	-	-	-	+
Тестирование	+	+	+	+	+	=
Руководство пользователя	+	=	=	=	=	=
Руководство по КСЗ	+	+	=	+	+	=
Тестовая документация	+	+	+	+	+	=
Конструкторская (проектная) документация	+	+	+	+	+	+

Обозначения:

“ - ” - нет требований к данному классу

“ + ” - новые или дополнительные требования

“ = ” - требования совпадают с требованиями к СВТ предыдущего класса

Седьмой класс присваивается средствам вычислительной техники, к которым предъявлялись требования по защите от несанкционированного доступа к информации, но при оценке защищенность средства оказалась ниже уровня требований шестого класса.

Показатели защищенности автоматизированных систем от НСД

Документы Гостехкомиссии устанавливают девять классов защищенности АС от НСД, распределенных по трем группам. Каждый класс характеризуется определенной совокупностью требований к средствам защиты. В пределах каждой группы соблюдается иерархия классов защищенности АС.

Третья группа включает АС, в которых работает один пользователь, допущенный ко всей информации АС, размещенной на носителях одного уровня конфиденциальности. Данная группа содержит два класса защищенности - 3Б (низший) и 3А (высший).

Вторая группа включает АС, в которых пользователи имеют одинаковые полномочия доступа ко всей информации, обрабатываемой и хранимой в АС на носителях различного уровня конфиденциальности. Данная группа содержит два класса защищенности - 2Б (низший) и 2А (высший).

Первая группа включает многопользовательские АС, в которых одновременно обрабатывается и хранится информация разных уровней конфиденциальности, не все пользователи имеют равные права доступа. Данная группа содержит пять классов защищенности - 1Д (низший), 1Г, 1В, 1Б и 1А (высший).

В таблице 10.2 приведены требования к подсистемам защиты для каждого класса защищенности.

Табл. 10.2. Показатели защищенности и требования к классам защиты АС от НСД

Подсистемы и требования	Классы								
	3Б	3А	2Б	2А	1Д	1Г	1В	1Б	1А
I. Подсистема управления доступом									
A. Идентификация, проверка подлинности и контроль доступа субъектов:									
• в систему	+	+	+	+	+	+	+	+	+
• к терминалам, ЭВМ, узлам сети ЭВМ, каналам связи, внешним устройствам ЭВМ				+		+	+	+	+
• к программам				+		+	+	+	+
• к томам, каталогам, файлам, записям, полям записей				+		+	+	+	+
B. Управление потоками информации				+			+	+	+
II. Подсистема регистрации и учета									
A. Регистрация и учет									
• входа/выхода субъектов доступа в/из системы (узла сети)	+	+	+	+	+	+	+	+	+
• выдачи печатных (графических) выходных документов		+		+		+	+	+	+
• запуска/завершения программ и процессов (заданий, задач)				+		+	+	+	+
• доступа программ субъектов доступа к защищаемым файлам, включая их создание и удаление, передачу по линиям и каналам связи				+		+	+	+	+
• доступа программ субъектов доступа к терминалам, ЭВМ, узлам сети ЭВМ, каналам связи, внешним устройствам ЭВМ, программам, томам, каталогам, файлам, записям, полям записей				+		+	+	+	+
• изменения полномочий субъектов доступа							+	+	+
• создаваемых защищаемых объектов доступа				+			+	+	+
B. Учет носителей информации	+	+	+	+	+	+	+	+	+
C. Очистка (обнуление, обезличивание) освобождаемых областей оперативной памяти ЭВМ и внешних накопителей		+		+		+	+	+	+
D. Сигнализация попыток нарушения защиты							+	+	+
III. Криптографическая подсистема									
A. Шифрование конфиденциальной информации				+				+	+
B. Шифрование информации, принадлежащей различным субъектам									+

Подсистемы и требования		Классы								
		3Б	3А	2Б	2А	1Д	1Г	1В	1Б	1А
	доступа (группам субъектов) на разных ключах									
C.	Использование аттестованных (сертифицированных) криптографических средств				+				+	+
IV. Подсистема обеспечения целостности										
A.	Обеспечение целостности программных средств и обрабатываемой информации	+	+	+	+	+	+	+	+	+
B.	Физическая охрана средств вычислительной техники и носителей информации	+	+	+	+	+	+	+	+	+
C.	Наличие администратора (службы) защиты информации в АС				+			+	+	+
D.	Периодическое тестирование СЗИ НСД	+	+	+	+	+	+	+	+	+
E.	Наличие средств восстановления СЗИ НСД	+	+	+	+	+	+	+	+	+
F.	Использование сертифицированных средств защиты		+		+			+	+	+

Вопросы для самоконтроля

1. Перечислите основные руководящие документы, посвященные проблемам защиты информации в СВТ и АС.
2. В чем концептуальное различие между АС и СВТ с точки зрения Гостехкомиссии?
3. Что Гостехкомиссия понимает под НСД к информации?
4. Охарактеризуйте четыре уровня нарушителя с точки зрения Гостехкомиссии.
5. Сколько классов защищенности установлено для СВТ? Перечислите показатели, характеризующие защищенность СВТ от НСД.
6. Охарактеризуйте три группы АС.
7. Сколько классов защищенности установлено для АС каждой группы? Перечислите показатели, характеризующие защищенность АС от НСД.

11. Инженерно-техническая защита информации

В настоящее время возможности перехвата информации злоумышленником по техническим каналам исключительно велики.

Классификация технических каналов представлена на рисунке 11.1

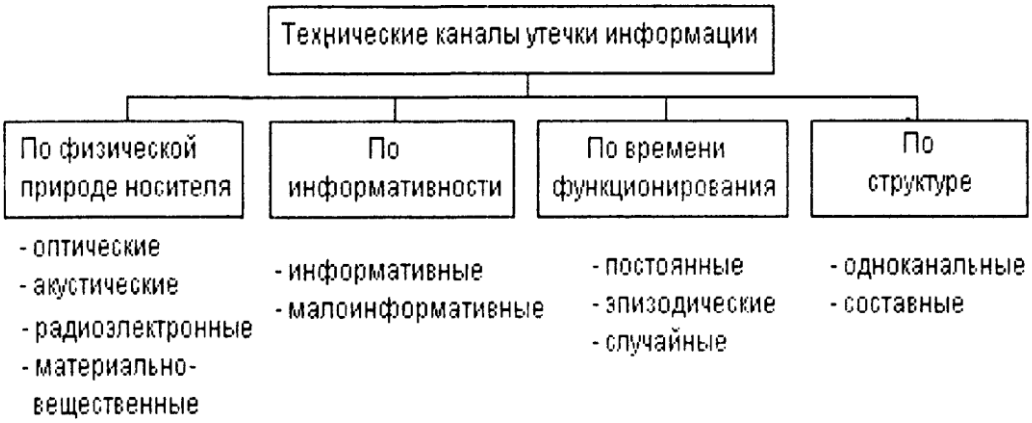


Рис. 11.1. Классификация каналов утечки информации

Наиболее информативным по возможностям утечки информации считается радиоэлектронный канал в силу следующих его особенностей [20, 21]:

- независимость функционирования канала от времени суток и года, существенно меньшая зависимость его параметров, по сравнению с другими каналами, от метеоусловий:

- высокая достоверность добываемой информации (за исключением случаев дезинформации);

- большой объем добываемой информации;

- оперативность получения информации;

- скрытность перехвата сигналов и радиотеплового наблюдения.

