С.К. Варлатая, М.В. Шаханова

Аппаратно-программные средства и методы защиты информации

Владивосток
2007

Федеральное агентство по образованию

Дальневосточный государственный технический университет
(ДВПИ им. В.В. Куйбышева)

С.К. Варлатая, М.В. Шаханова

Аппаратно-программные средства и методы защиты информации

Рекомендовано Дальневосточным региональным учебно-методическим центром в качестве учебного пособия для студентов специальности 090104 «Комплексная защита объектов информации»

Владивосток
2007

Одобрено научно-методическим советом ДВГТУ

УДК 614.2

Программно-аппаратная защита информации: учеб. пособие /С.К. Варлатая, М.В. Шаханова. - Владивосток: Изд-во ДВГТУ, 2007.

В учебном пособии последовательно излагаются основные понятия аппаратно-программных средств защиты информации. Рассматриваются основные понятия программно-аппаратной защиты информации, идентификация пользователей КС-субъектов доступа к данным, средства и методы ограничения доступа к файлам, аппаратно-программные средства криптографической защиты информации, методы и средства ограничения доступа к компонентам ЭВМ, защита программ от несанкционированного копирования, управление криптографическими ключами, защита программных средств от исследования.

Пособие предназначено для студентов специальности 090104 «Комплексная защита объектов информации» для изучения дисциплины «Программно-аппаратная защита информации».

Рецензенты: МГУ Каф. АИС к.т.н. проф. Глушков С.В., зав. Кафедрой информационной безопасности ДВГУ д.ф-м.н. П.Н. Корнюшин

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Быстро развивающиеся компьютерные информационные технологии вносят заметные изменения в нашу жизнь. Информация стала товаром, который можно приобрести, продать, обменять. При этом стоимость информации часто в сотни раз превосходит стоимость компьютерной системы, в которой она хранится.

По результатам одного исследования около 58% опрошенных пострадали от компьютерных взломов за последний год. Примерно 18% опрошенных из этого числа заявляют, что потеряли более миллиона долларов в ходе нападений, более 66% потерпели убытки в размере 50 тыс. долларов. Свыше 22% атак были нацелены на промышленные секреты или документы, представляющие интерес прежде всего для конкурентов.

От степени безопасности информационных технологий в настоящее время зависит благополучие, а порой и жизнь многих людей. Такова плата за усложнение и повсеместное распространение автоматизированных систем обработки информации.

Современная информационная система представляет собой сложную систему, состоящую из большого числа компонентов различной степени автономности, которые связаны между собой и обмениваются данными. Практически каждый компонент может подвергнуться внешнему воздействию или выйти из строя.

Несмотря на то, что современные ОС для персональных компьютеров, такие, как Windows 2000, Windows XP и Windows NT, имеют собственные подсистемы защиты, актуальность создания дополнительных средств защиты сохраняется. Дело в том, что большинство систем не способны защитить данные, находящиеся за их пределами. И в этих случаях для защиты данных используются аппаратно-программные средства защиты информации.

Учебное пособие представляет собой структурированную подборку материалов, посвященных рассмотрению наиболее передовых аппаратно-программных средств и методов защиты информации на середину 2007-го года. Пособие преследует целью своего создания довести до читателя большинство средств и методов в области защиты информации в информационных системах и персональных компьютеров, дать понятие о достижениях в области защиты данных.

Тема пособия является актуальной по следующим причинам:

- информация имеет ценность;
- аппаратно-программные средства защиты информации развиваются наиболее динамично, их развитие определяется спросом на те или иные разработки в области защиты данных;
- потребность в информации для пользователей ПЭВМ является особенно острой (недостаток подготовки и обилие «мифических» преимуществ);
- обилие низкокачественной «коммерческой» информации по теме при недостатке компетентной аналитики и справочных ресурсов.

Учебно-методический комплекс по дисциплине «Программно-аппаратная защита информации» включает в себя учебник, рабочую учебную программу по дисциплине, методические рекомендации к выполнению лабораторных практических работ и контрольно-измерительные материалы, представлены в виде вопросов с вариантами ответов. Вопросы для самоконтроля, приведенные после каждой главы методического пособия, помогут лучше разобраться в нем и глубже понять его смысл, так же эти вопросы могут быть использованы преподавателями с целью контроля усвоения материала учащимися. Такой учебный комплекс позволяет студентам наиболее полно изучить дисциплину, также материалы методического пособия могут быть использованы как справочная литература.

1. ОСНОВНЫЕ ПОНЯТИЯ ПРОГРАММНО-АППАРАТНОЙ ЗАЩИТЫ ИНФОРМАЦИИ

1.1 Предмет и задачи программно-аппаратной защиты информации

Предмет защиты

В Федеральном законе РФ «Об информации, информатизации и защите информации», принятом 25 января 1995 года Государственной Думой, определено, что «информация - сведения о лицах, предметах, фактах, событиях, явлениях и процессах, независимо от формы их представления». Информация имеет ряд особенностей:

- она нематериальна;
- информация хранится и передается с помощью материальных носителей;
- любой материальный объект содержит информацию о самом себе или о другом объекте.

Не материальность информации понимается в том смысле, что нельзя измерить ее параметры известными физическими методами и приборами. Информация не имеет массы, энергии и т. п.

Информация хранится и передается на материальных носителях. Такими носителями являются мозг человека, звуковые и электромагнитные волны, бумага, машинные носители (магнитные и оптические диски, магнитные ленты и барабаны) и др.

Информации присущи следующие свойства.

Информация доступна человеку, если она содержится на материальном носителе. Поэтому необходимо защищать материальные носители информации, так как с помощью материальных средств можно защищать только материальные объекты.

Информация имеет ценность. Ценность информации определяется степенью ее полезности для владельца. Обладание истинной (достоверной)

информацией дает ее владельцу определенные преимущества. Истинной или достоверной информацией является информация, которая с достаточной для владельца (пользователя) точностью отражает объекты и процессы окружающего мира в определенных временных и пространственных рамках.

Информация, искаженно представляющая действительность (недостоверная информация), может нанести владельцу значительный материальный и моральный ущерб. Если информация искажена умышленно, то ее называют *дезинформацией*.

Законом «Об информации, информатизации и защите информации» гарантируется право собственника информации на ее использование и защиту от доступа к ней других лиц (организаций). Если доступ к информации ограничивается, то такая информация является *конфиденциальной*. Конфиденциальная информация может содержать государственную или коммерческую тайну. *Коммерческую тайну* могут содержать сведения, принадлежащие частному лицу, фирме, корпорации и т. п. *Государственную* тайну могут содержать сведения, принадлежащие государству (государственному учреждению). В соответствии с законом «О государственной тайне» сведениям, представляющим ценность для государства, может быть присвоена одна из трех возможных степеней секретности. В порядке возрастания ценности (важности) информации ей может быть присвоена степень (гриф) «*секретно*», «*совершенно секретно*» или «*особой важности*». В государственных учреждениях менее важной информации может присваиваться гриф «*для служебного пользования*».

Для обозначения ценности конфиденциальной коммерческой информации используются три категории:

- «коммерческая тайна - строго конфиденциально»;
- «коммерческая тайна - конфиденциально»;
- «коммерческая тайна».

Используется и другой подход к градации ценности коммерческой информации:

- «строго конфиденциально - строгий учет»;
- «строго конфиденциально»;
- «конфиденциально».

Ценность информации изменяется во времени.

Как правило, со временем ценность информации уменьшается. Зависимость ценности информации от времени приближенно определяется в соответствии с выражением:

$$C(t) = C_0 e^{-2,3t/\tau}$$

где C_0 - ценность информации в момент ее возникновения (получения); t - время от момента возникновения информации до момента определения ее стоимости; τ - время от момента возникновения информации до момента ее устаревания.

Время, через которое информация становится устаревшей, меняется в очень широком диапазоне. Так, например, для пилотов реактивных самолетов, авто гонщиков информация о положении машин в пространстве устаревает за доли секунд. В то же время информация о законах природы остается актуальной в течение многих веков.

Информация покупается и продается.

Ее правомочно рассматривать как товар, имеющий определенную цену. Цена, как и ценность информации, связаны с полезностью информации для конкретных людей, организаций, государств. Информация может быть ценной для ее владельца, но бесполезной для других. В этом случае информация не может быть товаром, а, следовательно, она не имеет и цены. Например, сведения о состоянии здоровья обычного гражданина являются ценной информацией для него. Но эта информация, скорее всего, не заинтересует кого-то другого, а, следовательно, не станет товаром, и не будет иметь цены.

Информация может быть получена тремя путями:

- проведением научных исследований;
- покупкой информации;
- противоправным добыванием информации.

Как любой товар, информация имеет себестоимость, которая определяется затратами на ее получение. Себестоимость зависит от выбора путей получения информации и минимизации затрат при добывании необходимых сведений выбранным путем. Информация добывается с целью получения прибыли или преимуществ перед конкурентами, противоборствующими сторонами. Для этого информация:

- продается на рынке;
- внедряется в производство для получения новых технологий и товаров, приносящих прибыль;
- используется в научных исследованиях;
- позволяет принимать оптимальные решения в управлении.

Объект защиты информации.

Объектом защиты информации является компьютерная система или автоматизированная система обработки данных (АСОД). В работах, посвященных защите информации в автоматизированных системах, до последнего времени использовался термин АСОД, который все чаще заменяется термином КС. Что же понимается под этим термином?

Компьютерная система - это комплекс аппаратных и программных средств, предназначенных для автоматизированного сбора, хранения, обработки, передачи и получения информации. Наряду с термином «информация» применительно к КС часто используют термин «данные». Используется и другое понятие - «информационные ресурсы». В соответствии с законом РФ «Об информации, информатизации и защите информации» под информационными ресурсами понимаются отдельные документы и отдельные массивы документов в информационных системах (библиотеках, архивах, фондах, банках данных и других информационных системах).

Понятие КС очень широкое и оно охватывает следующие системы:

- ЭВМ всех классов и назначений;
- вычислительные комплексы и системы;
- вычислительные сети (локальные, региональные и глобальные).

Такой широкий диапазон систем объединяется одним понятием по двум причинам: во-первых, для всех этих систем основные проблемы защиты информации являются общими; во-вторых, более мелкие системы являются элементами более крупных систем. Если защита информации в каких-либо системах имеет свои особенности, то они рассматриваются отдельно.

Предметом защиты в КС является информация. Материальной основой существования информации в КС являются электронные и электромеханические устройства (подсистемы), а также машинные носители. С помощью устройств ввода или систем передачи данных (СПД) информация попадает в КС. В системе информация хранится в запоминающих устройствах, (ЗУ) различных уровней, преобразуется (обрабатывается) процессорами (ПЦ) и выводится из системы с помощью устройств вывода или СПД. В качестве машинных носителей используются бумага, магнитные ленты, диски различных типов. Ранее в качестве машинных носителей информации использовались бумажные перфокарты и перфоленты, магнитные барабаны и карты. Большинство типов машинных носителей информации являются съемными, т.е. могут сниматься с устройств и использоваться (бумага) или храниться (ленты, диски, бумага) отдельно от устройств. Таким образом, для защиты информации (обеспечения безопасности информации) в КС необходимо защищать устройства (подсистемы) и машинные носители от несанкционированных (неразрешенных) воздействий на них.

Однако такое рассмотрение КС с точки зрения защиты информации является неполным. Компьютерные системы относятся к классу человеко-машинных систем. Такие системы эксплуатируются специалистами (обслуживающим персоналом) в интересах пользователей. Причем, в последние

годы пользователи имеют самый непосредственный доступ к системе. В некоторых КС (например, ПЭВМ) пользователи выполняют функции обслуживающего персонала. Обслуживающий персонал и пользователи являются также носителями информации. Поэтому от несанкционированных воздействий необходимо защищать не только устройства и носители, но также обслуживающий персонал и пользователей.

При решении проблемы защиты информации в КС необходимо учитывать также противоречивость человеческого фактора системы. Обслуживающий персонал и пользователи могут быть как объектом, так и источником несанкционированного воздействия на информацию.

Понятие «**объект защиты**» или «**объект**» чаще трактуется в более широком смысле. Для сосредоточенных КС или элементов распределенных систем понятие «объект» включает в себя не только информационные ресурсы, аппаратные, программные средства, обслуживающий персонал, пользователей, но и помещения, здания, и даже прилегающую к зданиям территорию.

Одними из основных понятий теории защиты информации являются понятия «безопасность информации» и «защищенные КС». **Безопасность (защищенность) информации в КС** - это такое состояние всех компонент компьютерной системы, при котором обеспечивается защита информации от возможных угроз на требуемом уровне. Компьютерные системы, в которых обеспечивается безопасность информации, называются защищенными.

Безопасность информации в КС (информационная безопасность) является одним из основных направлений обеспечения безопасности государства, отрасли, ведомства, государственной организации или частной фирмы.

Информационная безопасность достигается проведением руководством соответствующего уровня **политики информационной безопасности**. Основным документом, на основе которого проводится политика информационной безопасности, является **программа информационной безопасности**. Этот документ разрабатывается и принимается как официальный руководящий документ высшими органами управления государством,

ведомством, организацией. В документе приводятся цели политики информационной безопасности и основные направления решения задач защиты информации в КС. В программах информационной безопасности содержатся также общие требования и принципы построения систем защиты информации в КС.

Под **системой защиты информации в КС** понимается единый комплекс правовых норм, организационных мер, технических, программных и криптографических средств, обеспечивающий защищенность информации в КС в соответствии с принятой политикой безопасности.

Сеть ЭВМ - это совокупность ЭВМ, взаимосвязанных каналами передачи данных, и необходимых для реализации этой взаимосвязи программного обеспечения и (или) технических средств, предназначенных для организации распределенной обработки данных. В такой системе любое из подключенных устройств может использовать ее для передачи или получения информации. По размерности различают локальные и глобальные сети.

Многие организации используют средства **Сетей ЭВМ** для обеспечения нужд обработки и передачи данных. До использования **Сетей ЭВМ** основная часть обработки и обмена данными была централизована; информация и управление ею были сосредоточены в одном месте и централизованы. Сейчас **Сети ЭВМ** логически и физически рассредоточили данные, а также вычислительную мощность и службы обмена сообщениями по всей организации.

Службы безопасности, защищающие данные, а также средства по их обработке и передаче, также должны быть распределены по всей **Сети.**

1.2 Основные понятия

Под ***информацией,*** применительно к задаче ее защиты, понимают сведения о лицах, предметах, фактах, событиях, явлениях и процессах независимо от формы их представления. В зависимости от формы

представления информация может быть разделена на речевую, телекоммуникационную и документированную.

Речевая информация возникает в ходе ведения в помещениях разговоров, работы систем связи, звукоусиления и звуковоспроизведения. ***Телекоммуникационная*** информация циркулирует в технических средствах обработки и хранения информации, а также в каналах связи при ее передаче. К ***документированной*** информации, или ***документам,*** относят информацию, представленную на материальных носителях вместе с идентифицирующими ее реквизитами.

К ***информационным процессам*** относят процессы сбора, обработки, накопления, хранения, поиска и распространения информации.

Под ***информационной системой*** понимают упорядоченную совокупность документов и массивов документов и информационных технологий, реализующих информационные процессы.

Информационными ресурсами называют документы и массивы документов, существующие отдельно или в составе информационных систем.

Процесс создания оптимальных условий для удовлетворения информационных потребностей граждан, организаций, общества и государства в целом называют ***информатизацией***.

Информацию разделяют на открытую и ограниченного доступа. К информации ограниченного доступа относятся государственная тайна и конфиденциальная информация. В соответствии с российским законодательством к конфиденциальной относится следующая информация:

- служебная тайна (врачебная, адвокатская, тайна суда и следствия и т.п.);
- коммерческая тайна;
- персональные данные (сведения о фактах, событиях и обстоятельствах жизни гражданина, позволяющие идентифицировать его личность).

Информация является одним из объектов гражданских прав, в том числе и прав собственности, владения и пользования. ***Собственник*** информационных ресурсов, систем и технологий — это субъект с полномочиями владения, пользования и распоряжения указанными объектами. ***Владельцем*** информационных ресурсов, систем и технологий является субъект с полномочиями владения и пользования указанными объектами. Под ***пользователем*** информации будем понимать субъекта, обращающегося к информационной системе за получением необходимой ему информации и пользующегося ею.

К ***защищаемой*** относится информация, являющаяся предметом собственности и подлежащая защите в соответствии с требованиями правовых документов или требованиями, устанавливаемыми собственником информации.

Зашитой информации называют деятельность по предотвращению утечки защищаемой информации, несанкционированных и непреднамеренных воздействий на защищаемую информацию.

Под ***утечкой*** понимают неконтролируемое распространение защищаемой информации путем ее разглашения, несанкционированного доступа к ней и получения разведками. ***Разглашение*** — это доведение защищаемой информации до неконтролируемого количества получателей информации (например, публикация информации на открытом сайте в сети Интернет или в открытой печати). ***Несанкционированный доступ*** — получение защищаемой информации заинтересованным субъектом с нарушением правил доступа к ней.

Несанкционированное воздействие на защищаемую информацию — воздействие с нарушением правил ее изменения (например, намеренное внедрение в защищаемые информационные ресурсы вредоносного программного кода или умышленная подмена электронного документа).

Под ***непреднамеренным воздействием*** на защищаемую информацию понимают воздействие на нее из-за ошибок пользователя, сбоя технических

или программных средств, природных явлений, иных нецеленаправленных воздействий (например, уничтожение документов в результате отказа накопителя на жестком магнитном диске компьютера).

Целью защиты информации (ее желаемым результатом) является предотвращение ущерба собственнику, владельцу или пользователю информации. Под ***эффективностью*** защиты информации понимают степень соответствия результатов защиты информации поставленной цели. ***Объектом защиты*** может быть информация, ее носитель или информационный процесс, в отношении которых необходимо обеспечивать защиту в соответствии с поставленной целью.

Под ***качеством информации*** понимают совокупность свойств, обусловливающих пригодность информации удовлетворять определенные потребности ее пользователей в соответствии с назначением информации. Одним из показателей качества информации является ее ***защищенность*** — поддержание на заданном уровне тех параметров информации, которые характеризуют установленный статус ее хранения, обработки и использования.

Основными характеристиками защищаемой информации являются конфиденциальность, целостность и доступность. ***Конфиденциальность*** информации — это известность ее содержания только имеющим соответствующие полномочия субъектам. Конфиденциальность является субъективной характеристикой информации, связанной с объективной необходимостью защиты законных интересов одних субъектов от других.

Шифрованием информации называют процесс ее преобразования, при котором содержание информации становится непонятным для не обладающих соответствующими полномочиями субъектов. Результат шифрования информации называют ***шифротекстом,*** или ***криптограммой***. Обратный процесс восстановления информации из шифротекста называют ***расшифрованием*** информации. Алгоритмы, используемые при шифровании и расшифровании информации, обычно не являются конфиденциальными, а

конфиденциальность шифротекста обеспечивается использованием при шифровании дополнительного параметра, называемого ***ключом шифрования***. Знание ключа шифрования позволяет выполнить правильное расшифрование шифротекста.

Целостностью информации называют неизменность информации в условиях ее случайного и (или) преднамеренного искажения или разрушения. Целостность является частью более широкой характеристики информации — ее достоверности, включающей помимо целостности еще полноту и точность отображения предметной области.

Хешированием информации называют процесс ее преобразования в хеш - значение фиксированной длины (дайджест). Одним из применений хеширования является обеспечение целостности информации.

Под ***доступностью*** информации понимают способность обеспечения беспрепятственного доступа субъектов к интересующей их информации. ***Отказом в обслуживании*** называют состояние информационной системы, при котором блокируется доступ к некоторому ее ресурсу. Совокупность информационных ресурсов и системы формирования, распространения и использования информации называют ***информационной средой*** общества.

Под ***информационной безопасностью*** понимают состояние защищенности информационной среды, обеспечивающее ее формирование и развитие.

Политика безопасности — это набор документированных норм, правил и практических приемов, регулирующих управление, защиту и распределение информации ограниченного доступа.

Целью данного учебного пособия является представление методов и средств защиты информации в компьютерных системах. ***Компьютерной,*** или ***автоматизированной,*** системой обработки информации называют организационно-техническую систему, включающую в себя:

- технические средства вычислительной техники и связи;

- методы и алгоритмы обработки информации, реализованные в виде программных средств;
- информацию (файлы, базы данных) на различных носителях;
- обслуживающий персонал и пользователей, объединенных по организационно-структурному, тематическому, технологическому или другим признакам.

Электронный документ (ЭД): Информация, зафиксированная в электронной форме, подтвержденная электронной цифровой подписью и имеющая другие реквизиты электронного документа, позволяющие его идентифицировать.

Реквизиты электронного документа: Обязательные данные или сведения, которые должен содержать официальный документ, чтобы обладать подлинной юридической силой, служить основанием для совершения операций.

Электронная цифровая подпись (ЭЦП): Подпись в электронном документе, полученная в результате специальных преобразований информации данного электронного документа с использованием закрытого ключа электронной цифровой подписи и позволяющая при помощи открытого ключа электронной цифровой подписи установить отсутствие искажения информации в электронном документе и идентифицировать владельца закрытого ключа электронной цифровой подписи.

Подтверждение подлинности ЭЦП: Положительный результат проверки принадлежности электронной цифровой подписи владельцу закрытого ключа электронной цифровой подписи и отсутствия искажений информации в электронном документе.

1.3 Уязвимость компьютерных систем.

Под ***угрозой*** безопасности информации в компьютерной системе (КС) понимают событие или действие, которое может вызвать изменение

функционирования КС, связанное с нарушением защищенности обрабатываемой в ней информации.

Уязвимость информации — это возможность возникновения на каком-либо этапе жизненного цикла КС такого ее состояния, при котором создаются условия для реализации угроз безопасности информации.

Атакой на КС называют действие, предпринимаемое нарушителем, которое заключается в поиске и использовании той или иной уязвимости. Иначе говоря, атака на КС является реализацией угрозы безопасности информации в ней.

Угрозы информационной безопасности могут быть разделены на угрозы, не зависящие от деятельности человека *(**естественные*** угрозы физических воздействий на информацию стихийных природных явлений), и угрозы, вызванные человеческой деятельностью *(**искусственные*** угрозы), которые являются гораздо более опасными.

Искусственные угрозы исходя из их мотивов разделяются на ***непреднамеренные*** (случайные) и ***преднамеренные*** (умышленные).

К непреднамеренным угрозам относятся:

- ошибки в проектировании КС;
- ошибки в разработке программных средств КС;
- случайные сбои в работе аппаратных средств КС, линий связи, энергоснабжения;
- ошибки пользователей КС;
- воздействие на аппаратные средства КС физических полей других электронных устройств (при несоблюдении условий их электромагнитной совместимости) и др.

К умышленным угрозам относятся:

- несанкционированные действия обслуживающего персонала КС (например, ослабление политики безопасности администратором, отвечающим за безопасность КС);

- несанкционированный доступ к ресурсам КС со стороны пользователей КС и посторонних лиц, ущерб от которого определяется полученными нарушителем полномочиями.
- В зависимости от целей преднамеренных угроз безопасности информации в КС угрозы могут быть разделены на три основные группы:
- угроза нарушения конфиденциальности, т.е. утечки информации ограниченного доступа, хранящейся в КС или передаваемой от одной КС к другой;
- угроза нарушения целостности, т. е. преднамеренного воздействия на информацию, хранящуюся в КС или передаваемую между КС (заметим, что целостность информации может быть также нарушена, если к несанкционированному изменению или уничтожению информации приводит случайная ошибка в работе программных или аппаратных средств КС; санкционированным является изменение или уничтожение информации, сделанное уполномоченным лицом с обоснованной целью);
- угроза нарушения доступности информации, т. е. отказа в об-служивании, вызванного преднамеренными действиями одного из пользователей КС (нарушителя), при котором блокируется доступ к некоторому ресурсу КС со стороны других пользователей КС (постоянно или на большой период времени).

Опосредованной угрозой безопасности информации в КС является угроза раскрытия параметров подсистемы защиты информации, входящей в состав КС. Реализация этой угрозы дает возможность реализации перечисленных ранее непосредственных угроз безопасности информации.

Результатом реализации угроз безопасности информации в КС может быть утечка (копирование) информации, ее утрата (разрушение) или искажение (подделка), блокирование информации. Поскольку сложно заранее

определить возможную совокупность угроз безопасности информации и результатов их реализации, модель потенциальных угроз безопасности информации в КС должна создаваться совместно собственником (владельцем) КС и специалистами по защите информации на этапе проектирования КС. Созданная модель должна затем уточняться в ходе эксплуатации КС.

Рассмотрим возможные каналы утечки информации в КС. ***Косвенными*** каналами утечки называют каналы, не связанные с физическим доступом к элементам КС:

- использование подслушивающих (радиозакладных) устройств;
- дистанционное видеонаблюдение;
- перехват побочных электромагнитных излучений и наводок (ПЭМИН).

Побочные электромагнитные излучения создаются техническими средствами КС при обработке информации, существуют в диапазоне от единиц герц до 1,5 ГГц и могут распространять обрабатываемую информацию с дальностью до 1 км. Наиболее опасными с точки зрения ПЭМИН являются дисплеи, кабельные линии связи, накопители на магнитных дисках, матричные принтеры. Для перехвата ПЭМИН используется специальная портативная аппаратура, включающая в себя широкополосный автоматизированный супергетеродинный приемник с устройством регистрации информации на магнитном носителе и (или) дисплеем.

Побочные электромагнитные наводки представляют собой сигналы в цепях электропитания и заземления аппаратных средств КС и в находящихся в зоне воздействия ПЭМИН работающих аппаратных средств КС кабелях вспомогательных устройств (звукоусиления, связи, времени, сигнализации), металлических конструкциях зданий, сантехническом оборудовании. Эти наведенные сигналы могут выходить за пределы зоны безопасности КС.

Другим классом каналов утечки информации являются ***непосредственные*** каналы, связанные с физическим доступом к элементам КС. К непосредственным каналам утечки, не требующим изменения элементов КС, относятся:

- хищение носителей информации;
- сбор производственных отходов с информацией (бумажных и магнитных носителей);
- намеренное копирование файлов других пользователей КС;
- чтение остаточной информации после выполнения заданий других пользователей (областей оперативной памяти, удаленных файлов, ошибочно сохраненных временных файлов);
- копирование носителей информации;
- намеренное использование для несанкционированного доступа к информации незаблокированных терминалов других пользователей КС;
- маскировка под других пользователей путем похищения их идентифицирующей информации (паролей, карт и т.п.);
- обход средств разграничения доступа к информационным ресурсам вследствие недостатков в их программном обеспечении и др.
- К непосредственным каналам утечки, предполагающим изменение элементов КС и ее структуры, относятся:
- незаконное подключение специальной регистрирующей аппаратуры к устройствам или линиям связи (пассивное для фиксации и сохранения передаваемых данных или активное для их уничтожения, искажения или подмены);
- злоумышленное изменение программ для выполнения ими несанкционированного копирования информации при ее обработке;
- злоумышленный вывод из строя средств защиты информации.

Пассивное подключение нарушителя к устройствам или линиям связи легко предотвратить (например, с помощью шифрования передаваемой

информации), но невозможно обнаружить. Активное подключение, напротив, легко обнаружить (например, с помощью хеширования и шифрования передаваемой информации), но невозможно предотвратить.

Помимо утечки информации в КС возможны также ее несанкционированное уничтожение или искажение (например, заражение компьютерными вирусами), а также несанкционированное использование информации при санкционированном доступе к ней (например, нарушение авторских прав владельцев или собственников программного обеспечения или баз данных).

Наличие в КС значительного числа потенциальных каналов утечки информации является объективным фактором и обусловливает уязвимость информации в подобных системах с точки зрения ее несанкционированного использования.

Поскольку наиболее опасные угрозы информационной безопасности вызваны преднамеренными действиями нарушителя, которые в общем случае являются неформальными, проблема защиты информации относится к формально не определенным проблемам. Отсюда следуют два основных вывода:

- надежная защита информации в КС не может быть обеспечена только формальными методами (например, только программными и аппаратными средствами);
- защита информации в КС не может быть абсолютной.
- При решении задачи защиты информации в КС необходимо применять так называемый системно-концептуальный подход. В соответствии с ним решение задачи должно подразумевать:
- системность целевую, при которой защищенность информации рассматривается как составная неотъемлемая часть ее качества;
- системность пространственную, предполагающую взаимосвязанность защиты информации во всех элементах КС;

- системность временную, предполагающую непрерывность защиты информации;
- системность организационную, предполагающую единство организации всех работ по защите информации в КС и управления ими.

Концептуальность подхода к решению задачи защиты информации в КС предусматривает ее решение на основе единой концепции (совокупности научно обоснованных решений, необходимых и достаточных для оптимальной организации защиты информации в КС).

Обеспечение информационной безопасности КС является непрерывным процессом, целенаправленно проводимым на всех этапах ее жизненного цикла с комплексным применением всех имеющихся методов и средств.

Существующие методы и средства защиты информации можно подразделить на четыре основные группы:

- методы и средства организационно-правовой защиты информации;
- методы и средства инженерно-технической защиты информации;
- криптографические методы и средства защиты информации;
- программно-аппаратные методы и средства защиты информации.

1.4 Политика безопасности в компьютерных системах. Оценка защищенности

Политика безопасности - набор законов, правил и практических рекомендаций, на основе которых строится управление, защита и распределение критичной информации в системе. Она должна охватывать все особенности процесса обработки информации, определяя поведение системы в различных ситуациях.

Политика безопасности представляет собой некоторый набор требований, прошедших соответствующую проверку, реализуемых при помощи организационных мер и программно-технических средств, и определяющих

архитектуру системы защиты. Ее реализация для конкретной КС осуществляется при помощи средств управления механизмами защиты.

Для конкретной организации политика безопасности должна быть индивидуальной, зависимой от конкретной технологии обработки информации, используемых программных и технических средств расположения организации т.д.

Перед тем, как приступит к изложению материала введем некоторые определения, чтобы избежать путаницы.

В этой главе под "системой" мы будем понимать некоторую совокупность субъектов и объектов и их отношений между ними.

Субъект - активный компонент системы, который может явиться причиной потока информации от объекта к объекту или изменения состояния системы.

Объект - пассивный компонент системы, хранящий, принимающий или передающий информацию. Доступ к объекту подразумевает доступ к содержащейся в нем информации.

Основу политики безопасности составляет способ управления доступом, определяющий порядок доступа субъектов системы к объектам системы. Название этого способа, как правило, определяет название политики безопасности.

Для изучения свойств способа управления доступом создается его формальное описание - математическая модель. При этом модель должна отражать состояния всей системы, ее переходы из одного состояния в другое, а также учитывать, какие состояния и переходы можно считать безопасными в смысле данного управления. Без этого говорить о каких-либо свойствах системы, и тем более гарантировать их, по меньшей мере некорректно. Отметим лишь, что для разработки моделей применяется широкий спектр математических методов (моделирования, теории информации, графов, автоматов и другие).

В настоящее время лучше всего изучены два вида политики безопасности: избирательная и полномочная, основанные, соответственно на избирательном и полномочном способах управления доступом. Особенности каждой из них, а также их отличия друг от друга будут описаны ниже.

Кроме того, существует набор требований, усиливающий действие этих политик и предназначенный для управления информационными потоками в системе.

Следует отметить, что средства защиты, предназначенные для реализации какого-либо из названных выше способа управления доступом, только предоставляют возможности надежного управления доступом или информационными потоками.

Определение прав доступа субъектов к объектам и/или информационным потокам (полномочий субъектов и атрибутов объектов, присвоение меток критичности и т.д.) входит в компетенцию администрации системы.

Избирательная политика безопасности

Основой избирательной политики безопасности является избирательное управление доступом (ИУД, Discretionary Access Control; DAC), которое подразумевает, что:

- все субъекты и объекты системы должны быть идентифицированы;
- права доступа субъекта к объекту системы определяются на основании некоторого внешнего (по отношению к системе) правила (свойство избирательности).

Для описания свойств избирательного управления доступом применяется модель системы на основе матрицы доступа (МД, иногда ее

называют матрицей контроля доступа). Такая модель получила название матричной.

Матрица доступа представляет собой прямоугольную матрицу, в которой объекту системы соответствует строка, а субъекту - столбец. На пересечении столбца и строки матрицы указывается тип (типы) разрешенного доступа субъекта к объекту. Обычно выделяют такие типы доступа субъекта к объекту как "доступ на чтение", "доступ на запись", "доступ на исполнение" и др.

Множество объектов и типов доступа к ним субъекта может изменяться в соответствии с некоторыми правилами, существующими в данной системе. Определение и изменение этих правил также является задачей ИУД. Например, доступ субъекта к конкретному объекту может быть разрешен только в определенные дни (дата - зависимое условие), часы (время - зависимое условие), в зависимости от других характеристик субъекта (контекстно-зависимое условие) или в зависимости от характера предыдущей работы. Такие условия на доступ к объектам обычно используются в СУБД. Кроме того, субъект с определенными полномочиями может передать их другому субъекту (если это не противоречит правилам политики безопасности).

Решение на доступ субъекта к объекту принимается в соответствии с типом доступа, указанным в соответствующей ячейке матрицы доступа. Обычно, избирательное управление доступом реализует принцип "что не разрешено, то запрещено", предполагающий явное разрешение доступа субъекта к объекту.

Матрица доступа - наиболее примитивный подход к моделированию систем, который, однако, является основой для более сложных моделей, наиболее полно описывающих различные стороны реальных КС.

Вследствие больших размеров и разреженности МД хранение полной матрицы представляется нецелесообразным, поэтому во многих средствах защиты используют более экономные представления МД. Каждый из этих

способов представления МД имеет свои достоинства и недостатки, обуславливающие область их применения. Поэтому в каждом конкретном случае надо знать, во-первых, какое именно представление использует средство защиты, и, во-вторых, какие особенности и свойства имеет это представление.

Избирательное управление доступом является основой требований к классам С2 и С1.

Избирательная политика безопасности наиболее широко применяется в коммерческом секторе, так как ее реализация на практике отвечает требованиям коммерческих организаций по разграничению доступа и подотчетности (accountability), а также имеет приемлемую стоимость и небольшие накладные расходы.

Полномочная политика безопасности

Основу полномочной политики безопасности составляет полномочное управление доступом (Mandatory Access Control; MAC), которое подразумевает что:

- все субъекты и объекты системы должны быть однозначно
- идентифицированы;
- каждому объекту системы присвоена метка критичности,
- определяющая ценность содержащейся в нем информации;
- каждому субъекту системы присвоен уровень прозрачности (security clearance), определяющий максимальное значение метки критичности объектов, к которым субъект имеет доступ.

В том случае, когда совокупность меток имеет одинаковые значения, говорят, что они принадлежат к одному уровню безопасности. Организация меток имеет иерархическую структуру и, таким образом, в системе можно реализовать иерархически ненисходящий (по ценности) поток информации (например, от рядовых исполнителей к руководству). Чем важнее объект или

субъект, тем выше его метка критичности. Поэтому наиболее защищенными оказываются объекты с наиболее высокими значениями метки критичности.

Каждый субъект кроме уровня прозрачности имеет текущее значение уровня безопасности, которое может изменяться от некоторого минимального значения до значения его уровня прозрачности.

Для моделирования полномочного управления доступом используется модель Белла-Лападула (Bell-LaPadulla model), включающая в себя понятия безопасного (с точки зрения политики) состояния и перехода. Для принятия решения на разрешение доступа производится сравнение метки критичности объекта с уровнем прозрачности и текущим уровнем безопасности субъекта. Результат сравнения определяется двумя правилами: простым условием защиты (simple security condition) и *-свойством (*-property). В упрощенном виде, они определяют, что информация может передаваться только "наверх", то есть субъект может читать содержимое объекта, если его текущий уровень безопасности не ниже метки критичности объекта, и записывать в него, - если не выше (*-свойство).

Простое условие защиты гласит, что любую операцию над объектом субъект может выполнять только в том случае, если его уровень прозрачности не ниже метки критичности объекта.

Полномочное управление доступом составляет основу требований к классу B1 , где оно используется совместно с избирательным управлением.

Основное назначение полномочной политики безопасности - регулирование доступа субъектов системы к объектам с различным уровнем критичности и предотвращение утечки информации с верхних уровней должностной иерархии на нижние, а также блокирование возможных проникновений с нижних уровней на верхние. При этом она функционирует на фоне избирательной политики, придавая ее требованиям иерархически упорядоченный характер (в соответствии с уровнями безопасности).

Изначально полномочная политика безопасности была разработана в интересах МО США для обработки информации с различными грифами

секретности. Ее применение в коммерческом секторе сдерживается следующими основными причинами:

- отсутствием в коммерческих организациях четкой классификации хранимой и обрабатываемой информации,
- аналогичной государственной классификации (грифы секретности сведений);
- высокой стоимостью реализации и большими накладными расходами.

Управление информационными потоками

Помимо управления доступом субъектов к объектам системы проблема защиты информации имеет еще один аспект.

Как уже отмечалось для того, чтобы получить информацию о каком-либо объекте системы, вовсе не обязательно искать пути несанкционированного доступа к нему. Можно получать информацию, наблюдая за работой системы и, в частности, за обработкой требуемого объекта. Иными словами, при помощи каналов утечки информации. По этим каналам можно получать информацию не только о содержимом объекта, но и о его состоянии, атрибутах и др. в зависимости от особенностей системы и установленной защиты. Эта особенность связана с тем, что при взаимодействии субъекта и объекта возникает некоторый поток информации от субъекта к объекту (информационный поток, information flow)

Информационные потоки существуют в системе всегда. Поэтому возникает необходимость определить, какие информационные потоки в системе являются "легальными", то есть не ведут к утечке информации, а какие - ведут. Таким образом, возникает необходимость разработки правил, регулирующих управление информационными потоками в системе.

Для этого необходимо построить модель системы, которая может описывать такие потоки. Такая модель разработана Гогеном и Мисгаером

(Goguen Meseguer model) и называется потоковой. Модель описывает условия и свойства взаимного влияния (интерференции) субъектов, а также количество информации, полученной субъектом в результате интерференции.

Управление информационными потоками в системе не есть самостоятельная политика, так как оно не определяет правил обработки информации. Управление информационными потоками применяется обычно в рамках избирательной или полномочной политики, дополняя их и повышая надежность системы защиты. В рамках полномочной политики оно является основой требований к классу B2 стандарта "Оранжевая книга".

Управление доступом (избирательное или полномочное) сравнительно легко реализуемо (аппаратно или программно), однако оно неадекватно реальным КС из-за существования в них скрытых каналов. Тем не менее управление доступом обеспечивает достаточно надежную защиту в простых системах, не обрабатывающих особо важную информацию. В противном случае средства защиты должны дополнительно реализовывать управление информационными потоками. Организация такого управления в полном объеме достаточна сложна, поэтому его обычно используют для усиления надежности полномочной политики: восходящие (относительно уровней безопасности) информационные потоки считаются разрешенными, все остальные - запрещенными.

Отметим, что кроме способа управления доступом политика безопасности включает еще и другие требования, такие как подотчетность, гарантии и т.д.

Избирательное и полномочное управление доступом, а также управление информационными потоками - своего рода три кита, на которых строится вся защита.

Достоверная вычислительная база

Для того, чтобы корректно воплотить в жизнь разработанную политику безопасности необходимо иметь надежные механизмы ее реализации. При

последующем изложении материала основное внимание обратим на то, каким образом должны быть реализованы средства защиты для выполнения требований политики безопасности (способа управления доступом).

Естественно предположить, что все средства, отвечающие за реализацию политики безопасности, сами должны быть защищены от любого вмешательства в их работу. В противном случае говорить о надежности защиты будет трудно. Можно изменять их параметры, но в своей основе они должны оставаться в неприкосновенности.

Поэтому все средства защиты и управления должны быть объединены в так называемую достоверную вычислительную базу.

Достоверная вычислительная база (ДВБ; Trusted Computing Base; TCB) - это абстрактное понятие, обозначающее полностью защищенный механизм вычислительной системы (включая аппаратные и программные средства), отвечающий за поддержку реализации политики безопасности.

Средства защиты должны создавать ДВБ для обеспечения надежной защиты КС. В различных средствах защиты ДВБ может быть реализована по-разному. Способность реализации ДВБ к безотказной работе зависит от ее устройства и корректного управления, а ее надежность является залогом соблюдения политики безопасности в защищаемой системе.

Таким образом, ДВБ выполняет двойную задачу - поддерживает реализацию политики безопасности и является гарантом целостности механизмов защиты, то есть самой себя. ДВБ совместно используется всеми пользователями КС, однако ее модификация разрешена только пользователям со специальными полномочиями. К ним относятся администраторы системы и другие привилегированные сотрудники организации.

Процесс, функционирующий от имени ДВБ, является достоверным. Это означает, что система защиты безоговорочно доверяет этому процессу и все его действия санкционированы политикой безопасности. Именно поэтому задача номер один защиты ДВБ - поддержание собственной

целостности; все программы и наборы данных ДВБ, должны быть надежно защищены от несанкционированных изменений.

Для поддержки политики безопасности и собственной защиты ДВБ должна обеспечить защиту субъектов (процессов) системы и защиту объектов системы в оперативной памяти и на внешних носителях.

Защита ДВБ строится на основе концепции иерархической декомпозиции системы. Сущность концепции заключается в том, что реальная система представляется как совокупность иерархически упорядоченных абстрактных уровней; при этом функции каждого уровня реализуются компонентами более низкого уровня. Компоненты определенного уровня зависят только от компонентов более низких уровней и их внутренняя структура полагается недоступной с более высоких уровней. Связь уровней организуется через межуровневый интерфейс (см. рис. 1.1).

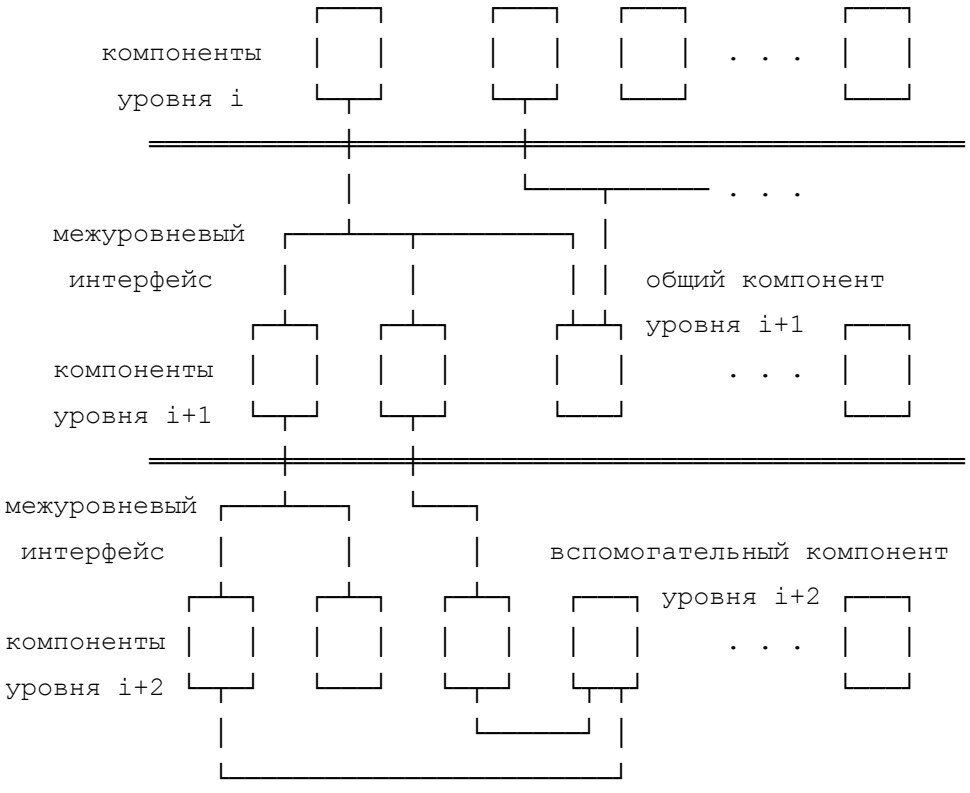


Рисунок 1.1 – Структура компонентов системы

Структура компонентов системы и связи между ними являются жестко фиксированными; их изменение, дублирование, уничтожение невозможны. Компоненты более высоких уровней привязаны к компонентам более низких уровней, те, в свою очередь, к элементам физической реализации (устройствам ввода-вывода, процессору и др.). Связи между различными компонентами определяются спецификациями межуровневого интерфейса и также не могут изменяться. Это является дополнительной мерой обеспечения целостности ДВБ.

Компоненты верхних уровней обычно описывают интерфейс пользователя. Сюда входят различные редакторы, компиляторы, интерпретаторы командных языков, утилиты и т.д. Средние уровни обычно реализуют ввод-вывод на уровне записей, работу с файлами и виртуальной памятью. Компоненты нижних уровней реализуют планирование и диспетчеризацию процессов, распределение ресурсов, ввод-вывод на физическом уровне, обработку прерываний и т.д. Компонентами нулевого уровня можно считать элементы физической реализации: особенности архитектуры процессора, состав и назначение регистров (общих и привилегированных), физическую реализацию некоторых функций и т.д. Множество компонентов всех уровней, кроме верхнего, а также средства управления ими и составляют ДВБ.

Пользователь, находясь на самом высоком уровне, может только послать запрос на выполнение какой-либо операции. Этот запрос будет разрешен к выполнению компонентами более низких уровней только в том случае, если, пройдя обработку корректности на всех промежуточных уровнях, он не был отвергнут, то есть не сможет нарушить существующую политику безопасности. При этом каждая функция может быть выполнена только определенными компонентами на определенном уровне, что определяется архитектурой системы в целом.

Например, пользователь из командного интерпретатора послал запрос на выполнение операции ввода-вывода (для редактирования файла,

размещающегося на диске). Этот запрос будет обработан интерпретатором и передан на более низкий уровень - в подсистему ввода-вывода. Та проверит корректность запроса (разрешен ли доступ к этому файлу?), обработает его и передаст дальше - примитивам ввода-вывода, которые выполнят операцию и сообщат о результатах. При этом спецификации межуровневого интерфейса гарантируют, что прямой вызов примитивов ввода-вывода пользователю недоступен. Он еще может иногда обращаться непосредственно к подсистеме ввода-вывода (из программы), но не на более низкий уровень. Таким образом, гарантируется невозможность доступа субъекта к объекту в обход средств контроля.

Необходимость защиты внутри отдельных компонентов системы очевидна: каждый из них должен проверять корректность обращения к реализуемой им функции.

Особенность применения концепции иерархической декомпозиции заключается в следующем:

1. Каждый компонент должен выполнять строго определенную функцию;

2. Каждая функция с помощью операции декомпозиции может быть разбита на ряд подфункций, которые реализуются и защищаются отдельно. Этот процесс может насчитывать несколько этапов;

3. Основная "тяжесть" защиты приходится на межуровневый интерфейс, связывающий декомпозированные подфункции в единое целое; горизонтальные ссылки должны быть сведены до минимума.

Помимо защиты самой себя ДВБ также должна обеспечить надежную защиту пользователей системы (в частности, друг от друга). Для защиты пользователей используются те же самые механизмы, что и для защиты ДВБ. Теми же остаются и цели защиты: субъектов и объектов пользователей, в оперативной памяти и на внешних носителях. Рассмотрим подробнее принципы такой защиты.

Защита субъектов осуществляется с помощью межуровневого интерфейса: в зависимости от выполняемой им функции система переводит

его на соответствующий уровень. Уровень, в свою очередь, определяет и степень управляемости процесса пользователем, который находится на самом верхнем уровне – чем ниже уровень процесса, тем меньше он управляем с более верхних уровней и тем больше он зависит от ОС.

Любые попытки защиты оперативной памяти приводят к необходимости создания виртуальной памяти в том или ином виде. Здесь используется та же концепция иерархической декомпозиции, чтобы отделить реальную память, содержащую информацию, от той, которая доступна пользователям. Соответствие между виртуальной и физической памятью обеспечивается диспетчером памяти. При этом различные области памяти могут являться компонентами разных уровней - это зависит от уровня программ, которые могут обращаться к этим областям.

Пользователи и их программы могут работать только с виртуальной памятью. Доступ к любому участку физической оперативной памяти (в том числе и принадлежащему ДВБ), контролируется диспетчером памяти. При трансляции виртуального адреса в физический проверяются права доступа к указанному участку. Надежность разделения оперативной памяти во многом обеспечивается за счет надежности функции, отображающей виртуальные адреса в физические: адресные пространства различных пользователей и системы не должны перекрываться в физической памяти.

Доступ к информации на внешних носителях осуществляется с помощью подсистемы ввода-вывода; программы этой подсистемы являются компонентами нижних и средних уровней ДВБ. При получении имени файла (адреса записи) в первую очередь проверяются полномочия пользователя на доступ к запрашиваемым данным. Принятие решение на осуществление доступа осуществляется на основе информации, хранящейся в базе данных защиты. Сама база данных является частью ДВБ, доступ к ней также контролируются.

ДВБ должна быть организована таким образом, чтобы только ее компоненты могли выполнить запрос, причем только тот, который содержит корректные параметры.

Одним из необходимых условий реализации ДВБ в средствах защиты является наличие мультирежимного процессора (то есть процессора, имеющего привилегированный и обычный режим работы) с аппаратной поддержкой механизма переключения режимов, и различных способов реализации виртуальной памяти.