К основным группам технических средств перехвата информации относятся следующие [19, 20].

1. Радиопередатчики с микрофоном (радиомикрофоны):

– с автономным питанием;

– с питанием от телефонной линии;

– с питанием от электросети;

– управляемые дистанционно;

– использующие функцию включения по голосу;

– полуактивные;

– с накоплением информации и передачей в режиме быстродействия.

2. Электронные "уши":

– микрофоны с проводами;

– электронные стетоскопы (прослушивающие через стены);

– микрофоны с острой диаграммой направленности;

– лазерные микрофоны;

– микрофоны с передачей через сеть 220 В;

– прослушивание через микрофон телефонной трубки;

– гидроакустические микрофоны.

4. Устройства перехвата телефонных сообщений:

– непосредственного подключения к телефонной линии;

– подключения с использованием индукционных датчиков (датчики Холла и др.);

– с использованием датчиков, расположенных внутри телефонного аппарата;

– телефонный радиотранслятор;

– перехвата сообщений сотовой телефонной связи;

– перехвата пейджерных сообщений;

– перехвата факс-сообщений;

– специальные многоканальные устройства перехвата телефонных сообщений.

4. Устройства приема, записи, управления:

– приемник для радиомикрофонов;

– устройства записи;

– ретрансляторы;

– устройства записи и передачи в ускоренном режиме;

– устройства дистанционного управления.

5. Видеосистемы записи и наблюдения.

6. Системы определения местоположения контролируемого объекта.

7. Системы контроля компьютеров и компьютерных сетей:

– программные закладки;

– снифферы.

Рассмотрим краткую характеристику основных устройств и систем перехвата, а также средств защиты от них.

11.1. Радиомикрофоны

Радиомикрофон это микрофон, объединенный с радиоканалом передачи звуковой информации. Данные устройства также называют иногда радиозакладками, радиожучками, радиокапсулами. В общем виде структурная схема радиомикрофона приведена на рис. 11.2.

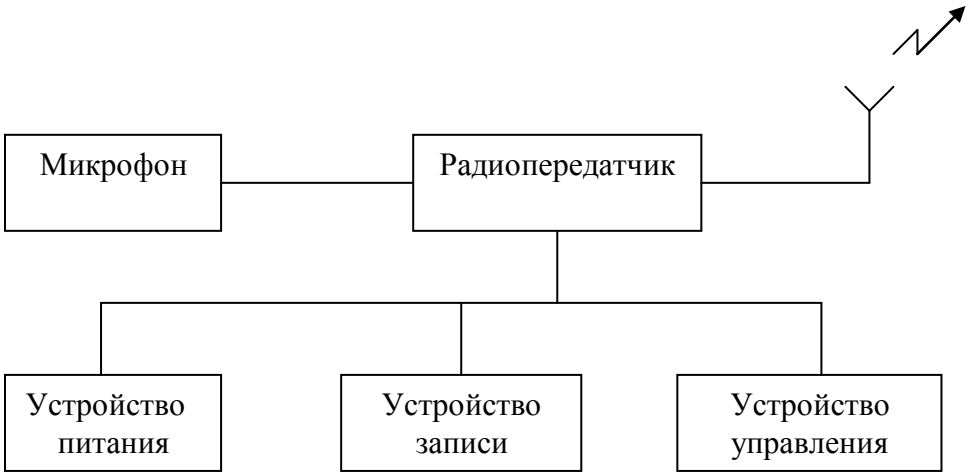


Рис. 11.2. Общая структурная схема радиомикрофона

Радиомикрофоны являются самыми распространенными техническими средствами ведения коммерческой разведки. Их популярность объясняется, прежде всего, удобством их оперативного использования, простотой применения, дешевизной и очень небольшими размерами. В самом простом случае радиомикрофон состоит из собственно микрофона, т.е. устройства для преобразования звуковых колебаний в электрические, а также радиопередатчика - устройства, излучающего в пространство электромагнитные колебания

радиодиапазона (несущую частоту), промодулированные электрическими сигналами с микрофона. Микрофон определяет зону акустической чувствительности (обычно она колеблется от нескольких до 20 - 30 метров), радиопередатчик - дальность действия радиолинии. Определяющими параметрами с точки зрения дальности действия для передатчика являются мощность, стабильность несущей частоты, диапазон частот, вид модуляции. Существенное влияние на длину радиоканала оказывает тип радиоприемного устройства.

Устройство управления не является обязательным элементом радиомикрофона. Оно предназначено для расширения его возможностей: дистанционного включения-выключения передатчика, микрофона, записывающего устройства, переключения режимов. Устройство записи также не является обязательным элементом.

Распространенным явлением является маскировка радиомикрофонов под какие-либо устройства двойного назначения: зажигалки, калькуляторы, часы и т.д. Интересной является схема оперативного применения радиомикрофона, реализованная в изделии SIPE-PS [19]. Это комплект, состоящий из бесшумного пистолета с прицельным расстоянием 25 м и радиомикрофона-стрелы. В реальных условиях города дальность действия радиомикрофона не превышает 50 м, и это обстоятельство снижает оперативную ценность системы. Аналогичный комплект фирмы CCS включает арбалет и несколько стрел-дротиков. Микрофон обеспечивает контроль разговора в радиусе до 10 м, а передатчик передает сигнал на приемник, находящийся на расстоянии до 100 м.

Дальность действия радиопередатчиков определяется в существенной степени качествами радиоприемных устройств, прежде всего, чувствительностью.

В качестве примера современного стационарного приемника можно привести радиоприемник ICOM R7100. Этот многофункциональный сканирующий приемник имеет возможность приема радиосигналов с любыми ви-

дами модуляции. Данный приемник оборудован системой автоматического поиска и записи в память значений обнаруженных частот и встроенными часами для управления режимами работы по программе.

11.2. Устройства перехвата телефонных сообщений

Ценность информации, передаваемой по телефонным линиям, и возможности организации несанкционированного прослушивания в коммерческих или других целях, вызывает большое беспокойство у предприятий и частных лиц за сохранение конфиденциальности своих переговоров. Рассмотрим наиболее вероятные технические возможности организации такого прослушивания.

В техническом плане самым простым способом прослушивания является контактное подключение. Возможно временное подключение к абонентской проводке с помощью стандартной «монтерской трубки».

Однако такое подключение легко обнаруживается с помощью простейших средств контроля напряжения телефонной сети. Существенными недостатками контактного способа подключения являются нарушение целостности проводов и влияние подключенного устройства на характеристики линий связи.

В настоящее время чрезвычайно популярны телефонные радиоретрансляторы, которые представляют собой радиоудлинители для передачи телефонных разговоров по радиоканалу. Большинство телефонных закладок автоматически включаются при поднятии телефонной трубки и передают информацию на пункт перехвата и записи. Источником питания для радиопередатчика является, как правило, напряжение телефонной сети. Так как в данном случае не требуется ни батареек, ни встроенного микрофона, размеры ретранслятора могут быть очень небольшими. Недостатком подобных устройств является то, что они могут быть обнаружены по радиоизлучению. На рис. 11.3 представлен пример подобного устройства.

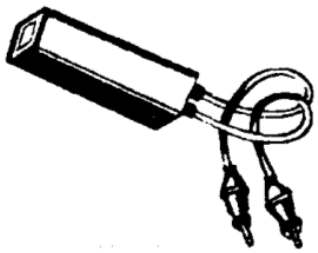


Рис. 11.3. Пример телефонного радиоретраслятора

Чтобы уменьшить возможность обнаружения радиоизлучения для радиоретранслятора, применяют тот же способ, что и в случае с радиомикрофоном - уменьшают мощность излучения передатчика, установленного на телефонной линии, а в безопасном месте устанавливают более мощный ретранслятор, переизлучающий сигнал на другой частоте и в зашифрованном виде.

Достаточно просто подслушать разговор, если используется телефон с радиоудлинителем, представляющим собой две радиостанции: одна смонтирована в трубке, другая - в самом телефонном аппарате. В этом случае нужно только настроить приемник на требуемую частоту. Для подобных целей выпускаются и специальные разведывательные приемники.

11.3. Специализированные устройства

К специализированным устройствам аудиоперехвата можно отнести направленные микрофоны, лазерные микрофоны, стетоскопы, СВЧ и ИК передатчики. Рассмотрим их особенности.

Направленные микрофоны

Обычные микрофоны способны регистрировать речевую информацию на расстоянии, не превышающем нескольких десятков метров. Для увеличения дистанции, на которой можно производить прослушивание, практикуют применение направленного микрофона. Это устройство собирает звуки только с одного направления, т.е. обладает узкой диаграммой направленности. Такие устройства широко применяются не только в разведке, но и журналистами, охотниками, спасателями и т.д.

Можно выделить два основных типа направленных микрофонов: с параболическим отражателем и резонансный микрофон. В микрофоне с параболическим отражателем собственно микрофон расположен в фокусе параболического отражателя звука. Резонансный микрофон основан на использовании явления резонанса в металлических трубках разной длины.

С точки зрения скрытого перехвата звука, применение направленных микрофонов затруднено из-за зачастую неприемлемых их габаритов и источников акустических помех. Кроме того, для того, чтобы не быть прослушанным, например, в комнате, достаточно просто закрыть форточку.

Лазерные микрофоны

В том случае, если поднято стекло в автомобиле или закрыта форточка, может быть использован лазерный микрофон. Структурная схема подобного устройства изображена на рисунке 11.4.

Луч лазера, отраженный от стекла помещения, в котором ведутся переговоры, оказывается промодулированным звуковой частотой. Принятый фотоприемником отраженный луч детектируется, звук усиливается и записывается. Подобные системы имеют очень высокую стоимость и, кроме того, требуют специального обучения персонала и использования компьютерной обработки речи для увеличения дальности.

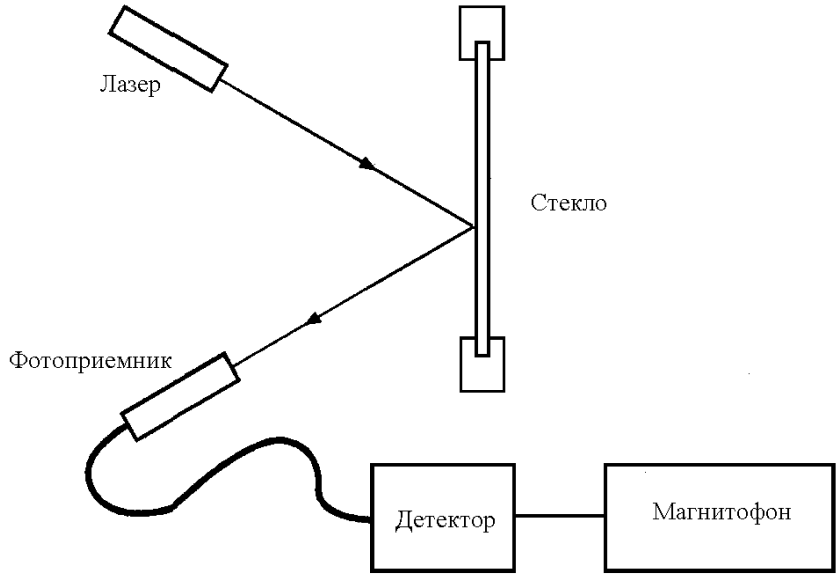


Рис. 11.4. Схема применения лазерного микрофона

Стетоскопы

Стетоскоп представляет собой совокупность вибродатчика, усилителя и головных телефонов. Вибродатчик специальной мастикой прикрепляется к стене, потолку и т.п. Стетоскоп может оснащаться проводом, радио или другим каналом передачи информации. Основным преимуществом стетоскопа можно считать трудность обнаружения, т.к. он может устанавливаться в соседних помещениях.

СВЧ и ИК передатчики

Данные передатчики используются для повышения их скрытности и затруднения обнаружения. В этом случае для передачи информации используется инфракрасный канал или СВЧ диапазон.

11.4. Обнаружение, локализация и подавление закладных подслушивающих устройств

Обнаружение закладных устройств производится по их демаскирующим признакам. Чем больше демаскирующих признаков в признаковой структуре и чем они информативнее, тем выше вероятность обнаружения объекта. Ка-

ждый вид закладных устройств имеет свою признаковую структуру, позволяющую с той или иной вероятностью обнаружить закладку.

Наиболее информативными признаками микрофонной закладки являются [20]:

– тонкий провод, проложенный от малогабаритного микрофона закладки в другое помещение;

– наличие в кожухе закладки одного или нескольких отверстий.

Признаковые структуры некамуфлированной радиозакладки включают:

– радиоизлучения с модуляцией радиосигнала акустическим сигналом, циркулирующим в помещении;

– признаки внешнего вида - малогабаритный предмет непонятного назначения в форме параллелепипеда, цилиндра без или с одним органом управления (выключателем питания) на поверхности;

– одно или несколько отверстий малого диаметра в кожухе;

– наличие, но не всегда, небольшого отрезка провода, выходящего из кожуха;

– присутствие полупроводниковых элементов, выявляемых при облучении обследуемых предметов нелинейными радиолокаторами;

– наличие в устройстве металлических проводников или других деталей, определяемых металлодетекторами, или при просвечивании предмета рентгеновскими лучами.

Для предотвращения утечки информации с помощью закладных подслушивающих устройств используют средства обнаружения, локализации и подавления закладных устройств.

Классификация обнаружителей радиоизлучений закладных устройств представлена на рисунке 11.5 [20].