Достоверная вычислительная база состоит из ряда механизмов защиты, позволяющих ей обеспечивать поддержку реализации политики безопасности.

1.5 Механизмы защиты

В этом пункте будут рассмотрены механизмы защиты и их свойства, входящие в состав ДВБ и обеспечивающие поддержку политики безопасности. Основное внимание уделим механизмам реализации избирательной политики безопасности поскольку, во-первых, она является основной для коммерческого сектора, а во-вторых, базовые механизмы поддержки этой политики также используются как для поддержки полномочной политики, так и для управления информационным потоком.

Основой ДВБ является ядро безопасности (security kernel) - элементы аппаратного и программного обеспечения, защищенные от модификации и проверенные на корректность, которые разделяют все попытки доступа субъектов к объектам.

Ядро безопасности является реализацией концепции монитора ссылок (reference monitor) - абстрактной концепции механизма защиты.

Помимо ядра безопасности ДВБ содержит другие механизмы, отвечающие за жизнедеятельность системы. К ним относятся планировщики процессов, диспетчеры памяти, программы обработки прерываний, примитивы

ввода-вывода и др. программно-аппаратные средства, а также системные наборы данных.

Под монитором ссылок понимают концепцию контроля доступа субъектов к объектам в абстрактной машине. Схематически монитор ссылок изображен на рис. 1.2.

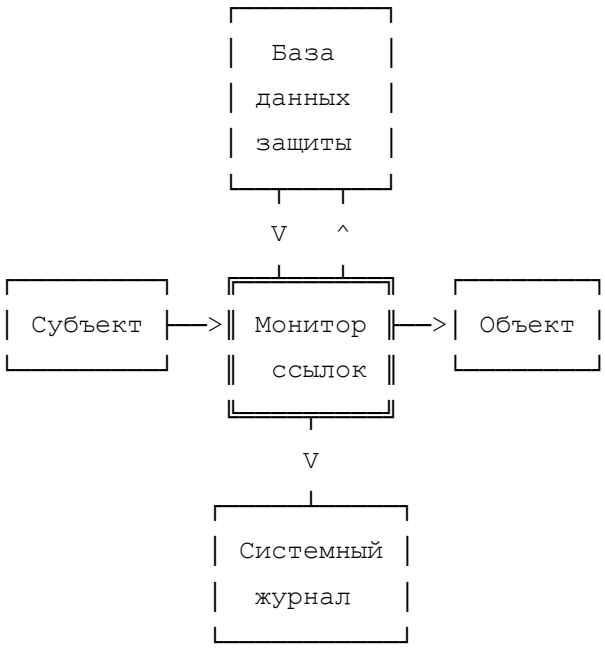


Рисунок 1.2 - Монитор ссылок

Под базой данных защиты (security database) понимают базу данных, хранящую информацию о правах доступа субъектов системы к объектам. Основу базы данных защиты составляет матрица доступа или ее представления, которая служит основой избирательной политики безопасности.

Любая операционная система, поддерживающая ИУД, использует МД и операции над ней, поскольку МД - удобный инструмент контроля использования и передачи привилегий. Однако, вследствие больших размеров и разреженности МД, хранение полной матрицы представляется

нецелесообразным, поэтому во многих системах используют более экономные представления МД: по строкам, по столбцам, поэлементно.

Рассмотрим эти способы более подробно:

1. Профиль (profile).

Профилем называется список защищаемых объектов системы и прав доступа к ним, ассоциированный с каждым субъектом. При обращении к объекту профиль субъекта проверяется на наличие соответствующих прав доступа. Таким образом МД представляется своими строками.

В системах с большим количеством объектов профили могут иметь большие размеры и, вследствие этого, ими трудно управлять; изменение профилей нескольких субъектов может потребовать большого количества операций и привести к трудностям в работе системы. Поэтому профили обычно используются лишь администраторами безопасности для контроля работы субъектов, и даже такое их применение весьма ограничено.

2. Список контроля доступа (access control list).

Это представление МД по столбцам - каждому объекту соответствует список субъектов вместе с их правами. В современных условиях списки контроля доступа (СКД) - лучшее направление реализации ИУД, поскольку это очень гибкая структура, предоставляющая пользователям много возможностей.

3. Мандат или билет (capability или ticket).

Это элемент МД, определяющий тип доступа определенного субъекта к определенному объекту (т.е. субъект имеет "билет" на доступ к объекту). Каждый раз билет выдается субъекту динамически - при запросе доступа, и так же динамически билет может быть изъят у субъекта. Поскольку распространение билетов происходит очень динамично, и они могут размещаться непосредственно внутри объектов, то вследствие этого контроль за ним очень затруднен. В чистом виде билетный механизм хранения и передачи привилегий используется редко. Однако реализация других

механизмов присвоения привилегий (например с использованием СКД) часто осуществляется с помощью билетов.

При реализации полномочной политики безопасности база данных защиты также содержит метки критичности всех объектов и уровни прозрачности субъектов системы.

Монитор ссылок должен выполнять следующие функции:

1. Проверять права доступа каждого субъекта к любому объекту на основании информации, содержащейся в базе данных защиты и положений политики безопасности (избирательной или полномочной);

2. При необходимости регистрировать факт доступа и его параметры в системном журнале.

Реализующее монитор ссылок ядро безопасности должно обладать следующими свойствами:

- контролировать все попытки доступа субъектов к объектам;
- иметь защиту от модификации, подделки, навязывания;
- быть протестировано и верифицировано для получения гарантий надежности;
- иметь небольшой размер и компактную структуру.

В терминах модели Белла-Лападулла (избирательной и полномочной политик безопасности) монитор ссылок должен контролировать состояния системы и переходы из одного в другое. Основными функциями, которые должно выполнять ядро безопасности совместно с другими службами ОС, являются:

1. Идентификация, аутентификация и авторизация субъектов и объектов системы.

Эти функции необходимы для подтверждения подлинности субъекта, законности его прав на данный объект или на определенные действия, а также для обеспечения работы субъекта в системе.

Идентификация - процесс распознавания элемента системы, обычно с помощью заранее определенного идентификатора или другой априорной информации; каждый субъект или объект должен быть однозначно идентифицируем.

Аутентификация - проверка идентификации пользователя, процесса, устройства или другого компонента системы (обычно осуществляется перед разрешением доступа); а также проверка целостности данных при их хранении или передаче для предотвращения несанкционированной модификации.

Авторизация - предоставление субъекту прав на доступ к объекту.

Эти функции необходимы для поддержания разрешительного порядка доступа к системе и соблюдения политики безопасности: авторизованный (разрешенный) доступ имеет только тот субъект, чей идентификатор удовлетворяет результатам аутентификации. Они выполняются как в процессе работы (при обращении к наборам данных, устройствам, ресурсам), так и при входе в систему. Во втором случае имеются отличия, которые мы рассмотрим в следующем пункте.

2. Контроль входа пользователя в систему и управление паролями.

Эти функции являются частным случаем перечисленных выше: при входе в систему и вводе имени пользователя осуществляется идентификация, при вводе пароля - аутентификация и, если пользователь с данными именем и паролем зарегистрирован в системе, ему разрешается доступ к определенным объектам и ресурсам (авторизация). Однако при входе в систему существуют отличия при выполнении этих функций. Они обусловлены тем, что в процессе работы система уже имеет информацию о том, кто работает, какие у него полномочия (на основе информации в базе данных защиты) и т.д. и поэтому может адекватно реагировать на запросы субъекта. При входе в систему это все только предстоит определить. В данном случае возникает необходимость организации "достоверного маршрута" (trusted path) - пути передачи идентифицирующей информации от

пользователя к ядру безопасности для подтверждения подлинности. Как показывает практика, вход пользователя в систему - одно из наиболее уязвимых мест защиты; известно множество случаев взлома пароля, входа без пароля, перехвата пароля и т.д. Поэтому при выполнении входа и пользователь, и система должны быть уверены, что они работают непосредственно друг с другом, между ними нет других программ и вводимая информация истинна.

Достоверный маршрут реализуется привилегированными процедурами ядра безопасности, чья работа обеспечивается механизмами ДВБ, а также некоторыми другими механизмами, выполняющими вспомогательные функции. Они проверяют, например, что терминал, с которого осуществляется вход в систему, не занят никаким другим пользователем, который имитировал окончание работы.

3. Регистрация и протоколирование. Аудит.

Эти функции обеспечивают получение и анализ информации о состоянии ресурсов системы с помощью специальных средств контроля, а также регистрацию действий, признанных администрацией потенциально опасными для безопасности системы. Такими средствами могут быть различные системные утилиты или прикладные программы, выводящие информацию непосредственно на системную консоль или другое определенное для этой цели устройство, а также системный журнал. Кроме того, почти все эти средства контроля могут не только обнаружить какое-либо событие, но и фиксировать его. Например, большинство систем имеет средства протоколирования сеансов работы отдельных пользователей (всего сеанса или его отдельных параметров).

Большинство систем защиты имеют в своем распоряжении средства управления системным журналом (audit trail). Как было показано выше, системный журнал является составной частью монитора ссылок и служит для контроля соблюдения политики безопасности. Он является одним из

основных средств контроля, помогающим администратору предотвращать возможные нарушения в связи с тем, что:

- способен оперативно фиксировать происходящие в системе события;
- может помочь выявить средства и априорную информацию, использованные злоумышленником для нарушения;
- может помочь определить, как далеко зашло нарушение, подсказать метод его расследования и способы исправления ситуации.

Содержимое системного журнала и других наборов данных, хранящих информацию о результатах контроля, должны подвергаться периодическому просмотру и анализу (аудит) с целью проверки соблюдения политики безопасности.

4. Противодействие "сборке мусора".

После окончания работы программы обрабатываемая информация не всегда полностью удаляется из памяти. Части данных могут оставаться в оперативной памяти, на дисках и лентах, других носителях. Они хранятся на диске до перезаписи или уничтожения. При выполнении этих действий на освободившемся пространстве диска находятся их остатки.

Хотя при искажении заголовка файла эти остатки прочитать трудно, однако, используя специальные программы и оборудование, такая возможность все-таки имеется. Этот процесс называется "сборкой мусора" (disk scavenging). Он может привести к утечке важной информации.

Для защиты от "сборки мусора" используются специальные средства, которые могут входить в ядро безопасности ОС или устанавливаться дополнительно.

5. Контроль целостности субъектов.

Согласно модели Белла-Лападулла множество субъектов системы есть подмножество множества объектов, то есть каждый субъект одновременно

является объектом. При этом под содержимым субъекта обычно понимают содержимое контекста процесса, куда входит содержимое общих и специальных регистров (контекст процесса постоянно изменяется). Кроме содержимого или значения субъект имеет ряд специфических атрибутов приоритет, список привилегий, набор идентификаторов и др. характеристики. В этом смысле поддержание целостности субъекта, то есть предотвращение его несанкционированной модификации, можно рассматривать как частный случай этой задачи для объектов вообще.

В то же время субъект отличается от объекта тем, что является, согласно определению, активным компонентом системы. В связи с этим для защиты целостности субъекта, в качестве представителя которого выступает процесс, вводится такое понятие как рабочая среда или область исполнения процесса. Эта область является логически защищенной подсистемой, которой доступны все ресурсы системы, относящиеся к соответствующему процессу. Другими словами, область исполнения процесса является виртуальной машиной. В рамках этой области процесс может выполнять любые санкционированные действия без опасения нарушения целостности. Таким образом, реализуется концепция защищенной области для отдельного процесса.

Контроль целостности обеспечивается процедурами ядра безопасности, контролируемыми механизмами поддержки ДВБ. Основную роль играют такие механизмы, как поддержка виртуальной памяти (для создания области данного процесса) и режим исполнения процесса (определяет его возможности в рамках данной области и вне ее).

Область исполнения процесса может содержать или вкладываться в другие подобласти, которые составляют единую иерархическую структуру системы. Процесс может менять области: это действие называется переключением области процесса (process switching). Оно всегда связано с переходом центрального процессора в привилегированный режим работы.

Механизмы поддержки областей исполнения процесса обеспечивают контроль их целостности достаточно надежно. Однако даже разделенные процессы должны иметь возможность обмениваться информацией. Для этого разработаны несколько специальных механизмов, чтобы можно было осуществлять обмен информацией между процессами без ущерба безопасности или целостности каждого из них. К таким механизмам относятся, например, кластеры флагов событий, почтовые ящики и другие системные структуры данных. Следует однако учитывать, что с их помощью может осуществляться утечка информации, поэтому если использование таких механизмов разрешено, их обязательно следует контролировать.

6. Контроль доступа.

Под контролем доступа будем понимать ограничение возможностей использования ресурсов системы программами, процессами или другими системами (для сети) в соответствии с политикой безопасности. Под доступом понимается выполнение субъектом некоторой операции над объектом из множества разрешенных для данного типа. Примерами таких операций являются чтение, открытие, запись набора данных, обращение к устройству и т.д.

Контроль должен осуществляться при доступе к:

- оперативной памяти;
- разделяемым устройствам прямого доступа;
- разделяемым устройствам последовательного доступа;
- разделяемым программам и подпрограммам;
- разделяемым наборам данных.

Основным объектом внимания средств контроля доступа являются совместно используемые наборы данных и ресурсы системы. Совместное использование объектов порождает ситуацию "взаимного недоверия" при которой разные пользователи одного объекта не могут до конца доверять друг

другу. Тогда, если с этим объектом что-нибудь случиться, все они попадают в круг подозреваемых.

Существует четыре основных способа разделения субъектов к совместно используемым объектам:

1. Физическое - субъекты обращаются к физически различным объектам (однотипным устройствам, наборам данных на разных носителях и т.д.).

2. Временное - субъекты с различными правами доступа к объекту получают его в различные промежутки времени.

3. Логическое - субъекты получают доступ к совместно используемому объекту в рамках единой операционной среды, но под контролем средств разграничения доступа, которые моделируют виртуальную операционную среду "один субъект - все объекты"; в этом случае разделение может быть реализовано различными способами разделение оригинала объекта, разделение с копированием объекта и т.д.

4. Криптографическое - все объекты хранятся в зашифрованном виде, права доступа определяются наличием ключа для расшифрования объекта.

Существует множество различных вариантов одних и тех же способов разделения субъектов, они могут иметь разную реализацию в различных средствах защиты.

Контроль доступа субъектов системы к объектам (не только к совместно используемым, но и к индивидуальным) реализуется с помощью тех же механизмов, которые реализуют ДВБ и осуществляется процедурами ядра безопасности.

Принципы реализации политики безопасности

Как уже отмечалось выше, настройка механизмов защиты дело сугубо индивидуальное для каждой системы и даже для каждой задачи. Поэтому дать ее подробное описание довольно трудно. Однако существуют общие

принципы, которых следует придерживаться, чтобы облегчить себе работу, так как они проверены практикой. Рассмотрим их.

1. Группирование.

Это объединение множества субъектов под одним групповым именем; всем субъектам, принадлежащим одной группе, предоставляются равные права. Принципы объединения пользователей в группы могут быть самые разные: ссылки на одни и те же объекты, одинаковый характер вычислений, работа над совместным проектом и т.д. При этом один и тот же субъект может входить в несколько различных групп, и, соответственно, иметь различные права по отношению к одному и тому же объекту.

Механизм группирования может быть иерархическим. Это означает, что каждый субъект является членом нескольких групп, упорядоченных по отношению "быть подмножеством". Контроль за состоянием групп очень важен, поскольку члены одной группы имеют доступ к большому числу объектов, что не способствует их безопасности. Создание групп и присвоение групповых привилегий должно производиться администратором безопасности, руководителем группы или каким-либо другим лицом, несущим ответственность за сохранность групповых объектов.

2. Правила умолчания.

Большое внимание при назначении привилегий следует уделять правилам умолчания, принятым в данных средствах защиты; это необходимо для соблюдения политики безопасности. Во многих системах, например, субъект, создавший объект и являющийся его владельцем, по умолчанию получает все права на него. Кроме того, он может эти права передавать кому-либо.

В различных средствах защиты используются свои правила умолчания, однако принципы назначения привилегий по умолчанию в большинстве систем одни и те же. Если в системе используется древовидная файловая структура, то необходимо принимать во внимание правила умолчания для каталогов.

Корректное использование правил умолчания способствуют поддержанию целостности политики безопасности.

3. Минимум привилегий.

Это один из основополагающих принципов реализации любой политики безопасности, используемый повсеместно. Каждый пользователь и процесс должен иметь минимальное число привилегий, необходимое для работы. Определение числа привилегий для всех пользователей, с одной стороны, позволяющих осуществлять быстрый доступ ко всем необходимым для работы объектам, а, с другой, - запрещающих доступ к чужим объектам - проблема достаточно сложная. От ее решения во многом зависит корректность реализации политики безопасности.

4. "Надо знать".

Этот принцип во многом схож с предыдущим. Согласно ему, полномочия пользователей назначаются согласно их обязанностям. Доступ разрешен только к той информации, которая необходима им для работы.

5. Объединение критичной информации.

Во многих системах сбор, хранение и обработка информации одного уровня производится в одном месте (узле сети, устройстве, каталоге). Это связано с тем, что проще защитить одним и тем же способом большой массив информации, чем организовывать индивидуальную защиту для каждого набора.

Для реализации этого принципа могут быть разработаны специальные программы, управляющие обработкой таких наборов данных. Это будет простейший способ построения защищенных областей.

6. Иерархия привилегий.

Контроль объектов системы может иметь иерархическую организацию. Такая организация принята в большинстве коммерческих систем.

При этом схема контроля имеет вид дерева, в котором узлы - субъекты системы, ребра - право контроля привилегий согласно иерархии, корень -

администратор системы, имеющий право изменять привилегии любого пользователя (см. рис.1.3).

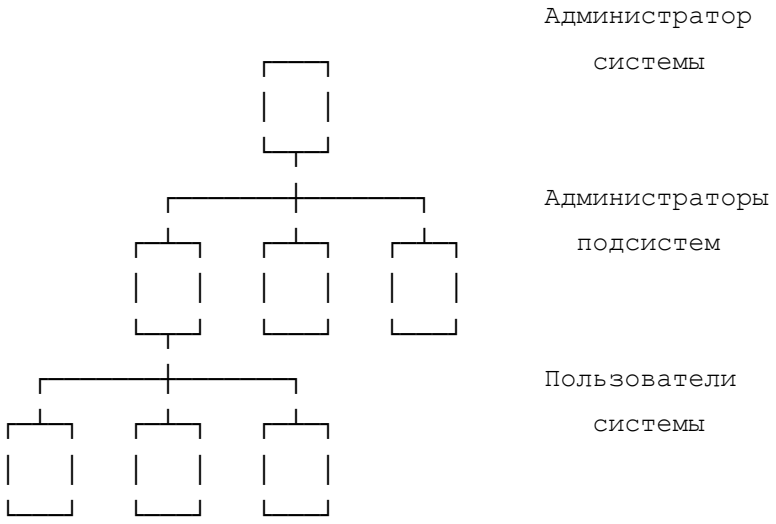


Рисунок 1.3 – Схема контроля объектов системы

Узлами нижележащих уровней являются администраторы подсистем, имеющие права изменять привилегии пользователей этих подсистем (в их роли могут выступать руководители организаций, отделов). Листьями дерева являются все пользователи системы. Вообще говоря, субъект, стоящий в корне любого поддерева, имеет право изменять защиту любого субъекта, принадлежащего этому поддереву.

Достоинство такой структуры - точное копирование схемы организации, которую обслуживает КС. Поэтому легко составить множество субъектов, имеющих право контролировать данный объект. Недостаток иерархии привилегий - сложность управления доступом при большом количестве субъектов и объектов, а также возможность получения доступа администратора системы (как высшего по иерархии) к любому набору данных.

7. Привилегии владельца.

При таком контроле каждому объекту соответствует единственный субъект с исключительным правом контроля объекта - владелец (owner). Как

правило, это его создатель. Владелец обладает всеми разрешенными для этого типа данных правами на объект, может разрешать доступ любому другому субъекту, но не имеет права никому передать привилегию на корректировку защиты. Однако такое ограничение не касается администраторов системы - они имеют право изменять защиту любых объектов.

Главным недостатком принципа привилегий владельца является то, что при обращении к объекту, пользователь должен предварительно получить разрешение у владельца (или администратора). Это может приводить к сложностям в работе (например, при отсутствии владельца или просто нежелании его разрешить доступ). Поэтому такой принцип обычно используется при защите личных объектов пользователей.

8. Свободная передача привилегий.

При такой схеме субъект, создавший объект, может передать любые права на него любому другому субъекту вместе с правом корректировки СКД этого объекта. Тот, в свою очередь, может передать все эти права другому субъекту.

Естественно, при этом возникают большие трудности в определении круга субъектов, имеющих в данный момент доступ к объекту (права на объект могут распространяться очень быстро и так же быстро исчезать), и поэтому такой объект легко подвергнуть несанкционированной обработке. В силу этих обстоятельств подобная схема применяется достаточно редко - в основном в исследовательских группах, работающих над одним проектом (когда все имеющие доступ к объекту заинтересованы в его содержимом).

В чистом виде рассмотренные принципы реализации политики безопасности применяются редко. Обычно используются их различные комбинации. Ограничение доступа к объектам в ОС включает в себя ограничение доступа к некоторым системным возможностям, например, ряду команд, программам и т.д., если при использовании их нарушается политика безопасности. Вообще набор полномочий каждого пользователя должен быть

тщательно продуман, исключены возможные противоречия и дублирования, поскольку большое количество нарушений происходит именно из-за этого. Может произойти утечка информации без нарушения защиты, если плохо была спроектирована или реализована политика безопасности.

Политика безопасности и механизмы поддержки ее реализации образуют единую защищенную среду обработки информации. Эта среда имеет иерархическую структуру, где верхние уровни представлены требованиями политики безопасности, далее следует интерфейс пользователя, затем идут несколько программных уровней защиты (включая уровни ОС) и, наконец, нижний уровень этой структуры представлен аппаратными средствами защиты. На всех уровнях, кроме верхнего, должны реализовываться требования политики безопасности, за что, собственно, и отвечают механизмы защиты.

В различных системах механизмы защиты могут быть реализованы по-разному; их конструкция определяется общей концепцией системы. Однако одно требование должно выполняться неукоснительно: эти механизмы должны адекватно реализовывать требования политики безопасности.

Основные критерии оценки безопасности систем

Для оценки надежности средств защиты применяются различные критерии оценки. Анализ некоторых критериев показал общность идеи, лежащей в основе подхода к оценке безопасности (степени защищенности) компьютерных систем. Ее сущность состоит в следующем. Для предоставления пользователям возможности обоснованного выбора средств защиты вводится некая система классификации их свойств. Задается иерархия функциональных классов безопасности. Каждому классу соответствует определенная совокупность обязательных функций. Конкретное средство разграничения доступа относится к такому классу безопасности, в котором реализованы все соответствующие ему функции безопасности, если оно не может быть отнесено к более высокому классу.

В разных странах за разработку этих документов и проверку средств разграничения доступа на соответствие им, отвечают различные организации. Например, в США это уже упоминаемый ранее Национальный Центр Компьютерной Безопасности, в России это Государственная техническая комиссия при Президенте Российской Федерации (в дальнейшем просто ГТК РФ).

Система документов России

Руководящие документы (в некоторой степени аналогичные разработанным NSCS) в области защиты информации разработаны ГТК РФ. Требования всех приведенных ниже документов обязательны для исполнения только в государственном секторе, либо коммерческими организациями, которые обрабатывают информацию, содержащую государственную тайну.

Для остальных коммерческих структур документы носят рекомендательно-консультативный характер. Все вопросы криптографической защиты информации находятся в компетенции Федерального агентства правительственной связи и информации Руководящие документы ГТК РФ включают:

1) Концепцию защиты средств вычислительной техники и автоматизированных систем от несанкционированного доступа (НСД) к информации. Этот документ содержит определение НСД, основные способы осуществления НСД, модель нарушителя, основные направления и принципы организации работ по защите информации от НСД;

2) Термины и определения в области защиты от НСД к информации. Этот документ вводит в действие основные термины и определения, используемые в других документах;

3) Показатели защищенности СВТ от НСД к информации. Этот документ устанавливает классификацию СВТ по уровню защищенности от НСД к

информации на базе перечня показателей защищенности и совокупности предъявляемым к ним требованиям;

4) Классификацию автоматизированных систем и требования по защите информации. Документ устанавливает классификацию автоматизированных систем (АС), подлежащих защите от НСД к информации, и требования по защите информации в АС различных классов.

5) Временное положение о государственном лицензировании деятельности в области защиты информации. Документ устанавливает основные принципы, организационную структуру системы лицензирования деятельности предприятий в сфере оказания услуг в области защиты информации, а также правила осуществления лицензирования и надзора за деятельностью предприятий, получивших лицензию.

1.6 Контрольные вопросы

1. Какие свойства присущи информации?
2. Дайте понятие объекта защиты информации.
3. Что относят к информационным процессам?
4. Что понимают под информационной системой?
5. Что называют информационными ресурсами?
6. Что понимают под угрозой информации, дайте понятие искусственных и естественных угроз, приведите примеры.
7. Что составляет основу политики безопасности?
8. Сделайте сравнительный анализ избирательной и полномочной политики безопасности.
9. Проанализируйте механизмы и свойства защиты информации.

2. ИДЕНТИФИКАЦИЯ ПОЛЬЗОВАТЕЛЕЙ КС-СУБЪЕКТОВ ДОСТУПА К ДАННЫМ

2.1. Основные понятия и концепции

С каждым объектом компьютерной системы (КС) связана некоторая информация, однозначно идентифицирующая его. Это может быть *число, строка символов, алгоритм,* определяющий данный объект. Эту информацию называют *идентификатором объекта.* Если объект имеет некоторый идентификатор, зарегистрированный в сети, он называется законным (легальным) объектом; остальные объекты относятся к незаконным (нелегальным).

Идентификация объекта - одна из функций подсистемы защиты. Эта функция выполняется в первую очередь, когда объект делает попытку войти в сеть. Если процедура идентификации завершается успешно, данный объект считается законным для данной сети.

Следующий шаг-аутентификация объекта (проверка подлинности объекта). Эта процедура устанавливает, является ли данный объект именно таким, каким он себя объявляет.

После того как объект идентифицирован и подтверждена его подлинность, можно установить сферу его действия и доступные ему ресурсы КС. Такую процедуру называют *предоставлением полномочий (авторизацией).*

Перечисленные три процедуры инициализации являются процедурами защиты и относятся к одному объекту КС.

При защите каналов передачи данных *подтверждение подлинности* (аутентификация) объектов означает взаимное установление подлинности объектов, связывающихся между собой по линиям связи. Процедура подтверждения подлинности выполняется обычно в начале сеанса в процессе установления соединения абонентов. (Термин "соединение" указывает на логическую связь (потенциально двустороннюю) между двумя объектами сети.

Цель данной процедуры - обеспечить уверенность, что соединение установлено с законным объектом и вся информация дойдет до места назначения.

После того как соединение установлено, необходимо обеспечить выполнение требований защиты при обмене сообщениями:

(а) получатель должен быть уверен в подлинности источника данных;

(б) получатель должен быть уверен в подлинности передаваемых данных;

(в) отправитель должен быть уверен в доставке данных получателю;

(г) отправитель должен быть уверен в подлинности доставленных данных.

Для выполнения требований (а) и (б) средством защиты является *цифровая подпись.* Для выполнения требований (в) и (г) отправитель должен получить *уведомление о вручении* с помощью удостоверяющей почты (certified mail). Средством защиты в такой процедуре является цифровая подпись подтверждающего ответного сообщения, которое в свою очередь является доказательством пересылки исходного сообщения.

Если эти четыре требования реализованы в КС, то гарантируется защита данных при их передаче по каналу связи и обеспечивается функция защиты, называемая функцией подтверждения (неоспоримости) передачи. В этом случае отправитель не может отрицать ни факта посылки сообщения, ни его содержания, а получатель не может отрицать ни факта получения сообщения, ни подлинности его содержания.

2.2. Идентификация и аутентификация пользователя

Прежде чем получить доступ к ресурсам компьютерной системы, пользователь должен пройти процесс представления компьютерной системе, который включает две стадии:

- идентификацию - пользователь сообщает системе по ее запросу свое имя (идентификатор);

- аутентификацию - пользователь подтверждает идентификацию, вводя в систему уникальную, не известную другим пользователям информацию о себе (например, пароль).

Для проведения процедур идентификации и аутентификации пользователя необходимы:

- наличие соответствующего *субъекта (модуля) аутентификации;*
- наличие *аутентифицирующего объекта,* хранящего уникальную информацию для аутентификации пользователя.

Различают две формы представления объектов, аутентифицирующих пользователя:

- внешний аутентифицирующий объект, не принадлежащий системе;
- внутренний объект, принадлежащий системе, в который переносится информация из внешнего объекта.

Внешние объекты могут быть технически реализованы на различных носителях информации - магнитных дисках, пластиковых картах и т. п. Естественно, что внешняя и внутренняя формы представления аутентифицирующего объекта должны быть семантически тождественны.

Типовые схемы идентификации и аутентификации пользователя

Рассмотрим структуры данных и протоколы идентификации и аутентификации пользователя. Допустим, что в компьютерной системе зарегистрировано n пользователей. Пусть i-й аутентифицирующий объект i-го пользователя содержит два информационных поля:

ID_i-неизменный идентификатор i-го пользователя, который является аналогом имени и используется для идентификации пользователя;

Ki-аутентифицирующая информация пользователя, которая может изменяться и служит для аутентификации (например, пароль $P_i=K_i$).

Описанная структура соответствует практически любому ключевому носителю информации, используемому для опознания пользователя. Например,

для носителей типа пластиковых карт выделяется неизменяемая информация ID_i первичной персонализации пользователя и объект в файловой структуре карты, содержащий $К_i$.

Совокупную информацию в ключевом носителе можно назвать первичной аутентифицирующей информацией i-го пользователя! Очевидно, что внутренний аутентифицирующий объект не должен существовать в системе длительное время (больше времени работы конкретного пользователя). Для длительного хранения следует использовать данные в защищенной форме.

Рассмотрим две типовые схемы идентификации и аутентификации.

Схема 1. В компьютерной системе выделяется объект-эталон для идентификации и аутентификации пользователей. Структура объекта-эталона для схемы 1 показана в табл. 5.1. Здесь $E_i=F(ID_i\ K_i)$, где F-функция, которая обладает свойством "невосстановимости" значения $К_i$ по E_i и ID_i. "Невоостановимость" K_i оценивается некоторой пороговой трудоемкостью T_0 решения задачи восстановления аутентифицирующей информации $К_i$ по E_i и ID_i. Кроме того, для пары K_i и Kj возможно совпадение соответствующих значений E. В связи с этим вероятность лажной аутентификации пользователя не должна быть больше некоторого порогового значения P_0.

На практике задают $T_o=10^{20}.,,10^{30}$, $P_o=10^{-7}...10^{-9}$

Таблица 2.1

Структура объекта-эталона для схемы 1

Номер пользователя	Информация для идентификации	Информация для аутентификации
1	**ID_1**	**E_1**
2	**ID_2**	**E_2**
N	**ID_n**	**$E_п$**

Протокол 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 допускается к работе, иначе (существует ID_i = ID) устанавливается, что пользователь, назвавшийся пользователем i, прошел идентификацию.

3. Субъект аутентификации запрашивает у пользователя его аутентификатор К.

4. Субъект аутентификации вычисляет значение $Y=F(ID_i , К)$.

5. Субъект аутентификации производит сравнение значений Y и E_i. При совпадении этих значений устанавливается, что данный пользователь успешно аутентифицирован в системе. Информация об этом пользователе передается в программные модули, использующие ключи пользователей (т.е. в систему шифрования, разграничения доступа и т.д.). В противном случае аутентификация отвергается - пользователь не допускается к работе.

Данная схема идентификации и аутентификации пользователя может быть модифицирована. Модифицированная схема 2 обладает лучшими характеристиками по сравнению со схемой 1.

Схема 2. В компьютерной системе выделяется модифицированный объект-эталон, структура которого показана в табл. 2.2.

Таблица 2.2

Структура модифицированного объекта-эталона

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 отличие от схемы 1, в схеме 2 значение E_i. равно $F(S_i\ K_i)$, где S_i- случайный вектор, задаваемый при создании идентификатора пользователя, т.е. при создании строки, необходимой для идентификации и аутентификации пользователя; F-функция, которая обладает свойством "невосстановимости" значения K_i по E_i и S_i.

Протокол идентификации и аутентификации (для схемы 2).

1. Пользователь предъявляет свой идентификатор ID.

2. Если ID не совпадает ни с одним IDi, зарегистрированным в компьютерной системе, то идентификация отвергается - пользователь не допускается к работе, иначе (существует IDi=ID) устанавливается, что пользователь, называвшийся пользователем i, прошел идентификацию.

3. По идентификатору IDi выделяется вектор Si.

4. Субъект аутентификации запрашивает у пользователя аутентификатор K.

5. Субъект аутентификации вычисляет значение Y= F(Si, K).

6. Субъект аутентификации производит сравнение значений Y и Ei. При совпадении этих значений устанавливается, что данный пользователь успешно аутентифицирован в системе. В противном случае аутентификация отвергается - пользователь не допускается к работе.

Вторая схема аутентификации применяется в ОС UNIX. В качестве идентификатора ID используется имя пользователя (запрошенное по Loqin), в качестве аутентификатора Ki - пароль пользователя (запрошенный по Password), функция F представляет собой алгоритм шифрования DES. Эталоны для идентификации и аутентификации содержатся в файле Etc/passwd.

Следует отметить, что необходимым требованием устойчивости схем аутентификации к восстановлению информации Ki является случайный равновероятный выбор Ki из множества возможных значений.

Системы парольной аутентификации имеют пониженную стойкость, поскольку в них выбор аутентифицирующей информации происходит из

относительно небольшого множества осмысленных слов. Мощность этого множества определяется энтропией соответствующего языка.

Особенности применения пароля для аутентификации пользователя

Традиционно каждый законный пользователь компьютерной системы получает идентификатор и/или пароль. В начале сеанса работы пользователь предъявляет свой идентификатор системе, которая затем запрашивает у пользователя пароль.

Простейший метод подтверждения подлинности с использованием пароля основан на сравнении представляемого пользователем пароля P_A с исходным значением P_A', хранящимся в компьютерном центре (рис. 2.1). Поскольку пароль должен храниться в тайне, он должен шифроваться перед пересылкой по незащищенному каналу. Если значения P_A и P_A' совпадают, то пароль P_A считается подлинным, а пользователь - законным.

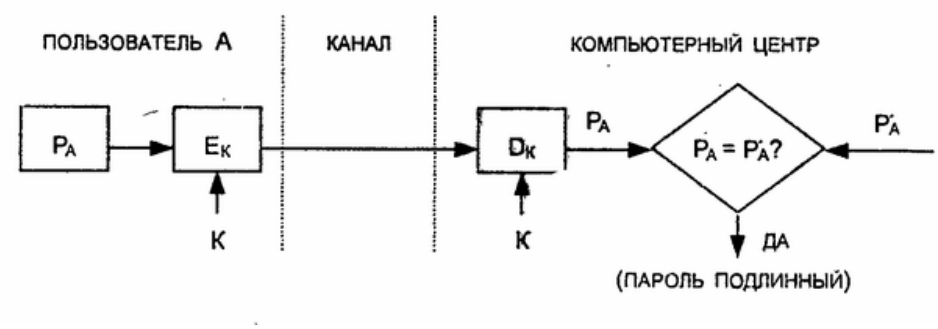


Рисунок 2.1 – Схема простой аутентификации с помощью пароля

Если кто-нибудь, не имеющий полномочий для входа в систему, узнает каким-либо образам пароль и идентификационный номер законного пользователя, он получает доступ в систему.

Иногда получатель не должен раскрывать исходную открытую форму пароля. В этом случае отправитель должен пересылать вместо открытой формы пароля отображение пароля, получаемое с использованием односторонней функции a(.) пароля. Это преобразование должно гарантировать невозможность раскрытия противником пароля по его отображению, так как противник наталкивается на неразрешимую числовую задачу.

Например, функция a(.) может быть определена следующим образом:

$$a(P)Ep(ID),$$

где P - пароль отправителя; ID-идентификатор отправителя; E_P - процедура шифрования, выполняемая с использованием пароля P в качестве ключа.

Такие функции особенно удобны, если длина пароля и ключа одинаковы. В этом случае подтверждение подлинности с помощью пароля состоит из пересылки получателю отображения a(P) и сравнения его с предварительно вычисленным и хранимым эквивалентом a'(P).

На практике пароли состоят только из нескольких букв, чтобы дать возможность пользователям запомнить их. Короткие пароли уязвимы к атаке полного перебора *всех* вариантов. Для того чтобы предотвратить такую атаку, функцию a(P) определяют иначе, а именно:

$$a(P)=Ep_{p+к}(ID),$$

где К и ID-соответственно ключ и идентификатор отправителя.

Очевидно, значение a(P) вычисляется заранее и хранится в виде a'(P) в идентификационной таблице у получателя (рис. 2.2). Подтверждение подлинности состоит из сравнения двух отображений пароля $a(P_A)$ и $a'(P_A)$ и признания пароля P_A, если эти отображения равны.

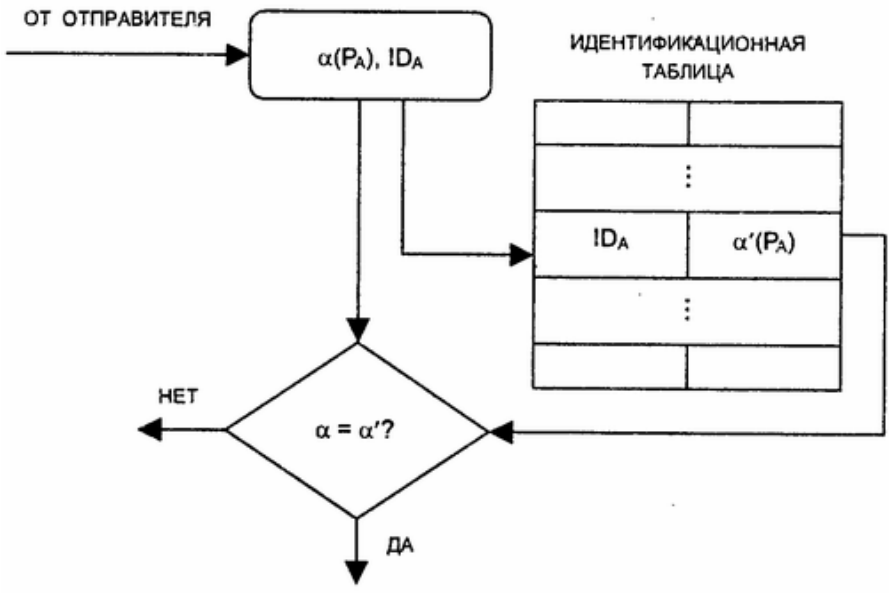


Рисунок 2.2 – Схема аутентификации с помощью пароля с использованием идентификационной таблицы

Конечно, любой, кто получит доступ к идентификационной таблице, может незаконно изменить ее содержимое, не опасаясь, что эти действия будут обнаружены.

Биометрическая идентификация и аутентификация пользователя

Процедуры идентификации и аутентификации пользователя могут базироваться не только на секретной информации, которой обладает пользователь (пароль, секретный ключ, персональный идентификатор и т.п.). В последнее время все большее распространение получает биометрическая идентификация и аутентификация пользователя, позволяющая уверенно идентифицировать потенциального пользователя путем измерения физиологических параметров и характеристик человека, особенностей его поведения.

Отметим основные достоинства биометрических методов идентификации и аутентификации пользователя по сравнению с традиционными:

- высокая степень достоверности идентификации по биометрическим признакам из-за их уникальности;
- неотделимость биометрических признаков от дееспособной личности;
- трудность фальсификации биометрических признаков.

В качестве биометрических признаков, которые могут быть использованы при идентификации потенциального пользователя, можно выделить следующие:

- узор радужной оболочки и сетчатки глаз;
- отпечатки пальцев;
- геометрическая форма руки;

- форма и размеры лица;
- особенности голоса;
- биомеханические характеристики рукописной подписи;
- биомеханические характеристики "клавиатурного почерка".

При регистрации пользователь должен продемонстрировать один или несколько раз свои характерные биометрические признаки. Эти признаки (известные как подлинные) регистрируются системой как контрольный "образ" законного пользователя. Этот образ пользователя хранится в электронной форме и используется для проверки идентичности каждого, кто выдает себя за соответствующего законного пользователя. В зависимости от совпадения или несовпадения совокупности предъявленных признаков с зарегистрированными в контрольном образе их предъявивший признается законным пользователем (при совпадении) или нет (при несовпадении).

Системы идентификации по узору радужной оболочки и сетчатки глаз могут быть разделены на два класса:

- использующие рисунок радужной оболочки глаза,
- использующие рисунок кровеносных сосудов сетчатки глаза.

Поскольку вероятность повторения данных параметров равна 10^{-78}, эти системы являются наиболее надежными среди всех биометрических систем. Такие средства идентификации применяются там, где требуется высокий уровень безопасности (например, в США в зонах военных и оборонных объектов).

Системы идентификации по отпечаткам пальцев являются самыми распространенными. Одна из основных причин широкого распространения таких систем заключается в наличии больших банков данных по отпечаткам пальцев. Основными пользователями подобных систем во всем мире являются полиция, различные государственные и некоторые банковские организации.

Системы идентификации по геометрической форме руки используют сканеры формы руки, обычно устанавливаемые на стенах. Следует отметить,

что подавляющее большинство пользователей предпочитают системы именно этого типа, а не описанные выше.

Системы идентификации по лицу и голосу являются наиболее доступными из-за их дешевизны, поскольку большинство современных компьютеров имеют видео- и аудиосредства. Системы данного класса широко применяются при удаленной идентификации субъекта доступа в телекоммуникационных сетях.

Системы идентификации личностей по динамике рукописной подписи учитывают интенсивность каждого усилия подписывающего, частотные характеристики написания каждого элемента подписи и начертание подписи в целом.

Системы идентификации по биомеханическим характеристикам "клавиатурного почерка" основываются на том, что моменты нажатия и отпускания клавиш при наборе текста на клавиатуре существенно различаются у разных пользователей. Этот динамический ритм набора ("клавиатурный почерк") позволяет построить достаточно надежные средства идентификации. В случае обнаружения изменения клавиатурного почерка пользователя ему автоматически запрещается работа на ЭВМ.

Следует отметить, что применение биометрических параметров при идентификации субъектов доступа автоматизированных систем пока не получило надлежащего нормативно-правового обеспечения, в частности в виде стандартов. Поэтому применение систем биометрической идентификации допускается только в автоматизированных системах, обрабатывающих и хранящих персональные данные, составляющие коммерческую и служебную тайну.

2.3. Взаимная проверка подлинности пользователей

Обычно стороны, вступающие в информационный обмен, нуждаются во взаимной проверке подлинности (аутентификации) друг друга. Этот процесс взаимной аутентификации выполняют в начале сеанса связи.

Для проверки подлинности применяют следующие способы:

- механизм запроса-ответа;
- механизм отметки времени ("временной штемпель").

Механизм запроса-ответа состоит в следующем. Если пользователь А хочет быть уверенным, что сообщения, получаемые им от пользователя В, не являются ложными, он включает в посылаемое для В сообщение непредсказуемый элемент-запрос X (например, некоторое случайное число). При ответе пользователь В должен выполнить над этим элементом некоторую операцию (например, вычислить некоторую функцию f(X)). Это невозможно осуществить заранее, так как пользователю В неизвестно, какое случайное число X придет в запросе. Получив ответ с результатом действий В, пользователь может быть уверен, что В - подлинный. Недостаток этого метода-возможность установления закономерности между запросом и ответом.

Механизм отметки времени подразумевает регистрацию времени для каждого сообщения. В этом случае каждый пользователь сети может определить, насколько "устарело" пришедшее сообщение, и решить не принимать его, поскольку оно может быть ложным.

В обоих случаях для защиты механизма контроля следует применять шифрование, чтобы быть уверенным, что ответ послан не злоумышленником.

При использовании отметок времени возникает проблема *допустимого временного интервала задержки* для подтверждения подлинности сеанса. Ведь сообщение с "временным штемпелем" в принципе не может быть передано мгновенно. Кроме того, компьютерные часы получателя и отправителя не могут быть абсолютно синхронизированы. Какое запаздывание "штемпеля" является подозрительным?

Для взаимной проверки подлинности обычно используют *процедуру "рукопожатия"*. Эта процедура базируется на указанных выше механизмах контроля и заключается во взаимной проверке ключей, используемых сторонами. Иначе говоря, стороны признают друг друга законными партнерами, если докажут друг другу, что обладают правильными ключами.

Процедуру рукопожатия обычно применяют в компьютерных сетях при организации сеанса связи между пользователями, пользователем и хост - компьютером, между хост - компьютерами и т.д.

Рассмотрим в качестве примера процедуру рукопожатия для двух пользователей А и В. (Это допущение не влияет на общность рассмотрения.

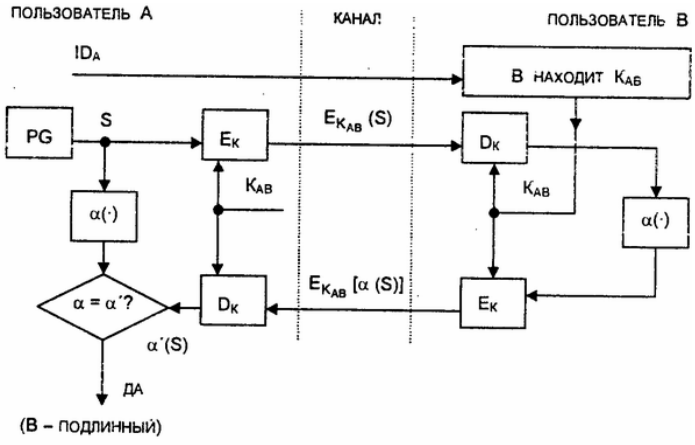


Рисунок 2.3 – Схема процедуры рукопожатия (пользователь А проверяет подлинность пользователя В)

Такая же процедура используется, когда вступающие в связь стороны не являются пользователями). Пусть приценяется симметричная криптосистема. Пользователи А и В разделяют один и тот же секретный ключ $К_{АВ}$. Вся процедура показана на рис. 2.3.

• Пусть пользователь А инициирует процедуру рукопожатия, отправляя пользователю В свой идентификатор ID_A в открытой форме.

• Пользователь В, получив идентификатор ID_A, находит в базе данных секретный *ключ* $К_{АВ}$ и вводит его в свою криптосистему.

• Тем временем *пользователь А* генерирует случайную *последовательность S с* помощью псевдослучайного генератора PG и отправляет ее *пользователю В* в виде криптограммы

$$E_{К_{АВ}}(S).$$

• Пользователь В расшифровывает эту криптограмму и раскрывает исходный вид последовательности S.

• Затем оба пользователя A и B преобразуют последовательность S, используя открытую одностороннюю функцию a(.).

• Пользователь B шифрует сообщение a(S) и отправляет эту криптограмму *пользователю A*.

• Наконец, пользователь A расшифровывает эту криптограмму и сравнивает полученное сообщение a'(S) с исходным a(S). Если эти сообщения равны, пользователь A признает подлинность пользователя B.

Очевидно, пользователь B проверяет подлинность пользователя A таким же способом. Обе эти процедуры образуют процедуру рукопожатия, которая обычно выполняется в самом начале любого сеанса связи между любыми двумя сторонами в компьютерных сетях.

Достоинством модели рукопожатия является то, что ни один из участников сеанса связи не получает никакой секретной информации во время процедуры подтверждения подлинности.

Иногда пользователи хотят иметь непрерывную проверку подлинности отправителей в течение всего сеанса связи. Один из простейших способов непрерывной проверки подлинности показан на рис. 5.4. Передаваемая криптограмма имеет вид

$$Ек(ID_A,M),$$

где ID_A-идентификатор отправителя A; M - сообщение.

Получатель B, принявший эту криптограмму, расшифровывает ее и раскрывает пару (ID_A, M). Если принятый идентификатор ID_A совпадает с хранимым значением ID_A, получатель B признает эту криптограмму.

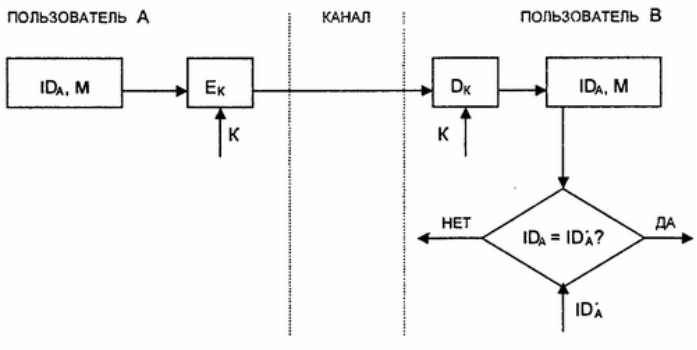


Рис. 5.4. Схема непрерывной проверки подлинности отправителя

Рисунок 2.4 – Схема непрерывной проверки подлинности отправителя

Другой вариант непрерывной проверки подлинности использует вместо идентификатора отправителя его секретный пароль. Заранее подготовленные пароли известны обеим сторонам. Пусть P_A и $P_в$-пароли пользователей А и В соответственно. Тогда пользователь А создает криптограмму

$$C = E_K(P_A,M).$$

Получатель криптограммы расшифровывает ее и сравнивает пароль, извлеченный из этой криптограммы, с исходным значением. Если они равны, получатель признает эту криптограмму.