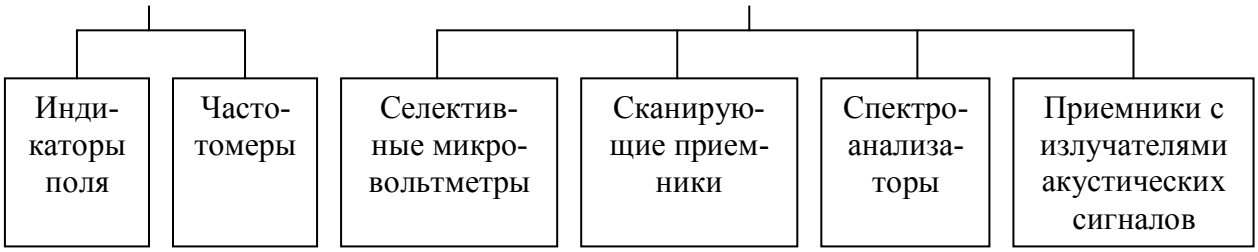


Рис. 11.5. Классификация средств обнаружения излучений закладных устройств

С помощью обнаружителей поля можно обнаруживать поля радиозакладок в непосредственной близости от источника излучения. *Индикаторы поля* позволяют информировать оператора о наличии в месте расположения антенны индикатора электромагнитного поля с напряженностью выше фоновой. *Частотомеры* обеспечивают кроме этого измерение частоты колебаний поля. На рис. 11.6. представлен внешний вид индикатора поля D006 и частотомера ST-007.

Возможности бытовых радиоприемников ограничены поиском радиозакладок в радиовещательном диапазоне. Для поиска радиозакладных устройств с неизвестной частотой данные средства неэффективны. Примером бытового радиоприемника, позволяющего осуществлять поиск отдельных радиозакладок, является Sony CFM-145.

D006 ST-007

Рис. 11.6. Примеры индикатора поля D006 и частотомера ST-007

Широкими возможностями по обнаружению радиозакладок обладают специальные приемники. Среди них все большую популярность приобретают радиоприемники с автоматизированным сканированием радиодиапазона. Они обеспечивают поиск в диапазоне частот, перекрывающем частоты почти всех применяемых радиозакладок - от долей МГц до единиц ГГц. Кроме

того, сканирующие радиоприемники имеют, как правило, оперативную память для запоминания частот не представляющих интерес источников излучения, прежде всего, радиовещательных и служебных радиостанций. На рис. 11.7 представлен внешний вид скоростного поискового приемника «Скорпион» и сканирующего радиоприемника AR-5000A.

СКОРПИОН AR-5000A

Рис. 12.7. Примеры скоростного поискового приемника «Скорпион» и сканирующего радиоприемника AR-5000A

Информационно-техническое сопряжение сканирующих приемников с переносными компьютерами послужило технической основой для создания автоматизированных комплексов быстрого и надежного поиска радиоизлучающих подслушивающих устройств.

Дистанционно управляемые радиозакладки и закладки, передающие информацию по проводам, не обнаруживаются аппаратурой радиоконтроля. Для их поиска используются демаскирующие признаки материала конструкции и элементов схемы закладного устройства, а также признаки сигналов, распространяющихся по проводам. Классификация средств обнаружения неизлучающих закладок представлена на рис. 11.8 [20].

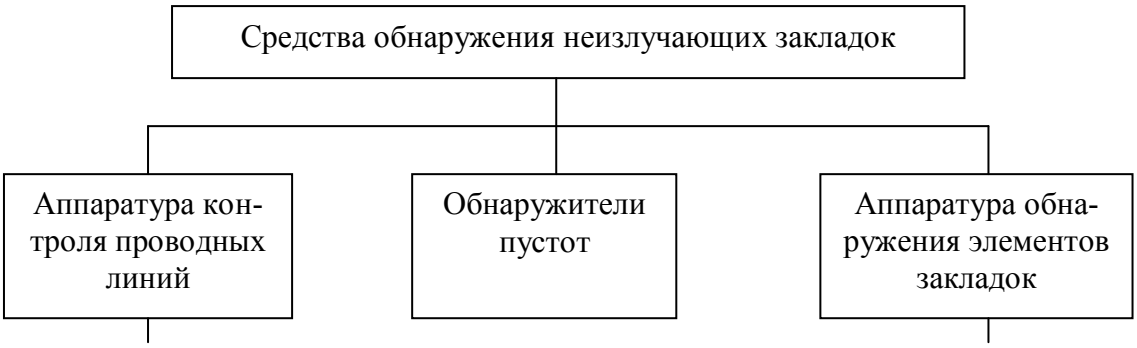


Рис. 11.8. Классификация средств обнаружения неизлучающих закладок

Аппаратура для контроля проводных линий предназначена для выявления в них опасных сигналов и их источников, в том числе закладных устройств. Так как основными направляющими линиями, по которым передаются от закладных устройств электрические сигналы с информацией, являются телефонные линии и цепи электропитания, то соответствующие средства контроля включают приборы контроля телефонных линий и линий электропитания. На рис. 11.9. представлен внешний вид контроллера телефонных линий «КТЛ-400», позволяющего обнаруживать гальванические подключения к телефонной линии и определять тип подключения: параллельное, последовательное.

Рис. 11.9. Контроллер телефонных линий КТЛ-400

Обнаружители пустот позволяют обнаруживать возможные места установки закладных устройств в пустотах стен или других деревянных или кирпичных конструкциях. На рис. 11.10. показан внешний вид прибора для обследования строительных конструкций «РАСКАН-2М», предназначенного для обнаружения скрытых предметов, арматуры, различных неоднородностей и инородных тел, пустот, а также получения изображения внутренней структуры исследуемого объекта. На этом же рисунке представлены изображения, сформированные данным прибором.

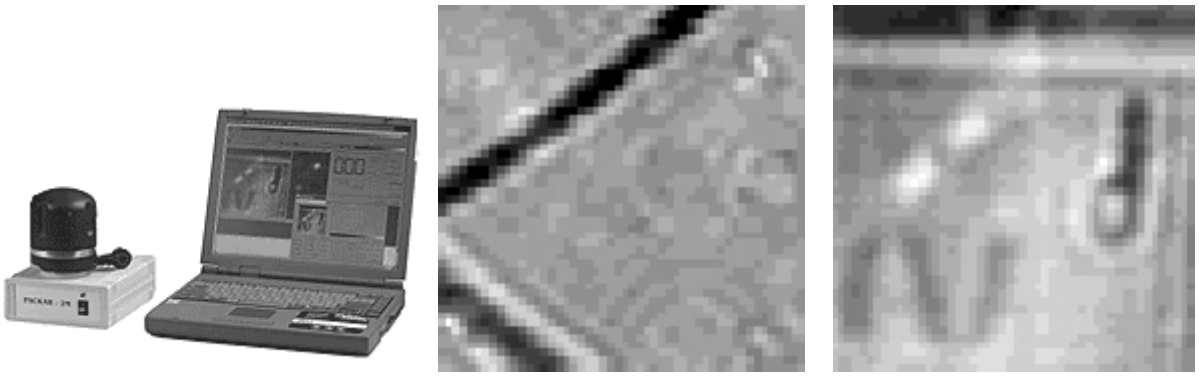


Рис. 11.10. Прибор для обследования строительных конструкций «РАСКАН-2М» и примеры сформированных им изображений

Большую группу образуют средства обнаружения или локализации закладных устройств по физическим свойствам элементов электрической схемы или конструкции. Такими элементами являются: полупроводниковые приборы, которые применяются в любых закладных устройствах, металлические детали конструкции, элементы, поглощающие рентгеновские лучи. Из этих средств наиболее достоверные результаты обеспечивают средства обнаружения полупроводниковых элементов по их нелинейным свойствам - *нелинейные радиолокаторы*. Принципы работы нелинейных радиолокаторов близки к принципам работы радиолокационных станций, широко применяемых для радиолокационного наблюдения различных объектов. Существенное отличие заключается в том, что если приемник радиолокационной станции принимает отраженный от объекта эхо-сигнал на частоте излучаемого сигнала, то приемник нелинейного локатора принимает 2-ю и 3-ю гармоники отраженного сигнала. Появление в отраженном сигнале этих гармоник обусловлено нелинейностью характеристик выход/вход полупроводников. В результате нелинейного преобразования электрического сигнала, индуцируемого в элементах схемы закладного устройства высокочастотным полем локатора, образуется сигнал, в спектре которого присутствуют кроме основной частоты ее гармоники. Количество и амплитуда гармоник зависят от характера нелинейности и мощности электромагнитного поля. На рис. 11.11 представлен внешний вид нелинейного локатора NR-900EM.

Рис. 11.11. Нелинейный локатор NR-900EM

Металлодетекторы (металлоискатели) реагируют на наличие в зоне поиска электропроводных материалов, прежде всего, металлов, и позволяют обнаруживать корпуса или другие металлические элементы закладки.

Переносные рентгеновские установки применяются для просвечивания предметов, назначение которых не удается выявить без их разборки, прежде всего тогда, когда разборка невозможна без разрушения найденного предмета. На рис. 11.12 представлен внешний вид малогабаритной рентгенотелевизионной установки «НОРКА».

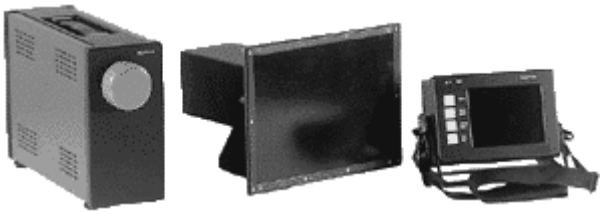


Рис. 11.12. Малогабаритная рентгенотелевизионная установка «НОРКА»

Подавление сигналов закладных устройств является одним из активных средств борьбы с радиопередающими закладками. Классификация *средств подавления закладных устройств* представлена на рис. 11.13 [20].

Линейного зашумления	Пространственного зашумления

Рис. 11.13. Классификация средств подавления закладок

Выходы *генератора помех* соединяются с проводами телефонной линии и электросети и в них подаются электрические сигналы, перекрывающие опасные сигналы по спектру и мощности. Генераторы пространственного зашумления повышают уровень электромагнитных помех в помещении и. следовательно, на входе приемника злоумышленника. Для эффективного подавления сигнала закладки уровень помехи в полосе спектра сигнала должен в несколько раз превышать уровень сигнала. На рис. 11.14. представлен внешний вид устройства защиты телефонных переговоров ЦИКАДА-М.

Рис. 11.14. Устройство защиты телефонных переговоров ЦИКАДА-М.

Один из способов физического повреждения закладок, подключенных к телефонной линии и линиям электропитания, - подача в линию коротких импульсов большой амплитуды. Так как в схемах закладок применяются миниатюрные низковольтные детали (транзисторы, конденсаторы), то высоковольтные импульсы их пробивают, и схема закладки выводится из строя. На рис. 11.15 представлен внешний вид выжигателя телефонных закладок «КОБРА».

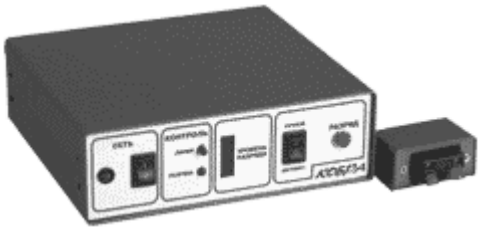


Рис. 11.15. Выжигатель телефонных закладок КОБРА

11.5. Противодействие перехвату речевой информации

Способы и средства противодействия подслушиванию направлены, прежде всего, на предотвращение утечки информации в акустическом канале. Кроме этого, для повышения дальности подслушивания применяют составные каналы утечки информации, содержащие наряду с акустическими также радиоэлектронные (с использованием закладных устройств) и оптические (с лазерными микрофонами).

В соответствии с общими методами защиты информации для защиты от прослушивания применяются следующие способы [20].

1. Информационное скрытие, предусматривающее:

−техническое закрытие и шифрование семантической речевой информации в функциональных каналах связи;

−дезинформирование.

2. Энергетическое скрытие путем:

−звукоизоляции акустического сигнала;

−звукопоглощения акустической волны;

−глушения акустических сигналов;

−зашумления помещения или твердой среды распространения другими широкополосными звуками (шумами, помехами), обеспечивающими

−маскировку акустических сигналов.

3. Обнаружение, локализация и изъятие закладных устройств.

Информационное скрытие речевой информации обеспечивается техническим закрытием (аналоговым скремблированием) и шифрованием сигналов речевой информации, передаваемых по кабелям и радиоканалам. При *аналоговом скремблировании* изменяются характеристики исходного речевого сообщения таким образом, что преобразованное сообщение становится нераспознаваемым «на слух», но занимает ту же частотную полосу. Это позволяет передавать скремблированные сигналы по обычным телефонным каналам связи. На рис. 11.15 представлен внешний вид скремблера «ГРОТ».

Рис. 11.15. Скремблер «ГРОТ»

Энергетическое скрытие акустических сигналов обеспечивается путем применения способов и средств, уменьшающих энергию носителя или увеличивающих энергию помех. Простейшим способом первого метода является уменьшение громкости речи во время разговора на конфиденциальные темы. Однако это возможно, если количество собеседников мало. В иных случаях применяют звукоизоляцию (применение звукоизолирующих материалов), звукопоглощение (применение звукопоглощающих материалов) и глушение звука. Третий метод предусматривает применение активных средств - генераторов акустических помех, вибрационное зашумление. На рис. 11.16. представлен внешний вид устройства постановки виброакустических и акустических помех «ШОРОХ-2».

Рис. 11.16. Устройство постановки виброакустических и акустических помех ШОРОХ-2

Пассивное энергетическое скрытие акустической информации от подслушивания лазерным микрофоном заключается в ослаблении энергии акустической волны, воздействующей на оконное стекло. Оно достигается использованием штор и жалюзей, а также двойных оконных рам. Активные способы энергетического скрытия акустической информации предусматривают применение генераторов шумов в акустическом диапазоне, датчики которых приклеиваются к стеклу и вызывают его колебание по случайному за-

кону с амплитудой, превышающей амплитуду колебаний стекла от акустической волны.

Для предотвращения несанкционированной записи речевой информации на диктофон необходимо:

–обнаружить работающий диктофон в кармане, портфеле, сумке или других носимых вещах участника переговоров или совещания;

–нарушить работу диктофона таким образом, чтобы качество записанной информации было ниже допустимого уровня.

Диктофон может быть обнаружен металлодетектором (ручным или стационарным). Но этот способ допустим перед проведением ответственного совещания по договоренности с его участниками. В обычной деятельности организации такой способ нецелесообразен, так как может вызвать негативную реакцию посетителя или участника переговоров.