Процедура рукопожатия была рассмотрена в предположении, что пользователи А и В уже имеют общий *секретный сеансовый ключ.* Реальные процедуры предназначены для распределения ключей между подлинными партнерами и включает как этап распределения ключей, так и этап собственно подтверждения подлинности партнеров по информационному обмену.

2.4. Протоколы идентификации с нулевой передачей знаний

Широкое распространение интеллектуальных карт (смарт-карт) для разнообразных коммерческих, гражданских и военных применений (кредитные карты, карты социального страхования, карты доступа в охраняемое помещение, компьютерные пароли и ключи, и т.п.) потребовало обеспечения безопасной идентификации таких карт и их владельцев. Во многих приложениях главная проблема заключается в том, чтобы при предъявлении интеллектуальной карты оперативно обнаружить обман и отказать обманщику в допуске, ответе или обслуживании.

Для безопасного использования интеллектуальных карт разработаны протоколы идентификации с нулевой передачей знаний. Секретный ключ владельца карты становится неотъемлемым признаком его личности. Доказательство знания этого секретного ключа с нулевой передачей этого знания служит доказательством подлинности личности владельца карты.

Упрощенная схема идентификации с нулевой передачей знаний

Схему идентификации с нулевой передачей знаний предложили в 1986 г. У. Фейге, А. Фиат и А. Шамир. Она является наиболее известным доказательством идентичности с нулевой передачей конфиденциальной информации.

Рассмотрим сначала упрощенный вариант схемы идентификации с нулевой передачей знаний для более четкого выявления ее основной концепции. Прежде всего, выбирают случайное значение модуля n, который является произведением двух больших простых чисел. Модуль n должен иметь длину 512... 1024 бит. Это значение n может быть представлено группе пользователей, которым придется доказывать свою подлинность. В процессе идентификации участвуют две стороны:

- сторона А, доказывающая свою подлинность,
- сторона В, проверяющая представляемое стороной А доказательство.

Для того чтобы сгенерировать открытый и секретный ключи для стороны А, доверенный арбитр (Центр) выбирает некоторое число V, которое является квадратичным вычетом по модулю n. Иначе говоря, выбирается такое число V, что сравнение

$$x^2 = V(\mathrm{mod}\ n)$$

имеет решение и существует целое число

$$V^{-1} \mathrm{mod}\ n.$$

Выбранное значение V является *открытым ключом* для А. Затем вычисляют наименьшее значение S, для которого

$$S = \mathrm{sqrt}(V^{-1})(\mathrm{mod} n).$$

Это значение S является *секретным ключом* для А.

Теперь можно приступить к выполнению протокола идентификации.

1. Сторона А выбирает некоторое случайное число r, r<n. Затем она вычисляет

$$x = r^2 \mathrm{mod}\ n$$

и отправляет x стороне В.

2. Сторона B посылает A случайный бит b.

3. Если b = 0, тогда A отправляет r стороне B. Если b = 1, то A отправляет стороне B

$$y = r * S \bmod n.$$

4. Если b = 0, сторона B проверяет, что

$$x = r^2 \bmod n,$$

чтобы убедиться, что A знает sqrt(x). Если b = 1, сторона B проверяет, что

$$x = y^2 * V \bmod n,$$

чтобы быть уверенной, что A знает sqrt(V^1).

Эти шаги образуют один цикл протокола, называемый *аккредитацией.* Стороны A и B повторяют этот цикл t раз при разных случайных значениях r и b до тех пор, пока B не убедится, что A знает значение S.

Если сторона A не знает значения S, она может выбрать такое значение r, которое позволит ей обмануть сторону B, если B отправит ей b = 0, либо A может выбрать такое r, которое позволит обмануть B, если B отправит ей b = 1. Но этого невозможно сделать в обоих случаях. Вероятность того, что A обманет B в одном цикле, составляет 1/2. Вероятность обмануть B в t циклах равна $(1/2)^t$

Для того чтобы этот протокол работал, сторона A никогда не должна повторно использовать значение r. Если A поступила бы таким образом, а сторона B отправила бы стороне A на шаге 2 другой случайный бит b, то B имела бы оба ответа A. После этого B может вычислить значение S, и для A все закончено.

Параллельная схема идентификации с нулевой передачей знаний

Параллельная схема идентификации позволяет увеличить число аккредитаций, выполняемых за один цикл, и тем самым уменьшить длительность процесса идентификации.

Как и в предыдущем случае, сначала генерируется число n как произведение двух больших чисел. Для того, чтобы сгенерировать открытый и

секретный ключи для стороны А, сначала выбирают К различных чисел V_1 V_2, ...,V_K, где каждое V; является квадратичным вычетом по модулю n. Иначе говоря, выбирают значение V_i таким, что сравнение

$$x^2 = V_i \bmod n$$

имеет решение и существует V_i^{-1}modn. Полученная строка V_1V_2, ..., V_K является *открытым ключом.*

Затем вычисляют такие наименьшие значения S,, что

$$S_i = \text{sqrt}\ (V_i^1) \bmod n.$$

Эта строка S_1 S_2,..., S_K является *секретным ключом* стороны А. Протокол процесса идентификации имеет следующий вид:

1. Сторона А выбирает некоторое случайное число r, r<n. Затем она вычисляет $x = r^2 \bmod n$ и посылает x стороне В.

2. Сторона В отправляет стороне А некоторую случайную двоичную строку из К бит: b_1, b_2,.... $b_к$.

3. Сторона А вычисляет

$$y = r * (S_1^{b1} * S_2^{b2} *...* S_K^{bk}) \bmod n.$$

Перемножаются только те значения S_i, для которых $b_i = 1$. Например, если $b_i = 1$, то сомножитель S_1 входит в произведение, если же $b_i = 0$, то S_1 не входит в произведение, и т.д. Вычисленное значение y отправляется стороне В.

4. Сторона В проверяет, что

$$x = y^2 * (V_1^{b1} * V_2^{b2} *...*V_k^{bk}) \bmod n.$$

Фактически сторона В перемножает только те значения V_i, для которых $b_i = 1$. Стороны А и В повторяют этот протокол t раз, пока В не убедится, что А знает S_1, S_2.....S_K.

Вероятность того, что. А может обмануть В, равна $(1/2)^{Kt}$. Авторы рекомендуют в качестве контрольного значения брать вероятность обмана В равной $(1/2)^{20}$ при К = 5 и t=4.

Пример. Рассмотрим работу этого протокола для небольших числовых значений . Если n = 35 (n - произведение двух простых чисел 5 и 7), то возможные квадратичные вычеты будут следующими:

Таблица 2.3

1	$x^2 = 1 \pmod{35}$	имеет решения: x = 1, 6. 29, 34;
4	$x^2 = 4 \pmod{35}$	имеет решения: x = 2,12, 23, 33;
9	$x^2 = 9 \pmod{35}$	имеет решения: x = 3, 17, 18, 32;
11	$x^2 = 11 \pmod{35}$	имеет решения: x = 9, 16.19, 26;
14	$x^2 = 14 \pmod{35}$	имеет решения: x = 7, 28;
15	$x^2 = 15 \pmod{35}$	имеет решения: x = 15, 20;
16	$x^2 * 16 \pmod{35}$	имеет решения: x = 4, 11, 24, 31;
21	$x^2 = 21 \pmod{35}$	имеет решения: x = 14, 21;
25	$x^2 = 25 \pmod{35}$	имеет решения: x = 5, 30;
29	$x^2 = 29 \pmod{35}$	имеет решения: x = 8, 13, 22, 27;
30	$x^2 = 30 \pmod{35}$	имеет решения: x - 10, 25.

Заметим, что 14, 15, 21, 25 и 30 не имеют обратных значений по модулю 35. потому что они не являются взаимно простыми с 35. Следует также отметить, что число квадратичных вычетов по модулю 35, взаимно простых с n = p*q = 5*7=35 (для которых НОД (x, 35) = 1), равно

$$(p -1) (q -1)/4 = (5 -1) (7-1)/4 = 6.$$

Составим таблицу квадратичных вычетов по модулю 35, обратных к ним значений по модулю 35 и их квадратных корней.

Таблица 2.4

V	V^{-1}	$S = sqrt(V^{-1}')$
1	1	1
4	9	3
9	4	2
11	16	4
16	11	9
29	29	8

Итак, сторона А получает открытый ключ, состоящий из К=4 значений V:

[4, 11, 16, 29]. Соответствующий секретный ключ, состоящий из K=4 значений S:

[3 4 9 8].

Рассмотрим один цикл протокола.

1. Сторона A выбирает некоторое случайное число r =16, вычисляет

$$x=16^2 \bmod 35=11$$

и посылает это значение x стороне B.

2. Сторона B отправляет стороне A некоторую случайную двоичную строку

[1, 1,0, 1].

3. Сторона A вычисляет значение

$$y = r * (S_1^{b1} * S_2^{b2} *...* S_K^{bk}) \bmod n =16 * (3^1 * 4^1 * 9° * 8^1) \bmod 35 = 31$$

и отправляет это значение y стороне B.

4. Сторона B проверяет, что

$$x = y^2 * (V_1^{b1} * V_2^{b2} *...*V_k^{bk}) \bmod n = 31^2 * (4^1 *11^1 *16° * 29^1) \bmod 35 =11.$$

Стороны A и B повторяют этот протокол t раз, каждый раз с разным случайным числом r, пока сторона B не будет удовлетворена.

При малых значениях величин, как в данном примере, не достигается настоящей безопасности. Но если n представляет собой число длиной 512 бит и более, сторона B не сможет узнать ничего о секретном ключе стороны A, кроме того факта, что сторона A знает этот ключ.

В этот протокол можно включить идентификационную информацию.

Пусть I-некоторая двоичная строка, представляющая идентификационную информацию о владельце карты (имя, адрес, персональный идентификационный номер, физическое описание) и о карте (дата окончания действия и т.п.). Эту информацию I формируют в Центре выдачи интеллектуальных карт по заявке пользователя A.

Далее используют одностороннюю функцию f() для вычисления f(i,j), где j-некоторое двоичное число, сцепляемое со строкой i. Вычисляют значения

$$Vj = f(l, j)$$

для небольших значений j, отбирают К разных значений j, для которых Vj являются квадратичными вычетами по модулю п. Затем для отобранных квадратичных вычетов Vj вычисляют наименьшие квадратные корни из Vj^{-1}(mod n). Совокупность из К значений Vj образует открытый ключ, а совокупность из К значений Sj-секретный ключ пользователя А.

2.5 Схема идентификации Гиллоу-Куискуотера

Алгоритм идентификации с нулевой передачей знания, разработанный Л. Гиллоу и Ж. Куискуотером, имеет несколько лучшие характеристики, чем предыдущая схема идентификации. В этом алгоритме обмены между сторонами А и В и аккредитации в каждом обмене доведены до абсолютного минимума для каждого доказательства требуется только один обмен с одной аккредитацией. Однако объем требуемых вычислений для этого алгоритма больше, чем для схемы Фейге-Фиата-Шамира.

Пусть сторона А-интеллектуальная карточка, которая должна доказать свою подлинность проверяющей стороне В. Идентификационная информация стороны А представляет собой битовую строку I, которая включает имя владельца карточки, срок действия, номер банковского счета и др. Фактически идентификационные данные могут занимать достаточно длинную строку, и тогда их хэшируют к значению I.

Строка I является аналогом открытого ключа. Другой открытой информацией, которую используют все карты, участвующие в данном приложении, являются модуль п и показатель степени V Модуль п является произведением двух секретных простых чисел.

Секретным ключом стороны А является величина G, выбираемая таким образом, чтобы выполнялось соотношение

$$l*G^{v}=1(modn)$$

Сторона А отправляет стороне В свои идентификационные данные I. Далее ей нужно доказать стороне В, что эти идентификационные данные

принадлежат именно ей. Чтобы добиться этого, сторона A должна убедить сторону B, что ей известно значение G.

Вот протокол доказательства подлинности A без передачи стороне B значения G:

1. Сторона A выбирает случайное целое r, такое, что 1<r<n-1. Она вычисляет

$$T=r^{v} \bmod n$$

и отправляет это значение стороне B.

2. Сторона B выбирает случайное целое d, такое, что 1<d<n-1, и отправляет это значение d стороне A.

3. Сторона A вычисляет

$$D = r * G^{d} \bmod n$$

и отправляет это значение стороне B.

4. Сторона B вычисляет значение

$$T'= D^{v}I^{d} \bmod n.$$

Если $T=T' \pmod n$,

то проверка подлинности успешно завершена.

Математические выкладки, использованные в этом протоколе, не очень сложны:

$$T'= D^{v}I^{d} = (rG^{d})^{v}\, I^{d} = r^{v}G^{dv}I^{d} = r^{v}(IG^{v})^{d} = r^{v}=T \pmod n,$$

поскольку G вычислялось таким образом, чтобы выполнялось соотношение

$$IG^{v}=1 \pmod n.$$

2.6 Контрольные вопросы

1. Каковы процедуры инициализации объекта информационной защиты?
2. Опишите типовые схемы идентификации и аутентификации пользователя.
3. Каковы недостатки и достоинства схемы простой аутентификации с помощью пароля?

4. Достоинства биометрических методов идентификации и аутентификации пользователя по сравнению с традиционными?

3. СРЕДСТВА И МЕТОДЫ ОГРАНИЧЕНИЯ ДОСТУПА К ФАЙЛАМ

3.1 Защита информации в КС от несанкционированного доступа

Для осуществления НСДИ злоумышленник не применяет никаких аппаратных или программных средств, не входящих в состав КС. Он осуществляет НСДИ, используя:

- знания о КС и умения работать с ней;
- сведения о системе защиты информации;
- сбои, отказы технических и программных средств;
- ошибки, небрежность обслуживающего персонала и пользователей.

Для защиты информации от НСД создается система разграничения доступа к информации. Получить несанкционированный доступ к информации при наличии системы разграничения доступа (СРД) возможно только при сбоях и отказах КС, а также используя слабые места в комплексной системе защиты информации. Чтобы использовать слабости в системе защиты, злоумышленник должен знать о них.

Одним из путей добывания информации о недостатках системы защиты является изучение механизмов защиты. Злоумышленник может тестировать систему защиты путем непосредственного контакта с ней. В этом случае велика вероятность обнаружения системой защиты попыток ее тестирования. В результате этого службой безопасности могут быть предприняты дополнительные меры защиты.

Гораздо более привлекательным для злоумышленника является другой подход. Сначала получается копия программного средства системы защиты или техническое средство защиты, а затем производится их исследование в лабораторных условиях. Кроме того, создание неучтенных копий на съемных носителях информации является одним из распространенных и удобных

способов хищения информации. Этим способом осуществляется несанкционированное тиражирование программ. Скрытно получить техническое средство защиты для исследования гораздо сложнее, чем программное, и такая угроза блокируется средствами и методами обеспечивающими целостность технической структуры КС.

Для блокирования несанкционированного исследования и копирования информации КС используется комплекс средств и мер защиты, которые объединяются в систему защиты от исследования и копирования информации (СЗИК).

Таким образом, СРД и СЗИК могут рассматриваться как подсистемы системы защиты от НСДИ.

3.2. Система разграничения доступа к информации в КС

Управление доступом

Исходной информацией для создания СРД является решение владельца (администратора) КС о допуске пользователей к определенным информационным ресурсам КС. Так как информация в КС хранится, обрабатывается и передается файлами (частями файлов), то доступ к информации регламентируется на уровне файлов (объектов доступа). Сложнее организуется доступ в базах данных, в которых он может регламентироваться к отдельным ее частям по определенным правилам. При определении полномочий доступа администратор устанавливает операции, которые разрешено выполнять пользователю (субъекту доступа).

Различают следующие операции с файлами:

- чтение (R);
- запись;
- выполнение программ (E).

Операция записи в файл имеет две модификации. Субъекту доступа может быть дано право осуществлять запись с изменением содержимого файла

(W). Другая организация доступа предполагает разрешение только дописывания в файл, без изменения старого содержимого (A).

В КС нашли применение два подхода к организации разграничения доступа:

- матричный;
- полномочный (мандатный).

Матричное управление доступом предполагает использование матриц доступа. Матрица доступа представляет собой таблицу, в которой объекту доступа соответствует столбец O_j, а субъекту доступа - строка S_i. На пересечении столбцов и строк записываются операция или операции, которые допускается выполнять субъекту доступа i с объектом доступа j (рис. 3.1).

	O_1	O_2	…	O_j	…	O_m
S_1	R	R,W		E		R
S_2	R, A	-		R		E
...						
S_i	R	-		-		R
...						
S_n	R, W	-		E		E

Рисунок 3.1 - Матрица доступа

Матричное управление доступом позволяет с максимальной детализацией установить права субъекта доступа по выполнению разрешенных операций над объектами доступа. Такой подход нагляден и легко реализуем. Однако в реальных системах из-за большого количества субъектов и объектов доступа матрица доступа достигает таких размеров, при которых сложно поддерживать ее в адекватном состоянии.

Полномочный или *мандатный* метод базируется на многоуровневой модели защиты. Такой подход построен по аналогии с «ручным»

конфиденциальным (секретным) делопроизводством. Документу присваивается уровень конфиденциальности (гриф секретности), а также могут присваиваться метки, отражающие категории конфиденциальности (секретности) документа. Таким образом, конфиденциальный документ имеет гриф конфиденциальности (конфиденциально, строго конфиденциально, секретно, совершенно секретно и т. д.) и может иметь одну или несколько меток, которые уточняют категории лиц, допущенных к этому документу («для руководящего состава», «для инженерно-технического состава» и т. д.). Субъектам доступа устанавливается уровень допуска, определяющего максимальный для данного субъекта уровень конфиденциальности документа, к которому разрешается допуск. Субъекту доступа устанавливаются также категории, которые связаны с метками документа.

Правило разграничения доступа заключается в следующем: лицо допускается к работе с документом только в том случае, если уровень допуска субъекта доступа равен или выше уровня конфиденциальности документа, а в наборе категорий, присвоенных данному субъекту доступа, содержатся все категории, определенные для данного документа.

В КС все права субъекта доступа фиксируются в его мандате. Объекты доступа содержат метки, в которых записаны признаки конфиденциальности. Права доступа каждого субъекта и характеристики конфиденциальности каждого объекта отображаются в виде совокупности уровня конфиденциальности и набора категорий конфиденциальности.

Мандатное управление позволяет упростить процесс регулирования доступа, так как при создании нового объекта достаточно создать его метку. Однако при таком управлении приходится завышать конфиденциальность информации из-за невозможности детального разграничения доступа.

Если право установления правил доступа к объекту предоставляется владельцу объекта (или его доверенному лицу), то та кой метод контроля доступа к информации называется дискреционным.

Состав системы разграничения доступа

Система разграничения доступа к информации должна содержать четыре функциональных блока:

- блок идентификации и аутентификации субъектов доступа;
- диспетчер доступа;
- блок криптографического преобразования информации при ее хранении и передаче;
- блок очистки памяти.

Идентификация и аутентификация субъектов осуществляется в момент их доступа к устройствам, в том числе и дистанционного доступа.

Диспетчер доступа реализуется в виде аппаратно-программных механизмов (рис. 3.2) и обеспечивает необходимую дисциплину разграничения доступа субъектов к объектам доступа (в том числе и к аппаратным блокам, узлам, устройствам). Диспетчер доступа разграничивает доступ к внутренним ресурсам КС субъектов, уже получивших доступ к этим системам.

Необходимость использования диспетчера доступа возникает только в многопользовательских КС.

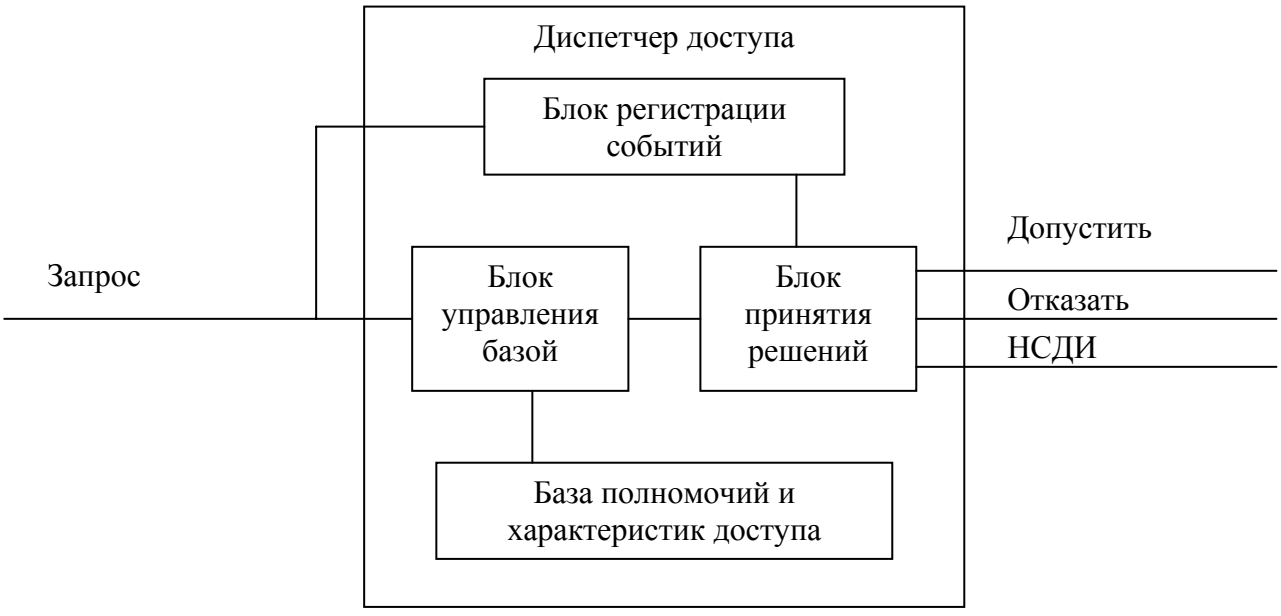


Рисунок 3.2 - Диспетчер доступа в виде аппаратно-программных механизмов

Запрос на доступ i-го субъекта и j-му объекту поступает в блок управления базой полномочий и характеристик доступа и в блок регистрации событий. Полномочия субъекта и характеристики объекта доступа анализируются в блоке принятия решения, который выдает сигнал разрешения выполнения запроса, либо сигнал отказа в допуске. Если число попыток субъекта допуска получить доступ к запрещенным для него объектам превысит определенную границу (обычно 3 раза), то блок принятия решения на основании данных блока регистрации выдает сигнал «НСДИ» администратору системы безопасности. Администратор может блокировать работу субъекта, нарушающего правила доступа в системе, и выяснить причину нарушений. Кроме преднамеренных попыток НСДИ диспетчер фиксирует нарушения правил разграничения, явившихся следствием отказов, сбоев аппаратных и программных средств, а также вызванных ошибками персонала и пользователей.

Следует отметить, что в распределенных КС криптографическое закрытие информации является надежным единственным способом защиты от НСДИ.

В СРД должна быть реализована функция очистки оперативной памяти и рабочих областей на внешних запоминающих устройствах после завершения выполнения программы, обрабатывающей конфиденциальные данные. Причем очистка должна производиться путем записи в освободившиеся участки памяти определенной последовательности двоичных кодов, а не удалением только учетной информации о файлах из таблиц ОС, как это делается при стандартном удалении средствами ОС.

3.3. Концепция построения систем разграничения доступа

В основе построения СРД лежит концепция разработки защищенной универсальной ОС на базе ядра безопасности. Под **ядром безопасности** понимают локализованную, минимизированную, четко ограниченную и

надежно изолированную совокупность программно-аппаратных механизмов, доказательно правильно реализующих функции диспетчера доступа. Правильность функционирования ядра безопасности доказывается путем полной формальной верификации его программ и пошаговым доказательством их соответствия выбранной математической модели защиты.

Применение ядра безопасности требует провести изменения ОС и архитектуры ЭВМ. Ограничение размеров и сложности ядра необходимо для обеспечения его верифицируемости.

Для аппаратной поддержки защиты и изоляции ядра в архитектуре ЭВМ должны быть предусмотрены:

- многоуровневый режим выполнения команд;
- использование ключей защиты и сегментирование памяти;
- реализация механизма виртуальной памяти с разделением адресных пространств;
- аппаратная реализация части функций ОС;
- хранение программ ядра в постоянном запоминающем устройстве (ПЗУ);
- использование новых архитектур ЭВМ, отличных от фон-неймановской архитектуры (архитектуры с реализацией абстрактных типов данных, теговые архитектуры с привилегиями и др.).

Обеспечение многоуровневого режима выполнения команд является главным условием создания ядра безопасности. Таких уровней должно быть не менее двух. Часть машинных команд ЭВМ должна выполняться только в режиме работы ОС. Основной проблемой создания высокоэффективной защиты от НСД является предотвращение несанкционированного перехода пользовательских процессов в привилегированное состояние. Для современных сложных ОС практически нет доказательств отсутствия возможности

несанкционированного получения пользовательскими программами статуса программ ОС.

Использование ключей защиты, сегментирование памяти и применение механизма виртуальной памяти предусматривает аппаратную поддержку концепции изоляции областей памяти при работе ЭВМ в мультипрограммных режимах. Эти механизмы служат основой для организации работы ЭВМ в режиме виртуальных машин. Режим виртуальных машин позволяет создать наибольшую изолированность пользователей, допуская использование даже различных ОС пользователями в режиме разделения времени.

Аппаратная реализация наиболее ответственных функций ОС и хранение программ ядра в ПЗУ существенно повышают изолированность ядра, его устойчивость к попыткам модификации. Аппаратно должны быть реализованы прежде всего функции идентификации и аутентификации субъектов доступа, хранения атрибутов системы защиты, поддержки криптографического закрытия информации, обработки сбоев и отказов и некоторые другие.

Универсальные ЭВМ и их ОС, используемые ранее, практически не имели встроенных механизмов защиты от НСД. Такие распространенные ОС как IBM System/370, MS-DOS и целый ряд других ОС не имели встроенных средств идентификации и аутентификации и разграничения доступа. Более современные универсальные ОС UNIX, VAX/VMS, Solaris и др. имеют встроенные механизмы разграничения доступа и аутентификации. Однако возможности этих встроенных функций ограничены и не могут удовлетворять требованиям, предъявляемым к защищенным ЭВМ.

Имеется два пути получения защищенных от НСД КС:

- создание специализированных КС;
- оснащение универсальных КС дополнительными средствами защиты.

Первый путь построения защищенных КС пока еще не получил широкого распространения в связи с нерешенностью целого ряда проблем. Основной из

них является отсутствие эффективных методов разработки доказательно корректных аппаратных и программных средств сложных систем. Среди немногих примеров специализированных ЭВМ можно назвать систему SCOMP фирмы «Honeywell», предназначенную для использования в центрах коммутации вычислительных сетей, обрабатывающих секретную информацию. Система разработана на базе концепции ядра безопасности. Узкая специализация позволила создать защищенную систему, обеспечивающую требуемую эффективность функционирования по прямому назначению.

Чаще всего защита КС от НСД осуществляется путем использования дополнительных программных или аппаратно-программных средств. Программные средства RACF, SECURC, TOPSECRET и другие использовались для защиты ЭВМ типа IBM-370.

В настоящее время появились десятки отдельных программ, программных и аппаратных комплексов, рассчитанных на защиту персональных ЭВМ от несанкционированного доступа к ЭВМ, которые разграничивают доступ к информации и устройствам ПЭВМ.

3.4. Организация доступа к ресурсам КС

Функционирование КСЗИ зависит не только от характеристик созданной системы, но и от эффективности ее использования на этапе эксплуатации КС. Основными задачами этапа эксплуатации являются максимальное использование возможностей КСЗИ, заложенных в систему при построении, и совершенствование ее защитных функций в соответствии с изменяющимися условиями.

Процесс эксплуатации КСЗИ можно разделить на применение системы по прямому назначению, что предполагает выполнение всего комплекса мероприятий, непосредственно связанных с защитой информации в КС, и техническую эксплуатацию (рис. 3.3). Применение по назначению предусматривает организацию доступа к ресурсам КС и обеспечение их целостности.

Под организацией доступа к ресурсам понимается весь комплекс мер, который выполняется в процессе эксплуатации КС для предотвращения несанкционированного воздействия на технические и программные средства, а также на информацию.

Организация доступа к ресурсам предполагает:

- разграничение прав пользователей и обслуживающего персонала по доступу к ресурсам КС в соответствии с функциональными обязанностями должностных лиц;
- организацию работы с конфиденциальными информационными ресурсами на объекте;
- защиту от технических средств разведки;
- охрану объекта;
- эксплуатацию системы разграничения доступа.

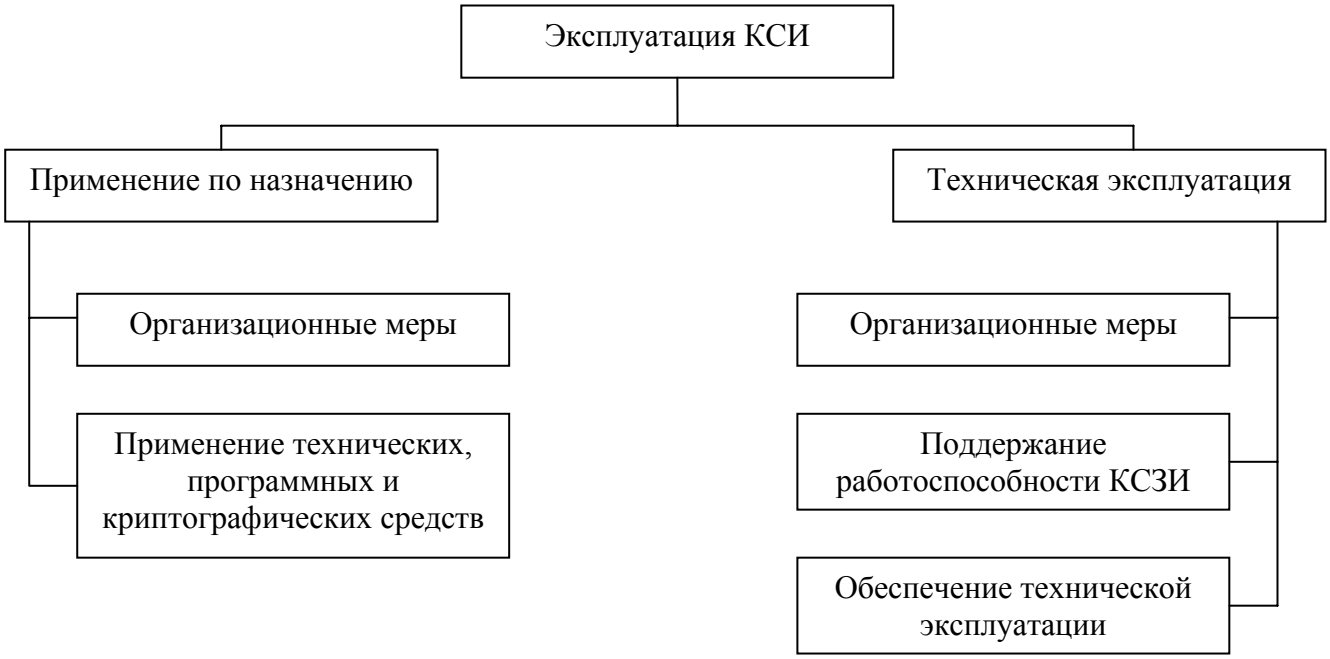


Рисунок 3.3 - Содержание процесса эксплуатации КСЗИ

Права должностных лиц по доступу к ресурсам КС устанавливаются руководством организации, в интересах которой используется КС. Каждому должностному лицу определяются для использования технические ресурсы

(рабочая станция, сервер, аппаратура передачи данных и т.д.), разрешенные режимы и время работы. Руководством устанавливается уровень компетенции должностных лиц по манипулированию информацией. Лицо, ответственное за ОБИ в КС, на основании решения руководителя о разграничении доступа должностных лиц обеспечивает ввод соответствующих полномочий доступа в систему разграничения доступа.

Руководство совместно со службой безопасности определяет порядок работы с конфиденциальными информационными ресурсами, не используемыми непосредственно в КС, хотя бы и временно. К таким ресурсам относятся конфиденциальная печатная продукция, в том числе и полученная с помощью КС, а также машинные носители информации, находящиеся вне устройств КС. Учетом, хранением и выдачей таких ресурсов занимаются должностные лица из службы безопасности, либо другие должностные лица по совместительству.

Службой безопасности выполняется весь комплекс мероприятий противодействия техническим средствам разведки. Контролируется применение пассивных средств защиты от ЭМИ и наводок. Активные средства защиты от угроз этого класса используются в соответствии с графиком работы объекта. Периодически осуществляются проверки помещений на отсутствие в них закладных устройств аудио- и видеоразведки, а также обеспечивается защищенность линий связи от прослушивания.

Охрана объекта КС обеспечивает разграничение непосредственного доступа людей на контролируемую территорию, в здания и помещения. Подразделение охраны (охранник) может находиться на объекте, а может охранять несколько объектов. В последнем случае на объекте находятся только технические средства охраны и сигнализации. В соответствии с принятой политикой безопасности руководство совместно со службой безопасности определяют структуру системы охраны. Количественный состав и режим работы подразделения охраны определяется важностью и онфиденциальностью

информации КС, а также используемыми техническими средствами охраны и сигнализации.

Система разграничения доступа (СРД) является одной из главных составляющих комплексной системы защиты информации. В этой системе можно выделить следующие компоненты:

- средства аутентификации субъекта доступа;
- средства разграничения доступа к техническим устройствам компьютерной системы;
- средства разграничения доступа к программам и данным;
- средства блокировки неправомочных действий;
- средства регистрации событий;
- дежурный оператор системы разграничения доступа.

Эффективность функционирования системы разграничения доступа во многом определяется надежностью механизмов аутентификации. Особое значение имеет аутентификация при взаимодействии удаленных процессов, которая всегда осуществляется с применением методов криптографии. При эксплуатации механизмов аутентификации основными задачами являются: генерация или изготовление идентификаторов, их учет и хранение, передача идентификаторов пользователю и контроль над правильностью выполнения процедур аутентификации в КС. При компрометации атрибутов доступа (пароля, персонального кода и т. п.) необходимо срочное их исключение из списка разрешенных. Эти действия должны выполняться дежурным оператором системы разграничения доступа.

В больших распределенных КС проблема генерации и доставки атрибутов идентификации и ключей шифрования не является тривиальной задачей. Так, например, распределение секретных ключей шифрования должно осуществляться вне защищаемой компьютерной системы. Значения идентификаторов пользователя не должны храниться и передаваться в системе в открытом виде. На время ввода и сравнения идентификаторов необходимо

применять особые меры защиты от подсматривания набора пароля и воздействия вредительских программ типа клавиатурных шпионов и программ-имитаторов СРД.

Средства разграничения доступа к техническим средствам препятствуют несанкционированным действиям злоумышленника, таким как включение технического средства, загрузка операционной системы, ввод-вывод информации, использование нештатных устройств и т. д. Разграничение доступа осуществляется оператором СРД путем использования технических и программных средств. Так оператор СРД может контролировать использование ключей от замков подачи питания непосредственно на техническое средство или на все устройства, находящиеся в отдельном помещении, дистанционно управлять блокировкой подачи питания на устройство или блокировкой загрузки ОС. На аппаратном или программном уровне оператор может изменять техническую структуру средств, которые может использовать конкретный пользователь.

Средства разграничения доступа к программам и данным используются наиболее интенсивно и во многом определяют характеристики СРД. Эти средства являются аппаратно-программными. Они настраиваются должностными лицами подразделения, обеспечивающего безопасность информации, и изменяются при изменении полномочий пользователя или при изменении программной и информационной структуры. Доступ к файлам регулируется диспетчером доступа. Доступ к записям и отдельным полям записей в файлах баз данных регулируется также с помощью систем управления базами данных.

Эффективность СРД можно повысить за счет шифрования файлов, хранящихся на внешних запоминающих устройствах, а также за счет полного стирания файлов при их уничтожении и стирания временных файлов. Даже если злоумышленник получит доступ к машинному носителю путем, например, несанкционированного копирования, то получить доступ к информации он не сможет без ключа шифрования.

В распределенных КС доступ между подсистемами, например удаленными ЛВС, регулируется с помощью межсетевых экранов. Межсетевой экран необходимо использовать для управления обменом между защищенной и незащищенной компьютерными системами. При этом регулируется доступ как из незащищенной КС в защищенную, так и доступ из защищенной системы в незащищенную. Компьютер, реализующий функции межсетевого экрана, целесообразно размещать на рабочем месте оператора КСЗИ.

Средства блокировки неправомочных действий субъектов доступа являются неотъемлемой компонентой СРД. Если атрибуты субъекта доступа или алгоритм его действий не являются разрешенными для данного субъекта, то дальнейшая работа в КС такого нарушителя прекращается до вмешательства оператора КСЗИ. Средства блокировки исключают или в значительной степени затрудняют автоматический подбор атрибутов доступа.

Средства регистрации событий также являются обязательной компонентой СРД. Журналы регистрации событий располагаются на ВЗУ. В таких журналах записываются данные о входе пользователей в систему и о выходе из нее, обо всех попытках выполнения несанкционированных действий, о доступе к определенным ресурсам и т. п. Настройка журнала на фиксацию определенных событий и периодический анализ его содержимого осуществляется дежурным оператором и вышестоящими должностными лицами из подразделения ОБИ. Процесс настройки и анализа журнала целесообразно автоматизировать программным путем.

Непосредственное управление СРД осуществляет дежурный оператор КСЗИ, который, как правило, выполняет и функции дежурного администратора КС. Он загружает ОС, обеспечивает требуемую конфигурацию и режимы работы КС, вводит в СРД полномочия и атрибуты пользователей, осуществляет контроль и управляет доступом пользователей к ресурсам КС.

3.5 Обеспечение целостности и доступности информации в КС

На этапе эксплуатации КС целостность и доступность информации в системе обеспечивается путем:

- дублирования информации;
- повышения отказоустойчивости КС;
- противодействия перегрузкам и «зависаниям» системы;
- использования строго определенного множества программ;
- контроля целостности информации в КС;
- особой регламентации процессов технического обслуживания и проведения доработок;
- выполнения комплекса антивирусных мероприятий.

Одним из главных условий обеспечения целостности и доступности информации в КС является ее дублирование. Стратегия дублирования выбирается с учетом важности информации, требований к непрерывности работы КС, трудоемкости восстановления данных. Дублирование информации обеспечивается дежурным администратором КС.

Целостность и доступность информации поддерживается также путем резервирования аппаратных средств, блокировок ошибочных действий людей, использования надежных элементов КС и отказоустойчивых систем. Устраняются также преднамеренные угрозы перегрузки элементов систем. Для этого используются механизмы измерения интенсивности поступления заявок на выполнение (передачу) и механизмы ограничения или полного блокирования передачи таких заявок. Должна быть предусмотрена также возможность определения причин резкого увеличения потока заявок на выполнение программ или передачу информации.

В сложных системах практически невозможно избежать ситуаций, приводящих к «зависаниям» систем или их фрагментов. В результате сбоев аппаратных или программных средств, алгоритмических ошибок, допущенных

на этапе разработки, ошибок операторов в системе происходят зацикливания программ, непредусмотренные остановы и другие ситуации, выход из которых возможен лишь путем прерывания вычислительного процесса и последующего его восстановления. На этапе эксплуатации ведется статистика и осуществляется анализ таких ситуаций. «Зависания» своевременно обнаруживаются, и вычислительный процесс восстанавливается. При восстановлении, как правило, необходимо повторить выполнение прерванной программы с начала или с контрольной точки, если используется механизм контрольных точек. Такой механизм используется при выполнении сложных вычислительных программ, требующих значительного времени для их реализации.

В защищенной КС должно использоваться только разрешенное программное обеспечение. Перечень официально разрешенных к использованию программ, а также периодичность и способы контроля их целостности должны быть определены перед началом эксплуатации КС.

В защищенных КС, сданных в эксплуатацию, как правило, нет необходимости использовать трансляторы и компиляторы, программы-отладчики, средства трассировки программ и тому подобные программные средства. Работы по созданию и модернизации программного обеспечения должны производиться в автономных КС или, как исключение, в сегментах защищенной КС, при условии использования надежных аппаратно-программных средств, исключающих возможность проведения мониторинга и несанкционированного внедрения исполняемых файлов в защищаемой КС.

Простейшим методом контроля целостности программ является метод контрольных сумм. Для исключения возможности внесения изменений в контролируемый файл с последующей коррекцией контрольной суммы необходимо хранить контрольную сумму в зашифрованном виде или использовать секретный алгоритм вычисления контрольной суммы.

Однако наиболее приемлемым методом контроля целостности информации является использование хэш-функции. Значение хэш-функции

практически невозможно подделать без знания ключа. Поэтому следует хранить в зашифрованном виде или в памяти, недоступной злоумышленнику, только ключ хеширования (стартовый вектор хеширования).

Контроль состава программного обеспечения и целостности (неизменности) программ осуществляется при плановых проверках комиссиями и должностными лицами, а также дежурным оператором КСЗИ по определенному плану, неизвестному пользователям. Для осуществления контроля используются специальные программные средства. В вычислительных сетях такая «ревизия» программного обеспечения может осуществляться дистанционно с рабочего места оператора КСЗИ.

Особое внимание руководства и должностных лиц подразделения ОБИ должно быть сосредоточено на обеспечении целостности структур КС и конфиденциальности информации, защите от хищения и несанкционированного копирования информационных ресурсов во время проведения технического обслуживания, восстановления работоспособности, ликвидации аварий, а также в период модернизации КС. Так как на время проведения таких специальных работ отключаются (или находятся в неработоспособном состоянии) многие технические и программные средства защиты, то их отсутствие компенсируется системой организационных мероприятий:

- подготовка КС к выполнению работ;
- допуск специалистов к выполнению работ;
- организация работ на объекте;
- завершение работ.
- Перед проведением работ, по возможности, должны предприниматься следующие шаги:
- отключить фрагмент КС, на котором необходимо выполнять работы, от функционирующей КС;
- снять носители информации с устройств;

- осуществить стирание информации в памяти КС;
- подготовить помещение для работы специалистов.

Перед проведением специальных работ необходимо всеми доступными способами изолировать ту часть КС, на которой предполагается выполнять работы, от функционирующей части КС. Для этого могут быть использованы аппаратные и программные блокировки и физические отключения цепей.

Все съемные носители с конфиденциальной информацией должны быть сняты с устройств и храниться в заземленных металлических шкафах в специальном помещении. Информация на несъемных носителях стирается путем трехкратной записи, например, двоичной последовательности чередующихся 1 и 0. На объекте необходимо определить порядок действий в случае не возможности стереть информацию до проведения специальных работ, например, при отказе накопителя на магнитных дисках. В этом случае восстановление работоспособности должно выполняться под непосредственным контролем должностного лица из подразделения ОБИ. При восстановлении функции записи на носитель первой же операцией осуществляется стирание конфиденциальной информации. Если восстановление работоспособности накопителя с несъемным носителем информации невозможно, то устройство подлежит утилизации, включая физическое разрушение носителя.

При оборудовании помещения для проведения специальных работ осуществляется подготовка рабочих мест и обеспечивается изоляция рабочих мест от остальной части КС. На рабочих местах должны использоваться сертифицированные и проверенные на отсутствие закладок приборы (если они не поставлялись в комплекте КС). Меры по обеспечению изолированности рабочих мест от остальной КС имеют целью исключить доступ сотрудников, выполняющих специальные работы, к элементам функционирующей КС.

Допуск специалистов осуществляется на рабочие места в определенное время, и после выполнения всех подготовительных операций.

При прибытии специалистов из других организаций, например, для проведения доработок, кроме обычной проверки лиц, допускаемых на объект, должны проверяться на отсутствие закладок приборы, устройства, которые доставлены для выполнения работ.

В процессе выполнения специальных работ необходимо исключить использование не проверенных аппаратных и программных средств, отклонения от установленной документацией технологии проведения работ, доступ к носителям с конфиденциальной информацией и к функционирующим в рабочих режимах элементам КС.

Специальные работы завершаются контролем работоспособности КС и отсутствия закладок. Проверка на отсутствие аппаратных закладок осуществляется путем осмотра устройств и тестирования их во всех режимах. Отсутствие программных закладок проверяется по контрольным суммам, а также путем тестирования. Результаты доработок принимаются комиссией и оформляются актом, в котором должны быть отражены результаты проверки работоспособности и отсутствия закладок. После проверок осуществляется восстановление информации и задействуются все механизмы защиты.

В автономных КС непосредственную ответственность за выполнение комплекса антивирусных мероприятий целесообразно возложить на пользователя КС. В ЛВС такая работа организуется должностными лицами подразделения ОБИ. Исполняемые файлы, в том числе саморазархивирующиеся и содержащие макрокоманды, должны вводиться в ЛВС под контролем дежурного оператора КСЗИ и подвергаться проверке на отсутствие вирусов.

Успех эксплуатации КСЗИ в большой степени зависит от уровня организации управления процессом эксплуатации. Иерархическая система управления позволяет организовать реализацию политики безопасности информации на этапе эксплуатации КС. При организации системы управления следует придерживаться следующих принципов:

- уровень компетенции руководителя должен соответствовать его статусу в системе управления;
- строгая регламентация действий должностных лиц;
- документирование алгоритмов обеспечения защиты информации;
- непрерывность управления;
- адаптивность системы управления.
- контроль над реализацией политики безопасности;

Каждое должностное лицо из руководства организации, службы безопасности или подразделения ОБИ должны иметь знания и навыки работы с КСЗИ в объеме, достаточном для выполнения своих функциональных обязанностей. Причем должностные лица должны располагать минимально возможными сведениями о конкретных механизмах защиты и о защищаемой информации. Это достигается за счет очень строгой регламентации их деятельности. Документирование всех алгоритмов эксплуатации КСЗИ позволяет, при необходимости, легко заменять должностных лиц, а также осуществлять контроль над их деятельностью. Реализация этого (принципа позволит избежать «незаменимости» отдельных сотрудников и наладить эффективный контроль деятельности должностных лиц.

Непрерывность управления КСЗИ достигается за счет организации дежурства операторов КСЗИ. Система управления должна быть гибкой и оперативно адаптироваться к изменяющимся условиям функционирования.

3.6 Контрольные вопросы

1. Что используют для блокирования несанкционированного исследования и копирования информации КС?
2. Проанализируйте матричное управление доступом.
3. Какие функциональные блоки содержит система разграничения доступа к информации?

4. Является ли криптографическое закрытие информации единственным надежным способом защиты от НСДИ в распределенных КС и почему?
5. Что понимают под ядром безопасности?
6. Что является основной проблемой создания высокоэффективной защиты от НСД?
7. Сделайте сравнительный анализ программных и аппаратных комплексов, рассчитанных на защиту персональных ЭВМ от несанкционированного доступа к ЭВМ, которые разграничивают доступ к информации и устройствам ПЭВМ.
8. Что предполагает организация доступа к ресурсам?
9. Каким образом поддерживается целостность и доступность информации?
10. Что является наиболее приемлемым методом контроля целостности информации?
11. Каких принципов следует придерживаться при организации системы управления?

4. АППАРАТНО-ПРОГРАММНЫЕ СРЕДСТВА КРИПТОГРАФИЧЕСКОЙ ЗАЩИТЫ ИНФОРМАЦИИ

4.1 Полностью контролируемые компьютерные системы

Любая компьютерная система (КС) использует стандартное и специализированное оборудование и программное обеспечение, выполняющее определенный набор функций: аутентификацию пользователя, разграничение доступа к информации, обеспечение целостности информации и ее защиты от уничтожения, шифрование и электронную цифровую подпись и др.

Целостность и ограничение доступа к информации обеспечиваются специализированными компонентами системы, использующими криптографические методы защиты. Для того чтобы компьютерной системе можно было полностью доверять, ее необходимо аттестовать, а именно:

- определить множество выполняемых функций;
- доказать конечность этого множества;
- определить свойства всех функций.

Отметим, что в процессе функционирования системы невозможно появление в ней новой функции, в том числе и в результате выполнения любой комбинации функций, заданных при разработке. Здесь мы не будем останавливаться на конкретном составе функций, поскольку они перечислены в соответствующих руководящих документах Федерального агентства правительственной связи и информации (ФАПСИ) и Государственной технической комиссии (ГТК) России.

При использовании системы ее функциональность не должна нарушаться, иными словами, необходимо обеспечить целостность системы в момент ее запуска и в процессе функционирования.

Надежность защиты информации в компьютерной системе определяется:

- конкретным перечнем и свойствами функций КС;
- используемыми в функциях КС методами;

- способом реализации функций КС.

Перечень используемых функций соответствует классу защищенности, присвоенному КС в процессе сертификации, и в принципе одинаков для систем одного класса. Поэтому при рассмотрении конкретной КС следует обратить внимание на используемые методы и способ реализации наиболее важных функций: аутентификацию и проверку целостности системы. Здесь следует отдать предпочтение криптографическим методам: шифрования (ГОСТ 28147-89), электронной цифровой подписи (ГОСТР 34.10-94) и функции хэширования (ГОСТР 34.11-94), надежность которых подтверждена соответствующими государственными организациями.

Программная реализация функций КС.

Большинство функций современных КС реализованы в виде программ, поддержание целостности которых при запуске системы и особенно в процессе функционирования является трудной задачей. Значительное число пользователей в той или иной степени обладают познаниями в программировании, осведомлены об ошибках в построении операционных систем. Поэтому существует достаточно высокая вероятность применения ими имеющихся знаний для "атак" на программное обеспечение.