В современных средствах защиты обнаружение и идентификация работающего диктофона производится путем выявления и анализа изменений параметров полей, измеренных в месте размещения посетителя (участника переговоров или совещания). Путем накопления изменений удается выделить регулярное поле двигателя диктофона на фоне даже более мощных случайных полей других источников.

Для исключения записи речи на диктофоны создано большое количество активных средств нарушения их работы. Принципы работы этих средств основаны на изменении под действием создаваемых ими полей режимов усилителей записи, в результате чего резко ухудшается разборчивость речи и становится невозможным ее восстановление при воспроизведении.

На рис. 11.17. представлен внешний вид устройства подавления диктофонов «САПФИР-2».

Рис. 11.17. Устройство подавления диктофонов «САПФИР-2»

11.6. Предотвращение утечки информации через побочные электромагнитные излучения и наводки

Способы и средства защиты информации от утечки через побочные электромагнитные излучения и наводки (ПЭМИН) должны удовлетворять следующим требованиям.

1. Опасные сигналы, которые могут содержать конфиденциальную информацию, должны быть ослаблены до уровня, исключающего съем с них информации на границе контролируемой зоны.

2. Средства защиты не должны вносить заметных искажений в работу функциональных устройств, используемых сотрудниками организации, и не усложнять процесс пользования ими.

Основной способ защиты от ПЭМИН - энергетическое скрытие.

Способы подавления опасных электрических сигналов, распространяющихся из контролируемой зоны по кабелям (электрическим проводам), могут быть пассивными и активными. Первые обеспечивают уменьшение уровня опасных сигналов, вторые - повышение уровня помех. Классификация данных способов защиты представлена на рис. 11.18 [20].

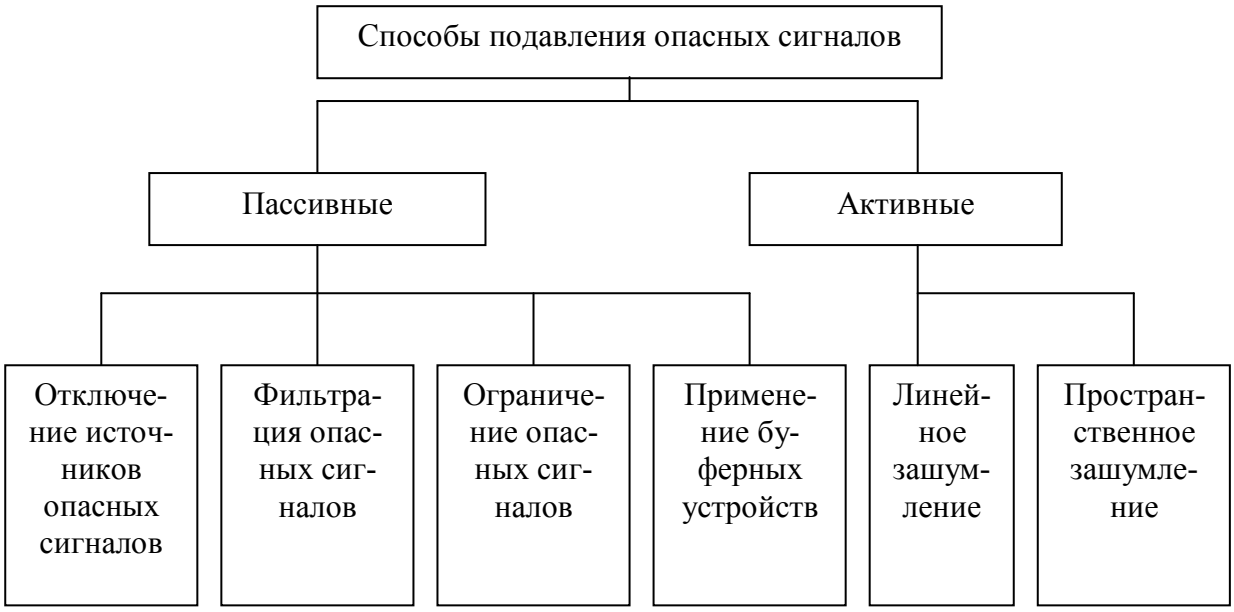


Рис. 11.18. Классификация способов подавления опасных сигналов

Отключение устройств с акустоэлектрическими преобразователями, создающими опасные сигналы, является наиболее простым и эффективным способом защиты информации. Необходимо отключать в помещении, в котором ведутся конфиденциальные разговоры, все радиоэлектронные средства и электрические приборы, без которых можно обойтись.

Фильтрация опасных сигналов эффективна, если частоты опасных сигналов существенно отличаются от частот полезных сигналов. Простейшим фильтром является конденсатор, устанавливаемый в звонковую цепь телефонных аппаратов устаревшей (с электромеханическим звонком) конструкции.

Возможность ограничения опасных сигналов основывается на нелинейных свойствах полупроводниковых элементов (диодов, транзисторов, тиристоров). Опасные сигналы, возникающие в защищаемых радиоэлектронных средствах и имеющие малую амплитуду по сравнению с полезным сигналом, проходя через фильтр, дополнительно ослабляются в тысячи раз, а полезные сигналы проходят через полупроводниковый ограничитель практически без затухания.

Активные способы защиты от опасных сигналов предусматривают генерирование помех в радиодиапазоне (для пространственного зашумления) и звуковом (для линейного зашумления).

На рис. 11.19. представлен внешний вид шумогенератора маскировки ПЭМИН «ГНОМ-3».

Рис. 11.19. Шумогенератор маскировки ПЭМИН «ГНОМ-3»

11.7. Вопросы для самоконтроля

1. Перечислите основные группы технических средств ведения разведки.

2. Каким образом производится обнаружение закладных устройств? Что такое демаскирующий признак?

3. Перечислите наиболее информативные демаскирующие признаки радиозакладок.

4. Перечислите основные обнаружители радиоизлучений закладных устройств.

5. В чем заключается функция индикаторов поля при обнаружении закладных устройств?

6. Перечислите способы обнаружения неизлучающих закладных устройств.

7. В чем заключаются функции обнаружителей пустот и нелинейных локаторов при обнаружении неизлучающих закладных устройств?

8. Перечислите способы подавления закладных устройств.

9. Какие способы предусмотрены для противодействия прослушиванию помещений?

10. Охарактеризуйте понятие «скремблер».

11. Какие способы предусмотрены для предотвращения несанкционированной записи на диктофон?

12. Перечислите способы подавления ПЭМИН.

12. ПРАВОВОЕ ОБЕСПЕЧЕНИЕ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ

Особенностью современного развития цивилизации становятся информационные ресурсы и инфокоммуникационные системы. Сейчас уже установлено [22], что для «…двукратного роста материального производства необходимо четырехкратное увеличение объема обеспечивающей информации». Таким образом, информация превращается в наиболее ценный продукт и один из основных товаров, а общество становится информационным. Естественно, что становление информационного общества возможно только через развитие законодательной базы.

Информация может иметь ценность, собственника, пользователя и владельца информационных ресурсов и, следовательно, являться объектом права.