Проверка целостности одних программ при помощи других не является надежной. Необходимо четко представлять, каким образом обеспечивается целостность собственно программы проверки целостности. Если обе программы находятся на одних и тех же носителях, доверять результатам такой проверки нельзя. В связи с этим к программным системам защиты от несанкционированного доступа (НСД) следует относиться с особой осторожностью.

Аппаратная реализация функций КС.

Использование аппаратных средств снимает проблему обеспечения целостности системы. В большинстве современных систем защиты от НСД применяется зашивка программного обеспечения в ПЗУ или в аналогичную микросхему. Таким образом, для внесения изменений в ПО необходимо получить доступ к соответствующей плате и заменить микросхему.

В случае использования универсального процессора реализация подобных действий потребует применения специального оборудования, что еще более затруднит проведение атаки. Использование специализированного процессора с реализацией алгоритма работы в виде интегральной микросхемы полностью снимает проблему нарушения целостности этого алгоритма.

На практике для повышения класса защищенности КС функции аутентификации пользователя, проверки целостности (платы Таблица 10.1 Аппаратные устройства криптографической защиты данных серии КРИПТОН, КРИПТОН-НСД, АККОРД и др.), криптографические функции (платы КРИПТОН-4, КРИПТОН-4К/8, КРИПТОН-4К/16, КРИПТОН-4/PCI, КРИПТОН-7/PCI, КРИПТОН-8/PCI), образующие ядро системы безопасности, реализуются аппаратно (табл. 10.1), все остальные функции - программно.

Таблица 4.1

Наименование	**Описание**
КРИПТОН-4	Шифрование по ГОСТ28147-89 (специализированным шифропроцессором "Блюминг-1"). Генерация случайных чисел. Хранение 3 ключей и 1 узла замены в шифраторе. Загрузка ключей в устройство до загрузки ОС с дискеты или со смарт-карты, минуя оперативную память ПК. Защита от НСД. Скорость шифрования до 350 Кбайт/с. Интерфейс шины ISA-8

КРИПТОН-4К/8	Функции устройства КРИПТОН-4. Более современная, чем в КРИПТОН-4, отечественная элементная база (шифропроцессор "Блюминг-1К'). Аппаратный журнал работы с устройством. Загрузка ключей с Touch-Memory. Скорость шифрования до 610 Кбайт/с. Интерфейс шины ISA-8
КРИПТОН-4К/16	Функции устройства КРИПТОН-4К/8. Функции электронного замка персонального компьютера - разграничение доступа, проверка целостности ОС. Скорость шифрования до 950 Кбайт/с. Интерфейс шины ISA-16
КРИПТОН-4/PCI	Функции устройства КРИПТОН-4К/16. Скорость шифрования до 1100 Кбайт/с. Интерфейс шины PCI Target. Возможность параллельной работы нескольких плат
КРИПТОН-7/PCI	Функции устройства КРИПТОН-4/PCI. Хранение до 1000 ключей (таблиц сетевых ключей) в защищенном ОЗУ. Управление доступом к ключам. Скорость шифрования до 1300 Кбайт/с. Интерфейс шины PCI Master/Target. Возможность параллельной работы нескольких плат
КРИПТОН-8/PCI	Функции устройства КРИПТОН-7/PCI. Хранение 32 ключей и 2 узлов замены в шифраторе, до 4000 ключей в защищенном ОЗУ. Аппаратная реализация быстрой смены ключей. Скорость шифрования до 8800 Кбайт/с. Интерфейс шины PCI Master/Target. Возможность параллельной работы нескольких плат

КРИПТОН-НСД	Шифрование по ГОСТ28147-89 (программой из ПЗУ). Генерация случайных чисел. Защита от НСД. Загрузка ключей с дискет, смарт-карт и Touch-Memory
Специализированная сетевая плата	Размещение коммуникационных модулей внутри платы для исключения их обхода (стадия разработки)

Для построения надежной системы защиты КС ее разработчик должен владеть возможно более полными знаниями о конкретной операционной системе (ОС), под управлением которой будет работать система. В настоящее время отечественные разработчики располагают относительно полной информацией только об одной операционной системе-DOS. Таким образом, к целиком контролируемым можно отнести КС, работающие в операционной системе DOS, или КС собственной разработки.

Частично контролируемые компьютерные системы.

Именно к таким системам можно отнести современные КС, использующие ОС Windows 95/98, Windows NT, различные версии UNIX, поскольку аттестовать их программное обеспечение полностью не представляется возможным. Сегодня вряд ли кто-нибудь возьмется достоверно утверждать, что в нем отсутствуют ошибки, программные закладки недобросовестных разработчиков или соответствующих служб.

Безопасность в таких КС может быть обеспечена:

- использованием специальных аттестованных (полностью контролируемых) аппаратно-программных средств, выполняющих ряд защищенных операций и играющих роль специализированных модулей безопасности;

- изоляцией от злоумышленника ненадежной компьютерной среды, отдельного ее компонента или отдельного процесса с помощью полностью контролируемых средств.

В частично контролируемых КС использование каких-либо программно реализованных функций, отвечающих за шифрование, электронную цифровую подпись, доступ к информации, доступ к сети и т.д., становится показателем наивности администратора безопасности. Основную опасность представляет при этом возможность перехвата ключей пользователя, используемых при шифровании и предоставлении полномочий доступа к информации.

Одним из наиболее известных и надежных аппаратных модулей безопасности являются платы серии КРИПТОН, обеспечивающие как защиту ключей шифрования и электронной цифровой подписи (ЭЦП), так и неизменность их алгоритмов. Все используемые в системе ключи могут шифроваться на мастер-ключе (загружаемом в плату минуя шину компьютера) и храниться на внешнем носителе в зашифрованном виде. Они расшифровываются только внутри платы, в которой применяются специальные методы фильтрации и зашумления для предотвращения возможности считывания ключей с помощью специальной аппаратуры. В качестве ключевых носителей используются дискеты, микропроцессорные электронные карточки (смарт-карты) и "таблетки" Touch-Memory.

В современных аппаратно-программных средствах защиты от НСД для частично контролируемых КС можно серьезно рассматривать только функции доступа к ПК, выполняемые до загрузки операционной системы, и аппаратные функции блокировки портов ПК. Таким образом, существуют широкие возможности для разработки модулей безопасности для защиты выбранных процессов в частично контролируемых системах.

Основная проблема защиты отечественных корпоративных и ведомственных сетей состоит в том, что их программное и аппаратное обеспечение в значительной степени является заимствованным,

приспособленным к ведомственным нуждам и производится за рубежом. Сертификация и аттестация компонентов этих сетей очень трудоемкий процесс. За время аттестации одной системы в продажу поступает, как правило, не одна, а несколько новых версий системы или отдельных ее элементов.

Для построения защищенной сети необходимо прежде всего обеспечить защиту ее компонентов. К основным компонентам сети относятся:

- абонентские места, персональные компьютеры или терминалы клиента;
- центры коммутации пакетов, маршрутизаторы, шлюзы и сетевые экраны;
- корпоративный сервер, локальные серверы и серверы приложений;
- отдельные сегменты сетей.

Защита каждого из компонентов (как правило, компьютера) складывается из:

- исключения несанкционированного доступа к компьютеру со стороны консоли;
- разграничения доступа к ресурсам компьютера со стороны консоли;
- исключения несанкционированного доступа к компьютеру со стороны сети;
- разграничения доступа к ресурсам компьютера со стороны сети;
- обеспечения секретности используемых для защиты криптографических ключей.

Кроме того, необходимо также защитить сеть целиком от проникновения извне и каналы обмена с другими сетями.

4.2. Основные элементы и средства защиты от несанкционированного доступа

Фирма АНКАД известна на отечественном рынке как разработчик, производитель и поставщик аппаратно-программных криптографических

средств защиты информации серии КРИПТОН. Традиционно они выпускались в виде устройств с минимальным программным обеспечением. Встраивание их в конечные системы осуществлялось пользователем. В настоящий момент наряду с производством и поставкой устройств фирма предлагает готовые решения: от программ абонентского шифрования и электронной подписи до защиты отдельных рабочих мест и систем в целом.

В состав средств криптографической защиты информации (СКЗИ) фирмы АНКАД включены (рис. 4.1):

- устройства криптографической защиты данных (УКЗД) и их программные эмуляторы;
- контроллеры смарт-карт;
- системы защиты информации от несанкционированного доступа (СЗИ НСД);
- программы абонентского шифрования, электронной подписи и защиты электронной почты;
- коммуникационные программы прозрачного шифрования IP-пакетов и ограничения доступа к компьютеру по сети;
- криптомаршрутизаторы;
- библиотеки поддержки различных типов смарт-карт;
- библиотеки функций шифрования и электронной цифровой подписи для различных операционных систем.

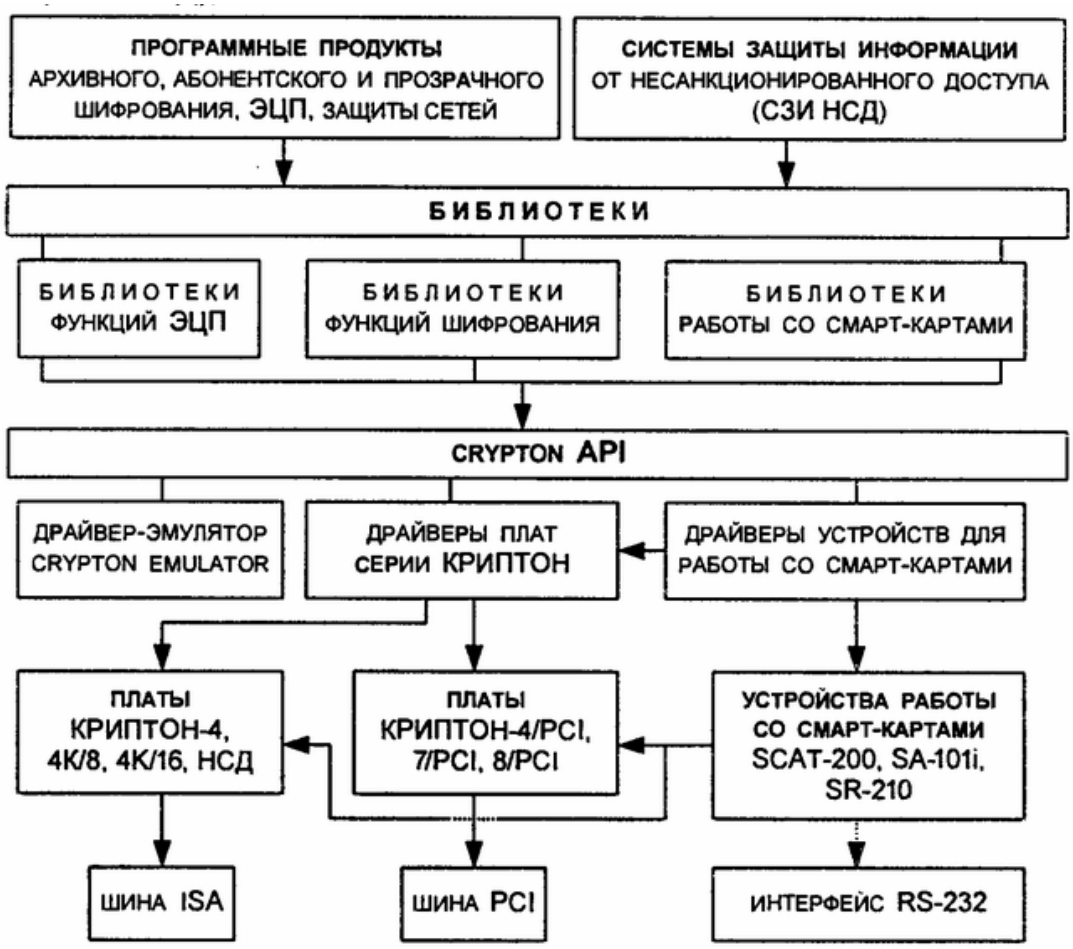


Рисунок 4.1 – Структура средств криптографической защиты информации

Отдельным рядом (семейством) устройств с использованием криптографических методов защиты являются специализированные модули безопасности для терминального оборудования, контрольно-кассовых машин, банкоматов и другого оборудования, используемого в палтежных и расчетных системах.

Устройства криптографической защиты данных серии КРИПТОН

Отличительной особенностью и в этом смысле уникальностью семейства УКЗД фирмы АНКАД является разработанная ею в рамках научно-технического сотрудничества с ФАПСИ отечественная специализированная микропроцессорная элементная база для наиболее полной и достоверной аппаратной реализации российского стандарта шифрования (см. табл. 10.1). В настоящее время серийно выпускаются УКЗД КРИПТОН-4, 4К/8 и 4К/16,

предназначенные для шифрования по ГОСТ28147-89 и генерации случайных чисел при формировании ключей. Началось производство устройств серии КРИПТОН с интерфейсом шины PCI.

В качестве ключевых носителей применяются дискеты, смарт-карты и Touch-Memory. Все ключи, используемые в системе, могут шифроваться на мастер-ключе и храниться на внешнем носителе в зашифрованном виде. Они расшифровываются только внутри платы. Устройство может выполнять проверку целостности программного обеспечения до загрузки операционной системы, а также играть роль электронного замка персонального компьютера, обеспечивая контроль и разграничение доступа к нему.

УКЗД семейства КРИПТОН аттестованы в ФАПСИ, широко применяются в разнообразных защищенных системах и сетях передачи данных и имеют сертификаты соответствия ФАПСИ в составе ряда АРМ абонентских пунктов при организации шифровальной связи I класса для защиты информации, содержащей сведения, составляющие государственную тайну.

Для систем защиты информации от несанкционированного доступа разработана специальная плата КРИПТОН-НСД, выполняющая программное шифрование по ГОСТ28147-89, аппаратную генерацию случайных чисел, загрузку ключей с дискет, смарт-карт или Touch Memory.

Для встраивания в конечные системы пользователя УКЗД имеют два уровня интерфейса в виде набора команд устройства и библиотеки функций. Команды выполняются драйверами устройств для операционных систем DOS, Windows 95/98 и NT4.0, UNIX. Функции реализованы на основе команд.

Наиболее важными особенностями рассматриваемых плат являются:

- наличие загружаемого до загрузки операционной системы мастер-ключа, что исключает его перехват;
- выполнение криптографических функций внутри платы, что исключает их подмену или искажение;
- наличие аппаратного датчика случайных чисел;

- реализация функций проверки целостности файлов операционной системы и разграничения доступа к компьютеру;
- высокая скорость шифрования: от 350 Кбайт/с (КРИПТОН-4) до 8800 Кбайт/с (КРИПТОН 8/PCI).

Допустимо параллельное подключение нескольких устройств одновременно в одном персональном компьютере, что может значительно повысить интегральную скорость шифрования и расширить другие возможности при обработке информации.

Средства серии КРИПТОН независимо от операционной среды обеспечивают:

- защиту ключей шифрования и электронной цифровой подписи (ЭЦП);
- неизменность алгоритма шифрования и ЭЦП.

Все ключи, используемые в системе, могут шифроваться на мастер-ключе и храниться на внешнем носителе в зашифрованном виде. Они расшифровываются только внутри платы. В качестве ключевых носителей используются дискеты, микропроцессорные электронные карточки (смарт-карты) и "таблетки" Touch-Memory.

Устройства для работы со смарт-картами

Для ввода ключей, записанных на смарт-карты, предлагаются разработанные фирмой АНКАД устройства для работы со смарт-картами, функции которых приведены в табл. 10.2.

Адаптер смарт-карт SA-101i предназначен для чтения и записи информации на смарт-картах. Адаптер подключается к УКЗД КРИПТОН и позволяет вводить в него ключи шифрования, хранящиеся на смарт-карте пользователя.

На одной смарт-карте могут быть размещены:

- таблица заполнения блока подстановок УЗ (ГОСТ28147-89);

- главный ключ шифрования;
- секретный и открытый ключи электронной цифровой подписи (ЭЦП) пользователя;

Таблица 4.2

Устройства для работы со смарт-картами

Наименование	Описание
SA-101i (Адаптер смарт-карт)	Запись/чтение информации на/с смарт-карты EEPROM (протокол I2C). Интерфейс с УКЗД серии КРИПТОН, обеспечивающий прямую загрузку ключей в устройство
SCAT-200 (Контроллер смарт-карт)	Шифрование по ГОСТ 28147-89, DES. Память для хранения одного мастер-ключа. Генерация случайных чисел. Запись/чтение информации на/с смарт-карты. Протоколы карт: I2C, GPM, ISO 781C T= 0. Интерфейс RS-232 с компьютером и специализированный интерфейс с УКЗД серии КРИПТОН
SR-210 (Контроллер смарт-карт)	Запись/чтение информации на/с смарт-карты. Протоколы карт: I2C, GPM, ISO 7816 T= 0, T= 1. Интерфейс RS-232 с компьютером и специализированный интерфейс с УКЗД серии КРИПТОН

- открытый ключ ЭЦП сертификационного центра;
- идентификатор пользователя системы защиты от несанкционированного доступа КРИПТОН-ВЕТО.

Адаптер SA-101i выпускается во внутреннем исполнении и легко встраивается в персональный компьютер на свободное место, предназначенное для дисковода.

Универсальный контроллер смарт-карт SCAT-200 предназначен для работы со смарт-картами. Контроллер SCAT-200 может подключаться как к УКЗД, так и к интерфейсу RS-232. Наиболее важными представляются следующие функции контроллера:

- запись информации на смарт-карту;
- чтение информации со смарт-карты;
- шифрование по ГОСТ28147-89 и DES;
- хранение секретных ключей (так же, как в плате КРИПТОН-4);
- генерация случайной последовательности;
- набор на клавиатуре PIN-кода.

В контроллере могут применяться электронные карточки:

- открытая память (протокол I2C);
- защищенная память (серия GPM);
- микропроцессорные карты (PCOS).

Универсальный контроллер SCAT-200 позволяет строить информационные системы на базе смарт-карт, что делает его полезным для систем:

- безналичных расчетов (дебетно/кредитные карты);
- контроля доступа (хранения прав доступа);
- хранения конфиденциальной информации (медицина, страхование, финансы);
- защиты информации (хранения идентификаторов, паролей и ключей шифрования).

Контроллер может использоваться в компьютерах, электронных кассовых аппаратах, электронных замках, торговых автоматах, бензоколонках,

платежных терминалах на базе IBM-совместимых компьютеров. Контроллер SCAT-200-совместный продукт фирмы АНКАД и АО "Скантек".

Универсальный контроллер смарт-карт SR-210 имеет те же возможности, что и SCAT-200, за исключением функций шифрования и генерации случайных последовательностей. Контроллер совместим с российскими интеллектуальными микропроцессорными карточками.

Программные эмуляторы функций шифрования устройств КРИПТОН

Для программной эмуляции функций шифрования УКЗД серии КРИПТОН разработаны и применяются:

- программа шифрования Crypton LITE для работы в среде MS-DOS;
- эмулятор Crypton Emulator для ОС Windows 95/98/NT.

Программа шифрования Crypton LITE предназначена для криптографической защиты (шифрования) информации, обрабатываемой ПЭВМ типа IBM PC/XT/AT 286, 384,486, Pentium в среде MS-DOS 3.0 и выше по алгоритму ГОСТ 28147-89.

Программа Crypton LITE полностью совместима с устройствами серии КРИПТОН, обеспечивающими гарантированную защиту информации. Crypton LITE и устройства серии КРИПТОН используют общее программное обеспечение.

Программа Crypton LITE рекомендуется для применений в компьютерах, где использование устройств КРИПТОН затруднено из-за конструктивных особенностей (например, в notebook). Crypton LITE применяется не только для защиты информации в компьютерах различного конструктивного исполнения, но и как средство поддержки при написании и отладке специализированного программного обеспечения к устройствам серии КРИПТОН.

Основные характеристики программы Crypton LITE:

Алгоритм шифрования	**ГОСТ 28147-89**
Скорость шифрования "память-память"	**до 3 Мбайт/с (для Pentium-2)**
Необходимая оперативная память	**2,5...8 Кбайт**

Длина ключа	**256 бит**
Ключевая система	**3-уровневая**

Программа Crypton LITE реализует все режимы алгоритма ГОСТ 28147-89:

- режим простой замены;
- режим гаммирования;
- режим гаммирования с обратной связью;
- режим вычисления имитовставки (имитоприставки).

Crypton LITE имеет встроенный датчик случайных чисел, используемый для генерации ключей. В программе Crypton LITE используются следующие ключевые элементы: К1 - первичный или файловый ключ (ключ данных), применяемый непосредственно для шифрования данных; К2 - вторичный ключ, применяемый для шифрования первичного ключа (в зависимости от используемой ключевой системы в качестве К2 выступают пользовательский ключ или сетевой ключ); ГК (или КЗ)-главный ключ (мастер-ключ), применяемый для шифрования других ключей; УЗ-узел замены, представляющий собой несекретный элемент, определяющий заполнение блока подстановки в алгоритме шифрования ГОСТ 28147-89.

Главный ключ и узел замены называют базовыми ключами. Базовые ключи загружаются при запуске программы Crypton LITE.

Дискета пользователя, на которой записаны базовые ключи ГК и УЗ, является ключом ко всей шифруемой информации. Для ключевой дискеты должен быть обеспечен специальный режим хранения и доступа. Следует отметить, что ГК может быть защищен от злоумышленников паролем (на случай потери ключевой дискеты).

Ключи К1 и К2 могут вводиться в программу Crypton LITE в любое время. В зашифрованном виде ключи К1 и К2 могут свободно храниться на внешних носителях и передаваться по каналам связи.

Открытый программный интерфейс программы Crypton LITE позволяет внедрять ее в любые системы без затруднений, а также разрабатывать

дополнительное программное обеспечение специального назначения для защиты информационных и финансовых, биржевых и банковских коммуникаций, баз данных и других массивов компьютерной информации.

Программные продукты фирмы АНКАД, совместимые с Crypton LITE, позволяют:

- прозрачно шифровать логические диски;
- разграничить доступ к компьютеру;
- осуществлять цифровую подпись электронных документов;
- передавать зашифрованную информацию по открытым каналам связи.

Программный эмулятор Crypton Emulator обеспечивает криптографическое преобразование данных по алгоритму шифрования ГОСТ 28147-89 в компьютере, работающем под управлением ОС Windows 95/98/NT. Основная задача данной программы заключается в эмуляции шифровальных функций устройств криптографической защиты данных серии КРИПТОН.

Для работы программы необходима операционная система Windows 95/98/NT 4.0. Перед установкой драйвера-эмулятора на компьютер необходимо установить программный интерфейс Crypton API версии 2.1 и выше. Никаких особых требований к компьютеру не предъявляется-драйвер-эмулятор будет работать на любом компьютере, где установлены вышеназванные ОС.

Win32-nporраммыi могут обращаться к функциям драйвера-эмулятора с помощью программного интерфейса Crypton API. Драйвер-эмулятор обеспечивает также возможность использования прерывания Ox4C в DOS-сессии Windows 95/98 или Windows NT 4.0. Драйвер-эмулятор находится на уровне ядра операционной системы, и все запросы на шифрование или расшифрование проходят через него при отсутствии в компьютере платы шифрования.

Входными данными для драйвера-эмулятора являются главный ключ (мастер-ключ) и узел замены (секретный элемент, определяющий заполнение блока подстановки в алгоритме ГОСТ 28147-89). Для инициализации драйвера-эмулятора необходимо загрузить базовые ключи ГК и УЗ с защищенной

ключевой дискеты. Эта загрузка выполняется с помощью специальной утилиты, поставляемой вместе с драйвером-эмулятором. В зависимости от применяемой операционной системы обмен данными между приложением Win32 или DOS и драйвером-эмулятором ведется двумя разными способами.

Рассмотрим, в частности, особенности обмена данными в Windows NT. При обращении приложения Win32 к драйверу-эмулятору запрос от приложения Win32 проходит три уровня:

1) уровень приложений;

2) уровень, обеспечивающий интерфейс приложений с драйвером;

3) уровень ядра ОС.

Драйвер эмулирует работу платы шифрования, т.е. каждое Win32-приложение имеет собственную виртуальную плату шифрования со своими ключами К1 и К2, однако ГК и УЗ являются общими для всех приложений.

Программные продукты фирмы АНКАД, совместимые с Crypton Emulator, позволяют эффективно решать разнообразные задачи защиты информации в компьютерных системах и сетях.

4.3. Системы защиты информации от несанкционированного доступа

Система криптографической защиты информации от НСД КРИПТОН-ВЕТО

Система предназначена для защиты ПК с процессором не ниже 386, работающего под управлением MS DOS 5.0 и выше, Windows 3.1. Персональный компьютер при этом может использоваться в качестве:

- абонентского пункта;
- центра коммутации пакетов;
- центра выработки ключей.

Система ограничивает круг лиц и их права доступа к информации на персональном компьютере. Ее реализация основана на технологиях "прозрачного" шифрования логических дисков по алгоритму ГОСТ 28147-89 и электронной цифровой подписи по ГОСТ 34.10/11-94. Согласно требованиям

ГТК России ее можно отнести к СЗ НСД класса 1В-1Б. (Сертификат №178 от 29 апреля 1998 г. на соответствие классу 1В, выдан ГТК при президенте Российской Федерации. Система также передана на сертификацию в ФАПСИ.)

В состав основных функций системы КРИПТОН-ВЕТО включены следующие (рис. 4.2):

- обеспечение секретности информации в случае кражи "винчестера" или ПК;
- обеспечение защиты от несанкционированного включения компьютера;
- разграничение полномочий пользователей по доступу к ресурсам компьютера;
- проверка целостности используемых программных средств системы в момент включения системы;
- проверка целостности программы в момент ее запуска на выполнение;
- запрещение запуска на выполнение посторонних программ;
- ведение системного журнала, регистрирующего события, возникающие в системе;
- обеспечение "прозрачного" шифрования информации при обращении к защищенному диску;
- обнаружение искажений, вызванных вирусами, ошибками пользователей, техническими сбоями или действиями злоумышленника.

Основным аппаратным элементом системы являются серийно выпускаемые аттестованные ФАПСИ платы серии КРИПТОН, с помощью которых проверяется целостность системы и выполняется шифрование по ГОСТ 28147-89. Система предполагает наличие администратора безопасности, который определяет взаимодействие между управляемыми ресурсами:

- пользователями;
- программами;

• логическими дисками;

• файлами (дискреционный и мандатный доступ);

• принтером;

• дисководами.

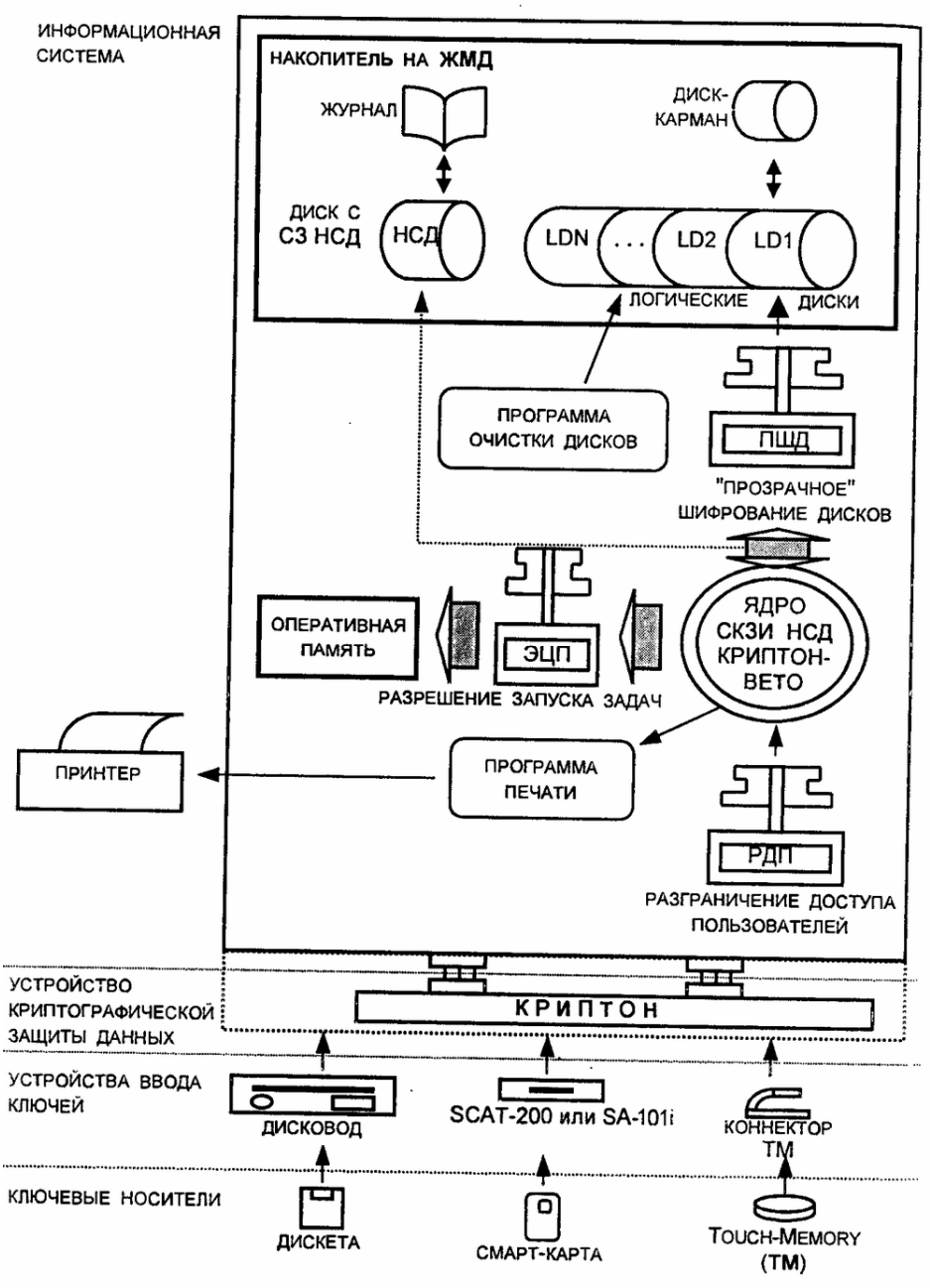


Рисунок 4.2 – Структура системы КРИПТОН-ВЕТО

Система обеспечивает защиту следующим образом. Жесткий диск разбивается на логические диски. Первый логический диск (C:) отводится для размещения системных программ и данных; последний логический диск-для

размещения СЗИ НСД и доступен только администратору. Остальные логические диски предназначены для хранения информации и программ пользователей. Эти диски можно разделить по пользователям и/или по уровню секретности размещаемой на них информации. Можно выделить отдельные диски с информацией различного уровня секретности (доступ к таким дискам осуществляется с помощью специальной программы, проверяющей допуск пользователя к документам-файлам). Сначала администратор устанавливает уровень секретности диска, а затем определяет круг лиц, имеющих доступ к этому диску. По форме хранения информации диски подразделяются на открытые и шифруемые; по уровню доступа - на доступные для чтения и записи, доступные только для чтения, недоступные (заблокированные).

Недоступный диск делается невидимым в DOS и, следовательно, не провоцирует пользователя на несанкционированный доступ к нему. Доступный только для чтения диск можно использовать для защиты не только от целенаправленного, но также от непреднамеренного (случайного) искажения (удаления) информации. Открытый диск ничем не отличается от обычного логического диска DOS. Очевидно, что системный диск должен быть открыт. Для шифруемых дисков используется шифрование информации в прозрачном режиме. При записи информации на диск она автоматически шифруется, при чтении с диска автоматически расшифровывается. Каждый шифруемый диск имеет для этого соответствующий ключ. Последнее делает бесполезными попытки улучшения своих полномочий пользователями, допущенными на ПК, поскольку они не имеют ключей доступа к закрытым для них дискам. Наличие шифрования обеспечивает секретность информации даже в случае кражи жесткого диска.

Для допуска к работе на ПК администратором формируется список пользователей, в котором:

- указывается идентификатор и пароль пользователя;
- определяется уровень допуска к секретной информации;
- определяются права доступа к логическим дискам.

В дальнейшем только администратор может изменить список пользователей и их полномочия.

Для исключения возможности установки на ПК посторонних программ с целью взлома защиты администратор определяет перечень программ, разрешенных к запуску на данном компьютере. Разрешенные программы подписываются администратором электронно-цифровой подписью (ЭЦП). Только эти программы могут быть запущены в системе. Использование ЭЦП одновременно с наличием разрешения позволяет отслеживать целостность запускаемых программ. Последнее исключает возможность запуска измененной программы, в том числе и произошедшего в результате непредвиденного воздействия "вируса".

Для входа в компьютер используются ключи, записанные на ключевой дискете, смарт-карте или на устройстве Touch-Memory. Ключи изготавливаются администратором системы и раздаются пользователям под расписку.

Для исключения загрузки компьютера в обход СЗ НСД загрузка осуществляется только с жесткого диска. При включении ПК (до загрузки операционной системы) с "винчестера" аппаратно проверяется целостность ядра системы безопасности КРИПТОН-ВЕТО, системных областей "винчестера", таблицы полномочий пользователей. Затем управление передается проверенному ядру системы безопасности, которая проверяет целостность операционной системы. Расшифрование полномочий пользователя, ключей зашифрованных дисков и дальнейшая загрузка операционной системы производятся лишь после заключения о ее целостности. В процессе работы в ПК загружены ключи только тех дисков, к которым пользователю разрешен доступ.

Для протоколирования процесса работы ведется журнал. В нем регистрируются следующие события:

- установка системы КРИПТОН-ВЕТО;
- вход пользователя в систему (имя, дата, время);
- попытка доступа к запрещенному диску (дата, время, диск);

- зашифрование диска;
- расшифрование диска;
- перешифрование диска;
- добавление нового пользователя;
- смена полномочий пользователя;
- удаление пользователя из списка;
- сброс причины останова системы;
- попытка запуска запрещенной задачи;
- нарушение целостности разрешенной задачи и т.д.

Журнал может просматриваться только администратором. Для проверки работоспособности системы используются программы тестирования. При необходимости пользователь может закрыть информацию на своем диске и от администратора, зашифровав последнюю средствами абонентского шифрования.

4.4. Комплекс КРИПТОН-ЗАМОК для ограничения доступа к компьютеру

Комплекс КРИПТОН-ЗАМОК предназначен для построения аппаратно-программных средств ограничения доступа к компьютеру с использованием УКЗД серии КРИПТОН. Комплекс позволяет организовать на базе персонального компьютера рабочее место с ограничением круга лиц, имеющих доступ к содержащейся в нем информации.

Для работы комплекса КРИПТОН-ЗАМОК необходим персональный компьютер IBM PC с процессором не ниже i386 и операционной системой-MS DOS, Windows 95/98/NT, UNIX и другими, для которых имеется соответствующий драйвер, позволяющий под управлением MS DOS понимать формат установленной на компьютере файловой системы.

Комплекс служит для защиты компьютеров с жесткими дисками, с файловыми системами в форматах FAT 12, FAT 16, FAT 32, NTFS, UNIX и т.д.

Работа с дисками с файловыми системами FAT 12, FAT 16 и FAT 32 обеспечивается средствами комплекса без дополнительных драйверов. Работа с

дисками с нестандартными файловыми системами NTFS, HTFS, UNIX и т.д., не поддерживаемыми операционной системой MS-DOS, может производиться только при наличии на компьютере соответствующих DOS-драйверов. Комплекс КРИПТОН-ЗАМОК выпускается в двух исполнениях:

- для жестких дисков объемом менее 8 Гбайт,
- для жестких дисков объемом более 8 Гбайт.

В базовый состав аппаратно-программных средств ограничения доступа к компьютеру входят:

- УКЗД серии КРИПТОН, поддерживающие режим работы комплекса ЗАМОК;
- комплект драйверов и библиотек УКЗД;
- комплекс ЗАМОК, включающий:

микросхему с программным обеспечением комплекса, устанавливаемую в УКЗД серии КРИПТОН;

инсталляционный дистрибутивный носитель с программным обеспечением, комплекса.

Установленный в персональный компьютер комплекс ограничения доступа КРИПТОН-ЗАМОК выполняет следующие функции:

- ограничивает доступ пользователей к компьютеру путем их идентификации и аутентификации;
- разделяет доступ пользователей к ресурсам компьютера в соответствии с их полномочиями;
- контролирует целостность ядра комплекса, программ операционной среды, прикладных программ и областей памяти в момент включения компьютера до загрузки его операционной системы;
- регистрирует события в защищенном электронном журнале;
- передает управление и параметры пользователя программному обеспечению (RUN-файлам), указанному администратором (например, ПО защиты от несанкционированного доступа).

В соответствии с выполняемыми функциями комплекс КРИПТОН-ЗАМОК содержит следующие основные подсистемы:

- подсистему управления доступом, состоящую из устройства КРИПТОН и программы обслуживания CRLOCK.EXE;
- подсистему регистрации и учета, включающую два журнала (аппаратный-на устройстве КРИПТОН, фиксирующий попытки входа в компьютер до запуска его операционной системы, и полный-на жестком диске, в котором после удачного входа в комплекс отображаются все события, в том числе и содержимое аппаратного журнала), управление которыми осуществляется программой обслуживания комплекса CRLOCK.EXE;
- подсистему обеспечения целостности, состоящую из устройства КРИПТОН и программы CHECKOS.EXE, проверяющей целостность главной ОС при работе комплекса.

При этом комплекс КРИПТОН-ЗАМОК обеспечивает выполнение следующих задач:

- в компьютер может войти только санкционированный пользователь;
- загружается достоверное ядро комплекса;
- загружается достоверная операционная система;
- проверяется целостность прикладного ПО, указанного администратором;
- • производится запуск программ, указанных администратором.

Рассмотрим штатную работу комплекса КРИПТОН-ЗАМОК. В начале работы с комплексом устройство КРИПТОН при инициализации его ключами с ключевого носителя (дискеты, смарт-карты или Touch Memory) загружает три файла: UZ.DB3 (УЗ, он один для всех пользователей данного компьютера); GK.DB3 (ГК, он уникален для каждого и может быть зашифрован на пароле пользователя) и файл-паспорт пользователя INIT.NSD.

Первые два файла обеспечивают выполнение устройством КРИПТОН криптографических процедур в соответствии с ГОСТ 28147-89 и формируются при помощи любой из программ генерации криптографических ключей, выпускаемых фирмой АНКАД для средств серии КРИПТОН (например, Crypton Soft, Crypton Tools или Cr Mng). Файл INIT.NSD уникален для каждого пользователя и используется при входе в комплекс для загрузки и проверки его ядра, поиска пользователя в файле полномочий, его аутентификации и расшифровки его записи. Файл INIT.NSD формируется на ключевом носителе пользователя: для администратора - автоматически программой INSTAL.EXE при установке комплекса на компьютер, а для всех остальных пользователей-администратором при помощи программ CRLOCK.EXE.

Алгоритм работы комплекса КРИПТОН-ЗАМОК включает следующие шаги:

• УКЗД КРИПТОН инициализируется файлами UZ.DB3 и GK.DB3.

• КРИПТОН загружает файл INIT.NSD и проверяет его целост-, ность по имитовставке. В случае нарушения целостности этого файла или при его отсутствии дальнейшая загрузка компьютера не производится.

• КРИПТОН производит поиск имени вошедшего пользователя в списке пользователей. В случае отсутствия пользователя в списке дальнейшая загрузка компьютера не производится.

• КРИПТОН производит аутентификацию пользователя-проверяет имитовставку его ключа. В случае несовпадения имитовставки пользователь считается несанкционированным и дальнейшая загрузка компьютера не производится.

• КРИПТОН производит загрузку ОС комплекса ЗАМОК с Flash-диска. При загрузке автоматически стартует программа проверки целостности защищаемой ОС компьютера (далее "главной ОС")-CHECKOS.EXE.

• CHECKOS.EXE получает параметры вошедшего пользователя от устройства КРИПТОН и:

- разблокирует клавиатуру;
- проверяет целостность файл-списка;
- проверяет целостность системных областей и файлов главной ОС;
- при наличии RUN-файлов проверяет их целостность и запускает на выполнение;
- по запросу пользователя меняет пароль ключей на его носителе; по запросу администратора запускает программу обслуживания комплекса CRLOCK.EXE;
- при успешном завершении всех проверок CHECKOS.EXE запускает главную ОС.

После загрузки главной ОС компьютера комплекс ограничения доступа к компьютеру прекращает свою деятельность и не вмешивается в дальнейшую работу компьютера (до следующей загрузки).

Далее устройство КРИПТОН может использоваться как обычный шифратор.

Механизм RUN-файлов позволяет в процессе работы комплекса КРИПТОН-ЗАМОК запускать любые программы с предварительной проверкой их целостности. В частности, механизм RUN-файлов может быть использован при проверке файлов, находящихся на логических дисках с нестандартными файловыми системами (NTFS, HPFS, UNIX и т.д.). Другой вариант использования -запуск из под комплекса КРИПТОН-ЗАМОК любого другого программного обеспечения: системы ЗНСД, криптомаршрутизатора, операционной системы и т.д. На этой основе может быть построена система защиты персонального компьютера с требуемыми свойствами.

Система защиты конфиденциальной информации Secret Disk

Система защиты конфиденциальной информации Secret Disk разработана компанией Aladdin при участии фирмы АНКАД и предназначена для широкого круга пользователей компьютеров: руководителей, менеджеров, бухгалтеров,

аудиторов, адвокатов, т. е. всех тех, кто должен заботиться о защите личной или профессиональной информации.

При установке системы Secret Disk на компьютере создаются новые логические диски, при записи на которые информация автоматически шифруется, а при чтении-расшифровывается. Работа с секретными дисками совершенно незаметна и равносильна встраиванию шифрования во все запускаемые приложения (например, бухгалтерскую программу, Word, Excel и т.п.).

В системе Secret Disk используется смешанная программно-аппаратная схема защиты с возможностью выбора, соответствующего российским нормативным требованиям криптографического алгоритма ГОСТ 28147-89 с длиной ключа 256 бит (программный эмулятор платы КРИПТОН или криптоплата КРИПТОН фирмы АНКАД).

Следует отметить, что применяемая в этой версии системы Secret Disk плата КРИПТОН сертифицирована ФАПСИ для защиты государственной тайны и поставляется по отдельному запросу фирмой АНКАД.

Система Secret Disk допускает также подключение внешнего криптомодуля того стандарта и с той длиной ключа, которую пользователь считает возможной для своих приложений.

Важная особенность системы Secret Disk заключается в том, что для доступа к защищенной информации необходим не только вводимый пользователем пароль, но и электронный идентификатор. В качестве такого идентификатора может использоваться обычный электронный ключ для параллельного порта, карточка PCMCIA для ноутбуков или смарт-карта (в этом случае необходимо установить в компьютер специальный считыватель смарт-карт).

Система Secret Disk подключается только после того, как пользователь введет пароль и система обнаружит соответствующий идентификатор. Поэтому, если пользователь вытащит из компьютера электронный ключ, злоумышленникам не поможет даже знание пароля.

При работе в критических ситуациях (например, под принуждением) система предоставляет пользователю специальный режим входа с помощью отдельного пароля и ряд блокировок, позволяющих не раскрывать информацию (т.е. доступ к диску будет открыт, но при попытке считать с него данные или переписать их на другой диск будут генерироваться системные ошибки Windows и будет разрушено содержимое памяти электронного идентификатора, без чего невозможно расшифровать содержимое секретного диска).

4.5 Система защиты данных Crypton Sigma

Система Crypton Sigma-это программный комплекс, предназначенный для защиты данных на персональном компьютере. По своим возможностям он во многом аналогичен системе Secret Disk. Будучи установленной на компьютере, система Crypton Sigma хранит конфиденциальные данные в зашифрованном виде, не допуская несанкционированный доступ и утечку данных. Для шифрования данных в системе Crypton Sigma используется алгоритм шифрования ГОСТ 28147-89.

Система защиты конфиденциальных данных Crypton Sigma ориентирована на широкий круг пользователей компьютеров-бизнесменов, менеджеров, бухгалтеров, адвокатов и др., т.е. всех тех, кто нуждается в защите профессиональной и личной информации.

Система Crypton Sigma легко устанавливается, проста и надежна в использовании, а также полностью "прозрачна"для всех программ и системных утилит операционной системы. При установке системы Crypton Sigma на компьютере создаются новые логические диски. При записи на эти диски информация автоматически шифруется, а при считывании-расшифровывается. Этот метод прозрачного шифрования позволяет полностью снять с пользователя заботу о защите данных. Работа с защищенными дисками незаметна для пользователя и равносильна встраиванию процедур шифрования/расшифрования в запускаемые приложения. Защищенные

системой диски на вид ничем не отличаются от обычных и могут использоваться в локальной или глобальной сети.

Поддерживаемые файловые системы-FAT 16, FAT 32 и NTFS. Система Crypton Sigma может работать как с УКЗД КРИПТОН, так и с его программным эмулятором. Криптографические ключи для защиты диска хранятся на съемном носителе (дискете), а при использовании УКЗД КРИПТОН возможно хранение ключевой информации на устройстве Touch Memory или смарт-карте. Кроме того, можно использовать устройство eТокеп (ключевой носитель для USB-порта). Применение УКЗД КРИПТОН не позволит злоумышленнику перехватить ключи пользователя с помощью внедренных программ.

Для работы системы Crypton Sigma требуется следующая минимальная конфигурация.

Компьютер:

• IBM PC/AT, PS/2 (с процессором X486 или выше) или полностью совместимый;

• минимум 8 Мбайт оперативной памяти;

• минимум 3 Мбайт свободного дискового пространства для установки и запуска системы Crypton Sigma.

Программно-аппаратное обеспечение:

• операционная система Windows 95/98 или Windows NT 4.0;

• интерфейс Crypton API v.2.2 или выше;

• УКЗД КРИПТОН с соответствующим драйвером или его программный эмулятор.

Система Crypton Sigma специально разрабатывалась так, чтобы сделать все процедуры управления максимально простыми и ясными. Все, что должен уметь пользователь,-это создать специальный файл (контейнер) для хранения зашифрованных данных и открыть его для доступа через логический диск системы Crypton Sigma.

Контейнер-это специальный файл, создаваемый при помощи Панели Управления системы Crypton Sigma. Контейнер можно открыть для доступа

через логический диск, обслуживаемый драйвером системы Crypton Sigma. Все файлы, находящиеся на этом логическом диске, хранятся в зашифрованном виде. Пользователь может создать любое количество контейнеров. Каждый контейнер имеет собственный пароль. Пользователь должен ввести этот пароль при создании контейнера и использовать его для получения доступа к тем данным, которые будут храниться в данном контейнере. Используя Панель Управления Crypton Sigma, пользователь может сменить пароль для выбранного контейнера при условии, что ему известен прежний пароль.

Схема работы системы Crypton Sigma показана на рис. 10.3. Логический диск системы Crypton Sigma создается и управляется драйвером этой системы. Этот логический диск используется для записи (чтения) данных в контейнер. Работа пользователя с таким логическим диском не отличается от работы с любыми другими дисками компьютера.

Рисунок 4.3 – Схема выполнения операций чтения (записи) с логических дисков системой Crypton Sigma

Драйвер системы Crypton Sigma обрабатывает запросы операционной системы на чтение (запись) с логических дисков, при этом драйвер автоматически производит шифрование/расшифрование данных. Следует отметить, что драйвер системы Crypton Sigma обрабатывает не все запросы на чтение/запись. Как уже упоминалось, система Crypton Sigma создает и обслуживает собственные логические диски. Драйвер системы обслуживает операции чтения (записи) только с этих логических дисков.

Эти диски доступны точно так же, как и остальные диски на компьютере, и могут обозначаться любыми незанятыми на данный момент буквами, например D:, E:, K:, Z:.

Данные, записываемые на логический диск, фактически записываются драйвером системы в контейнер системы. Естественно, размер доступной памяти логического диска равен размеру соответствующего контейнера. Максимальный размер контейнера, создаваемого

- на жестком диске с файловой системой FAT 16, равен 2 Гбайта;
- на жестком диске с файловой системой FAT 32. равен 4 Гбайта;
- на жестком диске с файловой системой NTFS, равен 512 Гбайт;
- на сетевом диске, равен 2 Гбайта.

Минимальный размер контейнера системы равен 5 Кбайт.

Для доступа к зашифрованным данным, хранящимся в контейнере, следует присоединить этот контейнер к выбранному логическому диску, например E:, и открыть его для доступа, введя соответствующий пароль. После завершения работы с данными следует закрыть этот логический диск для доступа. При этом данные, сохраненные в контейнере, станут недоступными.

Следует заметить, что если пользователь забудет пароль для доступа к данным, хранящимся в контейнере системы Crypton Sigma, то он полностью теряет возможность доступа к этим данным. Высокостойкие алгоритмы шифрования, используемые в системе Crypton Sigma, не позволяют восстановить информацию без знания пароля. Если существует опасность того, что пароль может быть забыт или утрачен, пользователь должен записать его и спрятать в надежном месте.

Отметим основные преимущества системы Crypton Sigma.

Надежная защита. Практически ни одна из существующих универсальных программ со встроенной защитой документов не имеет такой надежной защиты как Crypton Sigma. Компания Access Data (США) продает программный пакет, который вскрывает защиту данных в WordPerfect, Lotus 1-2-3, MS Excel, Symphony, Quattro Pro, Paradox, MS Word. Эта программа не

просто перебирает все возможные комбинации паролей она проводит математически обоснованный криптографический анализ и тратит на вскрытие защищенных данных всего лишь несколько секунд. Система Crypton Sigma выгодна отличается использованием стойких и надежных алгоритмов шифрования и не содержит встроенных программных блоков, позволяющих злоумышленнику совершить несанкционированный доступ к зашифрованным данным.

Высокая степень секретности. После того как данные записываются на логический диск системы Crypton Sigma, они уже никогда не хранятся на компьютере в открытом (расшифрованном) виде. Расшифрование данных происходит только в момент доступа к ним пользователей, знающих пароль. Система Sigma нигде не хранит паролей, необходимых для доступа к данным. Она лишь проверяет, подходит ли пароль для расшифрования данных, на которые претендует пользователь, точно так же, как замок нигде не хранит дубликат ключа, а только "проверяет", может ли данный ключ открыть его или нет.

Использование системы в локальных сетях. Программное обеспечение Crypton Sigma для Windows 95/98/NT позволяет использовать любой сетевой диск для создания на нем контейнеров и доступа к хранящимся на них данным. Эти сетевые диски могут быть выделены для доступа компьютерами с любой другой, отличной от Windows, операционной системой, например ОС семейства UNIX (OSF/1, LINUX, BSD, Sun OS, AIX и др.), а также Novell, Windows 3.xx и др.

Логические диски Crypton Sigma с точки зрения операционной системы или любого другого программного обеспечения выглядят точно так же, как обыкновенные локальные диски компьютера. Поэтому логические диски Crypton Sigma могут быть открыты для доступа через локальную компьютерную сеть. Таким образом, зашифрованная информация при необходимости может быть доступна для коллективного использования.

Удобство в использовании. Система Crypton Sigma проста в использовании и, следовательно, практически не позволяет совершать случайных действий, приводящих к появлению секретной информации на компьютере в открытом виде. Пользователю необходимо только ввести правильный пароль - об остальном позаботится система. После верификации пароля доступ к зашифрованной информации становится прозрачным для всех запускаемых пользователем программ. Все зашифрованные данные автоматически расшифровываются перед тем, как появиться перед пользователем, и автоматически зашифровываются перед записью их на диски, обслуживаемые системой Crypton Sigma.

4.6 Контрольные вопросы

1. Какие операционные системы можно отнести к частично контролируемым компьютерным системам и почему?
2. Какие вы знаете устройства для работы со смарт-картами?
3. Выделите основные преимущества и недостатки системы Crypton Sigma.

5. МЕТОДЫ И СРЕДСТВА ОГРАНИЧЕНИЯ ДОСТУПА К КОМПОНЕНТАМ ЭВМ

5.1 Защита информации в ПЭВМ

Усложнение методов и средств организации машинной обработки информации, а также широкое использование вычислительных сетей приводит к тому, что информация становится все более уязвимой.

В связи с этим **защита информации** в процессе ее сбора, хранения и обработки приобретает исключительно важное значение (особенно в коммерческих и военных областях).

Под **защитой информации** понимается совокупность мероприятий, методов и средств, обеспечивающих решение следующих основных задач:

- проверка целостности информации;
- исключение несанкционированного доступа к ресурсам ПЭВМ и хранящимся в ней программам и данным (с целью сохранения трех основных свойств защищаемой информации: целостности, конфиденциальности, готовности);
- исключение несанкционированного использования хранящихся в ПЭВМ программ (т.е. защита программ от копирования).

Возможные каналы утечки информации, позволяющие нарушителю получить доступ к обрабатываемой или хранящейся в ПЭВМ информации, принято классифицировать на три группы, в зависимости от типа средства, являющегося основным при получении информации. Различают 3 типа средств: человек, аппаратура, программа.

С первой группой, в которой основным средством является человек, связаны следующие основные возможные утечки:

- чтение информации с экрана посторонним лицом;
- расшифровка программой зашифрованной информации;

- хищение носителей информации (магнитных дисков, дискет, лент и т. д.).

Ко второй группе каналов, в которых основным средством является аппаратура, относятся следующие возможные каналы утечки:

- подключение к ПЭВМ специально разработанных аппаратных средств, обеспечивающих доступ к информации;

- использование специальных технических средств для перехвата

электромагнитных излучений технических средств ПЭВМ. В группе каналов, в которых основным средством является программа, можно выделить следующие возможные каналы утечки:

- несанкционированный доступ программы к информации;

- расшифровка программой зашифрованной информации;

- копирование программой информации с носителей.

Будем рассматривать средства защиты, обеспечивающие закрытие возможных каналов утечки, в которых основным средством является программа. Заметим, что такие средства в ряде случаев позволяют достаточно надежно закрыть некоторые возможные каналы утечки из других групп. Так, криптографические средства позволяют надежно закрыть канал, связанный с хищением носителей информации.

Обзор методов защиты информации

Проблемы защиты информации программного обеспечения имеют широкий диапазон: от законодательных аспектов защиты интеллектуальной собственности (прав автора) до конкретных технических устройств.

Средства защиты можно подразделить на следующие категории:

1 - средства собственной защиты;

2 - средства защиты в составе вычислительной системы;

3 - средства защиты с запросом информации;

4 - средства активной защиты;

5 - средства пассивной защиты.

Средства защиты информации

Собственной защиты	В составе ВС	С запросом информации	Активные		Пассивные
- документация - машинный код - сопровождение - авторское право - заказное проектирование	- защита магнитных дисков - специальная аппаратура - замки защиты - изменения функций	- пароли - шифры - сигнатура - аппаратура защиты (ПЗУ, преобразователи) генератор случайных чисел	- замки защиты (время, данные) - искаженные программы (программы вирусы, искажение функций)	- сигнал тревоги - запуск по ключам - авторская эстетика	- идентификация программ - частотный анализ - корреляционный анализ - «родимые пятна» - устройство контроля

Рисунок 5.1 - Классификация средств защиты информации

5.2 Защита информации, обрабатываемой ПЭВМ и ЛВС, от утечки по сети электропитания

Мероприятия по защите информации нередко требуют особого подхода к их применению. Для того, чтобы сделать правильный выбор в кризисной ситуации, предлагаем вам ознакомиться с наиболее распространенными из них.

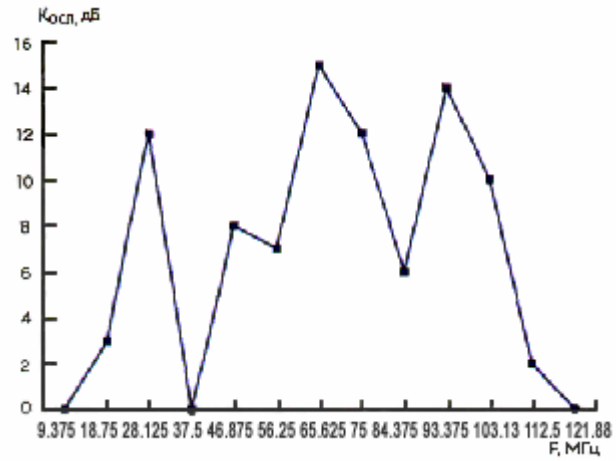


Рисунок 5.2 - Характеристика затухания информативного сигнала в линии электропитания длиной 16 м

Информативный сигнал в сети электропитания имеет достаточную для перехвата злоумышленником мощность и широкий частотный диапазон, что усложняет задачу защиты информации, обрабатываемой ПЭВМ и ЛВС. Таким образом, при соблюдении определенных энергетических и временных условий может возникнуть электромагнитный канал утечки конфиденциальной информации, обрабатываемой ПЭВМ и циркулирующей в ЛВС. Эти условия можно представить в виде:

$$\frac{P_{ис}}{P_ш} \geq \left(\frac{P_c}{P_ш}\right)_{пред} \qquad \Delta t \cong \Delta T, \tag{5.1}$$

где Рис – мощность информативного сигнала в точке приема;

Рш – мощность шумов в точке приема;

$\left(\frac{P_c}{P_ш}\right)_{пред}$ – предельное отношение мощности сигнала к мощности шума, при котором сигнал может быть перехвачен техническим средством злоумышленника;

T, - время обработки конфиденциальной информации;

t - время работы средства перехвата информации.

5.3 Виды мероприятий по защите информации

Мероприятия по инженерно-технической защите информации от утечки по электромагнитному каналу подразделяются на организационные и технические. Организационными мероприятиями предусматривается исключение нахождения в местах наличия информативного сигнала злоумышленника и контроль за его действиями и передвижением, а также первичные меры блокирования информативных сигналов, организуемые и выполняемые службой охраны объекта. Технические мероприятия направлены на недопущение выхода информативного сигнала за пределы контролируемой территории с помощью сертифицированных технических средств защиты. В качестве технических мероприятий могут использоваться как активные, так и пассивные способы защиты.

В соответствии с выражением (5.1) для защиты информации при ее утечке через сеть электропитания могут быть использованы:

1. Организационные мероприятия, ограничивающие присутствие злоумышленника в зоне возможного получения из сети электропитания информативного сигнала. Для этого вокруг объекта организуется контролируемая территория; ПЭВМ и кабели ЛВС размещаются с учетом радиуса зоны возможного перехвата информации; система электропитания строится в соответствии со специальными требованиями, а также используются различные разделительные системы для устранения утечки информативных сигналов.

2. Активные способы защиты, направленные на увеличение Рш (создание маскирующего шума). Данный способ защиты осуществляется за счет скрытия информативных излучений шумовыми помехами внутри самой ПЭВМ и в линиях электропитания. Для этого разработаны генераторы шума, встраиваемые в компьютер вместо FDD 3,5" в виде отдельной платы, а также генераторы для создания маскирующего шума в фазовых цепях и нейтрали системы электропитания.

3. Пассивные способы защиты, направленные на уменьшение (Рис 5.3). Для минимизации паразитных связей внутри ПЭВМ используются различные схемотехнические решения: применение радиоэкранирующих и радиопоглощающих материалов; экранирование корпусов элементов; оптимальное построение системы электропитания ПЭВМ; установка помехоподавляющих фильтров в цепях электропитания, в сигнальных цепях интерфейсов и на печатных платах ПЭВМ. Для предотвращения паразитной связи через электромагнитное поле совместно пролегающие кабели ЛВС и системы электропитания разносятся на безопасное расстояние. Также применяются фильтрация, прокладка цепей электропитания в экранирующих конструкциях, скрутка проводов электропитания и др.

Исследование сетей электропитания технических средств, используемых для обработки конфиденциальной информации, показало, что помимо

традиционных средств помехоподавления большое ослабление наведенных информативных сигналов обеспечивают и сами элементы сети электропитания – силовые кабели, трансформаторы, двигатели-генераторы, силовое оборудование трансформаторной подстанции и распределительных пунктов (сборные щиты, фидерные автоматы и т.п.).

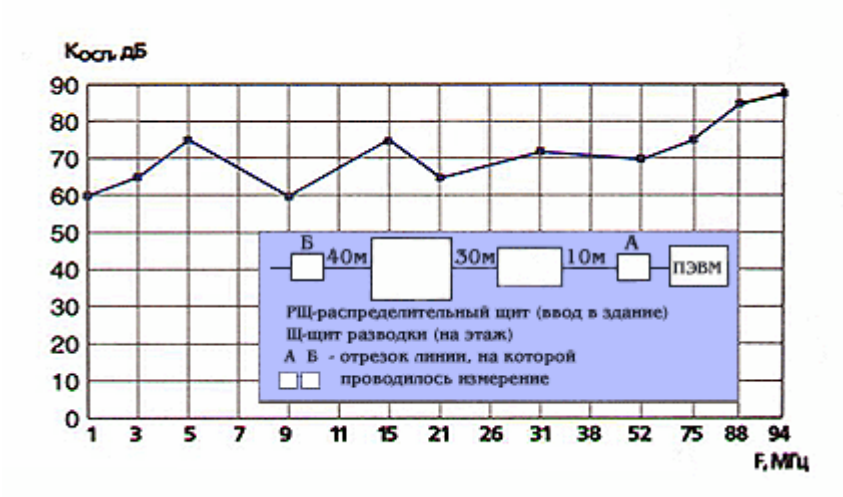


Рисунок 5.3 - Ослабление информативного сигнала на приведённом тракте его распространения по цепи электропитания

Для определения подобных характеристик в широком диапазоне частот было исследовано поведение линий электропитания различной конфигурации. На рис. 5.2 приведена характеристика затухания информативного сигнала в линии электропитания длиной 16 м, расположенной в воздушном пространстве. Эта характеристика свидетельствует о резонансном характере затухания, близком к эквивалентным схемам разомкнутой линии.

Резонансный характер затухания информативного сигнала в линиях электропитания наблюдался практически во всех вариантах конфигурации этих сетей (например, для участка цепи электропитания ПЭВМ, показанного на рис. 5.3 и состоящего из силового кабеля, соединяющего розетку электропитания ПЭВМ, щит разводки электропитания на этаже и распределительный щит на силовом вводе в здание).

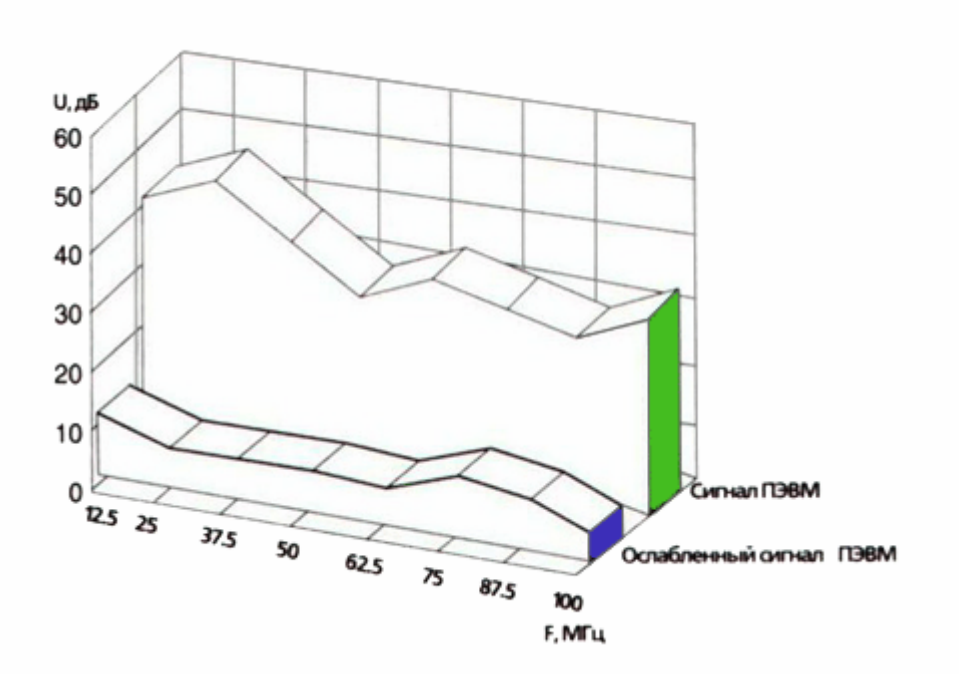


Рисунок 5.4 - Ослабление высокочастотного информативного сигнала

Однако ввиду того, что величина ослабления высокочастотного сигнала в силовых кабелях, входящих в тракты распространения информативных сигналов по сети электропитания, зависит как от линейной протяженности цепи, так и от конфигурации сети электропитания (длины ответвлений, наличия неоднородной трассы - кабельные вставки, места подключения приемников и т.д.), ее измерение необходимо проводить на каждом конкретном объекте на реальных трактах электропитания.

4. Комплексные мероприятия, включающие перечисленные выше с учетом их эффективности. Практика проведения защитных мероприятий показала, что объекты не всегда могут быть защищены от утечки информации за счет наводок информативного сигнала на цепи электропитания применением только пассивных или только активных способов защиты. Использование активных средств не всегда возможно из-за требований электромагнитной совместимости; кроме того, проведение защитных мероприятий нередко требует приобретения значительного количества средств защиты (как пассивных, так и активных), что зачастую ограничено финансовыми средствами. Исследования, проведенные в ходе защитных мероприятий, показали, что участок тракта, состоящий из силового кабеля, соединяющего

розетку электропитания ПЭВМ и распределительный щит, распределительного щита и кабеля, соединяющего распределительный щите трансформаторной подстанцией, обеспечивает минимальное ослабление высокочастотного информативного сигнала на 30-40 дБ (рис. 5.4).

Применение сетевого генератора шума позволяет создать уровень маскирующих помех порядка 40-60 дБ, что вполне достаточно для надежного закрытия этого канала утечки информации. Результаты проводимых мероприятий по защите ПЭВМ типа IBM PC AT 486 SX от утечки информативного сигнала по сети электропитания представлены на рис. 5.5.

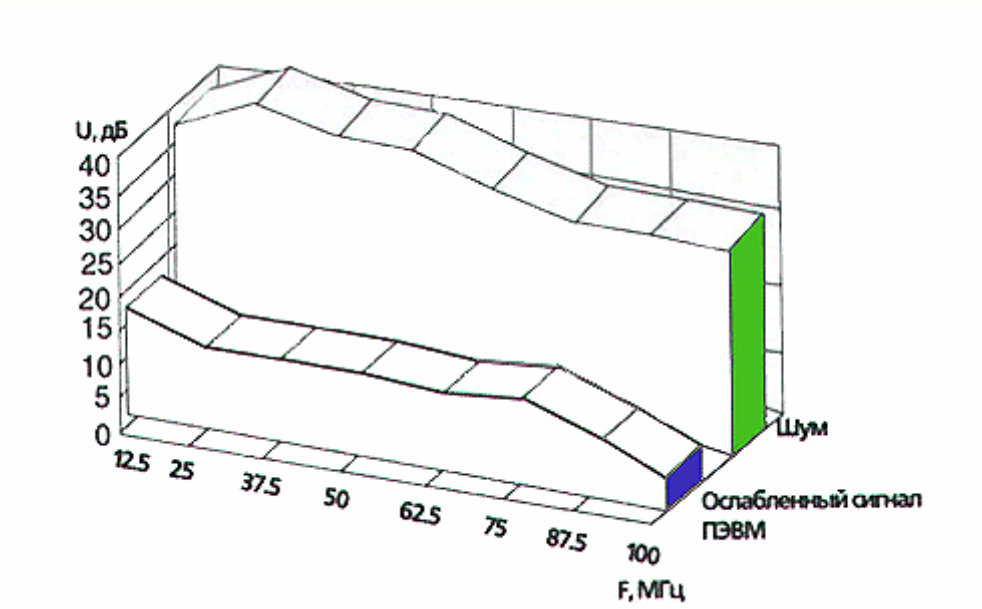


Рисунок 5.5 – Результаты мероприятий по защите ПЭВМ типа IBM PC AT 486 SX от утечки информативного сигнала по сети электропитания

5.4 Современные системы защиты ПЭВМ от несанкционированного доступа к информации

В качестве примеров отдельных программ, повышающих защищенность КС от НСД, можно привести утилиты из пакета Norton Utilities, такие как программа шифрования информации при записи на диск Diskreet или Secret disk, программа стирания информации с диска WipeInfo, программа контроля обращения к дискам DiskMonitor и др.

Отечественными разработчиками предлагаются программные системы защиты ПЭВМ «Снег-1.0», «Кобра», «Страж-1.1» и др. В качестве примеров отечественных аппаратно-программных средств защиты, имеющих сертификат Гостехкомиссии, можно привести системы «Аккорд-4», «DALLAS LOCK 3.1», «Редут», «ДИЗ-1». Аппаратно-программные комплексы защиты реализуют максимальное число защитных механизмов:

- идентификация и аутентификация пользователей;
- разграничение доступа к файлам, каталогам, дискам;
- контроль целостности программных средств и информации;
- возможность создания функционально замкнутой среды пользователя;
- защита процесса загрузки ОС;
- блокировка ПЭВМ на время отсутствия пользователя;
- криптографическое преобразование информации;
- регистрация событий;
- очистка памяти.

Программные системы защиты в качестве идентификатора используют, как правило, только пароль. Пароль может быть перехвачен резидентными программами двух видов. Программы первого вида перехватывают прерывания от клавиатуры, записывают символы в специальный файл, а затем передают управление ОС. После перехвата установленного числа символов программа удаляется из ОП. Программы другого вида выполняются вместо штатных программ считывания пароля. Такие программы первыми получают управление и имитируют для пользователя работу со штатной программой проверки пароля. Они запоминают пароль, имитируют ошибку ввода пароля и передают управление штатной программе парольной идентификации. Отказ при первом наборе пароля пользователь воспринимает как сбой системы или свою ошибку и осуществляет повторный набор пароля, который должен завершиться допуском его к работе. При перехвате пароля в обоих случаях пользователь не

почувствует, что его пароль скомпрометирован. Для получения возможности перехвата паролей злоумышленник должен изменить программную структуру системы. В некоторых программных системах защиты («Страж-1.1») для повышения достоверности аутентификации используются съемные магнитные диски, на которых записывается идентификатор пользователя.

Значительно сложнее обойти блок идентификации и аутентификации в аппаратно-программных системах защиты от НСД. В таких системах используются электронные идентификаторы, чаще всего - Touch Memory.

Для каждого пользователя устанавливаются его полномочия в отношении файлов, каталогов, логических дисков. Элементы, в отношении которых пользователю запрещены любые действия, становятся «невидимыми» для него, т. е. они не отображаются на экране монитора при просмотре содержимого внешних запоминающих устройств.

Для пользователей может устанавливаться запрет на использование таких устройств, как накопители на съемных носителях, печатающие устройства. Эти ограничения позволяют предотвращать реализацию угроз, связанных с попытками несанкционированного копирования и ввода информации, изучения системы защиты.

В наиболее совершенных системах реализован механизм контроля целостности файлов с использованием хэш-функции. При чем существуют системы, в которых контрольная характеристика хранится не только в ПЭВМ, но и в автономном ПЗУ пользователя. Постоянное запоминающее устройство, как правило, входит в состав карты или жетона, используемого для идентификации пользователя. Так в системе «Аккорд-4» хэш-функции вычисляются для контролируемых файлов и хранятся в специальном файле в ПЭВМ, а хэш-функция, вычисляемая для специального файла, хранится в Touch Memory.

После завершения работы на ПЭВМ осуществляется запись контрольных характеристик файлов на карту или жетон пользователя. При входе в систему осуществляется считывание контрольных характеристик из ПЗУ карты или

жетона и сравнение их с характеристиками, вычисленными по контролируемым файлам. Для того, чтобы изменение файлов осталось незамеченным, злоумышленнику необходимо изменить контрольные характеристики как в ПЭВМ, так и на карте или жетоне, что практически невозможно при условии выполнения пользователем простых правил.

Очень эффективным механизмом борьбы с НСДИ является создание функционально-замкнутых сред пользователей. Суть его состоит в следующем. Для каждого пользователя создается меню, в которое попадает пользователь после загрузки ОС. В нем указываются программы, к выполнению которых допущен пользователь. Пользователь может выполнить любую из программ из меню. После выполнения программы пользователь снова попадает в меню. Если эти программы не имеют возможностей инициировать выполнение других программ, а также предусмотрена корректная обработка ошибок, сбоев и отказов, то пользователь не сможет выйти за рамки установленной замкнутой функциональной среды. Такой режим работы вполне осуществим во многих АСУ.

Защита процесса загрузки ОС предполагает осуществление загрузки именно штатной ОС и исключение вмешательства в ее структуру на этапе загрузки. Для обеспечения такой защиты на аппаратном или программном уровне блокируется работа всех ВЗУ, за исключением того, на котором установлен носитель со штатной ОС. Если загрузка осуществляется со съемных носителей информации, то до начала загрузки необходимо удостовериться в том, что установлен носитель со штатной ОС. Такой контроль может быть осуществлен программой, записанной в ПЗУ ЭВМ.

Способы контроля могут быть разными: от контроля идентификатора до сравнения хэш-функций. Загрузка с несъемного носителя информации все же является предпочтительнее.

Процесс загрузки ОС должен исключать возможность вмешательства до полного завершения загрузки, пока не будут работать все механизмы системы

защиты. В КС достаточно блокировать на время загрузки ОС все устройства ввода информации и каналы связи.

При организации многопользовательского режима часто возникает необходимость на непродолжительное время отлучиться от рабочего места, либо передать ЭВМ другому пользователю. На это время необходимо блокировать работу ЭВМ. В этих случаях очень удобно использовать электронные идентификаторы, которые при работе должны постоянно находиться в приемном устройстве блока идентификации ЭВМ. При изъятии идентификатора гасится экран монитора и блокируются устройства управления. При предъявлении идентификатора, который использовался при доступе к ЭВМ, осуществляется разблокировка, и работа может быть продолжена. При смене пользователей целесообразно производить ее без выключения ЭВМ. Для этого необходим аппаратно-программный или программный механизм корректной смены полномочий. Если предыдущий пользователь корректно завершил работу, то новый пользователь получает доступ со своими полномочиями после успешного завершения процедуры аутентификации.

Одним из наиболее эффективных методов разграничения доступа является криптографическое преобразование информации. Этот метод является универсальным. Он защищает информацию от изучения, внедрения программных закладок, делает операцию копирования бессмысленной. Поэтому криптографические методы защиты информации рассматриваются довольно подробно в других главах. Здесь необходимо лишь отметить, что пользователи могут использовать одни и те же аппаратно-программные или программные средства криптографического преобразования или применять индивидуальные средства.

Для своевременного пресечения несанкционированных действий в отношении информации, а также для контроля за соблюдением установленных правил субъектами доступа, необходимо обеспечить регистрацию событий, связанных с защитой информации. Степень подробности фиксируемой информации может изменяться и обычно определяется администратором

системы защиты. Информация накапливается на ВЗУ. Доступ к ней имеет только администратор системы защиты.

Важно обеспечивать стирание информации в ОП и в рабочих областях ВЗУ. В ОП размещается вся обрабатываемая информация, причем, в открытом виде. Если после завершения работы пользователя не осуществить очистку рабочих областей памяти всех уровней, то к ней может быть осуществлен несанкционированный доступ.

5.5 Контрольные вопросы

1. Рассмотрите наиболее распространенные мероприятия по защите информации.
2. Какие мероприятия проводятся для защиты информации при ее утечке через сеть электропитания?
3. Проанализируйте результаты проводимых мероприятий по защите ПЭВМ типа IBM PC AT 486 SX от утечки информативного сигнала по сети электропитания представленые на рис. 5.5.
4. Приведите современные системы защиты ПЭВМ от несанкционированного доступа к информации.

6. ЗАЩИТА ПРОГРАММ ОТ НЕСАНКЦИОНИРОВАННОГО КОПИРОВАНИЯ

Дальнейшее развитие информационных технологий невозможно без создания новых программных средств различного назначения, баз данных, компьютерных средств обучения и других продуктов, предназначенных для корпоративного или персонального использования. При этом возникает проблема защиты авторских прав создателей и владельцев продуктов информационных технологий. Отсутствие такой защиты может привести к оттоку из сферы производства программного обеспечения части способных к творческой деятельности специалистов, снижению качества создаваемых информационных ресурсов и другим негативным социальным последствиям.

К сожалению, в настоящее время попытки нарушения авторских прав на объекты интеллектуальной собственности становятся достаточно регулярным и повсеместным явлением. Недостаточная эффективность правовых методов защиты интересов создателей и владельцев информационных ресурсов приводит к необходимости создания программных средств их защиты.

Под системой защиты от несанкционированного использования и копирования (защиты авторских прав, или просто защиты, от копирования) понимается комплекс программных или программно-аппаратных средств, предназначенных для усложнения или запрещения нелегального распространения, использования и (или) изменения программных продуктов и иных информационных ресурсов. Термин «нелегальное» здесь понимается как производимое без согласия правообладателя. Нелегальное изменение информационного ресурса может потребоваться нарушителю для того, чтобы измененный им продукт не подпадал под действие законодательства о защите авторских прав.

Под надежностью системы защиты от несанкционированного копирования понимается ее способность противостоять попыткам изучения

алгоритма ее работы и обхода реализованных в нем методов защиты. Очевидно, что любая программная или программно-аппаратная система защиты от копирования может быть преодолена за конечное время, так как процессорные команды системы защиты в момент своего исполнения присутствуют в оперативной памяти компьютера в открытом виде. Также очевидно, что надежность системы защиты равна надежности наименее защищенного из ее модулей.

Выделим принципы создания и использования систем защиты от копирования.

1. Учет условий распространения программных продуктов:
   - распространение дистрибутивных файлов на магнитных носителях через сеть торговых агентов или через сеть Интернет с последующей установкой самим пользователем, который при этом может пытаться копировать дистрибутивные магнитные диски, исследовать алгоритм работы системы защиты при помощи специальных программных средств (отладчиков и дисассемблеров), пытаться нарушить условия лицензионного соглашения и установить продукт на большем числе компьютеров, пытаться смоделировать алгоритм работы системы защиты для изготовления аналогичного варианта дистрибутивных файлов и распространения их от своего имени;
   - установка программного продукта официальным представителем правообладателя, при котором пользователь может пытаться нарушить условия лицензионного соглашения или исследовать алгоритм работы системы защиты;
   - приобретение и использование программного продукта лицами или организациями, не заинтересованными в его нелегальном распространении среди их коммерческих конкурентов — в этом случае возможны только попытки несанкционированного использования продукта другими лицами;

– приобретение программного продукта только для снятия с него системы защиты.

2. Учет возможностей пользователей программного продукта по снятию с него системы защиты (наличие достаточных материальных ресурсов, возможность привлечения необходимых специалистов и т.п.).

3. Учет свойств распространяемого программного продукта (предполагаемого тиража, оптовой и розничной цены, частоты обновления, специализированное™ и сложности продукта, уровня послепродажного сервиса для легальных пользователей, возможности применения правовых санкций к нарушителю и др.).

4. Оценка возможных потерь при снятии защиты и нелегальном использовании.

5. Учет особенностей уровня знаний и квалификации лиц, снимающих систему защиты.

6. Постоянное обновление использованных в системе защиты средств.

При добавлении к программному продукту системы его защиты от копирования возможен выбор уже имеющейся системы, что минимизирует издержки на установку системы защиты. Однако имеющаяся система защиты от копирования будет более легко сниматься с программного продукта (в силу ее пристыкованности к нему, см. подразд. 1.5), а также может оказаться несовместимой с защищаемой программой и имеющимся у пользователя программно-аппаратным обеспечением. Поэтому более целесообразной является разработка специализированной системы защиты от копирования конкретного программного продукта, что, однако, более заметно увеличит затраты на его производство.

Основные требования, предъявляемые к системе защиты от копирования:

- обеспечение некопируемости дистрибутивных дисков стандартными средствами (для такого копирования нарушителю по требуется тщательное изучение структуры диска с помощью

специализированных программных или программно-аппаратных средств);

- обеспечение невозможности применения стандартных отладчиков без дополнительных действий над машинным кодом программы или без применения специализированных программно-аппаратных средств (нарушитель должен быть специалистом высокой квалификации);
- обеспечение некорректного дисассемблирования машинного кода программы стандартными средствами (нарушителю потребуется использование или разработка специализированных дисассемблеров);
- обеспечение сложности изучения алгоритма распознавания индивидуальных параметров компьютера, на котором установлен программный продукт, и его пользователя или анализа применяемых аппаратных средств защиты (нарушителю будет сложно эмулировать легальную среду запуска защищаемой программы).

Выделим основные компоненты системы защиты программных продуктов от несанкционированного копирования:

- модуль проверки ключевой информации (некопируемой метки на дистрибутивном диске, уникального набора характеристик компьютера, идентифицирующей информации для легального пользователя) — может быть добавлен к исполнимому коду защищаемой программы по технологии компьютерного вируса, в виде отдельного программного модуля или в виде отдельной функции проверки внутри защищаемой программы;
- модуль защиты от изучения алгоритма работы системы защиты;
- модуль согласования с работой функций защищаемой программы в случае ее санкционированного использования;

- модуль ответной реакции в случае попытки несанкционированного использования (как правило, включение такого модуля в состав системы защиты нецелесообразно по морально-этическим соображениям).

6.1 Методы, затрудняющие считывание скопированной информации

Создание копий программных средств для изучения или несанкционированного использования осуществляется с помощью устройств вывода или каналов связи.

Одним из самых распространенных каналов несанкционированного копирования является использование накопителей на съемных магнитных носителях. Угроза несанкционированного копирования информации блокируется методами, которые могут быть распределены по двум группам:

- методы, затрудняющие считывание скопированной информации;
- методы, препятствующие использованию информации.

Методы из первой группы основываются на придании особенностей процессу записи информации, которые не позволяют считывать полученную копию на других накопителях, не входящих в защищаемую КС. Таким образом, эти методы направлены на создание совместимости накопителей только внутри объекта. В КС должна быть ЭВМ, имеющая в своем составе стандартные и нестандартные накопители. На этой ЭВМ осуществляется ввод (вывод) информации для обмена с другими КС, а также переписывается информация со стандартных носителей на нестандартные, и наоборот. Эти операции осуществляются под контролем администратора системы безопасности. Такая организация ввода-вывода информации существенно затрудняет действия злоумышленника не только при несанкционированном копировании, но и при попытках несанкционированного ввода информации.

Особенности работы накопителей на съемных магнитных носителях должны задаваться за счет изменения программных средств, поддерживающих

их работу, а также за счет простых аппаратных регулировок и настроек. Такой подход позволит использовать серийные образцы накопителей.

Самым простым решением является нестандартная разметка (форматирование) носителя информации. Изменение длины секторов, межсекторных расстояний, порядка нумерации секторов и некоторые другие способы нестандартного форматирования дискет затрудняют их использование стандартными средствами операционных систем. Нестандартное форматирование защищает только от стандартных средств работы с накопителями. Использование специальных программных средств (например, DISK EXPLORER. для IBM-совместимых ПЭВМ) позволяет получить характеристики нестандартного форматирования.

Перепрограммирование контроллеров ВЗУ, аппаратные регулировки и настройки вызывают сбой оборудования при использовании носителей на стандартных ВЗУ, если форматирование и запись информации производились на нестандартном ВЗУ. В качестве примеров можно привести изменения стандартного алгоритма подсчета контрольной суммы и работы системы позиционирования накопителей на гибких магнитных дисках.

В контроллерах накопителей подсчитывается и записывается контрольная сумма данных сектора. Если изменить алгоритм подсчета контрольной суммы, то прочитать информацию на стандартном накопителе будет невозможно из-за сбоев.

Позиционирование в накопителях на магнитных дисках осуществляется следующим образом. Определяется номер дорожки, на которой установлены магнитные головки. Вычисляется количество дорожек, на которое необходимо переместить головки и направление движения. Если нумерацию дорожек магнитного Диска начинать не с дорожек с максимальным радиусом, как это Делается в стандартных накопителях, а нумеровать их в обратном направлении, то система позиционирования стандартного накопителя не сможет выполнять свои функции при установке на него такого диска. Направление движения

будет задаваться в направлении, обратном фактически записанным на дискете номерам дорожек, и успешное завершение позиционирования невозможно.

Выбор конкретного метода изменения алгоритма работы ВЗУ (или их композиции) осуществляется с учетом удобства практической реализации и сложности повторения алгоритма злоумышленником. При разработке ВЗУ необходимо учитывать потребность использования устройств в двух режимах: в стандартном режиме и в режиме совместимости на уровне КС. Выбор одного из режимов, а также выбор конкретного алгоритма нестандартного использования должен осуществляться, например, записью в ПЗУ двоичного кода. Число нестандартных режимов должно быть таким, чтобы исключался подбор режима методом перебора. Процесс смены режима должен исключать возможность автоматизированного подбора кода. Установку кода на ВЗУ всего объекта должен производить администратор системы безопасности.

6.2 Методы, препятствующие использованию скопированной информации

Эта группа методов имеет целью затруднить использование полученных копированием данных. Скопированная информация может быть программой или данными. Данные и программы могут быть защищены, если они хранятся на ВЗУ в преобразованном криптографическими методами виде. Программы, кроме того, могут защищаться от несанкционированного исполнения и тиражирования, а также от исследования.

Наиболее действенным (после криптографического преобразования) методом противодействия несанкционированному выполнению скопированных программ является использование блока контроля среды размещения программы. Блок контроля среды размещения является дополнительной частью программ. Он создается при инсталляции (установке) программ. В него включаются характеристики среды, в которой размещается программа, а также средства получения и сравнения характеристик.

В качестве характеристик используются характеристики ЭВМ или носителя информации, или совместно, характеристики ЭВМ и носителя. С

помощью характеристик программа связывается с конкретной ЭВМ и (или) носителем информации. Программа может выполняться только на тех ЭВМ или запускаться только с тех носителей информации, характеристики которых совпадут с характеристиками, записанными в блоке контроля среды выполнения.

В качестве характеристик ЭВМ используются особенности архитектуры: тип и частота центрального процессора, номер процессора (если он есть), состав и характеристики внешних устройств, особенности их подключения, режимы работы блоков и устройств и т. п.

Сложнее осуществляется привязка программ к носителям информации, так как они стандартны и не имеют индивидуальных признаков. Поэтому такие индивидуальные признаки создаются искусственно путем нанесения физических повреждений или изменением системной информации и структуры физических записей на носителе. Например, на гибких магнитных дисках могут прожигаться лазером отверстия, используется нестандартное форматирование, пометка некоторых секторов как дефектных. Приведенные средства защиты от несанкционированного использования дискет эффективны против стандартных способов создания копий (COPY, XCOPY, Diskcopy, Pctools, Norton Utilities в MS-DOS и др.).

Однако существуют программные средства (COPYWRITE, DISK EXPLORER), позволяющие создавать полностью идентичные копии дискет с воспроизведением всех уникальных характеристик. Все же приведенный метод защиты нельзя считать абсолютно неэффективным, так как трудоемкость преодоления защиты велика и требования, предъявляемые к квалификации взломщика, высоки.

Общий алгоритм механизма защиты от несанкционированного использования программ в «чужой» среде размещения сводится к выполнению следующих шагов.

Шаг 1. Запоминание множества индивидуальных контрольных характеристик ЭВМ и (или) съемного носителя информации на этапе инсталляции защищаемой программы.

Шаг 2. При запуске защищенной программы управление передается на блок контроля среды размещения. Блок осуществляет сбор и сравнение характеристик среды размещения с контрольными характеристиками.

Шаг 3. Если сравнение прошло успешно, то программа выполняется, иначе - отказ в выполнении. Отказ в выполнении может быть дополнен выполнением деструктивных действий в отношении этой программы, приводящих к невозможности выполнения этой программы, если такую самоликвидацию позволяет выполнить ОС.

Привязка программ к среде размещения требует повторной их инсталляции после проведения модернизации, изменения структуры или ремонта КС с заменой устройств.

Для защиты от несанкционированного использования программ могут применяться и электронные ключи. Электронный ключ «HASP» имеет размеры со спичечный коробок и подключается к параллельному порту принтера. Принтер подключается к компьютеру через электронный ключ. На работу принтера ключ не оказывает никакого влияния. Ключ распространяется с защищаемой программой. Программа в начале и в ходе выполнения считывает контрольную информацию из ключа. При отсутствии ключа выполнение программы блокируется.

6.3 Основные функции средств защиты от копирования

При защите программ от несанкционированного копирования применяются методы, которые позволяют привносить в защищаемую программу функции привязки процесса выполнения кода программы только на тех ЭВМ, на которые они были инсталлированы. Инсталлированная программа для защиты от копирования при каждом запуске должна выполнять следующие действия:

- анализ аппаратно-программной среды компьютера, на котором она запущена, формирование на основе этого анализа текущих характеристик своей среды выполнения;
- проверка подлинности среды выполнения путем сравнения ее текущих характеристик с эталонными, хранящимися на винчестере;
- блокирование дальнейшей работы программы при несовпадении текущих характеристик с эталонными.

Этап проверки подлинности среды является одним из самых уязвимых с точки зрения защиты. Можно детально не разбираться с логикой защиты, а немного "подправить" результат сравнения, и защита будет снята.

При выполнении процесса проверки подлинности среды возможны три варианта: с использованием множества операторов сравнения того, что есть, с тем, что должно быть, с использованием механизма генерации исполняемых команд в зависимости от результатов работы защитного механизма и с использованием арифметических операций. При использовании механизма генерации исполняемых команд в первом байте хранится исходная ключевая контрольная сумма BIOS, во второй байт записывается подсчитанная контрольная сумма в процессе выполнения задачи. Затем осуществляется вычитание из значения первого байта значение второго байта, а полученный результат добавляется к каждой ячейки оперативной памяти в области операционной системы. Понятно, что если суммы не совпадут, то операционная система функционировать не будет. При использовании арифметических операций осуществляется преобразование над данными арифметического характера в зависимости от результатов работы защитного механизма.

Для снятия защиты от копирования применяют два основных метода: статический и динамический.

Статические методы предусматривают анализ текстов защищенных программ в естественном или преобразованном виде. Динамические методы

предусматривают слежение за выполнением программы с помощью специальных средств снятия защиты от копирования.

6.4 Основные методы защиты от копирования

Криптографические методы

Для защиты инсталлируемой программы от копирования при помощи криптографических методов инсталлятор программы должен выполнить следующие функции:

- анализ аппаратно-программной среды компьютера, на котором должна будет выполняться инсталлируемая программа, и формирование на основе этого анализа эталонных характеристик среды выполнения программы;
- запись криптографически преобразованных эталонных характеристик аппаратно-программной среды компьютер на винчестер.

Преобразованные эталонные характеристики аппаратно-программной среды могут быть занесены в следующие области жесткого диска:

- в любые места области данных (в созданный для этого отдельный файл, в отдельные кластеры, которые должны помечаться затем в FAT как зарезервированные под операционную систему или дефектные);
- в зарезервированные сектора системной области винчестера;
- непосредственно в файлы размещения защищаемой программной системы, например, в файл настройки ее параметров функционирования.

Можно выделить два основных метода защиты от копирования с использованием криптографических приемов:

- с использованием односторонней функции;
- с использованием шифрования.

Односторонние функции это функции, для которых при любом x из области определения легко вычислить f(x), однако почти для всех y из ее области значений, найти y=f(x) вычислительно трудно.

Если эталонные характеристики программно-аппаратной среды представить в виде аргумента односторонней функции x, то y - есть "образ" этих характеристик, который хранится на винчестере и по значению которого вычислительно невозможно получить сами характеристики. Примером такой односторонней функции может служить функция дискретного возведения в степень, описанная в разделах 2.1 и 3.3 с размерностью операндов не менее 512 битов.

При шифровании эталонные характеристики шифруются по ключу, совпадающему с этими текущими характеристиками, а текущие характеристики среды выполнения программы для сравнения с эталонными также зашифровываются, но по ключу, совпадающему с этими текущими характеристиками. Таким образом, при сравнении эталонные и текущие характеристики находятся в зашифрованном виде и будут совпадать только в том случае, если исходные эталонные характеристики совпадают с исходными текущими.

Метод привязки к идентификатору

В случае если характеристики аппаратно-программной среды отсутствуют в явном виде или их определение значительно замедляет запуск программ или снижает удобство их использования, то для защиты программ от несанкционированного копирования можно использовать методов привязки к идентификатору, формируемому инсталлятором. Суть данного метода заключается в том, что на винчестере при инсталляции защищаемой от копирования программы формируется уникальный идентификатор, наличие которого затем проверяется инсталлированной программой при каждом ее запуске. При отсутствии или несовпадении этого идентификатора программа блокирует свое дальнейшее выполнение.

Основным требованием к записанному на винчестер уникальному идентификатору является требование, согласно которому данный идентификатор не должен копироваться стандартным способом. Для этого идентификатор целесообразно записывать в следующие области жесткого диска: в отдельные кластеры области данных, которые должны помечаться затем в FAT как зарезервированные под операционную систему или как дефектные; в зарезервированные сектора системной области винчестера.

Некопируемый стандартным образом идентификатор может помещаться на дискету, к которой должна будет обращаться при каждом своем запуске программа. Такую дискету называют ключевой. Кроме того, защищаемая от копирования программа может быть привязана и к уникальным характеристикам ключевой дискеты. Следует учитывать, что при использовании ключевой дискеты значительно увеличивается неудобство пользователя, так как он всегда должен вставлять в дисковод эту дискету перед запуском защищаемой от копирования программы.

Методы, основанные на работа с переходами и стеком

Данные методы основаны на включение в тело программы переходов по динамически изменяемым адресам и прерываниям, а также самогенерирующихся команд (например, команд, полученных с помощью сложения и вычитания). Кроме того, вместо команды безусловного перехода (JMP) может использоваться возврат из подпрограммы (RET). Предварительно в стек записывается адрес перехода, который в процессе работы программы модифицируется непосредственно в стеке.

При работе со стеком, стек определяется непосредственно в области исполняемых команд, что приводит к затиранию при работе со стеком. Этот способ применяется, когда не требуется повторное исполнение кода программы. Таким же способом можно генерировать исполняемые команды до начала вычислительного процесса.

Манипуляции с кодом программы

При манипуляциях с кодом программы можно привести два следующих способа:

- включение в тело программы "пустых" модулей;
- изменение защищаемой программы.

Первый способ заключается во включении в тело программы модулей, на которые имитируется передача управления, но реально никогда не осуществляется. Эти модули содержат большое количество команд, не имеющих никакого отношения к логике работы программы. Но "ненужность" этих программ не должна быть очевидна потенциальному злоумышленнику.

Второй способ заключается в изменении начала защищаемой программы таким образом, чтобы стандартный дизассемблер не смог ее правильно дизассемблировать. Например, такие программы, как Nota и Copylock, внедряя защитный механизм в защищаемый файл, полностью модифицируют исходный заголовок EXE-файла.

Все перечисленные методы были, в основном направлены на противодействия статическим способам снятия защиты от копирования. В следующем подразделе рассмотрим методы противодействия динамическим способам снятия защиты.

6.5 Методы противодействия динамическим способам снятия защиты программ от копирования

Набор методов противодействия динамическим способам снятия защиты программ от копирования включает следующие методы.

Периодический подсчет контрольной суммы, занимаемой образом задачи области оперативной памяти, в процессе выполнения. Это позволяет:

- заметить изменения, внесенные в загрузочный модуль;
- в случае, если программу пытаются "раздеть", выявить контрольные точки, установленные отладчиком.

Проверка количества свободной памяти и сравнение и с тем объемом, к которому задача "привыкла" или "приучена". Это действия позволит застраховаться от слишком грубой слежки за программой с помощью резидентных модулей.

Проверка содержимого незадействованных для решения защищаемой программы областей памяти, которые не попадают под общее распределение оперативной памяти, доступной для программиста, что позволяет добиться "монопольного" режима работы программы.

Проверка содержимого векторов прерываний (особенно 13h и 21h) на наличие тех значений, к которым задача "приучена". Иногда бывает полезным сравнение первых команд операционной системы, отрабатывающих этим прерывания, с теми командами, которые там должны быть. Вместе с предварительной очисткой оперативной памяти проверка векторов прерываний и их принудительное восстановление позволяет избавиться от большинства присутствующих в памяти резидентных программ.

Переустановка векторов прерываний. Содержимое некоторых векторов прерываний (например, 13h и 21h) копируется в область свободных векторов. Соответственно изменяются и обращения к прерываниям. При этом слежение за известными векторами не даст желаемого результата. Например, самыми первыми исполняемыми командами программы копируется содержимое вектора 21h (4 байта) в вектор 60h, а вместо команд int 21h в программе везде записывается команда int 60h. В результате в явном виде в тексте программы нет ни одной команды работы с прерыванием 21h.

Постоянное чередование команд разрешения и запрещения прерывания, что затрудняет установку отладчиком контрольных точек.

Контроль времени выполнения отдельных частей программы, что позволяет выявить "остановы" в теле исполняемого модуля.

Многие перечисленные защитные средства могут быть реализованы исключительно на языке Ассемблер. Одна из основных отличительных особенностей этого языка заключается в том, что для него не существует

ограничений в области работы со стеком, регистрами, памятью, портами ввода/вывода и т.п.

Автокорреляция представляет значительный интерес, поскольку дает некоторую числовую характеристику программы. По всей вероятности автокорреляционные функции различного типа можно использовать и тестировании программ на технологическую безопасность, когда разработанную программу еще не с чем сравнивать на подобие с целью обнаружения программных дефектов. Таким образом, программы имеют целую иерархию структур, которые могут быть выявлены, измерены и использованы в качестве характеристик последовательности данных. При этом в ходе тестирования, измерения не должны зависеть от типа данных, хотя данные, имеющие структуру программы, должны обладать специфическими параметрами, позволяющими указать меру распознавания программы. Поэтому указанные методы позволяют в определенной мере выявить те изменения в программе, которые вносятся нарушителем либо в результате преднамеренной маскировки, либо преобразованием некоторых функций программы, либо включением модуля, характеристики которого отличаются от характеристик программы, а также позволяют оценить степень обеспечения безопасности программ при внесении программных закладок.

6.6 Контрольные вопросы

1. Перечислите основные требования, предъявляемые к системе защиты от копирования.
2. Назовите методы, затрудняющие считывание скопированной информации.
3. Отобразите схематично общий алгоритм механизма защиты от несанкционированного использования программ в «чужой» среде размещения.
4. Приведите примеры статических и динамических методов для снятия защиты от копирования.

5. Сделайте сравнительный анализ основных методов защиты от копирования.
6. Почему многие перечисленные в этой главе защитные средства могут быть реализованы исключительно на языке Ассемблер?