Но ценность влечет за собой и возможность неправомерных действий и неправомерного доступа к информации как внутри государства, так и на межгосударственном уровне. Последний случай называют «информационной войной», под которой понимается [22]: особый вид отношений между государствами, при котором для разрешения существующих межгосударственных противоречий используются методы, средства и технологии силового воздействия на информационную сферу этих государств. Особенность информационной войны является ее скрытность, латентность. Ясно, что война осуществляется при помощи информационного оружия. Предлагается следующая классификация информационного оружия [22]:

– стратегическое - совокупность информации, технологий, средств реализации технологий, способных нанести неприемлемый ущерб политическим, экономическим и военным интересам страны, а также структурам образующим ее стратегический потенциал, в рамках стратегической операции вооруженных сил государства;

– оперативное – совокупность видов информационно оружия, способного обеспечить решение важных задач при проведении операций вооруженных сил на определенном театре военных действий;

– тактическое - совокупность видов информационно оружия способного обеспечить решение важных задач в ходе боевых действий или боя.

Появились понятия информационного терроризма, «кибертерроризма».

Все это потребовало юридического толкования информационного оружия и его элементарных составляющих.

К видам информационного оружия, которые воздействуют непосредственно на информацию и программное обеспечение ЭВМ, можно отнести специальные компьютерные программы или части программ, именуемые

компьютерными вирусами и логическими бомбами, и аналогичные негативные средства. Приведем примеры.

Компьютерный вирус – это специальная программ, целью которой является негативное воздействие (разрушение, семантические изменения …) на хранимую в ЭВМ информацию и программное обеспечение.

Программная закладка – включенная в состав программ для ЭВМ последовательность команд, активизирующаяся при определенных условиях и выполняющая хищение информации.

Специфическим видом оружия является электромагнитное оружие и, соответственно, способы ведения войны: радиоэлектронная борьба, заключающаяся в создании помех средствам связи противника и радиолокации.

Столь же специфичной является и «хакерская» война - организация атак на вычислительные системы и сети специально обученными лицами.

Элементами негативных, проводящихся для нанесения ущерба действий, являются: уничтожение, блокирование, модификация и копирование информации, нарушение работы средства.

Фактически, становление инфокоммуникационных систем привело к пониманию необходимости создания системы информационного права как самостоятельной отрасли. В нее вошли набор правовых институтов, основным из которых является институт государственной тайны.

Законодательной базой информационного права стали такие документы, как: «Доктрина информационной безопасности», упоминавшийся ранее закон «Об информации,..», законы «О государственной тайне», «О связи», «Об оружии», «О безопасности», кодексы: «Уголовный» «Уголовно - процессуальный», «Гражданский» и др.

Пожалуй, впервые «Уголовный кодекс» включил информационно правовые статьи. Основными из них являются статьи 272, 273, 274. Рассмотрим их.

12.1. Статья 272 УК РФ

Ст. 272 УК РФ – «Неправомерный доступ к компьютерной информации».

1. Неправомерный доступ к охраняемой законом компьютерной информации, то есть информации на машинном носителе, в электронно-вычислительной машине (ЭВМ), системе ЭВМ или их сети, если это деяние повлекло уничтожение, блокирование, модификацию либо копирование информации, нарушение работы ЭВМ, системы ЭВМ или их сети,

- наказывается штрафом в размере от двухсот до пятисот минимальных размеров оплаты труда или в размере заработной платы или иного дохода осужденного за период от двух до пяти месяцев, либо исправительными работами на срок от шести месяцев до одного года, либо лишением свободы на срок до двух лет.

2. То же деяние, совершенное группой лиц по предварительному сговору или организованной группой либо лицом с использованием своего служебного положения, а равно имеющим доступ к ЭВМ, системе ЭВМ или их сети,

- наказывается штрафом в размере от пятисот до восьмисот минимальных размеров оплаты труда или в размере заработной платы или иного дохода осужденного за период от пяти до восьми месяцев, либо исправительными работами на срок от одного года до двух лет, либо арестом на срок от трех до шести месяцев, либо лишением свободы на срок до пяти лет.

12.2. Статья 273 УК РФ

Ст. 273 УК РФ – «Создание, использование и распространение вредоносных программ для ЭВМ»

1. Создание программ для ЭВМ или внесение изменений в существующие программы, заведомо приводящие к несанкционированному уничтожению, блокированию, модификации либо копированию информации, нарушению ра-

боты ЭВМ, системы ЭВМ, а равно использование либо распространение таких программ или машинных носителей с такими программами,-

- наказывается лишением свободы на срок до трех лет со штрафом в размере от двухсот до пятисот минимальных размеров оплаты труда или в размере заработной платы или иного дохода осужденного за период от двух до пяти месяцев.

2. Те же деяния, повлекшие по неосторожности тяжкие последствия,

- наказываются лишением свободы на срок от трех до семи лет.

Отметим, что негативное воздействие могут иметь и последствия, выражающиеся в нанесении ущерба материального, например, физическое разрушение ресурса (диска, «винчестера» и т.п.) или интеллектуальной собственности. В связи с этим при рассмотрении в судах компьютерных правонарушений, привлекаются другие статьи уголовного или гражданского кодексов.

12.3. Статья 274 УК РФ

Ст. 274 УК РФ – «Нарушение правил эксплуатации ЭВМ, системы ЭВМ или их сети»

1. Нарушение правил эксплуатации ЭВМ, системы ЭВМ или их сети лицами, имеющим доступ к ЭВМ, системе ЭВМ или их сети, повлекшие уничтожение, блокирование или модификацию охраняемой законом информации ЭВМ, если это деяние причинило существенный вред,

- наказывается лишением права занимать определенные должности или заниматься определенной деятельностью на срок до пяти лет, либо обязательными работами на срок от ста восьмидесяти до двухсот сорока часов, либо ограничением свободы на срок до двух лет.

12.4. Статья 146. Нарушение авторских и смежных прав

1. Незаконное использование объектов авторского права или смежных прав, а равно присвоение авторства, если эти деяния причинили крупный ущерб,

- наказываются штрафом в размере от двухсот до четырехсот минимальных размеров оплаты труда или в размере заработной платы или иного дохода осужденного за период от двух до четырех месяцев, либо обязательными работами на срок от ста восьмидесяти до двухсот сорока часов, либо лишением свободы на срок до двух лет.

2. Те же деяния, совершенные неоднократно либо группой лиц по предварительному сговору или организованной группой,

- наказываются штрафом в размере от четырехсот до восьмисот минимальных размеров оплаты труда или в размере заработной платы или иного дохода осужденного за период от четырех до восьми месяцев, либо арестом на срок от четырех до шести месяцев, либо лишением свободы на срок до пяти лет.

12.5. Статья 147. Нарушение изобретательских и патентных прав

1. Незаконное использование изобретения, полезной модели или промышленного образца, разглашение без согласия автора или заявителя сущности изобретения, полезной модели или промышленного образца до официальной публикации сведений о них, присвоение авторства или принуждение к соавторству, если эти деяния причинили крупный ущерб,

- наказываются штрафом в размере от двухсот до четырехсот минимальных размеров оплаты труда или в размере заработной платы или иного дохода осужденного за период от двух до четырех месяцев, либо обязательными работами на срок от ста восьмидесяти до двухсот сорока часов, либо лишением свободы на срок до двух лет.