7. УПРАВЛЕНИЕ КРИПТОГРАФИЧЕСКИМИ КЛЮЧАМИ

Любая криптографическая система основана на использовании криптографических ключей. В симметричной криптосистеме отправитель и получатель сообщения используют один и тот же секретный ключ. Этот ключ должен быть неизвестен всем остальным и должен периодически обновляться одновременно у отправителя и получателя. Процесс распределения (рассылки) секретных ключей между участниками информационного обмена в симметричных криптосистемах имеет весьма сложный характер.

Асимметричная криптосистема предполагает использование двух ключей открытого и личного (секретного). Открытый ключ можно разглашать, а личный надо хранить в тайне. При обмене сообщениями необходимо пересылать только открытый ключ. Важным требованием является обеспечение подлинности отправителя сообщения. Это достигается путем взаимной аутентификации участников информационного обмена.

Под *ключевой информацией* понимают совокупность всех действующих в АСОИ ключей. Если не обеспечено достаточно надежное управление ключевой информацией, то, завладев ею, злоумышленник получает неограниченный доступ ко всей информации.

Управление ключами - информационный процесс, включающий реализацию следующих основных функций:

- генерация ключей;
- хранение ключей;
- распределение ключей.

7.1 Генерация ключей

Безопасность любого криптографического алгоритма определяется используемым криптографическим ключом. Добротные криптографические ключи должны иметь достаточную длину и случайные значения битов. Для

получения ключей используются аппаратные и программные средства генерации случайных значений ключей. Как правило, применяют датчики псевдослучайных чисел (ПСЧ). Однако степень случайности генерации чисел должна быть достаточно высокой. Идеальными генераторами являются устройства на основе "натуральных" случайных процессов, например на основе *белого радиошума.*

В АСОИ со средними требованиями защищенности вполне приемлемы программные генераторы ключей, которые вычисляют ПСЧ как сложную функцию от текущего времени и (или) числа, введенного пользователем.

Один из методов генерации сеансового ключа для симметричных криптосистем описан в стандарте ANSI X 9.17. Он предполагает использование криптографического алгоритма DES (хотя можно применить и другие симметричные алгоритмы шифрования).

Обозначения:

$E_к$ (X) - результат шифрования алгоритмом DES значения X;

К- ключ, зарезервированный для генерации секретных ключей;

V_0-секретное 64-битовое начальное число;

Т- временная отметка.

Схема генерации случайного сеансового ключа Rj в соответствии со стандартом ANSI X 9.17 показана на рис.7.1. Случайный ключ Ri генерируют, вычисляя значение

$R_i=E_K(E_K(T_i)eV_i)$.

Следующее значение V_{l+1} вычисляют так:

$V_{i+1}=E_K(E_K(T_i)eR_i)$.

Если необходим 128-битовый случайный ключ, генерируют пару ключей Rj, R_{i+1} и объединяют их вместе. Если ключ не меняется регулярно, это может привести к его раскрытию и утечке информации. Регулярную замену ключа можно осуществить, используя процедуру модификации ключа.

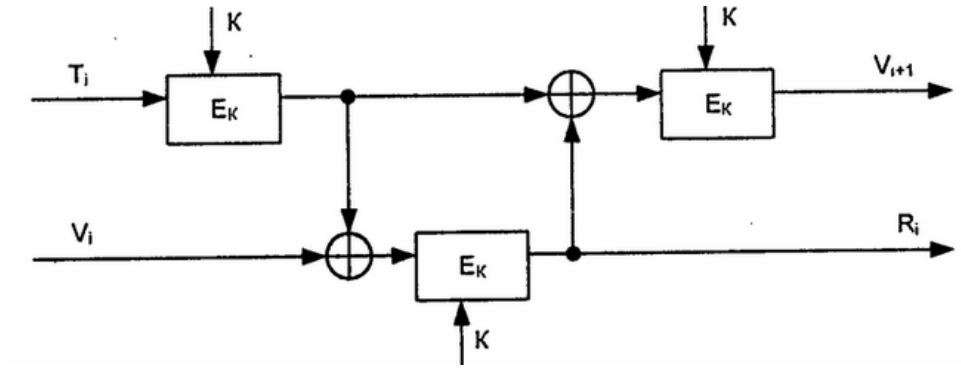


Рисунок 7.1 – Схема генерации случайного ключа R_i в соответствии со стандартом ANSI X 9.17

Модификация ключа - это генерирование нового ключа из предыдущего значения ключа с помощью односторонней (однонаправленной) функции. Участники информационного обмена разделяют один и тот же ключ и одновременно вводят его значение в качестве аргумента в одностороннюю функцию, получая один и тот же результат. Затем они берут определенные биты из этих результатов, чтобы создать новое значение ключа.

Процедура модификации ключа работоспособна, но надо помнить, что новый ключ безопасен в той же мере, в какой был безопасен прежний ключ. Если злоумышленник сможет добыть прежний ключ, то он сможет выполнить процедуру модификации ключа.

Генерация ключей для асимметричных криптосистем с открытыми ключами много сложнее, потому что эти ключи должны обладать определенными математическими свойствами (они должны быть очень большими и простыми и т.д.).

7.2 Хранение ключей

Под *функцией хранения ключей* понимают организацию их безопасного хранения, учета и удаления. Ключ является самым привлекательным для злоумышленника объектом, открывающим ему путь к конфиденциальной информации. Поэтому вопросам безопасного хранения ключей следует уделять

особое внимание. Секретные ключи никогда не должны записываться в явном виде на носителе, который может быть считан или скопирован.

Носители ключевой информации

Ключевой носитель может быть технически реализован различным образом на разных носителях информации-магнитных дисках, устройствах хранения ключей типа Touch Memory, пластиковых картах и т.д.

Магнитные диски представляют собой распространенный тип носителя ключевой информации. Применение магнитного диска (МД) в качестве носителя ключа позволяет реализовать необходимое свойство отчуждаемости носителя ключа от защищенной компьютерной системы, т.е. осуществить временное изъятие МД из состава технических средств компьютерной системы. Особенно целесообразно использование в качестве ключевых носителей съемных накопителей-гибких магнитных дисков, съемных магнитооптических носителей и т.д.

Основное преимущество МД по сравнению с другими носителями ключевой информации заключается в том, что оборудование для взаимодействия с МД (дисковод) входит в состав штатных средств компьютера. Другая важная особенность, определяющая широкое распространение МД - стандартный формат хранения информации на дисках и стандартные программные средства доступа к дискам. Кроме того, из всех средств хранения ключевой информации гибкие магнитные диски имеют самую низкую стоимость.

Для обеспечения надежного хранения ключевой информации на МД применяют как минимум двукратное резервирование объектов хранения. Это позволяет защитить ключевую информацию от ошибок при считывании с МД и от сбоев программной и аппаратной части.

Для предотвращения возможности перехвата ключевой информации в процессе ее чтения с МД используют хранение ключевой информации на МД в зашифрованном виде.

Устройство хранения ключей типа Touch Memory является относительно новым носителем ключевой информации, предложенным американской компанией Dallas Semiconductor. Носитель информации Touch Memory (TM) представляет собой энергонезависимую память, размещенную в металлическом корпусе, с одним сигнальным контактом и одним контактом земли. Корпус TM имеет 'диаметр 16,25 мм и толщину 3,1 или 5,89 мм (в зависимости от модификации прибора).

В структуру TM входят следующие основные блоки:

- Постоянное запоминающее устройство (ПЗУ) хранит 64-разрядный код, состоящий из байтового кода типа прибора, 48-битового уникального серийного номера и 8-битовой контрольной суммы. Содержимое ПЗУ уникально и не может быть изменено в течение всего срока службы прибора.
- Оперативное запоминающее устройство (ОЗУ) емкостью от 128 до 8192 байт содержат практически все модификации TM. В одной из модификаций оперативная память аппаратно защищена от несанкционированного доступа.
- Встроенная миниатюрная литиевая батарейка со сроком службы не менее 10 лет обеспечивает питанием все блоки устройства.

Особенностью технологии хранения и обмена ключевой информации между носителем TM и внешними устройствами является сравнительно низкая скорость (обусловленная последовательной передачей данных) и высокая вероятность сбоя в тракте чтения-записи, обусловленная тем, что контакт устройства TM с устройством чтения осуществляется пользователем вручную без дополнительной фиксации (простое касание, что и определило название прибора TM). В связи с этим особое значение приобретают вопросы надежного обмена между программами обработки ключевой информации пользователей и носителем TM.

В устройстве ТМ конструктивно отработаны вопросы надежности функционирования и вопросы интерфейса со считывающим устройством на основе одного сигнального контакта. Для обеспечения достоверного чтения применяются корректирующие коды, для обеспечения достоверной записи в приборе предусмотрена технология буферизации. При проведении операции записи первоначально вектор передаваемой в ТМ информации помещается в буфер, далее выполняется операция чтения из буфера, затем прочтенная из буфера информация сравнивается с записываемой и в случае совпадения подается сигнал переноса информации из буфера в память долговременного хранения.

Таким образом, носитель ТМ является микроконтроллерным устройством без собственной вычислительной мощности и с ограниченным объемом хранения, но с достаточно высокими надежностными характеристиками. Поэтому применение ТМ вполне обосновано в случае повышенных требований к надежности носителя ключа и небольшого объема ключевой информации, хранимой в ТМ.

Электронные пластиковые карты становятся в настоящее время наиболее распространенным и универсальным носителем конфиденциальной информации, который позволяет идентифицировать и аутентифицировать пользователей, хранить криптографические ключи, пароли и коды.

Интеллектуальные карты (смарт-карты), обладающие наибольшими возможностями, не только эффективно применяются для хранения ключевой информации, но и широко используются в электронных платежных системах, в комплексных решениях для медицины, транспорта, связи, образования и т.п.

Концепция иерархии ключей

Любая информация об используемых ключах должна быть защищена, в частности храниться в зашифрованном виде.

Необходимость в хранении и передаче ключей, зашифрованных с помощью других ключей, приводит к концепции *иерархии ключей.* В стандарте

ISO 8532 (Banking-Key Management) подробно изложен метод главных/сеансовых ключей (master/session keys). Суть метода состоит в том, что вводится иерархия ключей: главный ключ (ГК), ключ шифрования ключей (КК), ключ шифрования данных (КД).

Иерархия ключей может быть:

• двухуровневой (КК/КД);

• трехуровневой (ГК/КК/КД);

Самым нижним уровнем являются *рабочие или сеансовые КД,* которые применяются для шифрования данных, персональных идентификационных номеров (PIN) и аутентификации сообщений.

Когда эти ключи надо зашифровать с целью защиты при передаче или хранении, используют ключи следующего уровня - *ключи шифрования ключей.* Ключи шифрования ключей никогда не должны использоваться как сеансовые (рабочие) КД, и наоборот.

Такое разделение функций необходимо для обеспечения максимальной безопасности. Фактически стандарт устанавливает, что различные типы рабочих ключей (например, для шифрования данных, для аутентификации и т.д.) должны всегда шифроваться с помощью различных версий ключей шифрования ключей. В частности, ключи шифрования ключей, используемые для пересылки ключей между двумя узлами сети, известны также как *ключи обмена между узлами сети* (cross domain keys). Обычно в канале используются два ключа для обмена между узлами сети, по одному в каждом направлении. Поэтому каждый узел сети будет иметь *ключ отправления* для обмена с узлами сети и *ключ получения* для каждого канала, поддерживаемого другим узлом сети.

На верхнем уровне иерархии ключей располагается *главный ключ, мастер-ключ.* Этот ключ применяют для шифрования КК, когда требуется сохранить их на диске. Обычно в каждом компьютере используется только один мастер-ключ.

Мастер-ключ распространяется между участниками обмена неэлектронным способом - при личном контакте, чтобы исключить его перехват и/или компрометацию. Раскрытие противником значения мастер - ключа полностью уничтожает защиту компьютера.

Значение мастер - ключа фиксируется на длительное время (до нескольких недель или месяцев). Поэтому генерация и хранение мастер - ключей являются критическими вопросами криптографической защиты. На практике мастер-ключ компьютера создается истинно случайным выбором из всех возможных значений ключей. Мастер-ключ помещают в защищенный от считывания и записи и от механических воздействий блок криптографической системы таким образом, чтобы раскрыть значение этого ключа было невозможно. Однако все же должен существовать способ проверки, является ли значение ключа правильным.

Проблема аутентификации мастер - ключа может быть решена различными путями. Один из способов аутентификации показан на рис.7.2

Администратор, получив новое значение мастер - ключа $К_н$ хост - компьютера, шифрует некоторое сообщение М ключом $К_н$ Пара (криптограмма $Е_{Кн}(М)$, сообщение М) помещается в память компьютера. Всякий раз, когда требуется аутентификация мастер - ключа хост - компьютера, берется сообщение М из памяти и подается в криптографическую систему. Получаемая криптограмма сравнивается с криптограммой, хранящейся в памяти. Если они совпадают, считается, что данный ключ является правильным.

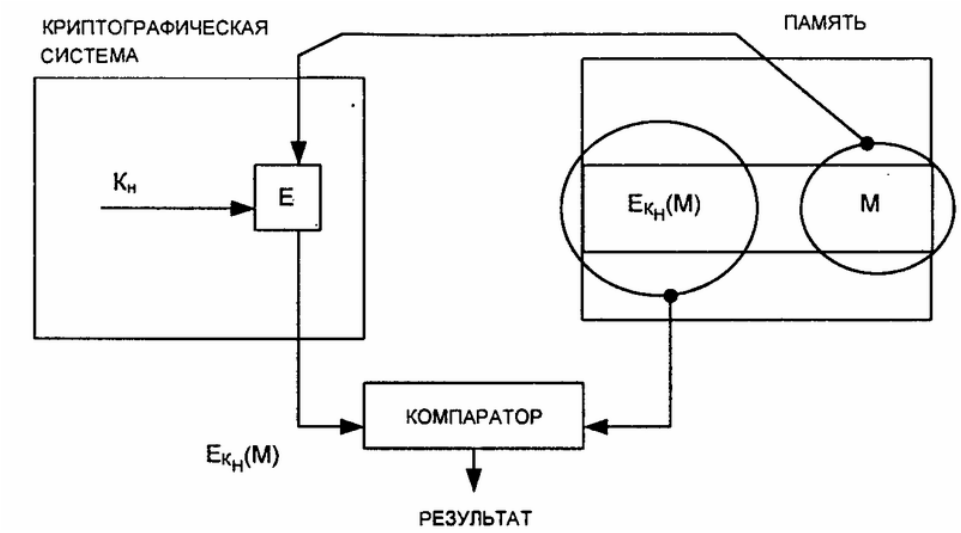


Рисунок 7.2 – Схема аутентификации мастер-ключа хост-компьютера

Рабочие ключи (например, сеансовый) обычно создаются с помощью псевдослучайного генератора и могут храниться в незащищенном месте. Это возможно, поскольку такие ключи генерируются в форме соответствующих криптограмм, т.е. генератор ПСЧ выдает вместо ключа K_s его криптограмму $E_{KH}(K_S)$, получаемую с помощью мастер - ключа хост - компьютера. Расшифровывание такой криптограммы выполняется только перед использованием ключа K_s.

Схема защиты рабочего (сеансового) ключа показана на рис.7.3. Чтобы зашифровать сообщение M ключом K_s, на соответствующие входы криптографической системы подается криптограмма $E_{K_H}(K_S)$ и сообщение M. Криптографическая система сначала восстанавливает ключ K_S а затем шифрует сообщение M, используя открытую форму сеансового ключа K_s.

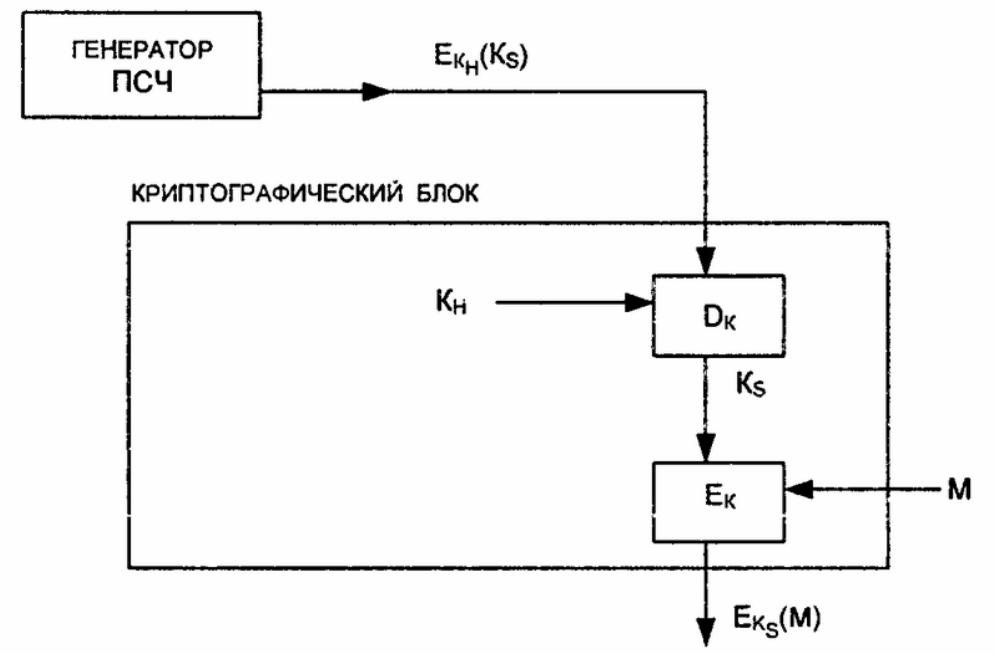


Рисунок 7.3 – Схема защиты ключа K_S

Таким образом, безопасность сеансовых ключей зависит от безопасности криптографической системы. Криптографический блок может быть спроектирован как единая СБИС и помещен в физически защищенное место.

Очень важным условием безопасности информации является периодическое обновление ключевой информации в АСОИ. При этом должны переназначаться как рабочие ключи, так и мастер - ключи. В особо ответственных АСОИ обновление ключевой информации (сеансовых ключей) желательно делать ежедневно. Вопрос обновления ключевой информации тесно связан с третьим элементом управления ключами - распределением ключей.

7.3 Распределение ключей

Распределение ключей - самый ответственный процесс в управлении ключами. К нему предъявляются следующие требования:

- оперативность и точность распределения;
- скрытность распределяемых ключей.

Распределение ключей между пользователями компьютерной сети реализуется двумя способами:

1) использованием одного или нескольких центров распределения ключей;

2) прямым обменом сеансовыми ключами между пользователями сети.

Недостаток первого подхода состоит в том, что центру распределения ключей известно, кому и какие ключи распределены, и это позволяет читать все сообщения, передаваемые по сети. Возможные злоупотребления существенно влияют на защиту. При втором подходе проблема состоит в том, чтобы надежно удостоверить подлинность субъектов сети.

В обоих случаях должна быть обеспечена подлинность сеанса связи. Это можно осуществить, используя механизм запроса-ответа или механизм отметки времени.

Механизм запроса-ответа заключается в следующем. Пользователь А включает в посылаемое сообщение (запрос) для пользователя В непредсказуемый элемент (например, случайное число). При ответе пользователь В должен выполнить некоторую операцию с этим элементом (например, добавить единицу), что невозможно осуществить заранее, поскольку неизвестно, какое случайное число придет в запросе. После получения результата действий пользователя В (ответ) пользователь А может быть уверен, что сеанс является подлинным.

Механизм отметки времени предполагает фиксацию времени для каждого сообщения. Это позволяет каждому субъекту сети определить, насколько старо пришедшее сообщение, и отвергнуть его, если появится сомнение в его подлинности. При использовании отметок времени необходимо установить допустимый временной интервал задержки.

В обоих случаях для защиты элемента контроля используют шифрование, чтобы быть уверенным, что ответ отправлен не злоумышленником и не изменен штемпель отметки времени.

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

- взаимное подтверждение подлинности участников сеанса;
- подтверждение достоверности сеанса механизмом запроса-ответа или отметки времени;
- использование минимального числа сообщений при обмене ключами;
- возможность исключения злоупотреблений со стороны центра распределения ключей (вплоть до отказа от него).

В основу решения задачи распределения ключей целесообразно положить принцип отделения процедуры подтверждения подлинности партнеров от процедуры собственно распределения ключей. Цель такого подхода состоит в создании метода, при котором после установления подлинности участники сами формируют сеансовый ключ без участия центра распределения ключей с тем, чтобы распределитель ключей не имел возможности выявить содержание сообщений.

Распределение ключей с участием центра распределения ключей

При распределении ключей между участниками предстоящего информационного обмена должна быть гарантирована подлинность сеанса связи. Для взаимной проверки подлинности партнеров приемлема *модель рукопожатия:* В этом случае ни один из участников не будет получать никакой секретной информации во время процедуры установления подлинности.

Взаимное установление подлинности гарантирует вызов нужного субъекта с высокой степенью уверенности, что связь установлена с требуемым адресатом и никаких попыток подмены не было. Реальная процедура организации соединения между участниками информационного обмена включает как этап распределения, так и этап подтверждения подлинности партнеров.

При включении в процесс распределения ключей центра распределения ключей (ЦРК) осуществляется его взаимодействие с одним или обоими

участниками сеанса с целью распределения секретных или открытых ключей, предназначенных для использования в последующих сеансах связи.

Следующий этап-подтверждение подлинности участников - содержит обмен удостоверяющими сообщениями, чтобы иметь возможность выявить любую подмену или повтор одного из предыдущих вызовов.

Рассмотрим протоколы для симметричных криптосистем с секретными ключами и для асимметричных криптосистем с открытыми ключами. Вызывающий (исходный объект) обозначается через А, а вызываемый (объект назначения)-через В. Участники сеанса А и В имеют уникальные идентификаторы ld_A и ld_B соответственно.

7.4 Протокол аутентификации и распределения ключей для симметричных криптосистем

Рассмотрим в качестве примера протокол аутентификации и распределения ключей Kerberos (по-русски - Цербер). Первоначально протокол Kerberos был разработан в Массачусетском технологическом институте (США) для проекта Athena. Протокол Kerberos спроектирован для работы в сетях TCP/IP и предполагает участие в аутентификации и распределении ключей третьей доверенной стороны. Kerberos обеспечивает надежную аутентификацию в сети, разрешая законному пользователю доступ к различным машинам в сети. Протокол Kerberos основывается на симметричной криптографии (реализован алгоритм DES, хотя возможно применение и других симметричных криптоалгоритмов). Kerberos разделяет отдельный секретный ключ с каждым субъектом сети. Знание такого секретного ключа равносильно Доказательству подлинности субъекта сети.

Основной протокол Kerberos является, вариантом протокола аутентификации и распределения ключей Нидхема-Шредера. В основном протоколе Kerberos (версия 5) участвуют две взаимодействующие стороны А и В и доверенный сервер KS (Kerberos Server). Стороны А и В, каждая по

отдельности, разделяют свой секретный ключ с сервером KS. Доверенный сервер KS выполняет роль центра распределения ключей ЦРК.

Пусть сторона A хочет получить сеансовый ключ для информационного обмена со стороной B.

Сторона A инициирует фазу распределения ключей, посылая по сети серверу KS идентификаторы ld_A и ld_B:

A -> KS: ld_A, ld_B. (7.1)

Сервер KS генерирует сообщение с временной отметкой T, сроком действия L, случайным сеансовым ключом K и идентификатором ld_A. Он шифрует это сообщение секретным ключом, который разделяет со стороной B.

Затем сервер KS берет временную отметку T, срок действия L, сеансовый ключ K, идентификатор ld_B стороны B и шифрует все это секретным ключом, который разделяет со стороной A. Оба эти зашифрованные сообщения он отправляет стороне A:

KS -> A: $E_A(T, L, K, ld_B)$, $E_B(T, L, K, ld_A)$. (7.2)

Сторона A расшифровывает первое сообщение своим секретным ключом, проверяет отметку времени T, чтобы убедиться, что это сообщение не является повторением предыдущей процедуры распределения ключей.

Затем сторона A генерирует сообщение со своим идентификатором ld_A и отметкой времени T, шифрует его сеансовым ключом K и отправляет стороне B. Кроме того, A отправляет для B сообщение от KS, зашифрованное ключом стороны B:

A->B:$E_K(ld_A,T)$,$E_B(T, L, K, ld_A)$. (7.3)

Только сторона B может расшифровать сообщения (7.3). Сторона B получает отметку времени T, срок действия L, сеансовый ключ K и идентификатор ld_A. Затем сторона B расшифровывает сеансовым ключом K вторую часть сообщения (7.3). Совпадение значений T и ld_A в двух частях сообщения подтверждают подлинность A по отношению к B.

Для взаимного подтверждения подлинности сторона B создает сообщение, состоящее из отметки времени T плюс 1, шифрует его ключом K и отправляет стороне A:

$$B->A:E_к(T+1). \qquad (7.4)$$

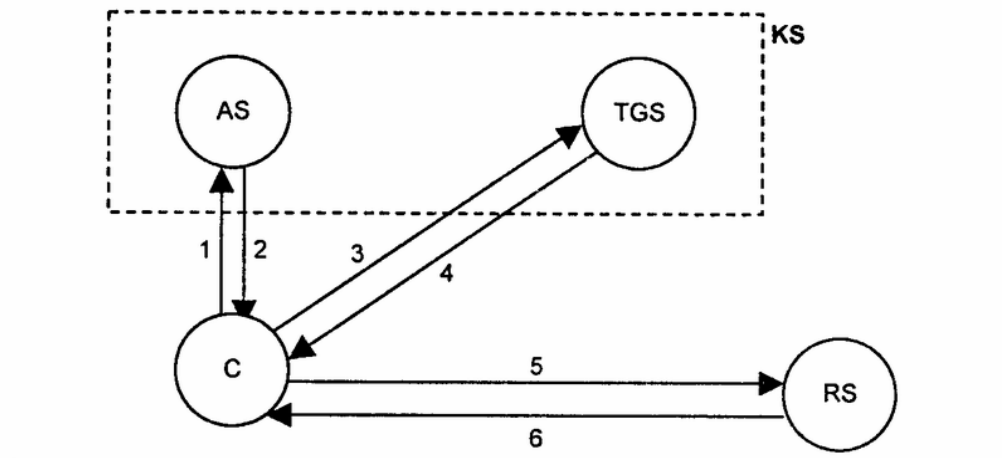
Если после расшифрования сообщения (7.4) сторона A получает ожидаемый результат, она знает, что на другом конце линии связи находится действительно B.

Этот протокол успешно работает при условии, что часы каждого участника синхронизированы с часами сервера KS. Следует отметить, что в этом протоколе необходим обмен с KS для получения сеансового ключа каждый раз, когда A желает установить связь с B. Протокол обеспечивает надежное соединение объектов A и B при условии, что ни один из ключей не скомпрометирован и сервер KS защищен.

Система Kerberos обеспечивает защиту сети от несанкционированного доступа, базируясь исключительно на программных решениях, и предполагает многократное шифрование передаваемой по сети управляющей информации.

Система Kerberos имеет структуру типа клиент-сервер и состоит из клиентских частей C, установленных на все машины сети (рабочие станции пользователей и серверы), и Kerberos-сервера KS, располагающегося на каком-либо (не обязательно выделенном) компьютере.

Kerberos-сервер, в свою очередь, можно разделить на две. части: сервер идентификации AS (Authentication Server) и сервер выдачи разрешений TGS (Ticket Granting Server). Информационными ресурсами, необходимыми клиентам C, управляет сервер информационных ресурсов RS (рис.7.4).



Обозначения:

KS	– сервер системы Kerberos
AS	– сервер идентификации
TGS	– сервер выдачи разрешений
RS	– сервер информационных ресурсов
C	– клиент системы Kerberos
1 : C → AS:	– запрос разрешить обратиться к TGS
2 : AS → C:	– разрешение обратиться к TGS
3 : C → TGS:	– запрос на допуск к RS
4 : TGS → C:	– разрешение на допуск к RS
5 : C → RS:	– запрос на получение информационного ресурса от RS
6 : RS → C:	– подтверждение подлинности сервера RS и предоставление информационного ресурса

Рисунок 7.4 – Схема и шаги протокола Kerberos

Область действия системы Kerberos распространяется на тот участок сети, все пользователи которого зарегистрированы под своими именами и паролями в базе данных Kerberos-сервера.

Укрупненно процесс идентификации и аутентификации пользователя в системе Kerberos можно описать следующим образом. Пользователь (клиент) C, желая получить доступ к ресурсу сети, направляет запрос серверу идентификации AS. Последний идентифицирует пользователя с помощью его имени и пароля и выдает разрешение на доступ к серверу выдачи разрешений TGS, который, в свою очередь, по запросу клиента C разрешает использование необходимых ресурсов сети с помощью целевого сервера информационных ресурсов RS.

Данная модель взаимодействия клиента с серверами может функционировать только при условии обеспечения конфиденциальности и целостности передаваемой управляющей информации. Без строгого

обеспечения информационной безопасности клиент не может отправлять серверам AS.TGS и RБ свои запросы и получать разрешения на доступ к обслуживанию в сети. Чтобы избе жать возможности перехвата и несанкционированного использования информации, Kerberos применяет при передаче любой управляющей информации в сети сложную систему многократного шифрования с использованием комплекса секретных ключей (секретный ключ клиента, секретный ключ сервера, секретные сеансовые ключи, клиент-сервер).

7.5 Протокол для асимметричных криптосистем с использованием сертификатов открытых ключей

В этом протоколе используется идея сертификатов открытых ключей.

Сертификатом открытого ключа С называется сообщение ЦРК, удостоверяющее целостность некоторого открытого ключа объекта. Например, сертификат открытого ключа для пользователя А, обозначаемый C_A, содержит отметку времени Т, идентификатор ld_A и открытый ключ $К_A$, зашифрованные секретным ключом ЦРК Кцрк, т. е. $C_A = E_{кцрк}(T, ld_A, К_A)$.

Отметка времени Т используется для подтверждения актуальности сертификата и тем самым предотвращает повторы прежних сертификатов, которые содержат открытые ключи и для которых соответствующие секретные ключи несостоятельны.

Секретный ключ $к_{ЦРК}$ известен только менеджеру ЦРК. Открытый ключ Кцрк известен участникам А и В. ЦРК поддерживает таблицу открытых ключей всех объектов сети, которые он обслуживает.

Вызывающий объект А инициирует стадию установления ключа, запрашивая у ЦРК сертификат своего открытого ключа и открытого ключа участника В:

А -> ЦРК: ld_A,ld_B ,"Вышлите сертификаты ключей А и В'". (7.5)

Здесь ld_A и ld_B- уникальные идентификаторы соответственно участников А и В.

Менеджер ЦРК отвечает сообщением

ЦРК-> А: $E_{кцрк}(T, ld_A. К_д>,, E_{кцрк}(T, ld_B, K_B)$. (7.6)

Участник А, используя открытый ключ ЦРК $Кц_{РК}$, расшифровывает ответ ЦРК, проверяет оба сертификата. Идентификатор ld_B убеждает А, что личность вызываемого участника правильно зафиксирована в ЦРК и $К_в$ - действительно открытый ключ участника В, поскольку оба зашифрованы ключом $к_{ЦРК}$.

Хотя открытые ключи предполагаются известными всем, посредничество ЦРК позволяет подтвердить их целостность. Без такого посредничества злоумышленник может снабдить А своим открытым ключом, который А будет считать ключом участника В. Затем злоумышленник может подменить собой В и установить связь с А, и его никто не сможет выявить. Следующий шаг протокола включает установление связи А с В:

A->B:$C_{A1}E_{кА}(T),Ек_в(г_1)$. (7.7)

Здесь C_A-сертификат открытого ключа пользователя А; $E_{кд}(T)$-отметка времени, зашифрованная секретным ключом участника А и являющаяся подписью участника А, поскольку никто другой не может создать такую подпись; г - случайное число, генерируемое А и используемое для обмена с В в ходе процедуры подлинности.

Если сертификат C_A и подпись А верны, то участник В уверен, что сообщение пришло от А. Часть сообщения $E_{кв}(г,)$ может расшифровать только В, поскольку никто другой не знает секретного ключа $к_в$, соответствующего открытому ключу $К_в$. Участник В расшифровывает значение числа г, и, чтобы подтвердить свою подлинность, посылает участнику А сообщение

B->Ека(r,). (7.8)

Участник А восстанавливает значение г, расшифровывая это сообщение с использованием своего секретного ключа $к_А$. Если это ожидаемое значение г-,, то А получает подтверждение, что вызываемый участник действительно В.

Протокол, основанный на симметричном шифровании, функционирует быстрее, чем протокол, основанный на криптосистемах с открытыми ключами. Однако способность систем с открытыми ключами генерировать цифровые

подписи, обеспечивающие различные функции защиты, компенсирует избыточность требуемых вычислений.

Прямой обмен ключами между пользователями

При использовании для информационного обмена криптосистемы с симметричным секретным ключом два пользователя, желающие обменяться криптографически защищенной информацией, должны обладать общим секретным ключом. Пользователи должны обменяться общим ключом по каналу связи безопасным образом. Если пользователи меняют ключ достаточно часто, то доставка ключа превращается в серьезную проблему.

Для решения этой проблемы можно применить два способа:

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

2)использование системы открытого распределения ключей Диффи-Хеллмана.

Второй способ основан на применении системы открытого распределения ключей. Эта система позволяет пользователям обмениваться ключами по незащищенным каналам связи. Интересно отметить, что система открытого распределения ключей базируется на тех же принципах, что и система шифрования с открытыми ключами.

Алгоритм открытого распределения ключей Диффи-Хеллмана

Алгоритм Диффи-Хеллмана был первым алгоритмом с открытыми ключами (предложен в 1976 г.). Его безопасность обусловлена трудностью вычисления дискретных логарифмов в конечном поле, в отличие от легкости дискретного возведения в степень в том же конечном поле.

Предположим, что два пользователя A и B хотят организовать защищенный коммуникационный канал.

1. Обе стороны заранее уславливаются о модуле N (N должен быть простым числом) и примитивном элементе geZ_N, (1<g<N-1), который образует все ненулевые элементы множества Z_N, т.е. $\{g.g^2.....g^{N-1}=i\}=z_N-\{0\}$.

Эти два целых числа N и g могут не храниться в секрете. Как правило, эти значения являются общими для всех пользователей системы.

2. Затем пользователи А и В независимо друг от друга выбирают собственные секретные ключи $к_A$ и $к_в$ ($к_A$ и $к_в$-случайные большие целые числа, которые хранятся пользователями А и В в секрете).

3. Далее пользователь А вычисляет открытый ключ

$y_A = g^{kA}(\text{mod } N)$, а пользователь В - открытый ключ

$Ув = g^{kB}(\text{mod } N)$.

4. Затем стороны А и В обмениваются вычисленными значениями открытых ключей y_A и $y_в$ по незащищенному каналу. (Мы считаем, что все данные, передаваемые по незащищенному каналу связи, могут быть перехвачены злоумышленником.)

5. Далее пользователи А и В вычисляют общий секретный ключ, используя следующие сравнения:

пользователь А: $К = (y_в)^{ka}= (g^{kв})^{ka} (\text{mod } N)$;

пользователь В: $К'= (y_A)^{кв} = (g^{ka})^{kв}(\text{mod } N)$.

При этом К=К', так как $(g^{kв})^{ka}= (д^{ka})^{kв} (\text{mod } N)$.

Схема реализации алгоритма Диффи-Хеллмана показана на рис. 7.5.

Ключ К может использоваться в качестве общего секретного ключа (ключа шифрования ключей) в симметричной криптосистеме.

Кроме того, обе стороны А и В могут шифровать сообщения, используя следующее преобразование шифрования (типа RSA): $C = E_K(M) = M^k(\text{mod}N)$.

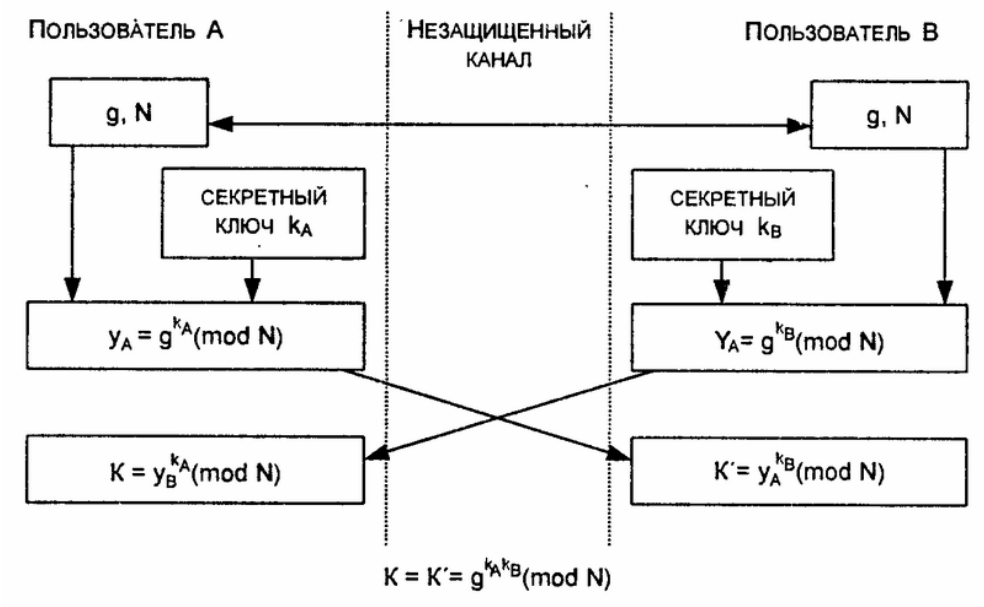


Рисунок 7.5 – Схема реализации алгоритма Диффи-Хеллмана

Для выполнения расшифрования получатель сначала находит ключ расшифрования К* с помощью сравнения

К*К*=1(modN-1),

а затем восстанавливает сообщение

$M = D_K (C) = C^{K'} (mod\ N)$.

Пример. Допустим, модуль N=47 а примитивный элемент д = 23. Предположим, что пользователи А и В выбрали свои секретные ключи: kA=12(mod47) и k_B=33(mod47).

Для того чтобы иметь общий секретный ключ К, они вычисляют сначала значения частных открытых ключей:

$y_A = g^{ka} = 23^{12} = 27 (mod47)$,

$y_B = g^{kB} = 23^{33} = 33 (mod47)$.

После того, как пользователи А и В обменяются своими значениями y_A и $y_в$, они вычисляют общий секретный ключ

$K = (y_s)^{kA} = (y_A)^{кв} = 33^{12} = 27^{м} = 23^{12"33} = 25\ (mod\ 47)$.

Кроме того, они находят секретный ключ расшифрования, используя следующее сравнение:

К*К*=1(modN-1), откуда К*=35 (mod 46).

Теперь, если сообщение M =16, то криптограмма

$C=M^K=16^{25} = 21(mod47\}$. Получатель восстанавливает сообщение так:

$M = C^{K'} = 21^{39}=16(mod47)$.

Злоумышленник, перехватив значения N, g, y_A и $y_в$, тоже хотел бы определить значение ключа К. Очевидный путь для решения этой задачи состоит в вычислении такого значения k_A no N, g, y_A, что $g^{kA}modN = y_A$ (поскольку в этом случае, вычислив $к_A$, можно найти $K = (y_B)^{kA}modN$). Однако нахождение k_A no N,g и y_A-задача нахождения дискретного логарифма в конечном поле, которая считается неразрешимой.

Выбор значений N ид может иметь существенное влияние на безопасность этой системы. Модуль N должен быть большим и простым числом. Число (N-1)/2 также должно быть простым числом. Число g желательно выбирать таким, чтобы оно было примитивным элементом множества Z_N. (В принципе достаточно, чтобы число g генерировало большую подгруппу мультипликативной группы по mod N).

Алгоритм открытого распределения ключей Диффи-Хеллмана позволяет обойтись без защищенного канала для передачи ключей. Однако, работая с этим алгоритмом, необходимо иметь гарантию того, что пользователь А получил открытый ключ именно от пользователя В, и наоборот. Эта проблема решается с помощью электронной подписи, которой подписываются сообщения об открытом ключе.

Метод Диффи-Хеллмана дает возможность шифровать данные при каждом сеансе связи на новых ключах. Это позволяет не хранить секреты на дискетах или других носителях. Не следует забывать, что любое хранение секретов повышает вероятность попадания их в руки конкурентов или противника.

Преимущество метода Диффи-Хеллмана по сравнению с методом RSA заключается в том, что формирование общего секретного ключа происходит в сотни раз быстрее. В системе RSA генерация новых секретных и открытых

ключей основана на генерации новых простых чисел, что занимает много времени.

Протокол SKIP управления криптоключами

Протокол SKIP (Simple Key management for Internet Protocol) может использоваться в качестве интегрирующей среды и системы управления криптоключами.

Протокол SKIP базируется на криптографии открытых ключей Диффи-Хеллмана и обладает рядом достоинств:

• обеспечивает высокую степень защиты информации;

• обеспечивает быструю смену ключей;

• поддерживает групповые рассылки защищенных сообщений;

• допускает модульную замену систем шифрования;

• вносит минимальную избыточность.

Концепция SKIP-протокола основана на организации множества двухточечных обменов (по алгоритму Диффи-Хеллмана) в компьютерной сети.

• Узел I имеет секретный ключ i(i=k1) и сертифицированный открытый ключ g'mod N.

• Подпись сертификата открытого ключа производится при помощи надежного алгоритма (ГОСТ, DSA и др.). Открытые ключи свободно распространяются центром распределения ключей из общей базы данных.

• Для каждой пары узлов I, J вычисляется совместно используемый секрет (типичная длина 1024 бита): g^{ij} mod N.

• Разделяемый ключ $К_{ij}$ вычисляется из этого секрета путем уменьшения его до согласованной в рамках протокола длины 64...128 бит.

• Узел вычисляет ключ $К_{ij}$ (используемый как ключ шифрования ключей) для относительно длительного применения и размещает его в защищенной памяти.

Следует отметить, что если сеть содержит п узлов, то в каждом узле должно храниться (п-1) ключей, используемых исключительно для организации связи с соответствующими узлами.

7.6 Контрольные вопросы

1. Чем отличаются симметричные криптосистемы от асимметричных?
2. Какие основные функции включает управление ключами?
3. Генерация ключей для асимметричных криптосистем с открытыми ключами много сложнее, потому что эти ключи должны обладать определенными математическими свойствами, какими?
4. Перечислите носители ключевой информации.
5. Понятие концепции иерархии ключей.
6. Для чего нужен мастер-ключ?
7. Почему процесс распределения ключей самый ответственный в управлении ключами, какие требования к нему предъявляются?
8. Какие протоколы аутентификации и распределения ключей для симметричных криптосистем вы можете назвать?
9. Опишите суть алгоритма Диффи-Хеллмана, чем обусловлена его безопасность?

8. ЗАЩИТА ПРОГРАММНЫХ СРЕДСТВ ОТ ИССЛЕДОВАНИЯ

Изучение логики работы программы может выполняться в одном из двух режимов: статическом и динамическом. Сущность статического режима заключается в изучении исходного текста программы. Для получения листингов исходного текста выполняемый программный модуль дизассемблируют, то есть получают из программы на машинном языке программу на языке Ассемблер.

Динамический режим изучения алгоритма программы предполагает выполнение трассировки программы. Под трассировкой программы понимается выполнение программы на ЭВМ с использованием специальных средств, позволяющих выполнять программу в пошаговом режиме, получать доступ к регистрам, областям памяти, производить остановку программы по определенным адресам и т. д. В динамическом режиме изучение алгоритма работы программы осуществляется либо в процессе трассировки, либо по данным трассировки, которые записаны в запоминающем устройстве.

Средства противодействия дизассемблированию не могут защитить программу от трассировки и наоборот: программы, защищенные только от трассировки, могут быть дизассемблированы. Поэтому для защиты программ от изучения необходимо иметь средства противодействия как дизассемблированию, так и трассировке.

Существует несколько методов противодействия дизассемблированию:

- шифрование;
- архивация:
- использование самогенерирующих кодов;
- «обман» дизассемблера.

Архивацию можно рассматривать как простейшее *шифрование*. Причем архивация может быть объединена с шифрованием. Комбинация таких методов

позволяет получать надежно закрытые компактные программы. Зашифрованную программу невозможно дизассемблировать без расшифрования. Зашифрование (расшифрование) программ может осуществляться аппаратными средствами или отдельными программами. Такое шифрование используется перед передачей программы по каналам связи или при хранении ее на ВЗУ. Дизассемблирование программ в этом случае возможно только при получении доступа к расшифрованной программе, находящейся в ОП перед ее выполнением (если считается, что преодолеть криптографическую защиту невозможно).

Другой подход к защите от дизассемблирования связан с совмещением процесса расшифрования с процессом выполнения программ. Если расшифрование всей программы осуществляется блоком, получающим управление первым, то такую программу расшифровать довольно просто. Гораздо сложнее расшифровать и дизассемблировать программу, которая поэтапно расшифровывает информацию, и этапы разнесены по ходу выполнения программы. Задача становится еще более сложной, если процесс расшифрования разнесен по тексту программы.

Сущность метода, основанного на использовании *самогенерируемых* кодов, заключается в том, что исполняемые коды про граммы получаются самой программой в процессе ее выполнения. Самогенерируемые коды получаются в результате определенных действий над специально выбранным массивом данных. В качестве исходных данных могут использоваться исполняемые коды самой программы или специально подготовленный массив данных. Данный метод показал свою высокую эффективность, но он сложен в реализации.

Под *«обманом»* дизассемблера понимают такой стиль программирования, который вызывает нарушение правильной работы стандартного дизассемблера за счет нестандартных приемов использования отдельных команд, нарушения общепринятых соглашений. «Обман» дизассемблера осуществляется несколькими способами:

- нестандартная структура программы;
- скрытые переходы, вызовы процедур, возвраты из них и из прерываний;
- переходы и вызовы подпрограмм по динамически изменяемым адресам;
- модификация исполняемых кодов.

Для дезориентации дизассемблера часто используются скрытые переходы, вызовы и возвраты за счет применения нестандартных возможностей команд.

Маскировка скрытых действий часто осуществляется с применением стеков.

Трассировка программ обычно осуществляется с помощью программных продуктов, называемых отладчиками. Основное назначение их - выявление ошибок в программах. При анализе алгоритмов программ используются такие возможности отладчиков как пошаговое (покомандное) выполнение программ, возможность останова в контрольной точке.

Покомандное выполнение осуществляется процессором при установке пошагового режима работы. Контрольной точкой называют любое место в программе, на котором обычное выполнение программы приостанавливается, и осуществляется переход в особый режим, например, в режим покомандного выполнения. Для реализации механизма контрольной точки обычно используется прерывание по соответствующей команде ЭВМ (для IBM-совместных ПЭВМ такой командой является команда INT). В современных процессорах можно использовать специальные регистры для установки нескольких контрольных точек при выполнении определенных операций: обращение к участку памяти, изменение участка памяти, выборка по определенному адресу, обращение к определенному порту ввода-вывода и т. д.

При наличии современных средств отладки программ полностью исключить возможность изучения алгоритма программы невозможно, но

существенно затруднить трассировку возможно. Основной задачей противодействия трассировке является увеличение числа и сложности ручных операций, которые необходимо выполнить программисту-аналитику.

Для противодействия трассировке программы в ее состав вводятся следующие механизмы:

- изменение среды функционирования;
- модификация кодов программы;
- «случайные» переходы.

Под изменением среды функционирования понимается запрет или переопределение прерываний (если это возможно), изменение режимов работы, состояния управляющих регистров, триггеров и т. п. Такие изменения вынуждают аналитика отслеживать изменения и вручную восстанавливать среду функционирования.

Изменяющиеся коды программ, например, в процедурах приводят к тому, что каждое выполнение процедуры выполняется по различным ветвям алгоритма.

«Случайные» переходы выполняются за счет вычисления адресов переходов. Исходными данными для этого служат характеристики среды функционирования, контрольные суммы процедур (модифицируемых) и т. п. Включение таких механизмов в текст программ существенно усложняет изучение алгоритмов программ путем их трассировки.

8.1 Классификация средств исследования программ

В этом подразделе мы будем исходить из предположения, что на этапе разработки программная закладка была обнаружена и устранена, либо ее вообще не было. Для привнесения программных закладок в этом случае необходимо взять готовый исполняемый модуль, дизассемблировать его и после внесения закладки подвергнуть повторной компиляции. Другой способ заключается в незаконном получении текстов исходных программ, их анализе,

внесении программных дефектов и дальнейшей замене оригинальных программ на программы с приобретенными закладками. И, наконец, может осуществляться полная замена прикладной исполняемой программы на исполняемую программу нарушителя, что впрочем, требует от последнего необходимость иметь точные и полные знания целевого назначения и конечных результатов прикладной программы.

Все средства исследования ПО можно разбить на 2 класса: статические и динамические. Первые оперируют исходным кодом программы как данными и строят ее алгоритм без исполнения, вторые же изучают программу, интерпретируя ее в реальной или виртуальной вычислительной среде. Отсюда следует, что первые являются более универсальными в том смысле, что теоретически могут получить алгоритм всей программы, в том числе и тех блоков, которые никогда не получат управления. Динамические средства могут строить алгоритм программы только на основании конкретной ее трассы, полученной при определенных входных данных. Поэтому задача получения полного алгоритма программы в этом случае эквивалентна построению исчерпывающего набора текстов для подтверждения правильности программы, что практически невозможно, и вообще при динамическом исследовании можно говорить только о построении некоторой части алгоритма.

Два наиболее известных типа программ, предназначенных для исследования ПО, как раз и относятся к разным классам: это отладчик (динамическое средство) и дизассемблер (средство статистического исследования). Если первый широко применяется пользователем для отладки собственных программ и задачи построения алгоритма для него вторичны и реализуются самим пользователем, то второй предназначен исключительно для их решения и формирует на выходе ассемблерный текст алгоритма.

Помимо этих двух основных инструментов исследования, можно использовать:

- "дискомпиляторы", программы, генерирующие из исполняемого кода программу на языке высокого уровня;

- "трассировщики", сначала запоминающие каждую инструкцию, проходящую через процессор, а затем переводящие набор инструкций в форму, удобную для статического исследования, автоматически выделяя циклы, подпрограммы и т.п.;
- "следящие системы", запоминающие и анализирующие трассу уже не инструкции, а других характеристик, например вызванных программой прерывания.

8.2 Методы защиты программ от исследования

Для защиты программ от исследования необходимо применять методы защиты от исследования файла с ее исполняемым кодом, хранящемся на внешнем носителе, а также методы защиты исполняемого кода, загружаемого в оперативную память для выполнения этой программы.

В первом случае защита может быть основана на шифровании секретной части программы, а во втором - на блокировании доступа к исполняемому коду программы в оперативной памяти со стороны отладчиков. Кроме того, перед завершением работы защищаемой программы должен обнуляться весь ее код в оперативной памяти. Это предотвратит возможность несанкционированного копирования из оперативной памяти дешифрованного исполняемого кода после выполнения защищаемой программы.

Таким образом, защищаемая от исследования программа должна включать следующие компоненты:

- инициализатор;
- зашифрованную секретную часть;
- деструктор (деициниализатор).

Инициализатор должен обеспечивать выполнение следующих функций:

- сохранение параметров операционной среды функционирования (векторов прерываний, содержимого регистров процессора и т.д.);
- запрет всех внутренних и внешних прерываний, обработка которых не может быть запротоколирована в защищаемой программе;

- загрузка в оперативную память и дешифрование кода секретной части программы;
- передача управления секретной части программы.

Секретная часть программы предназначена для выполнения основных целевых функций программы и защищается шифрованием для предупреждения внесения в нее программной закладки.

Деструктор после выполнения секретной части программы должен выполнить следующие действия:

- обнуление секретного кода программы в оперативной памяти;
- восстановление параметров операционной системы (векторов прерываний, содержимого регистров процессора и т.д.), которые были установлены до запрета неконтролируемых прерываний;
- выполнение операций, которые невозможно было выполнить при запрете неконтролируемых прерываний;
- освобождение всех незадействованных ресурсов компьютера и завершение работы программы.

Для большей надежности инициализатор может быть частично зашифрован и по мере выполнения может дешифровать сам себя. Дешифроваться по мере выполнения может и секретная часть программы. Такое дешифрование называется динамическим дешифрованием исполняемого кода. В этом случае очередные участки программ перед непосредственным исполнением расшифровываются, а после исполнения сразу уничтожаются.

Для повышения эффективности защиты программ от исследования необходимо внесение в программу дополнительных функций безопасности, направленных на защиту от трассировки. К таким функциям можно отнести:

- периодический подсчет контрольной суммы области оперативной памяти, занимаемой защищаемым исходным кодом; сравнение текущей контрольной суммы с предварительно сформированной эталонной и принятие необходимых мер в случае несовпадения;

- проверку количества занимаемой защищаемой программой оперативной памяти; сравнение с объемом, к которому программа адаптирована, и принятие необходимых мер в случае несоответствия;
- контроль времени выполнения отдельных частей программы;
- блокировку клавиатуры на время отработки особо секретных алгоритмов.

Для защиты программ от исследования с помощью дизассемблеров можно использовать и такой способ, как усложнение структуры самой программы с целью запутывания злоумышленника, который дизассемблирует эту программу. Например, можно использовать разные сегменты адреса для обращения к одной и той же области памяти. В этом случае злоумышленнику будет трудно догадаться, что на самом деле программа работает с одной и той же областью памяти.

Анализ программ на этапе их эксплуатации

В данном разделе будут рассмотрены методы поиска и нейтрализации РПС с помощью дизассемблеров и отладчиков на этапе эксплуатации программ. То есть задача защиты в отличии задач защиты в предыдущих разделах здесь решается "с точностью до наоборот".

Основная схема анализа исполняемого кода, в данном случае, может выглядеть следующим образом:

- выделение чистого кода, то есть удаление кода, отвечающего за защиту этой программы от несанкционированного запуска, копирования и т.п. и преобразования остального кода в стандартный правильно интерпретируемый дизассемблером;
- лексический анализ;
- дизассемблирование;
- семантический анализ;

- перевод в форму, удобную для следующего этапа (в том числе и перевод на язык высокого уровня);
- синтаксический анализ.

После снятия защиты осуществляется поиск сигнатур (лексем) РПС. Примеры сигнатур РПС приведены в работе. Окончание этапа дизассемблирования предшествует синтаксическому анализу, то есть процессу отождествлению лексем, найденных во входной цепочке, одной из языковых конструкций, задаваемых грамматикой языка, то есть синтаксический анализ исполняемого кода программ состоит в отождествлении сигнатур, найденных на этапе лексического анализа, одному из видов РПС.

При синтаксическом анализе могут встретиться следующие трудности:

- могут быть не распознаны некоторые лексемы. Это следует из того, что макроассемблерные конструкции могут быть представлены бесконечным числом регулярных ассемблерных выражений;
- порядок следования лексем может быть известен с некоторой вероятностью или вообще не известен;
- грамматика языка может пополняться, так как могут возникать новые типы РПС или механизмы их работы.

Таким образом, окончательное заключение об отсутствии или наличии РПС можно дать только на этапе семантического анализа, а задачу этого этапа можно конкретизировать как свертку терминальных символов в нетерминалы как можно более высокого уровня там, где входная цепочка задана строго.

Так как семантический анализ удобнее вести на языке высокого уровня далее проводится этап перевода ассемблерного текста в текст на языке более высокого уровня, например, на специализированном языке макроассемблера, который нацелен на выделение макроконструкций, используемых в РПС.

На этапе семантического анализа дается окончательный ответ на вопрос о том, содержит ли входной исполняемый код РПС, и если да, то какого типа. При этом используется вся информация, полученная на всех предыдущих этапах. Кроме того, необходимо учитывать, что эта информация может

считаться правильной лишь с некоторой вероятностью, причем не исключены вообще ложные факты, или умозаключения исследователей. В целом, задача семантического анализа является сложной и ресурсоемкой и скорее не может быть полностью автоматизирована.

8.3 Общая характеристика и классификация компьютерных вирусов

Под *компьютерным вирусом* (или просто вирусом) понимается автономно функционирующая программа, обладающая способностью к самостоятельному внедрению в тела других программ и последующему самовоспроизведению и самораспространению в информационно-вычислительных сетях и отдельных ЭВМ. Предшественниками вирусов принято считать так называемые *троянские программы*, тела которых содержат скрытые последовательности команд (модули), выполняющие действия, наносящие вред пользователям. Наиболее распространенной разновидностью троянских программ являются широко известные программы массового применения (редакторы, игры, трансляторы и т.д.), в которые встроены так называемые "логические бомбы", срабатывающие по наступлении некоторого события. Следует отметить, что троянские программы не являются саморазмножающимися.

Принципиальное отличие вируса от троянской программы состоит в том, что вирус после его активизации существует самостоятельно (автономно) и в процессе своего функционирования заражает (инфицирует) программы путем включения (имплантации) в них своего текста. Таким образом, компьютерный вирус можно рассматривать как своеобразный "генератор троянских программ". Программы, зараженные вирусом, называются *вирусоносителями*.

Заражение программы, как правило, выполняется таким образом, чтобы вирус получил управление раньше самой программы. Для этого он либо встраивается в начало программы, либо имплантируется в ее тело так, что первой командой зараженной программы является безусловный переход на компьютерный вирус, текст которой заканчивается аналогичной командой

безусловного перехода на команду вирусоносителя, бывшую первой до заражения. Получив управление, вирус выбирает следующий файл, заражает его, возможно, выполняет какие-либо другие действия, после чего отдает управление вирусоносителю.

"Первичное" заражение происходит в процессе поступления инфицированных программ из памяти одной машины в память другой, причем в качестве средства перемещения этих программ могут использоваться как магнитные носители (дискеты), так и каналы вычислительных сетей. Вирусы, использующие для размножения сетевые средства, принято называть сетевыми. Цикл жизни вируса обычно включает следующие периоды: внедрение, инкубационный, репликации (саморазмножения) и проявления. В течение инкубационного периода вирус пассивен, что усложняет задачу его поиска и нейтрализации. На этапе проявления вирус выполняет свойственные ему целевые функции, например необратимую коррекцию информации в компьютере или на магнитных носителях.

Физическая структура компьютерного вируса достаточно проста. Он состоит из головы и, возможно, хвоста. Под головой вируса понимается его компонента, получающая управление первой. Хвост - это часть вируса, расположенная в тексте зараженной программы отдельно от головы. Вирусы, состоящие из одной головы, называют несегментированными, тогда как вирусы, содержащие голову и хвост - сегментированными.

Наиболее существенные признаки компьютерных вирусов позволяют провести следующую их классификацию. По режиму функционирования:

- *резидентные вирусы* - вирусы, которые после активизации постоянно находятся в оперативной памяти компьютера и контролируют доступ к его ресурсам;
- *транзитные вирусы* - вирусы, которые выполняются только в момент запуска зараженной программы.

По объекту внедрения:

- *файловые вирусы* - вирусы, заражающие файлы с программами;

- *загрузочные (бутовые) вирусы* - вирусы, заражающие программы, хранящиеся в системных областях дисков.

В свою очередь файловые вирусы подразделяются на вирусы, заражающие:

- исполняемые файлы;
- командные файлы и файлы конфигурации;
- составляемые на макроязыках программирования, или файлы, содержащие макросы (макровирусы);
- файлы с драйверами устройств;
- файлы с библиотеками исходных, объектных, загрузочных и оверлейных модулей, библиотеками динамической компоновки и т.п.

Загрузочные вирусы подразделяются на вирусы, заражающие:

- системный загрузчик, расположенный в загрузочном секторе дискет и логических дисков;
- внесистемный загрузчик, расположенный в загрузочном секторе жестких дисков.

По степени и способу маскировки:

- вирусы, не использующие средств маскировки;
- *stealth-вирусы* - вирусы, пытающиеся быть невидимыми на основе контроля доступа к зараженным элементам данных;
- *вирусы-мутанты (MtE-вирусы)* - вирусы, содержащие в себе алгоритмы шифрования, обеспечивающие различие разных копий вируса.

MtE-вирусы делятся на

- обычные вирусы-мутанты, в разных копиях которых различаются только зашифрованные тела, а расшифровщики совпадают;
- полиморфные вирусы, в разных копиях которых различаются не только зашифрованные тела, но их дешифровщики.

Наиболее распространенные типы вирусов характеризуются следующими основными особенностями.

Файловый транзитный вирус целиком размещается в исполняемом файле, в связи с чем он активизируется только в случае активизации вирусоносителя, а по выполнении необходимых действий возвращает управление самой программе. При этом выбор очередного файла для заражения осуществляется вирусом посредством поиска по каталогу. Файловый резидентный вирус отличается от нерезидентного логической структурой и общим алгоритмом функционирования. Резидентный вирус состоит из так называемого инсталлятора и программ обработки прерываний. Инсталлятор получает управление при активизации вирусоносителя и инфицирует оперативную память путем размещения в ней управляющей части вируса и замены адресов в элементах вектора прерываний на адреса своих программ, обрабатывающих эти прерывания. На так называемой фазе слежения, следующей за описанной фазой инсталляции, при возникновении какого-либо прерывания управление получает соответствующая подпрограмма вируса. В связи с существенно более универсальной по сравнению с нерезидентными вирусами общей схемой функционирования, резидентные вирусы могут реализовывать самые разные способы инфицирования.

Наиболее распространенными способами являются инфицирование запускаемых программ, а также файлов при их открытии или чтении. Отличительной особенностью последних является инфицирование загрузочного сектора (бут-сектора) магнитного носителя. Голова бутового вируса всегда находится в бут-секторе (единственном для гибких дисков и одном из двух - для жестких), а хвост - в любой другой области носителя. Наиболее безопасным для вируса способом считается размещение хвоста в так называемых псевдосбойных кластерах, логически исключенных из числа доступных для использования.

Существенно, что хвост бутового вируса всегда содержит копию оригинального (исходного) бут-сектора. Механизм инфицирования, реализуемый бутовыми вирусами, например, при загрузке MS DOS, таков. При загрузке операционной системы с инфицированного диска вирус, в силу своего

положения на нем (независимо от того, с дискеты или с винчестера производится загрузка), получает управление и копирует себя в оперативную память. Затем он модифицирует вектор прерываний таким образом, чтобы прерывание по обращению к диску обрабатывались собственным обработчиком прерываний вируса, и запускает загрузчик операционной системы. Благодаря перехвату прерываний бутовые вирусы могут реализовывать столь же широкий набор способов инфицирования и целевых функций, сколь и файловые резидентные вирусы.

Stealth-вирусы пользуются слабой защищенностью некоторых операционных систем и заменяют некоторые их компоненты (драйверы дисков, прерывания) таким образом, что вирус становится невидимым (прозрачным) для других программ. Для этого заменяются функции DOS таким образом, что для зараженного файла подставляются его оригинальная копия и содержание, каким они были до заражения.

Полиморфные вирусы содержат алгоритм порождения дешифровщиков (с размером порождаемых дешифровщиков от 0 до 512 байтов) непохожих друг на друга. При этом в дешифровщиках могут встречаться практически все команды процессора Intel и даже использоваться некоторые специфические особенности его реального режимы функционирования.

Макровирусы распространяются под управлением прикладных программ, что делает их независимыми от операционной системы. Подавляющее число макровирусов функционируют под управлением системы Microsoft Word for Windows. В то же время, известны макровирусы, работающие под управлением таких приложений как Microsoft Exel for Windows, Lotus Ami Pro, Lotus 1-2-3, Lotus Notes, в операционных системах фирм Microsoft и Apple.

Сетевые вирусы, называемые также автономными репликативными программами, или, для краткости, репликаторами, используют для размножения средства сетевых операционных систем. Наиболее просто реализуется размножение в тех случаях, когда сетевыми протоколами предусмотрен обмен программами. Однако, размножение возможно и в тех

случаях, когда указанные протоколы ориентированы только на обмен сообщениями. Классическим примером реализации процесса размножения с использованием только стандартных средств электронной почты является уже упоминаемый репликатор Морриса. Текст репликатора передается от одной ЭВМ к другой как обычное сообщение, постепенно заполняющее буфер, выделенный в оперативной памяти ЭВМ-адресата. В результате переполнения буфера, инициированного передачей, адрес возврата в программу, вызвавшую программу приема сообщения, замещается на адрес самого буфера, где к моменту возврата уже находится текст вируса.

Тем самым вирус получает управление и начинает функционировать на ЭВМ-адресате.

"Лазейки", подобные описанной выше и обусловленные особенностями реализации тех или иных функций в программном обеспечении, являются объективной предпосылкой для создания и внедрения репликаторов злоумышленниками. Эффекты, вызываемые вирусами в процессе реализации ими целевых функций, принято делить на следующие группы:

- искажение информации в файлах либо таблице размещения файлов (FAT-таблице), которое может привести к разрушению файловой системы в целом;
- имитация сбоев аппаратных средств;
- создание звуковых и визуальных эффектов, включая, например, отображение сообщений, вводящих оператора в заблуждение или затрудняющих его работу;
- инициирование ошибок в программах пользователей или операционной системы.

Теоретически возможно создание "вирусных червей" - разрушающих программ, которые незаметно перемещаются между узлами вычислительной сети, не нанося никакого вреда до тех пор, пока не доберутся до целевого узла. В нем программа размещается и перестает размножаться.

Поскольку в будущем следует ожидать появления все более и более скрытых форм компьютерных, уничтожение очагов инфекции в локальных и глобальных сетях не станет проще. Время компьютерных вирусов "общего назначения" уходит в прошлое.

8.4 Общая характеристика средств нейтрализации компьютерных вирусов

Наиболее распространенным средством нейтрализации ПВ являются антивирусные программы (антивирусы). Антивирусы, исходя из реализованного в них подхода к выявлению и нейтрализации вирусов, принято делить на следующие группы:

- детекторы;
- фаги;
- вакцины;
- прививки;
- ревизоры;
- мониторы.

Детекторы обеспечивают выявление вирусов посредством просмотра исполняемых файлов и поиска так называемых сигнатур - устойчивых последовательностей байтов, имеющихся в телах известных вирусов. Наличие сигнатуры в каком-либо файле свидетельствует о его заражении соответствующим вирусом. Антивирус, обеспечивающий возможность поиска различных сигнатур, называют полидетектором.

Фаги выполняют функции, свойственные детекторам, но, кроме того, "излечивают" инфицированные программы посредством "выкусывания" вирусов из их тел. По аналогии с полидетекторами, фаги, ориентированные на нейтрализацию различных вирусов, именуют полифагами.

В отличие от детекторов и фагов, вакцины по своему принципу действия подобны вирусам. Вакцина имплантируется в защищаемую программу и запоминает ряд количественных и структурных характеристик последней. Если вакцинированная программа не была к моменту вакцинации инфицированной,

то при первом же после заражения запуске произойдет следующее. Активизация вирусоносителя приведет к получению управления вирусом, который, выполнив свои целевые функции, передаст управление вакцинированной программе. В последней, в свою очередь, сначала управление получит вакцина, которая выполнит проверку соответствия запомненных ею характеристик аналогичным характеристикам, полученным в текущий момент. Если указанные наборы характеристик не совпадают, то делается вывод об изменении текста вакцинированной программы вирусом. Характеристиками, используемыми вакцинами, могут быть длина программы, ее контрольная сумма и т.д.

Принцип действия прививок основан на учете того обстоятельства, что любой вирус, как правило, помечает инфицируемые программы каким-либо признаком с тем, чтобы не выполнять их повторное заражение. В ином случае имело бы место многократное инфицирование, сопровождаемое существенным и поэтому легко обнаруживаемым увеличением объема зараженных программ. Прививка, не внося никаких других изменений в текст защищаемой программы, помечает ее тем же признаком, что и вирус, который, таким образом, после активизации и проверки наличия указанного признака, считает ее инфицированной и "оставляет в покое".

Ревизоры обеспечивают слежение за состоянием файловой системы, используя для этого подход, аналогичный реализованному в вакцинах. Программа-ревизор в процессе своего функционирования выполняет применительно к каждому исполняемому файлу сравнение его текущих характеристик с аналогичными характеристиками, полученными в ходе предшествующего просмотра файлов. Если при этом обнаруживается, что, согласно имеющейся системной информации, файл с момента предшествующего просмотра не обновлялся пользователем, а сравниваемые наборы характеристик не совпадают, то файл считается инфицированным. Характеристики исполняемых файлов, получаемые в ходе очередного просмотра, запоминаются в отдельном файле (файлах), в связи с чем

увеличения длин исполняемых файлов, имеющего место при вакцинации, в данном случае не происходит. Другое отличие ревизоров от вакцин состоит в том, что каждый просмотр исполняемых файлов ревизором требует его повторного запуска.

Монитор представляет собой резидентную программу, обеспечивающую перехват потенциально опасных прерываний, характерных для вирусов, и запрашивающую у пользователей подтверждение на выполнение операций, следующих за прерыванием. В случае запрета или отсутствия подтверждения монитор блокирует выполнение пользовательской программы. Антивирусы рассмотренных типов существенно повышают вирусозащищенность отдельных ПЭВМ и вычислительных сетей в целом, однако, в связи со свойственными им ограничениями, естественно, не являются панацеей. В работе приведены основные недостатки при использовании антивирусов.

В связи с этим необходима реализация альтернативных подходов к нейтрализации вирусов: создание операционных систем, обладающих высокой вирусозащищенностью по сравнению с наиболее "вирусодружественной" MS DOS, разработка аппаратных средств защиты от вирусов и соблюдение технологии защиты от вирусов.

8.5 Классификация методов защиты от компьютерных вирусов

Проблему защиты от вирусов необходимо рассматривать в общем контексте проблемы защиты информации от несанкционированного доступа и технологической и эксплуатационной безопасности ПО в целом. Основной принцип, который должен быть положен в основу разработки технологии защиты от вирусов, состоит в создании многоуровневой распределенной системы защиты, включающей:

- регламентацию проведения работ на ПЭВМ,
- применение программных средств защиты,
- использование специальных аппаратных средств.

При этом количество уровней защиты зависит от ценности информации, которая обрабатывается на ПЭВМ. Для защиты от компьютерных вирусов в настоящее время используются следующие методы: Архивирование. Заключается в копировании системных областей магнитных дисков и ежедневном ведении архивов измененных файлов.

Архивирование является одним из основных методов защиты от вирусов. Остальные методы защиты дополняют его, но не могут заменить полностью.

Входной контроль. Проверка всех поступающих программ детекторами, а также проверка длин и контрольных сумм вновь поступающих программ на соответствие значениям, указанным в документации. Большинство известных файловых и бутовых вирусов можно выявить на этапе входного контроля. Для этой цели используется батарея (несколько последовательно запускаемых программ) детекторов. Набор детекторов достаточно широк, и постоянно пополняется по мере появления новых вирусов. Однако при этом могут быть обнаружены не все вирусы, а только распознаваемые детектором. Следующим элементом входного контроля является контекстный поиск в файлах слов и сообщений, которые могут принадлежать вирусу (например, Virus, COMMAND.COM, Kill и т.д.). Подозрительным является отсутствие в последних 2-3 килобайтах файла текстовых строк - это может быть признаком вируса, который шифрует свое тело.

Рассмотренный контроль может быть выполнен с помощью специальной программы, которая работает с базой данных "подозрительных" слов и сообщений, и формирует список файлов для дальнейшего анализа. После проведенного анализа новые программы рекомендуется несколько дней эксплуатировать в карантинном режиме. При этом целесообразно использовать ускорение календаря, т.е. изменять текущую дату при повторных запусках программы. Это позволяет обнаружить вирусы, срабатывающие в определенные дни недели (пятница, 13 число месяца, воскресенье и т.д.).

Профилактика. Для профилактики заражения необходимо организовать раздельное хранение (на разных магнитных носителях) вновь поступающих и

ранее эксплуатировавшихся программ, минимизация периодов доступности дискет для записи, разделение общих магнитных носителей между конкретными пользователями.

Ревизия. Анализ вновь полученных программ специальными средствами (детекторами), контроль целостности перед считыванием информации, а также периодический контроль состояния системных файлов.

Карантин. Каждая новая программа проверяется на известные типы вирусов в течение определенного промежутка времени. Для этих целей целесообразно выделить специальную ПЭВМ, на которой не проводятся другие работы. В случае невозможности выделения ПЭВМ для карантина программного обеспечения, для этой цели используется машина, отключенная от локальной сети и не содержащая особо ценной информации.

Сегментация. Предполагает разбиение магнитного диска на ряд логических томов (разделов), часть из которых имеет статус READ_ONLY (только чтение). В данных разделах хранятся выполняемые программы и системные файлы. Базы данных должны хранится в других секторах, отдельно от выполняемых программ. Важным профилактическим средством в борьбе с файловыми вирусами является исключение значительной части загрузочных модулей из сферы их досягаемости. Этот метод называется сегментацией и основан на разделении магнитного диска (винчестера) с помощью специального драйвера, обеспечивающего присвоение отдельным логическим томам атрибута READ_ONLY (только чтение), а также поддерживающего схемы парольного доступа. При этом в защищенные от записи разделы диска помещаются исполняемые программы и системные утилиты, а также системы управления базами данных и трансляторы, т.е. компоненты ПО, наиболее подверженные опасности заражения. В качестве такого драйвера целесообразно использовать программы типа ADVANCED DISK MANAGER (программа для форматирования и подготовки жесткого диска), которая не только позволяет разбить диск на разделы, но и организовать доступ к ним с помощью паролей. Количество используемых логических томов и их размеры зависят от

решаемых задач и объема винчестера. Рекомендуется использовать 3 - 4 логических тома, причем на системном диске, с которого выполняется загрузка, следует оставить минимальное количество файлов (системные файлы, командный процессор, а также программы - ловушки).

Фильтрация. Заключается в использовании программ - сторожей, для обнаружения попыток выполнить несанкционированные действия.

Вакцинация. Специальная обработка файлов и дисков, имитирующая сочетание условий, которые используются некоторым типом вируса для определения, заражена уже программа или нет.

Автоконтроль целостности. Заключается в использовании специальных алгоритмов, позволяющих после запуска программы определить, были ли внесены изменения в ее файл.

Терапия. Предполагает дезактивацию конкретного вируса в зараженных программах специальными программами (фагами). Программы фаги "выкусывают" вирус из зараженной программы и пытаются восстановить ее код в исходное состояние (состояние до момента заражения). В общем случае технологическая схема защиты может состоять из следующих этапов:

- входной контроль новых программ;
- сегментация информации на магнитном диске;
- защита операционной системы от заражения;
- систематический контроль целостности информации.

Необходимо отметить, что не следует стремиться обеспечить глобальную защиту всех файлов, имеющихся на диске. Это существенно затрудняет работу, снижает производительность системы и, в конечном счете, ухудшает защиту из-за частой работы в открытом режиме. Анализ показывает, что только 20-30% файлов должно быть защищено от записи.

При защите операционной системы от вирусов необходимо правильное размещение ее и ряда утилит, которое можно гарантировать, что после начальной загрузки операционная система еще не заражена резидентным файловым вирусом. Это обеспечивается при размещении командного

процессора на защищенном от записи диске, с которого после начальной загрузки выполняется копирование на виртуальный (электронный) диск. В этом случае при вирусной атаке будет заражен дубль командного процессора на виртуальном диске. При повторной загрузке информация на виртуальном диске уничтожается, поэтому распространение вируса через командный процессор становится невозможным.

Кроме того, для защиты операционной системы может применяться нестандартный командный процессор (например, командный процессор 4DOS, разработанный фирмой J.P.Software), который более устойчив к заражению. Размещение рабочей копии командного процессора на виртуальном диске позволяет использовать его в качестве программы-ловушки. Для этого может использоваться специальная программа, которая периодически контролирует целостность командного процессора, и информирует о ее нарушении. Это позволяет организовать раннее обнаружение факта вирусной атаки.

В качестве альтернативы MS DOS было разработано несколько операционных систем, которые являются более устойчивыми к заражению. Из них следует отметить DR DOS и Hi DOS. Любая из этих систем более "вирусоустойчива", чем MS DOS. При этом, чем сложнее и опаснее вирус, тем меньше вероятность, что он будет работать на альтернативной операционной системе.

Анализ рассмотренных методов и средств защиты показывает, что эффективная защита может быть обеспечена при комплексном использовании различных средств в рамках единой операционной среды. Для этого необходимо разработать интегрированный программный комплекс, поддерживающий рассмотренную технологию защиты. В состав программного комплекса должны входить следующие компоненты.

- *Каталог детекторов.* Детекторы, включенные в каталог, должны запускаться из операционной среды комплекса. При этом должна быть обеспечена возможность подключения к каталогу новых детекторов, а также указание параметров их запуска из диалоговой среды. С

помощью данной компоненты может быть организована проверка ПО на этапе входного контроля.

- *Программа-ловушка вирусов.* Данная программа порождается в процессе функционирования комплекса, т.е. не хранится на диске, поэтому оригинал не может быть заражен. В процессе тестирования ПЭВМ программа - ловушка неоднократно выполняется, изменяя при этом текущую дату и время (организует ускоренный календарь). Наряду с этим программа-ловушка при каждом запуске контролирует свою целостность (размер, контрольную сумму, дату и время создания). В случае обнаружения заражения программный комплекс переходит в режим анализа зараженной программы - ловушки и пытается определить тип вируса.
- *Программа для вакцинации.* Предназначена для изменения среды функционирования вирусов таким образом, чтобы они теряли способность к размножению. Известно, что ряд вирусов помечает зараженные файлы для предотвращения повторного заражения. Используя это свойство возможно создание программы, которая обрабатывала бы файлы таким образом, чтобы вирус считал, что они уже заражены.
- *База данных о вирусах и их характеристиках.* Предполагается, что в базе данных будет храниться информация о существующих вирусах, их особенностях и сигнатурах, а также рекомендуемая стратегия лечения. Информация из БД может использоваться при анализе зараженной программы-ловушки, а также на этапе входного контроля ПО. Кроме того, на основе информации, хранящейся в БД, можно выработать рекомендации по использованию наиболее эффективных детекторов и фагов для лечения от конкретного типа вируса.
- *Резидентные средства защиты.* Отдельная компонента может резидентно разместиться в памяти и постоянно контролировать целостность системных файлов и командного процессора. Проверка

может выполняться по прерываниям от таймера или при выполнении операций чтения и записи в файл.

8.6 Контрольные вопросы

1. Какие методы противодействия дизассемблированию вы можете назвать?
2. В чем заключается сущность метода, основанного на использовании самогенерируемых кодов?
3. Опишите методы защиты программ от исследования.
4. В чем состоит принципиальное отличие вируса от троянской программы?
5. Составьте схему классификации вирусов.
6. На какие группы принято делить антивирусы, исходя из реализованного в них подхода к выявлению и нейтрализации вирусов? Приведите примеров антивирусных программ на каждую из групп.
7. Изложите классификацию методов защиты от компьютерных вирусов.

ЗАКЛЮЧЕНИЕ

Разработанное учебное пособие содержит актуальный материал справочно-аналитического характера по следующим темам аппаратно-программных средств и методов защиты информации: идентификация пользователей кс-субъектов доступа к данным, средства и методы ограничения доступа к файлам, аппаратно-программные средства криптографической защиты информации, методы и средства ограничения доступа к компонентам ЭВМ, защита программ от несанкционированного копирования, управление криптографическими ключами, защита программных средств от исследования

Ценность содержания определяется постоянной потребностью защиты информации.

Вся информация структурирована и снабжена вопросами для самоконтроля по каждой теме.

Глубокое изучение рассматриваемых в учебном пособии методов использования программно-аппаратных средств защиты информации будет способствовать становлению студента как специалиста в выбранной им области.

9. РАБОЧАЯ УЧЕБНАЯ ПРОГРАММА

9.1 Организационно-методический раздел

Цели и задачи дисциплины

Цель изучения дисциплины – Предмет курса "Программно-аппаратная защита информации" -механизмы и практические методы защиты информации в компьютерах.

Цель курса - ознакомление студентов с современными средствами защиты информации в компьютерных системах, овладение методами решения профессиональных задач.

Изучение дисциплины "Программно-аппаратная защита информации" должно способствовать воспитанию у них профессиональной компетентности и профессионального кругозора, умению ориентироваться в продуктах и тенденциях развития средств защиты информационных технологий.

Требования к уровню освоения дисциплины

В результате изучения дисциплины студенты должны знать:

- возможные действия противника, направленные на нарушение поли тики безопасности информации;
- наиболее уязвимые для атак противника элементы компьютерных систем;
- механизмы решения типовых задач защиты информации.

Студенты должны уметь:

- анализировать механизмы реализации методов защиты конкретных объектов и процессов для решения профессиональных задач;
- применять штатные средства защиты и специализированные продукты для решения типовых задач;
- квалифицированно оценивать область применения конкретных механизмов защиты;
- грамотно использовать аппаратные средства защиты при решении практических задач.

Кроме того, студенты должны иметь навыки освоения и внедрения новых систем защиты, сопровождения систем защиты.

Объем дисциплины (в часах) и виды учебной работы, 8 и 9 семестры

Виды учебной работы	Всего часов
Общая трудоёмкость дисциплины	120
Лекции	30
Лабораторные занятия	30
Практические занятия	20
Всего самостоятельной работы	51

Разделы дисциплины, виды и объем занятий (в часах)

№ п.п.	Наименование раздела дисциплины	Распределение по видам (час)		
		Лекции	ПЗ	ЛР
1	Введение	2		
2	Идентификация пользователей КС – объектов доступа к данным	2	2	2
3	Средства и методы ограничения доступа к файлам	4	2	4
4	Программно-аппаратные средства шифрования	4	2	4
5	Методы и средства ограничения доступа к компонентам ЭВМ.	6	4	6
6	Защита программ от несанкционированного копирования	4	4	4
7	Хранение ключевой информации	4	2	4
8	Защита программ от изучения	4	4	6

Виды итогового контроля – зачет (восьмой семестр) и экзамен (девятый семестр)

9.2 Содержание дисциплины

РАЗДЕЛ 1. Введение

1.1 Предмет и задачи программно-аппаратной защиты информации

Компьютерная система (КС). Структура и компоненты КС. Классы КС. Сети ЭВМ.

1.2 Основные понятия

Электронный документ (ЭД). Виды информации в КС. Информационные потоки в КС. Понятие ЭД. Типы ЭД. Понятие исполняемого модуля.

1.3 Уязвимость компьютерных систем

Понятие доступа, субъект и объект доступа. Понятие несанкционированного доступа (НСД). Классы и виды НСД. Несанкционированное копирование программ как особый вид НСД. Понятие злоумышленника; злоумышленник в криптографии и при решении проблем компьютерной безопасности (КБ).

1.4 Политика безопасности в компьютерных системах. Оценка защищенности

Способы защиты конфиденциальности, целостности и доступности в КС. Руководящие документы Гостехкомиссии по оценке защищенности от НСД.

РАЗДЕЛ 2. Идентификация пользователей КС - субъектов доступа к данным

2.1 Понятие идентификации пользователя

Задача идентификации пользователя. Понятие протокола идентификации. Локальная и удаленная идентификация. Идентифицирующая информация. Понятие идентифицирующей информации. Способы хранения идентифицирующей информации. Связь с ключевыми системами.

РАЗДЕЛ 3. Средства и методы ограничения доступа к файлам

3.1 Основные подходы к защите данных от НСД

Шифрование. Контроль доступа. Разграничения доступа. Файл как объект доступа. Оценка надежности систем ограничения доступа - сведение к задаче оценки стойкости.

3.2 Организация доступа к файлам

Иерархический доступ к файлам. Понятие атрибутов доступа. Организация доступа к файлам в различных ОС. Защита сетевого файлового ресурса на примерах организации доступа в ОС UNIX, Novell NetWare и т. д.

3.3 Фиксация доступа к файлам

Способы фиксации фактов доступа. Журналы доступа. Критерии информативности журналов доступа. Выявление следов несанкционированного доступа к файлам, метод инициированного НСД.

3.4 Доступ к данным со стороны процесса

Понятие доступа к данным со стороны процесса: отличия от доступа со стороны пользователя. Понятие и примеры скрытого доступа. Надежность систем ограничения доступа.

3.5 Особенности защиты данных от изменения

Защита массивов информации от изменения (имитозащита). Криптографическая постановка защиты от изменения данных. Подходы к решению задачи защиты данных от изменения. Подход на основе формирования имитоприставки (MAC), способы построения MAC. Подход на основе формирования хэш-функции, требования к построению и способы реализации. Формирование электронной цифровой подписи (ЭЦП). Особенности защиты ЭД и исполняемых файлов. Проблема самоконтроля исполняемых модулей.

РАЗДЕЛ 4. Программно-аппаратные средства шифрования

4.1 Построение программно-аппаратных комплексов шифрования

Аппаратные и программно-аппаратные средства криптозащиты данных. Построение аппаратных компонент криптозащиты данных, специализированные СБИС как носители алгоритма шифрования. Защита алгоритма шифрования; принцип чувствительной области и принцип главного ключа. Необходимые и достаточные функции аппаратного средства криптозащиты. Проектирование модулей криптопреобразований на основе сигнальных процессоров.

4.2 Плата Криптон-3 (Криптон-4)

Архитектура платы. Организация интерфейса с приложениями. Другие программно-аппаратные СКЗД.

РАЗДЕЛ 5. Методы и средства ограничения доступа к компонентам ЭВМ

5.1 Компоненты ПЭВМ

Классификация защищаемых компонент ПЭВМ: отчуждаемые и неотчуждаемые компоненты ПЭВМ. Процесс начальной загрузки ПЭВМ, взаимодействие аппаратной и программной частей. Механизмы расширения BIOS, структура расширенного BIOS. Преимущества и недостатки программных и аппаратных средств. Проблемы использования расширении BIOS: эмуляция файловой системы до загрузки ОС и т. д.

5.2 Проблема защиты отчуждаемых компонентов ПЭВМ

Способы защиты информации на съемных дисках. Организация прозрачного режима шифрования.

5.3 Надежность средств защиты компонент

Понятие временной и гарантированной надежности.

РАЗДЕЛ 6. Защита программ от несанкционированного копирования

6.1 Несанкционированное копирование программ

Несанкционированное копирование программ как тип НСД. Юридические аспекты несанкционированного копирования программ. Общее понятие зашиты от копирования. Разновидности задач защиты от копирования.

6.2 Подходы к задаче защиты от копирования

Привязка ПО к аппаратному окружению и физическим носителям как единственное средство защиты от копирования ПО. Привязка программ к гибким магнитным дискам (ГМД). Структура данных на ГМД. Управление контроллером ГМД. Способы создания некопируемых меток. Точное измерение

характеристик форматирования дорожки. Технология "слабых битов". Физические метки и технология работы с ними. Привязка программ к жестким магнитным дискам (ЖМД). Особенности привязки к ЖМД. Виды меток на ЖМД. Привязка к прочим компонентам штатного оборудования ПЭВМ. Привязка к внешним (добавляемым) элементам ПЭВМ. Привязка к портовым ключам. Использование дополнительных плат расширения. Методы "водяных знаков" и методы "отпечатков пальцев".

РАЗДЕЛ 7. Хранения ключевой информации

7.1 Пароли и ключи

Секретная информация, используемая для контроля доступа: ключи и пароли. Злоумышленник и ключи. Классификация средств хранения ключей и идентифицирующей информации.

7.2 Организация хранения ключей (с примерами реализации)

Магнитные диски прямого доступа. Магнитные и интеллектуальные. Средство TouchMemory.

7.3 Типовые решения в организации ключевых систем

Открытое распределение ключей. Метод управляемых векторов.

РАЗДЕЛ 8. Защита программ от изучения

8.1 Изучение и обратное проектирование ПО

Понятие изучения и обратного проектирования ПО. Цели и задачи изучения работы ПО. Способы изучения ПО: статическое и динамическое изучение. Роль программной и аппаратной среды. Временная надежность (невозможность обеспечения гарантированной надежности).

8.2 Задачи защиты от изучения и способы их решения

Защита от отладки. Динамическое преобразование кода. Итеративный программный замок А. Долгина. Принцип ловушек и избыточного кода. Защита от дизассемблирования. Принцип внешней загрузки файлов. Динамическая модификация программы. Защита от трассировки по прерываниям.

8.3 Аспекты проблемы защиты от исследования

Способы ассоциирования защиты и программного обеспечения. Оценка надежности защиты от отладки.

8.4 Вирусы

Защита of разрушающих программных воздействий. Вирусы как особый класс разрушающих программных воздействий. Необходимые и достаточные условия недопущения разрушающего воздействия. Понятие изолированной программной среды.

Рекомендуемый перечень тем лабораторных работ

1. Аппаратные решения для выявления и предотвращения утечек конфиденциальной информации.
2. Программное средство КЗИ "Верба-О", рабочее место администратора безопасности системы.
3. Программное средство "PGP".
4. Средство защиты информации "Secret Net".
5. Разграничение доступа в ОС Novell Netware.

9.3 Учебно-методическое обеспечение дисциплины

Основная литература

1. Гайковыч В., Першин А. Безопасность электронных банковских систем. М., 1994.
2. Грушо А.А., Тимонина Е.Е. Теоретические основы защиты информации. М.: Агентство "Яхтсмен", 1996.
3. Соколов А.В., Шаньгин В.Ф. Защита информации в распределенных корпоративных сетях и системах. – М.: ДМК Пресс, 2002.
4. Вильям Столлингс Криптографическая защита сетей. – М.: Издательсткий дом “Вильямс”, 2001.
5. Кузьминов В.И. Криптографические методы защиты информации. – Новосибирск: Высшая школа, 1998.

6. Гостехкомиссия России. Руководящий документ. Защита от несанкционированного доступа к информации. Термины и определения. М.: ГТК, 1992.
7. Гостехкомиссия России. Руководящий документ. Автоматизированные системы. Защита от несанкционированного доступа к информации. Классификация автоматизированных систем и требования по защите информации. М.: ГТК, 1992.
8. Гостехкомиссия России. Руководящий документ. Средства вычислительной техники. Защита от несанкционированного доступа к информации. Показатели защищенности от НСД к информации. М.: ГТК, 1992.
9. Гостехкомиссия России. Руководящий документ. Средства вычислительной техники. Межсетевые экраны. Показатели защищенности от несанкционированного доступа. М.: ГТК, 1997.
10. Расторгуев СП. Защита информации в компьютерных системах, М., 1993.

Дополнительная литература

1. Дейтел Г. Введение в операционные системы: В 2 т.: Пер. с англ. М,: Мир, 1987.
2. Зегжда Д.П., Иеашко А.М. Основы безопасности информационных систем. М.: Горячая линия - Телеком, 2000.
3. Зубанов Ф. Windows NT - выбор профи. М.: Изд. отд. "Русская редакция" ТОО "Channel Trading Ltd.", 1996.
4. МакМален Дж. UNIX. М.: Компьютер, Изд. об-ние "ЮНИТИ", 1996.
5. Мафтик С. Механизмы защиты в сетях ЭВМ. М.: Мир, 1993.
6. Методы и теоретические средства обеспечения безопасности информации: Тезисы докладов, СПб.: Изд-во СПб. ГТУ 2000.
7. Домашев А.В., Грунтович М.М., Попов В.О. Программирование алгоритмов защиты информации. – М.: Издательство “Нолидж”, 2002.

Средства обеспечения освоения дисциплины

Дисплейный класс.

Материально-техническое обеспечение дисциплины

Класс ПЭВМ не ниже Intel Pentium 166, 64 Mb RAM, 2 Gb HDD с установленным программным обеспечением: Microsoft Windows NT 4.0, Microsoft Windows 2000 Professional, Microsoft Visual C++, Linux. Из расчета одна ПЭВМ на человека.

Рекомендуемый перечень тем практических занятий

1. Уязвимость компьютерных систем.
2. Идентификация пользователей КС — субъектов доступа к данным.
3. Основные подходы к защите данных от НСД.
4. Организация доступа к файлам.
5. Особенности защиты данных от изменения.
6. Построение программно-аппаратных комплексов шифрования.
7. Плата Криптон-3 (Криптон-4).
8. Защита программ от несанкционированного копирования.
9. Организация хранения ключей.
10. Защита программ от изучения.
11. Вирусы.

10. МЕТОДИЧЕСКИЕ УКАЗАНИЯ К ЛАБОРАТОРНЫМ И ПРАКТИЧЕСКИМ РАБОТАМ

Лабораторная работа № 1
Аппаратные решения для выявления и предотвращения утечек конфиденциальной информации

Цель работы: рассмотреть аппаратные решения для выявления и предотвращения утечек и сделать сравнительный анализ программных компонентов.

Теоретические сведения

Сегодня защита конфиденциальных данных - одна из главных задач любого бизнеса. Почти каждая компания располагает торговыми или промышленными секретами, приватными сведениями своих сотрудников, клиентов и партнеров, а в некоторых случаях интеллектуальной собственностью и другими цифровыми активами. Чтобы защитить всю эту информацию от несанкционированного доступа, предприятия берут на вооружение брандмауэры, системы обнаружения и предотвращения вторжений, средства двухфакторной аутентификации, а также другие продукты и технологии. Однако от инсайдеров - обширной категории служащих компании, имеющих легальный доступ к конфиденциальной информации в силу своих должностных обязанностей, - данные, не подлежащие разглашению, чаще всего остаются беззащитными. Тому, как обеспечить внутреннюю IT-безопасность, зафиксировать и предотвратить утечку или нецелевое использование информационных активов, посвящена эта статья.

Сейчас утечка конфиденциальной информации представляет самую опасную угрозу IT-безопасности. Так, по данным CXO Media и PricewaterhouseCoopers, на долю инсайдеров приходится 60% всех инцидентов

IT-безопасности. В то же самое время по сведениям компании InfoWatch, опросившей более 300 представителей российского бизнеса, 64% респондентов считают кражу данных главной угрозой IT-безопасности, при этом на втором месте со значительным отставанием оказалась угроза вредоносных кодов (49%).

В дальнейшем проблема защиты чувствительных данных только усилится. Это связано, прежде всего, с ужесточением законодательных требований, как по всему миру, так и в России.

Комплексный подход к выявлению и предотвращению утечек

Конфиденциальная информация может "покинуть" корпоративный периметр самыми разными путями. Среди самых распространенных каналов утечки следует отметить мобильные устройства или накопители, электронную почту и веб. Разумеется, никто не мешает нечистому на руку сотруднику воспользоваться более изощренными способами, скажем, переписать данные посредством беспроводных сетей (Bluetooth или Wi-Fi), поменять жесткий диск персонального компьютера и забрать с собой оригинальный и т. д. Таким образом, защита от утечки требует комплексного подхода: учета всех возможных коммуникационных каналов, обеспечения физической безопасности, шифрования резервных копий и информации, покидающей корпоративный периметр, и других организационных мероприятий (создание политики IT-безопасности, разрешение юридических вопросов и модификация трудовых договоров, тренинги и т. д.).

Сегодня на рынке существует довольно много решений, позволяющих детектировать и предотвращать утечку конфиденциальной информации по тем или иным каналам. Однако комплексных решений, покрывающих все существующие каналы, значительно меньше. Некоторые разработчики предоставляют продукты лишь для контроля над почтовым трафиком или коммуникационными портами рабочей станции. Такой подход обладает всего одним преимуществом: заказчик покупает автономный продукт, который требует минимум усилий при внедрении и сопровождении. Тем не менее

слабых сторон намного больше: компания должна сама позаботиться об оставшихся непокрытыми каналах передачи информации (что нередко просто невозможно), а также самостоятельно провести комплекс организационных мероприятий (для чего штатным специалистам часто не хватает опыта и знаний). Другими словами, при выборе конкретного решения заказчик должен обратить самое пристальное внимание на диапазон покрываемых каналов утечки и наличие важных сопроводительных услуг.

Еще один важный параметр, который необходимо учитывать, - наличие или отсутствие аппаратных модулей в комплексном решении либо в автономном продукте. Самые продвинутые поставщики сегодня предлагают на выбор программные и аппаратные компоненты для контроля над теми коммуникационными каналами, где это возможно. Так, ни один разработчик не предложит сегодня аппаратных модулей для предотвращения утечек через ресурсы рабочих станций (порты, принтеры, приводы и т. д.), поскольку эффективность подобной технологии сомнительна. Однако обеспечить контроль над почтовым или веб-трафиком с помощью отдельного устройства, а не выделенного сервера вполне логично. Дополнительным преимуществом такого подхода является возможность более эффективной защиты информационных активов крупной компании, имеющей обширную сеть филиалов. В этом случае можно настроить и протестировать аппаратные компоненты в штаб-квартире, а потом быстро внедрить их в филиалах. В отличие от программных модулей автономные устройства могут быть легко развернуты и не требуют серьезного сопровождения (следовательно, филиалу не обязательно иметь специалистов по IT-безопасности). К тому же в большинстве случаев аппаратное решение обладает более высокой производительностью. Хотя программные компоненты, работающие на выделенных серверах, в некоторых случаях обладают большей гибкостью и возможностями более тонкой настройки. Вдобавок программные модули чаще всего обходятся значительно дешевле аппаратных.

Порядок выполнения работы

1. Изучить аппаратные решения для выявления и предотвращения утечек информации.

2. Сделать сравнительный анализ программных компонентов выявления и предотвращения утечек информации.

Содержание отчета

В отчете необходимо привести:

1. Теоретические сведения.

2. Таблицу сравнительного анализа программных компонентов.

3. Выводы по работе.

Литература

1. Проскурин В.Г. и др. Программно-аппаратные средства обеспечения информационной безопасности. Защита в операционных системах. –М.: Радио и связь, 2000.
2. Программно-аппаратные средства обеспечения информационной безопасности. Защита программ и данных /П.Ю.Белкин, О.О. Михальский, А.С. Першаков и др.- М.: Радио и связь, 1999.
3. Хисамов Ф.Г. Макаров Ю.П. Оптимизация аппаратных средств криптографической защиты информации //Системы безопасности. - 2004. – февраль-март №1 (55). –стр.108.
4. Анин Б.Ю. Защита компьютерной информации. – Санкт-Петербург, 2000.
5. Теоретические основы компьютерной безопасности. Учебное пособие для вузов. Деревянин П.Н., Михальский О.О., Правиков Д.И. – Радио и связь, 2000.

Лабораторная работа № 2
Программное средство КЗИ "Верба-О"

Цель работы: ознакомиться и изучить программное средство КЗИ "Верба-О" (рабочее место оператора и рабочее место администратора безопасности системы).