2. Те же деяния, совершенные неоднократно либо группой лиц по предварительному сговору или организованной группой,

- *наказываются штрафом в размере от четырехсот до восьмисот минимальных размеров оплаты труда или в размере заработной платы или иного дохода осужденного за период от четырех до восьми месяцев, либо арестом на срок от четырех до шести месяцев, либо лишением свободы на срок до пяти лет.*

Несанкционированные действия с информацией могут иметь большие последствия, особенно при работе со сведениями, составляющими государственную тайну. Нормативно – законодательная база в настоящее время совершенствуется и развивается.

12.6. Вопросы для самоконтроля

1. Что понимают под информацией, информатизацией и документов в законе «Об информации, информатизации и защите информации»?

2. Охарактеризуйте понятие «информационная война».

3. Перечислите и охарактеризуйте основные статьи УК РФ, относящиеся к компьютерным правонарушениям.

ЛИТЕРАТУРА

1. Д.П. Зегжда, А.М. Ивашко. Основы безопасности информационных систем. М.: Горячая линия – Телеком, 2000.

2. П.Н. Девянин, О.О. Михальский, Д.И. Правиков, А.Ю. Щербаков. Теоретические основы компьютерной безопасности. М.: Радио и связь, 2000.

3. А.А. Грушо, Е.Е. Тимонина. Теоретические основы защиты информации. М.: Яхтсмен, 1996.

4. В. Жельников. Криптография от папируса до компьютера. М.: ABF, 1996.

5. В. В. Мельников. Защита информации в компьютерных системах. М.: Финансы и статистика, 1997.

6. М.А. Иванов. Криптографические методы защиты информации в компьютерных системах и сетях. М.: КУДИЦ-ОБРАЗ, 2001.

7. А.А. Молдовян, Н.А. Молдовян, Б.Я. Советов. Криптография. М.: Лань, 2000.

8. Б. Шнайдер. Прикладная криптография. М.:Трикмф, 2003.

9. American National Standard for Information Technology – Role Based Access Control // NIST 359. 2003.

10. ГОСТ 28147-89. Система обработки информации. Защита криптографическая. Алгоритм криптографического преобразования.

11. ГОСТ 34.10-94. Информационная технология. Криптографическая защита информации. Процедуры выработки и проверки электронно-цифровой подписи на базе асимметричного криптографического алгоритма.

12. В.С. Горбатов, О.Ю. Полянская. Основы технологии PKI. М.: Горячая линия – Телеком, 2004.

13. Ю.В. Романец, П.А. Тимофеев, В.Ф. Шаньгин. Защита информации в компьютерных системах и сетях. М.: Радио и связь,1999.

14. П.Ю. Белкин, О.О. Михальский и др. Программно-аппаратные средства обеспечения информационной безопасности. Защита программ и данных. М.: Радио и связь, 1999.

15. И.В. Аникин, В.И. Глова. Программно-аппаратная защита информации. Защита программного обеспечения от отладки и дизассемблирования. Учебное пособие. Казань, Изд. КГТУ им. А.Н. Туполева, 2003.

16. О.В. Бурдаев, М.А. Иванов, И.И. Тетерин. Ассемблер в задачах защиты информации. М.: КУДИЦ-ОБРАЗ, 2002.

17. С.П. Расторгуев, Н.Н. Дмитриевский. Искусство защиты и раздевания программ. М:Совмаркет, 1991.

18. В.Г. Проскурин, С.В. Крутов, И.В. Мацкевич. Программно-аппаратные средства обеспечения информационной безопасности. Защита в операционных системах. М.: Радио и связь, 2000.

19. В.И. Андрианов, В.А. Бородин, А.В. Соколов. «Шпионские штучки» и устройства для защиты объектов и информации. СПб, Лань, 1996.

20. А.А. Торокин. Основы инженерно-технической защиты информации. М.: Ось-89, 1998.

21. Петровский В.И., Петровский В.В. Помехи в технологии обеспечения информационной безопасности. – Казань: Изд-во Казан. Гос. техн. ун-та, 2004. 282с.

22. Фатьянов А.А. Правовое обеспечение безопасности информации в Российской федерации. М.: Юрист, 2001.

23. А.В. Черемушкин. Вычисления в алгебре и теории чисел. М.: 2002.

24. К. Касперски. Техника и философия хакерских атак. М.: 2000.

25. Р.Э. Смит. Аутентификация: от паролей до открытых ключей. Вильямс, 2002.

26. О.Ю. Пескова. Теория и практика организации защиты информационных систем. Методическое пособие. Таганрог: Изд-во ТРТУ, 2001.

27. В..Е. Козлов. Теория и практика борьбы с компьютерной преступностью. М.: Горячая линия-Телеком, 2002.

28. А.П. Баранов, Н.П. Борисенко, П.Д. Зегжда, С.С. Корт, А.Г. Ростовцев. Математические основы информационной безопасности. Орел: ВИПС, 1997.

29. П.Д. Зегжда, Д.П. Зегжда, П.В. Семьянов, С.С. Корт, В.М.Кузьмич, И.Д. Медведовский, А.М. Ивашко, А.П. Баранов. Теория и практика обеспечения информационной безопасности. М.: Яхтсмен, 1996.

30. У. Диффи, М.Э. Хеллман. Защищенность и имитостойкость: введение в криптографию // ТИИЭР, Т. 67, № 3, 1979.

31. С.А. Петренко, С.В. Симонов. Управление информационными рисками. Экономически оправданная безопасность.М.: АйТи, 2004.

32. А. Щербаков. Разрушающие программные воздействия. М.: ЭДЕЛЬ, 1993.

33. Защита информации в компьютерных системах. Теоретические аспекты защиты от вирусов / Под ред. Э.М. Шмакова, СПб: Изд-во СпбГТУ, 1993.

34. В.А. Матвеев, С.В. Молотков, Д.П. Зегжда, А.В. Мешков, П.В. Семьянов, Д.В. Шведов. Основы верификационного анализа безопасности исполняемого кода программ / Под редакцией проф. П.Д. Зегжды СПб.: СПбГТУ, 1994.

35. Н.Г. Милославская, А.И. Толстой. Интрасети. Доступ в INTERNET, защита. М.: Юнити, 2000.

36. В. Зима, А. Молдовян, Н. Молдовян. Безопасность глобальных сетевых технологий. БХВ-Петербург, 2000.

37. К. Закер. Компьютерные сети. Модернизация. Поиск неисправностей. СПб, БХВ, 2001.

АНИКИН Игорь Вячеславович
ГЛОВА Виктор Иванович
НИГМАТУЛЛИНА Алия Науфальевна

МЕТОДЫ И СРЕДСТВА ЗАЩИТЫ КОМПЬЮТЕРНОЙ ИНФОРМАЦИИ

Учебное пособие

Ответственный за выпуск
Технический редактор
Компьютерный набор и верстка
ЛР № 020678 от 09.12.97

Формат 60x84 1/16. Бумага газетная. Печать офсетная.
Печ. л. Усл.печ.л. Уч.-изд.л. Усл.кр.-отт.
Тираж . Заказ

Издательство Казанского государственного технического университета
Типография Издательства Казанского государственного технического университета
420111, Казань, К. Маркса,10