Теоретические сведения

Постановка задачи

Расширение сфер применения современных информационных технологий выдвигает новые требования к принципам построения и свойствам информационных систем. Сегодня все большую важность приобретает проблема обеспечения безопасности. Объясняется это в первую очередь внедрением и модернизацией информационных технологий в организациях и предприятиях, которые осознали необходимость обеспечения конфиденциальности своих данных, а также появлением новых технических приемов, таких как имеющая юридическую силу электронно-цифровая подпись. Кроме того, как это ни парадоксально, совершенствование компьютерных технологий привело к образованию множества лазеек для утечки ценной информации. Так, развитие локальных, а затем корпоративных и глобальных компьютерных сетей значительно увеличивает возможность проникновения в информационную систему предприятия злоумышленников, если не предпринять соответствующих мер. Учитывая, что тенденции развития четко обозначили преимущество комплексных, территориально распределенных компьютерных систем, можно сделать заключение, что задача по обеспечению безопасности также носит комплексный характер. Она может быть решена только в случае применения средств защиты в контексте всех эксплуатируемых операционных систем и прикладных программ.

Здесь следует выделить два основных направления работ, разделенных по принципу расположения данных. Во-первых, необходимы средства обеспечения безопасной связи внутренней сети масштаба предприятия с внешним миром, т. е. защита от несанкционированного доступа извне и надежное шифрование данных, передаваемых через сетевую инфраструктуру

внешнего мира; во-вторых, разграничение прав доступа и шифрование данных внутренних ресурсов.

Стоит отметить, что все предлагаемые компанией ВЕСТЬ АО программные продукты и интеграционные решения на их основе используют в качестве базы надежные сетевые операционные системы и промышленные СУБД, которые имеют встроенные средства идентификации пользователя при подключении к сети и обращении к данным, а также средства по протоколированию выполняемых действий. В результате этого пользователи получают строго ограниченные права доступа к ресурсам и все их попытки несанкционированного доступа регистрируются в системных журналах и журналах прикладного программного обеспечения. На некоторых платформах допустимо усиление защиты за счет встраивания в операционную среду дополнительных модулей контроля доступа. Например, система управления документами DOCS Open дает возможность применять в среде Windows NT Server специальное программное обеспечение DOCS Open Document Sentry Agent, которое предотвращает прямое обращение к объектам файловой системы в обход системы безопасности DOCS Open (можно сказать, что Document Sentry Agent выполняет функции своеобразного брандмауэра на уровне файлов).

Вместе с тем приходится констатировать, что, хотя большинство современных сетевых операционных систем и обеспечивают некоторый набор средств для решения задач безопасности, они не могут гарантировать полной конфиденциальности информации, поскольку используют незащищенные сервисы, такие как электронная почта или хранение данных в виде незашифрованных файлов. В связи с этим прикладное и системное программное обеспечение должно быть усилено специальными программными и аппаратными комплексами, гарантирующими криптографическую защиту информации (КЗИ).

Повышение безопасности с помощью криптозащиты

Шифрование информации с помощью системы КЗИ (СКЗИ) позволяет надежно сохранить ее от прочтения неуполномоченными лицами. Зашифрованные данные могут свободно передаваться через открытые каналы связи, пересылаться на дискетах или храниться в виде файлов. В любом случае у владельца данных есть гарантия, что расшифровать данные сможет только то лицо, которому он послал эту информацию.

Современные СКЗИ выполняют функции аутентификации пользователя, шифрования данных, формирования и проверки электронно-цифровой подписи (ЭЦП). Аутентификация может осуществляться не только с помощью ввода пароля, но и посредством ЭЦП. ЭЦП в свою очередь представляет собой некий блок данных, вырабатываемый на основе содержания подписываемого документа и личного секретного ключа пользователя. В случае проведения аутентификации пользователь подписывает формируемый случайным образом документ. Что же касается шифрования сообщений или файлов, то оно, как правило, основывается на принципе формирования ключей — информационных объектов, обеспечивающих уникальное пребразование данных, препятствующее их несанкционированному просмотру. Существует две наиболее известные схемы формирования ключей: первая из них — симметричная — требует наличия одного и того же ключа на двух концах канала связи; вторая — асимметричная, которая имеет пару ключей (открытый и закрытый). Открытый ключ подлежит свободному распространению для организации следующей схемы взаимодействия. Каждый пользователь при отправке конфиденциальной информации шифрует ее с помощью связки «личный закрытый ключ — открытый ключ адресата». Адресат применяет для расшифровки сообщения обратную связку «закрытый ключ адресата — открытый ключ отправителя». Надежность защиты, предоставляемой КЗИ, строится на статистических свойствах применяемых математических методов: без знания ключа расшифровать послание можно только за значительный период времени (несколько лет).

Важность проблемы защиты информации послужила поводом для создания множества как отечественных, так и зарубежных стандартов КЗИ. В числе зарубежных следует прежде всего упомянуть DES (Data Encryption Standard — стандарт шифрования данных), реализующий симметричную схему с закрытым ключом и утвержденный правительством США в качестве государственного стандарта (он использует блочное кодирования с длиной блока 64 бит и ключом в 56 бит), RSA (Rivest, Shamir, Adleman) — криптосистема с открытым ключом, блочные шифры IDEA (International Data Encryption Algorithm), RC-2 и его усовершенствованные версии RC-4 и RC-5. Объединение отдельных алгоритмов в один дало PGP (Pretty Good Privacy), который использует RSA для безопасного обмена ключами, IDEA для шифрования сообщений, RSA для цифровой подписи и MD5 для вычисления хеш-функций.

Все отечественные стандарты КЗИ оформлены в виде ГОСТов. Например, ГОСТ 28147-89 описывает алгоритмы криптографической обработки информации (шифрование и расшифрование, генерирование имитовставки с целью контроля целостности данных и предотвращения случайных или преднамеренных искажений), ГОСТ Р34.10-94 – процедуры выработки и проверки электронной цифровой подписи на базе асимметричного алгоритма, ГОСТ Р34.11-94 – вычисление хеш-функций произвольных блоков данных.

Обычно комплексы КЗИ производятся в виде встраиваемых в операционные системы или прикладное ПО исполняемых модулей или библиотек. Выбор здесь достаточно широк, но следует обратить особое внимание, сертифицирован ли тот или иной комплекс. Не стоит забывать, что каждая страна, как правило, имеет некоторый институт, который сертифицирует системы КЗИ, и обычно лишь малая часть предлагаемых на рынке систем такие сертификаты получает. В России подобную работу ведет ФАПСИ и до последнего времени ни одна из иностранных систем КЗИ (например, Microsoft CriptoAPI, использующийся в Windows NT, RSA

CRYPTOKI, Generic Security Services API, Independent Data Unit Protection API, Generic Crypto Service API) не получила сертификат. Это означает, что применение этих СКЗИ противопоказано государственным структурам и учреждениям и не обеспечивает на территории нашей страны юридической силы подписанным с помощью их средств ЭЦП документам. Кроме того, сертификация косвенно свидетельствует о высокой надежности СКЗИ.

Среди сертифицированных ФАПСИ СКЗИ можно назвать аппаратно-программный криптографический комплекс ШИП (шифратор IP-потоков), комплекс ТИТАН, предназначенный для создания защищенных сетей X.25 с использованием аутентификации абонентов (узлов сети) и шифрования сетевых пакетов X.25, комплекс «Криптографический сервер», «Янтарь» и другие.

«Верба» в составе продуктов от ВЕСТЬ АО

Одной из наиболее известных сертифицированных СКЗИ в России является комплекс «Верба», разработанный в московском отделении Пензенского научно-исследовательского электротехнического института, который позволяет обеспечить высокую степень защиты информации от несанкционированного доступа и выявления ее искажения при хранении на дисках или в результате передачи по каналам связи. Существует несколько версий, функционирующих в различных операционных системах, что дает возможность внедрять «Вербу» в гетерогенные информационные системы. Сегодня доступны версии для DOS, Windows 3.1x, Windows 95, Windows NT и нескольких диалектов Unix.

«Верба» позволяет шифровать информацию практически любых приложений, будь то программа бухгалтерского учета, электронная почта, офисный пакет, система удаленных электронных расчетов с банком или другая прикладная программа. Это, в частности, позволяет интегрировать СКЗИ «Верба» с современными системами управления документами (СУД) и системами автоматизации управления деловыми процессами (САДП), которые становятся неотъемлемой частью корпоративных информационных систем.

Так, компания ВЕСТЬ АО предлагает интеграционное решение, основанное на встраивании криптографических функций СКЗИ «Верба» в СУД DOCS Open (PC DOCS, Inc.), и относящуюся к классу САДП workflow-систему собственной разработки WorkRoute II. Это позволяет создавать прекрасно защищенные электронные архивы, поддерживать защиту информации в рамках автоматизированных деловых процессов (например, значения внутренних переменных с информацией о контрагентах или суммах договоров, а также другие сведения из прикладных программ, интегрированных с WorkRoute II) и формировать конфиденциальный документооборот. Скажем, использование WorkRoute II в комплексе с СКЗИ «Верба», за счет криптозащищенной аутентификации пользователей, обеспечивает достоверность формируемых заданий, а также защищает сами задания и участвующие в документообороте документы от несанкционированного просмотра и гарантирует их целостность (т. е. исключает возможность подмены, намеренного или случайного повреждения).

Программное средство КЗИ «Верба» представляет собой библиотеку динамической компоновки, функции которой после ее установки становятся доступны для использования из системы DOCS Open и WorkRoute II.

Важно, что все разработанные компанией ВЕСТЬ АО комплексы автоматизации делопроизводства, офисного и инженерного документооборота, а также управления предприятием могут быть легко усовершенствованы в целях повышения их безопасности. Таким образом, например, комплекс PowerDOCS, объединяющий в себе СУД DOCS Open, систему маршрутизации заданий и документов с контролем их исполнения WorkRoute II и систему для работы с образами документов DeltaImage, интегрируется со СКЗИ «Верба» путем установки на клиентские места соответствующей библиотеки и встраивания вызовов функций криптозащиты в интерфейс комплекса на основании требований заказчика. Стоит напомнить, что все применяемые в технических решениях компании ВЕСТЬ АО программные продукты (например, та же система DOCS Open) в свою очередь характеризуются

открытостью, т.е. снабжены развитыми средствами, способствующими расширению функционала системы. Точно так же криптозащита может быть встроена в комплекс по организации электронного архива инженерно-технической документации и сопутствующего документооборота TechnoDOCS или в систему автоматизации инвестиционной компании StockRoute. Последняя строится на основе системы WorkRoute II и программ бухгалтерского учета.

В число основных криптографических функций СКЗИ «Вербы» входит шифрование и расшифровка информации на уровне файлов и блоков памяти, формирование ключей электронной цифровой подписи и ключей шифрования (для этой цели существует специальное рабочее место администратора безопасности — АРМ АБ), формирование и проверка ЭЦП файлов и блоков памяти, а также обнаружение искажений, вносимых злоумышленниками или вирусами в защищаемую информацию.

Особо следует отметить, что большинство встраиваемых СКЗИ не гарантирует контроль целостности программного обеспечения СКЗИ при подключении ее к прикладным программам, о чем должны позаботиться интеграторы и разработчики комплексных информационных систем. В этом смысле «Верба» выгодно отличается от своих конкурентов, поскольку имеет систему встроенного контроля целостности и может дополнятся системой защиты от несанкционированного доступа «Аккорд».

Среди прочих возможностей СКЗИ «Верба» выделяется автоматическая загрузка рабочего ключа по его идентификатору в процессе расшифровки файла и формирования ЭЦП, подпись файла несколькими (от 1 до 255) корреспондентами, а также выполнение ряда специальных операций. В их число входит шифрование группы файлов на одном ключе с объединением их в один закрытый файл и выборочная расшифровка одного из них, а также ведение журналов регистрации протокола проверки ЭЦП, подписанных файлов, шифрования файлов и вывода расшифрованных файлов на печать.

Технические характеристики СКЗИ «Верба»

СКЗИ «Верба» использует ключи длиной 256 бит. Государственные организации обычно применяют модификацию «Вербы», обеспечивающую работу с закрытыми симметричными ключами. Для коммерческих организаций предлагается «Верба-О», реализующая шифрование с помощью асимметричных ключей.

СКЗИ «Верба» требует 200 Кб оперативной памяти и наличия накопителя на гибком магнитном диске. Ключи шифрования и ЭЦП могут храниться на гибком и жестком магнитных дисках. К Windows-версии СКЗИ «Верба» прилагается пример Windows-приложения в исходных текстах, использующего функции DLL.

При обработке информации на компьютере PC/AT 486/100 СКЗИ «Верба-О» обеспечивает следующие показатели быстродействия.

Операции	**Значение**
Шифрование	500 Кб/с
Вычистение хеш-функции	400 Кб/с
Формированеи ЭЦП	0,04 с
Проверка ЭЦП	0,2 с

Алгоритм шифрования выполнен в соответствии с требованиями ГОСТ 28147-89 «Системы обработки информации. Защита криптографическая».

Цифровая подпись выполнена согласно требованиям ГОСТ Р34.10-94 «Информационная технология. Криптографическая защита информации. Процедуры выработки и проверки электронной цифровой подписи на базе асимметричного криптографического алгоритма».

СКЗИ «Верба» может комплектоваться платой аппаратной поддержки «Кулон-1», дополняющей комплекс мер по защите от несанкционированного доступа.

Система шифрования и ЭЦП удовлетворяет требованиям ГОСТ Р34.10-94, ГОСТ Р34.11-94, ГОСТ 28147-89 и имеет сертификат ФАПСИ №СФ/114-0009 от 10.04.1996. Автоматизированное рабочее место администратора безопасности имеет сертификат СФ/114-0063 от 27.05.1996.

Важное замечание

Следует отметить, что предлагаемые компаний ВЕСТЬ АО комплексы автоматизации могут интегрироваться практически с любыми СКЗИ, в соответствии с пожеланиями клиента. Большинство современных СКЗИ обеспечивают легкое встраивание своей функциональной части в прикладные системы. Кроме того, для повышения надежности защиты программные средства могут быть дополнены различными аппаратными и биометрическими средствами, предоставляющими дополнительные данные для аутентификации. Это могут быть, например, смарт-карты, цифровые ключи, устройства распознавания отпечатков пальцев, сетчатки глаза, голоса, лица, оцифрованной подписи.

В заключение хотелось бы напомнить, что одних только технических средств обеспечения безопасности недостаточно. Руководство любой организации должно понимать, что наиболее уязвимым звеном любой системы является человек. Таким образом, вместе с внедрением комплекса криптозащиты (а лучше предварительно) необходимо провести с сотрудниками разъяснительную работу, издать соответствующие организационно-распорядительные документы, определить степень ответственности и, может быть, взять подписку о неразглашении информации, предназначенной для служебного пользования, а также разработать комплекс мер по соблюдению режима секретности и контролю дисциплины. Все это позволит снизить вероятность отрицательных последствий от возможной халатности или злого умысла сотрудников.

Порядок выполнения работы

1. Ознакомиться с программным средством КЗИ "Верба-О".

2. Изучить рабочее место оператора.
3. Изучить рабочее место администратора безопасности системы.
4. Сделать выводы по техническим характеристикам программного средства КЗИ "Верба-О".

Содержание отчета

В отчете необходимо привести:

1. Теоретические сведения.
2. Методику организации рабочего места оператора.
3. Организационные меры работы администратора безопасности системы.
4. Выводы по работе.

Литература

1. Программно-аппаратные средства обеспечения информационной безопасности. Защита программ и данных / П.Ю. Белкин, О.О. Михальский, А.С. Першаков и др.- М.: Радио и связь, 1999.
2. Зегжда Д.П., Иеашко А.М. Основы безопасности информационных систем. М.: Горячая линия - Телеком, 2000.
3. Мафтик С. Механизмы защиты в сетях ЭВМ. М.: Мир, 1993.
4. Ростовцев А.Г., Матвеев В.А. Защита информации в компьютерных системах. – СПб.: Издательство СПбГТУ, 1993.
5. Девянин П.Н., Михальский О.О. и др. Теоретические основы компьютерной безопасности.-М.: Радио и связь, 2000.-192 с.
6. Вильям Столлингс Криптографическая защита сетей. – М.: Издательсткий дом “Вильямс”, 2001.

Лабораторная работа № 3

Система защиты информации "Secret Net"

Цель работы: Изучить программно-аппаратный комплекс для обеспечения информационной безопасности в локальной вычислительной сети Secret Net.

Теоретические сведения

Назначение:

Программно-аппаратный комплекс для обеспечения информационной безопасности в локальной вычислительной сети, рабочие станции и сервера которой работают под управлением следующих операционных систем: Windows'9x (Windows 95, Windows 98 и их модификаций); Windows NT версии 4.0; UNIX MP-RAS версии 3.02.00.

Безопасность рабочих станций и серверов сети обеспечивается с помощью всевозможных механизмов защиты:

- усиленная идентификация и аутентификация,
- полномочное и избирательное разграничение доступа,
- замкнутая программная среда,
- криптографическая защита данных,

- другие механизмы защиты.

Администратору безопасности предоставляется единое средство управления всеми защитными механизмами, позволяющее централизованно управлять и контролировать исполнение требований политики безопасности.

Вся информация о событиях в информационной системе, имеющих отношение к безопасности, регистрируется в едином журнале регистрации. О попытках свершения пользователями неправомерных действий администратор безопасности узнает немедленно.

Существуют средства генерации отчетов, предварительной обработки журналов регистрации, оперативного управления удаленными рабочими станциями.

Компоненты Secret Net

Система Secret Net состоит из трех компонент: Клиентская часть Сервер безопасности Подсистема управления Особенностью системы Secret Net является клиент-серверная архитектура, при которой серверная часть обеспечивает централизованное хранение и обработку данных системы защиты, а клиентская часть обеспечивает защиту ресурсов рабочей станции или сервера и хранение управляющей информации в собственной базе данных.

Клиентская часть системы защиты

Клиент Secret Net (как автономный вариант, так и сетевой) устанавливается на компьютер, содержащий важную информацию, будь то рабочая станция в сети или какой-либо сервер (в том числе и сервер безопасности).

Основное назначение клиента Secret Net:

Защита ресурсов компьютера от несанкционированного доступа и разграничение прав зарегистрированных пользователей. Регистрация событий, происходящих на рабочей станции или сервере сети, и передача информации на сервер безопасности. Выполнение централизованных и децентрализованных управляющих воздействий администратора безопасности.

Клиенты Secret Net оснащаются средствами аппаратной поддержки (для идентификации пользователей по электронным идентификаторам и управления загрузкой с внешних носителей).

Сервер безопасности

Сервер безопасности устанавливается на выделенный компьютер или контроллер домена и обеспечивает решение следующих задач:

Ведение центральной базы данных (ЦБД) системы защиты, функционирующую под управлением СУБД Oracle 8.0 Personal Edition и содержащую информацию, необходимую для работы системы защиты. Сбор информации о происходящих событиях со всех клиентов Secret Net в единый журнал регистрации и передача обработанной информации подсистеме управления. Взаимодействие с подсистемой управления и передача управляющих команд администратора на клиентскую часть системы защиты.

Подсистема управления Secret Net

Подсистема управления Secret Net устанавливается на рабочем месте администратора безопасности и предоставляет ему следующие возможности: Централизованное управление защитными механизмами клиентов Secret Net. Контроль всех событий имеющих отношение к безопасности информационной системы. Контроль действий сотрудников в ИС организации и оперативное реагирование на факты и попытки НСД. Планирование запуска процедур копирования ЦБД и архивирования журналов регистрации. Схема управления, реализованная в Secret Net, позволяет управлять информационной безопасностью в терминах реальной предметной области и в полной мере обеспечить жесткое разделение полномочий администратора сети и администратора безопасности.

Автономный и сетевой вариант

Система защиты информации Secret Net выпускается в автономном и сетевом вариантах.

Автономный вариант - состоит только из клиентской части Secret Net и предназначен для обеспечения защиты автономных компьютеров или рабочих станций и серверов сети, содержащих важную информацию.

С*етевой вариант* - состоит из клиентской части, подсистемы управления, сервера безопасности и позволяет реализовать защиту, как всех компьютеров сети, так и только тех рабочих станций и серверов, которые хранят и обрабатывают важную информацию. Причем в сетевом варианте, благодаря наличию сервера безопасности и подсистемы управления, будет обеспечено централизованное управление и контроль работы всех компьютеров, на которых установлены клиенты Secret Net.

Сферы применения Secret Net

Основными сферами применения системы Secret Net являются: Защита информационных ресурсов; Централизованное управление информационной безопасностью; Контроль состояния информационной безопасности.

Порядок выполнения работы

1. Ознакомиться с программно-аппаратным комплексом для обеспечения информационной безопасности в локальной вычислительной сети Secret Net.
2. Исследовать возможные механизмы защиты обеспечивающие защиту и безопасность рабочих станций и серверов сети.
3. Описать структурную схему системы Secret Net, в которую входят: клиентская часть; сервер безопасности; подсистема управления.
4. Обосновать технические характеристики программно-аппаратного комплекса для обеспечения информационной безопасности в локальной вычислительной сети на базе Secret Net.

Содержание отчета

1. Теоретические сведения.

2. Полное описание компонент системы Secret Net.
3. Примеры автономного и сетевого варианта системы Secret Net.
4. Расчет технических характеристик программно-аппаратного комплекса для обеспечения информационной безопасности в локальной вычислительной сети на базе Secret Net.
5. Выводы по работе.

Литература

1. Методы и теоретические средства обеспечения безопасности информации: Тезисы докладов, СПб.: Изд-во СПб. ГТУ 2000.
2. Домашев А.В., Грунтович М.М., Попов В.О. Программирование алгоритмов защиты информации. – М.: Издательство “Нолидж”, 2002.
3. Соколов А.В., Шаньгин В.Ф. Защита информации в распределенных корпоративных сетях и системах. – М.: ДМК Пресс, 2002.
4. Вильям Столлингс Криптографическая защита сетей. – М.: Издательсткий дом “Вильямс”, 2001.
5. «Безопасность информационных технологий: Методология создания систем защиты», Домарев В.В., BHV, 2002 г.
6. «Комплексная защита информации», Завгородний И.В., «Логос», 2001.

Лабораторная работа № 4
Программное средство PGP

Цель работы: На основе приведенного теоретического материала и с использованием базы знаний предыдущих дисциплин по информационной

безопасности разобраться в принципе действия программного средства PGP и решить поставленные задачи.

Теоретические сведения

Система PGP (Pretty Good Privacy — вполне надежная секретность) представляет собой весьма замечательное явление. В значительной степени являясь плодом усилий одного человека, Фила Циммермана (Phil Zimmermann), PGP обеспечивает конфиденциальность и сервис аутентификации, которые можно использовать для электронной почты и приложений хранения файлов. По существу, Циммерман сделал следующее.

1. Выбрал в качестве строительных блоков лучшие из доступных криптографических алгоритмов.

2. Интегрировал эти алгоритмы в одном универсальном приложении, построенном на использовании небольшого числа простых команд и независимом от процессора и операционной системы.

3. Сделал соответствующий пакет, включающий документацию и исходный текст программы, свободно доступным через Internet, электронные доски объявлений и коммерческие сети типа CompuServe.

4. Заключил соглашение с некоторой компанией (бывшей Viacrypt, теперь Network Associates) о разработке и поддержке недорогой коммерческой версии PGP, полностью совместимой с бесплатной.

Система PGP быстро получила признание и теперь используется очень широко. Среди причин такого быстрого признания системы PGP можно назвать следующие.

1. Она широко доступна в версиях, выполняемых на множестве платформ, включая DOS/Windows, UNIX, Macintosh и многие другие. Кроме того, имеется коммерческая версия, призванная удовлетворить пользователей, предпочитающих иметь поддержку производителя.

2. Система PGP основана на алгоритмах, которые выдержали проверку практикой и считаются исключительно надежными. В частности, в пакет

включены алгоритмы шифрования с открытым ключом RSA, DSS и алгоритм Диффи-Хеллмана, алгоритмы традиционного шифрования CAST-128, IDEA и 3DES, а также алгоритм хэширования SHA-1.

3. Система PGP имеет очень широкую область применения — от корпораций, которые хотят иметь стандартизованную схему шифрования файлов и сообщений, до простых пользователей, которые нуждаются в защите своей переписки с другими пользователями в Internet или какой-то другой сети.

4. Система PGP не была разработана правительственной или некоторой другой официальной организацией, и поэтому неподконтрольна им. Поэтому PGP имеет дополнительную привлекательность для людей с инстинктивным недоверием к "аппарату".

Сначала мы рассмотрим общие принципы работы PGP, затем выясним, как создаются и хранятся криптографические ключи и наконец обсудим жизненно важный вопрос управления открытыми ключами.

Обозначения

K_s — сеансовый ключ, используемый в схеме традиционного шифрования,

KR_a — личный ключ A, используемый в схеме шифрования с открытым ключом,

KU_a — открытый ключ A, используемый в схеме шифрования с открытым ключом,

EP — шифрование в схеме с открытым ключом,

DP — дешифрование в схеме с открытым ключом,

EC — шифрование в схеме традиционного шифрования,

DC — дешифрование в схеме традиционного шифрования,

H — функция хэширования,

|| — конкатенация,

Z — сжатие с помощью алгоритма ZIP,

R64 — преобразование в формат radix-64 ASCII.

В документации PGP часто используется термин *секретный ключ,* означающий ключ, составляющий пару с открытым ключом в схеме шифрования с открытым ключом. В связи с этим существует возможность перепутать такой ключ с секретным ключом» используемым для традиционного шифрования. Поэтому мы используем вместо этого термин *личный ключ.*

Описание работы системы

Сервис PGP, если не рассматривать управление ключами, складывается из пяти функций: аутентификации, конфиденциальности, сжатия, совместимости на уровне электронной почты и сегментации (табл. 10.1). Мы рассмотрим каждую из них по очереди.

Аутентификация

На рис. 10.1(а) показана схема сервиса цифровой подписи, предлагаемая PGP. Эта схема соответствует схеме цифровой подписи. При этом выполняется следующая последовательность действий.

1. Отправитель создает сообщение.

2. Используется алгоритм SHA-1, в результате чего получается 160-битовый хэш-код сообщения.

Таблица 10.1

Краткая характеристика функций PGP

Функция	Используемые алгоритмы	Описание
Цифровая подпись	DSS/SHA или RSA/SHA	С помощью SHA-1 создается хэш-код сообщения. Полученный таким образом профиль сообщения шифруется с помощью DSS или RSA с использованием личного ключа отправителя и включается в сообщение
Шифрование сообщения	CAST либо IDEA, либо "тройной" DES с тремя ключами и алгоритмом Диффи-Хеллмана или RSA	Сообщение шифруется с помощью CAST-128 или IDEA, или 3DES с одноразовым сеансовым ключом, генерируемым отправителем. Сеансовый ключ шифруется с помощью алгоритма Диффи-Хеллмана или RSA с использованием открытого ключа получателя и включается в сообщение
Сжатие	ZIP	Сообщение можно сжать для хранения или передачи, используя ZIP
Совместимость на уровне электронной почты	Преобразование в формат radix-64	Чтобы обеспечить прозрачность для всех приложений электронной почты, шифрованное сообщение можно превратить в строку ASCII, используя преобразование в формат radix-64
Сегментация	—	Чтобы удовлетворить ограничениям максимального размера сообщений, PGP выполняет сегментацию и обратную сборку сообщения

3. Полученный хэш-код шифруется с помощью алгоритма RSA с использованием личного ключа отправителя, и результат добавляется в начало сообщения.

4. Получатель использует RSA с открытым ключом отправителя, чтобы дешифровать и восстановить хэш-код.

5. Получатель генерирует новый хэш-код полученного сообщения и сравнивает его с дешифрованным хэш-кодом. Если хэш-коды совпадают, сообщение считается подлинным.

Комбинация SHA-1 и RSA обеспечивает эффективную схему цифровой подписи. Ввиду надежности RSA получатель уверен в том, что только владелец соответствующего секретного ключа мог создать эту подпись. Надежность SHA-1 дает получателю уверенность в том, что никто другой не мог создать другое сообщение с соответствующим хэш-кодом и, следовательно, с подписью из оригинального сообщения.

Подписи могут также генерироваться с помощью DSS/SHA-1.

Хотя подписи обычно присоединяются к сообщениям или файлам, для которых они создаются, дело не всегда обстоит так: поддерживаются и отделенные подписи. Отделенная подпись может храниться и передаваться отдельно от самого сообщения. Это оказывается полезным в целом ряде случаев. Пользователь может иметь отдельный протокол подписей всех посылаемых и получаемых им

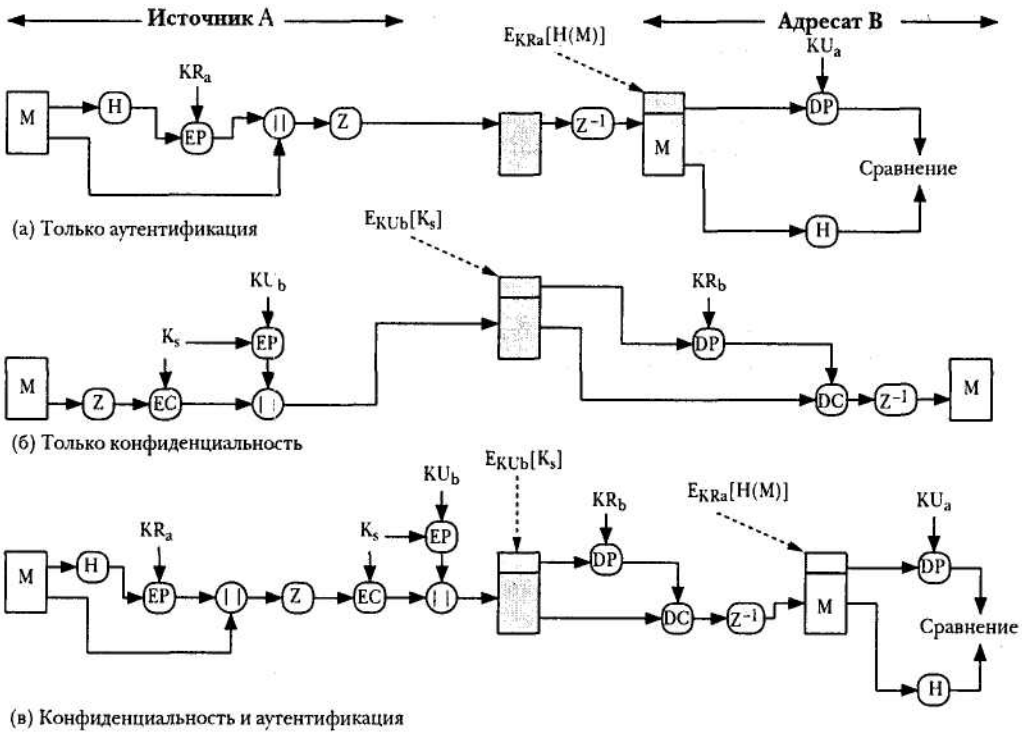


Рисунок 10.1 - Криптографические функции PGP

сообщений. Отделенная подпись выполняемой программы может впоследствии помочь обнаружить заражение программы вирусом. Наконец такие подписи могут использоваться тогда, когда подписывать документ должна не одна, а более сторон, как, например, в случае контракта. Подпись каждой из сторон оказывается тогда независимой и, таким образом, применимой только к данному документу. Иначе подписи должны быть вложенными, так что вторая сторона подписывала бы документ вместе с подписью первой стороны и т.д.

Конфиденциальность

Другим основным сервисом, предлагаемым PGP, является конфиденциальность, обеспечиваемая шифрованием сообщений, предназначенных для передачи или хранения в виде файлов. В обоих случаях можно использовать традиционное шифрование с помощью алгоритма CAST-128. Альтернативой является применение алгоритмов IDEA или 3DES. Может использоваться и режим обратной связи шифрованных 64-битовых блоков (режим CFB).

Как всегда, необходимо решать проблему распределения ключей. В PGP каждый ключ схемы традиционного шифрования применяется только один раз. Это значит, что для каждого сообщения генерируется новый ключ в виде случайного 128-битового числа. Таким образом, хотя в документации такой ключ называется сеансовым, на самом деле он является одноразовым. Ввиду того, что ключ задействуется только один раз, такой сеансовый ключ присоединяется к сообщению и передается вместе с сообщением. Чтобы защитить ключ, он шифруется с использованием открытого ключа получателя. На рис. 10.1(б) показана соответствующая схема, которая может быть описана следующим образом.

1. Отправитель генерирует сообщение и случайное 128-битовое число, которое выступает в качестве сеансового ключа только для этого сообщения.

2. Сообщение шифруется с помощью алгоритма CAST-128 (или IDEA, или 3DES) и данного сеансового ключа.

3. Сеансовый ключ шифруется с помощью алгоритма RSA и открытого ключа получателя и присоединяется к началу сообщения.

4. Получатель использует RSA с личным ключом, чтобы дешифровать и тем самым восстановить сеансовый ключ.

5. Сеансовый ключ применяется для дешифрования сообщения.

Чтобы обеспечить альтернативу использованию RSA для шифрования ключа, в PGP предлагается параметр *Diffie-Hellman* (алгоритм Диффи-Хеллмана). Как уже отмечалось в главе 6, алгоритм Диффи-Хеллмана является алгоритмом обмена ключами. На самом деле в PGP используется вариант этого алгоритма с возможностями шифрования/дешифрования, известный как алгоритм Эль-Гамаля (ElGamal).

В связи с этим можно сделать несколько замечаний. Во-первых, чтобы уменьшить время шифрования, преимущество отдается использованию комбинации традиционного шифрования и шифрования с открытым ключом, а не простому использованию RSA или алгоритма Эль-Гамаля, когда сообщение шифруется непосредственно: CAST-128 и другие алгоритмы традиционной схемы шифрования значительно быстрее, чем RSA или алгоритм Эль-Гамаля. Во-вторых, использование алгоритмов схемы шифрования с открытым ключом решает проблему распределения сеансовых ключей, так как только для получателя оказывается возможным восстановить сеансовый ключ, присоединенный к сообщению. Обратите внимание на то, что в таком случае не требуется использовать протокол обмена сеансовыми ключами типа описанного в главе 6, поскольку здесь не требуется начинать сеанс обмена данными. В этой ситуации, скорее, каждое сообщение является одиночным независимым событием со своим собственным ключом. К тому же вследствие самой природы электронной почты, являющейся системой с промежуточным хранением данных, использование процедуры подтверждения связи для того, чтобы убедиться в идентичности сеансового ключа обеих сторон, не является практически удобным решением. Наконец, использо-

вание одноразовых ключей в традиционной схеме шифрования еще более усиливает и без того достаточно надежный алгоритм традиционного шифрования. Только небольшой объем открытого текста шифруется с использованием одного ключа, и между ключами нет никакой связи. Таким образом, вся схема оказывается защищенной в той мере, в какой защищен алгоритм схемы шифрования с открытым ключом. Поэтому PGP предлагает пользователю выбор для длины ключа от 768 до 3072 битов (длина ключа DSS для подписей ограничивается величиной в 1024 бита).

Как показано на рис. 4.1(в), для одного сообщения можно использовать обе службы. Сначала для сообщения в виде открытого текста генерируется подпись, которая добавляется в начало сообщения. Затем открытый текст сообщения и подпись шифруются с помощью алгоритма CAST-128 (или IDEA, или 3DES), а сеансовый ключ шифруется с помощью RSA (или алгоритма Эль-Гамаля). Такая схема предпочтительнее обратной, т.е. схеме, когда сначала шифруется сообщение, а затем генерируется подпись для шифрованного сообщения. В общем случае оказывается более удобным хранить подпись с открытым текстом сообщения. К тому же с точки зрения возможностей трехсторонней верификации, если сначала генерируется подпись, третьей стороне не нужно заботиться о ключе традиционного шифрования, чтобы проверить подпись.

Короче говоря, при использовании обеих служб отправитель сначала подписывает сообщение с помощью собственного личного ключа, потом шифрует сообщение с помощью сеансового ключа и наконец шифрует сеансовый ключ с помощью открытого ключа получателя.

Сжатие

По умолчанию PGP сжимает сообщение после создания подписи, но перед шифрованием. Это имеет смысл с точки зрения уменьшения объема данных, как при передаче электронной почты, так и при хранении в виде файлов.

Очень важным оказывается выбор места применения алгоритма сжатия, обозначенного на рис. 1 как Z при сжатии и как Z^{-1} при распаковке данных.

1. Подпись генерируется до сжатия по следующим причинам.

- Предпочтительнее подписывать несжатое сообщение, чтобы в будущем иметь возможность хранить сообщение в несжатом виде вместе с подписью. Если подписать сжатый документ, то для верификации необходимо будет либо хранить сжатую версию сообщения, либо сжимать сообщение всякий раз, когда требуется верификация.
- Даже при наличии возможности динамически повторно сжимать сообщение для верификации такой подход несет в себе дополнительные трудности из-за самого алгоритма сжатия PGP: алгоритм не является детерминированным и различные реализации алгоритма выбирают разные варианты выполнения для оптимизации соотношения скорости выполнения и сжатия, а в результате получаются сжатые файлы разной формы. Такие разные алгоритмы сжатия являются переносимыми из-за того, что любая версия алгоритма может правильно восстановить данные, полученные с помощью любой другой версии. Применение функции хэширования и создания подписи после сжатия заставило бы во всех реализациях PGP применять один и тот же алгоритм сжатия.

2. Шифрование сообщения применяется после сжатия для того, чтобы усилить криптографическую защиту сообщения. Ввиду того, что сжатое сообщение имеет меньшую избыточность по сравнению с оригинальным открытым текстом, криптоанализ оказывается более трудным делом.

В качестве алгоритма сжатия применяется ZIP.

Совместимость на уровне электронной почты

При использовании PGP шифруется по крайней мере часть передаваемого блока. Если требуется только цифровая подпись, то шифруется профиль сообщения (с использованием личного ключа отправителя). Если имеет место сервис конфиденциальности, шифруется (с использованием одноразового симметричного ключа) сообщение плюс подпись (при наличии последней). Таким

образом, часть или весь выходной блок сообщения представляет собой поток произвольных 8-битовых байтов. Однако многие системы электронной почты позволяют использовать только блоки, состоящие из символов текста ASCII. Чтобы удовлетворить такому ограничению, PGP обеспечивает сервис конвертирования сырого 8-битового двоичного потока в поток печатаемых символов ASCII.

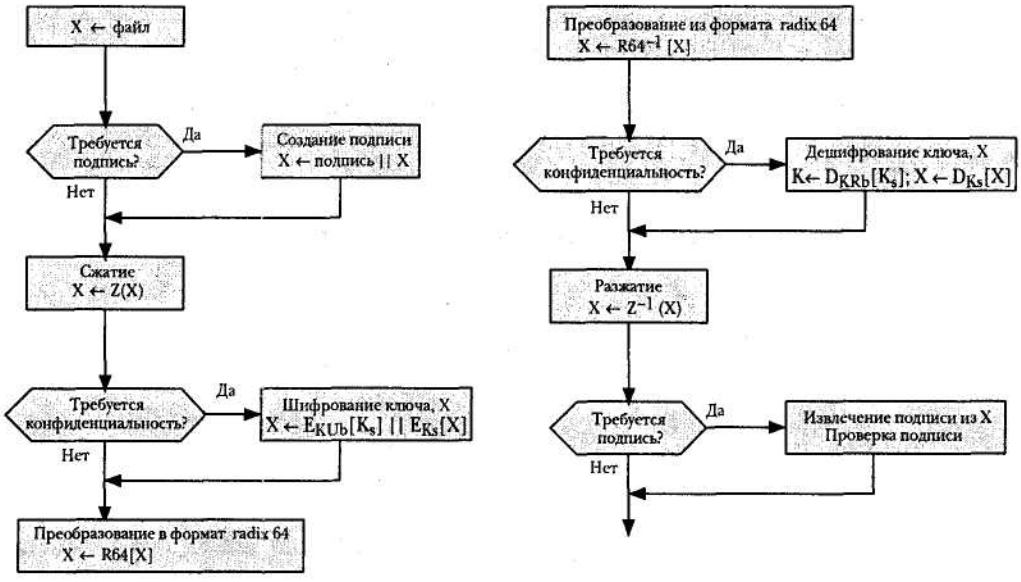
Для этого используется схема конвертирования radix-64. Каждая группа из трех байтов двоичных данных преобразуется в четыре символа ASCII, к которым присоединяется контрольная сумма (CRC), позволяющая обнаружить ошибки при передаче данных.

Конвертирование в формат radix-64 увеличивает длину передаваемого сообщения на 33%. К счастью, сеансовый ключ и порция подписи сообщения относительно компактны, а открытый текст сообщения сжимается. Фактически сжатие с избытком компенсирует расширение, получаемое вследствие перевода в формат radix-64. Например, сообщается о среднем коэффициенте сжатия для ZIP около 2,0. Если игнорировать относительно небольшую подпись и компоненты ключа, типичное полное влияние сжатия и расширение для файла длины X должно быть приблизительно равно 1,33 x 0,5 x X = 0,665 x X. Таким образом, имеет место общее сжатие примерно на одну треть.

Заслуживающим упоминания аспектом алгоритма radix-64 является то, что он слепо конвертирует входной поток в формат radix-64, невзирая на содержимое, даже если ввод оказывается текстом ASCII. Таким образом, если сообщение подписано, но не шифруется и конвертирование применяется ко всему блоку, то выходной поток данных будет непонятен случайному наблюдателю, что уже обеспечивает определенный уровень конфиденциальности. PGP можно сконфигурировать так, чтобы конвертирование в формат radix-64 выполнялось только для порции подписи открытого сообщения. Это дает получателю возможность прочитать сообщение без использования PGP. Но PGP все же придется использовать, если необходимо проверить подпись.

На рис. 10.2 показана связь между четырьмя описанными выше службами. При передаче, если это требуется, подпись генерируется с помощью хэш-кода открытого текста. Затем открытый текст и подпись, если последняя имеется, сжимаются. Далее, если требуется конфиденциальность, блок (сжатый открытый текст или сжатые подпись и открытый текст) шифруется и в начало добавляется шифрованный открытым ключом ключ шифрования традиционной схемы. Наконец весь полученный блок конвертируется в формат radix-64.

На стороне получателя поступающий блок сначала конвертируется обратно из формата radix-64 в двоичный. Затем, если сообщение зашифровано, получатель восстанавливает сеансовый ключ и дешифрует сообщение. Полученный в результате блок разжимается. Если сообщение подписано, получатель восстанавливает полученный хэш-код и сравнивает его с хэш-кодом, вычисленным им самим.



(а) Общая схема отправки (на стороне А)

(б) Общая схема получения (на стороне В)

Рисунок 10.2 - Отправка и прием сообщений PGP

Сегментация и обратная сборка сообщения

Средства электронной почты часто ограничивают максимально допустимую длину сообщения. Например, многие средства электронной почты,

доступные через Internet, допускают пересылку сообщений длиной не более 50000 байтов. Любое более длинное сообщение должно быть разбито на сегменты меньшей длины, каждый из которых посылается отдельно.

Чтобы соответствовать такому ограничению, PGP автоматически разбивает слишком длинные сообщения на сегменты, достаточно малые для того, чтобы их можно было переслать с помощью электронной почты. Сегментация проводится после выполнения всех других операций, включая преобразование в формат radix-64. В результате компоненты ключа и подписи появляются только один раз, в начале первого сегмента. На стороне получателя система PGP должна отбросить заголовок электронной почты и вновь собрать весь оригинальный блок сообщения перед выполнением шагов, показанных на рис. 10.2(6).

Криптографические ключи и связки ключей

PGP использует четыре типа ключей: одноразовые сеансовые ключи схемы традиционного шифрования, открытые ключи, личные ключи и парольные ключи схемы традиционного шифрования, описанные ниже. В отношении этих ключей можно сформулировать следующие три требования.

1. Наличие средств генерирования непредсказуемых сеансовых ключей.

Желательно, чтобы пользователь мог иметь несколько пар открытых/личных ключей. Одной из причин такого требования является то, что пользователь может время от времени менять пару ключей. В результате все сообщения в пути следования окажутся созданными со старым ключом. К тому же получатели будут знать только старый открытый ключ до тех пор, пока ими не будет получена новая версия ключа. В дополнение к необходимости время от времени менять ключи пользователь может иметь несколько пар ключей одновременно, чтобы взаимодействовать с различными группами получателей или просто для того, чтобы усилить защиту, ограничивая объем материала, шифруемого одним и тем же ключом. В результате однозначного соответствия между пользователями и их открыты

ми ключами нет. Таким образом, возникает необходимость в средствах, позволяющих идентифицировать конкретные ключи.

2. Каждый объект системы PGP должен поддерживать файл собственных пар открытых/личных ключей, а также открытых ключей корреспондентов.

Рассмотрим эти требования по порядку.

Генерирование сеансовых ключей

Каждый сеансовый ключ связывается с одним сообщением и используется только для шифрования и дешифрования этого сообщения. Вспомните, что шифрование/дешифрование сообщения выполняется с помощью алгоритма симметричной схемы шифрования. При этом алгоритмы CAST-128 и IDEA используют 128-битовые ключи, а 3DES — 168-битовый ключ. В дальнейшем обсуждении мы предполагаем использование CAST-128.

Случайные 128-битовые числа генерируются с помощью самого алгоритма CAST-128. Ввод для генератора случайных чисел складывается из 128-битового ключа и двух 64-битовых блоков, которые рассматриваются как открытый текст, подлежащий шифрованию. Используя режим шифрованной обратной связи, шифровальщик CAST-128 порождает два 64-битовых блока шифрованного текста, которые связываются конкатенацией, в результате чего формируется 128-битовый сеансовый ключ. Алгоритм, который при этом используется, основан на алгоритме, описанном в документе ANSI XI2.17.

"Открытый текст" для генератора случайных чисел, формируемый из двух 64-битовых блоков, извлекается из рандомизованного потока 128-битовых чисел. Эти числа строятся на основе ввода с клавиатуры от пользователя. Для создания рандомизованного потока используются как время между нажатиями, так и информация о фактически нажатых клавишах. Таким образом, если пользователь нажимает случайные клавиши в своем обычном темпе, будет порожден достаточно "случайный" поток для ввода. Этот случайный ввод объединяется с предыдущим сеансовым ключом, выданным алгоритмом CAST-128, чтобы сформировать данные для ввода генератору. В результате, ввиду хороших перемеши-

вающих свойств CAST-128, порождается последовательность сеансовых ключей, которая оказывается практически непредсказуемой.

Идентификаторы ключей

Как уже говорилось выше, шифрованное сообщение сопровождается использованным для шифрования сеансовым ключом в зашифрованном виде. Сеансовый ключ шифруется с помощью открытого ключа получателя. Следовательно, только получатель может расшифровать сеансовый ключ и, таким образом, прочесть сообщение. Если бы каждый пользователь использовал одну пару открытого и личного ключей, то получатель сразу бы знал, с помощью какого из ключей можно дешифровать сеансовый ключ — это единственный личный ключ получателя. Однако мы выдвинули требование, чтобы любой пользователь мог иметь любое число пар открытых/личных ключей.

Как в этом случае получателю узнать, какой из открытых ключей использовался для шифрования сеансового ключа? Простейшим решением является передача открытого ключа вместе с сообщением. Получатель мог бы тогда удостовериться, что это действительно один из открытых ключей, а затем продолжить обработку сообщения. Эта схема должна работать, но при этом пересылается слишком много лишних данных. Открытый ключ RSA может иметь длину в сотни десятичных разрядов. Другим решением является связывание с каждым открытым ключом некоторого идентификатора, уникального по крайней мере для одного пользователя. Для этой цели вполне подойдет, например, комбинация идентификатора пользователя и идентификатора ключа. Тогда придется пересылать только значительно более короткий идентификатор ключа. Такое решение, однако, порождает проблему управления и перегрузки: идентификаторы ключей должны приписываться и храниться так, чтобы как отправитель, так и получатель могли установить соответствие между идентификаторами ключей и самими открытыми ключами. Это кажется нежелательным и несколько обременительным.

Решением, принятым в PGP, является присвоение каждому открытому ключу такого идентификатора, который с очень высокой вероятностью должен

оказаться уникальным для данного пользователя. Идентификатор, связываемый с каждым открытым ключом, размещается в младших 64 разрядах ключа. Это значит, что идентификатор открытого ключа KU_a равен (KU. mod $2^{й4}$). Этой длины достаточно для того, чтобы вероятность дублирования идентификаторов ключей оказалась очень мала.

Идентификатор ключа требуется и для цифровой подписи PGP. Из-за того что отправитель может воспользоваться одним из нескольких личных ключей для шифрования профиля сообщения, получатель должен знать, какой открытый ключ ему следует использовать. Поэтому раздел цифровой подписи сообщения включает 64-битовый идентификатор соответствующего открытого ключа. При получении сообщения получатель проверяет, что идентификатор соответствует известному ему открытому ключу отправителя, а затем продолжает проверку подписи.

Теперь, определив понятие идентификатора ключа, мы можем более пристально взглянуть на формат передаваемого сообщения, который показан на рис. 3. Сообщение складывается из трех компонентов: собственно сообщения, его подписи (необязательно) и компонента сеансового ключа (необязательно).

Компонент сообщения включает фактические данные, предназначенные для хранения или передачи, а также имя файла и метку даты-времени, указывающую время создания сообщения.

Компонент подписи включает следующие компоненты.

■ **Метка даты-времени.** Время создания подписи.

■ **Профиль сообщения.** 160-битовый профиль сообщения, созданный с помощью SHA-1 и шифрованный с использованием личного ключа подписи отправителя. Профиль вычисляется для метки даты-времени подписи, связанной конкатенацией с порцией данных компонента сообщения. Включение метки даты-времени подписи в профиль обеспечивает защиту от атак

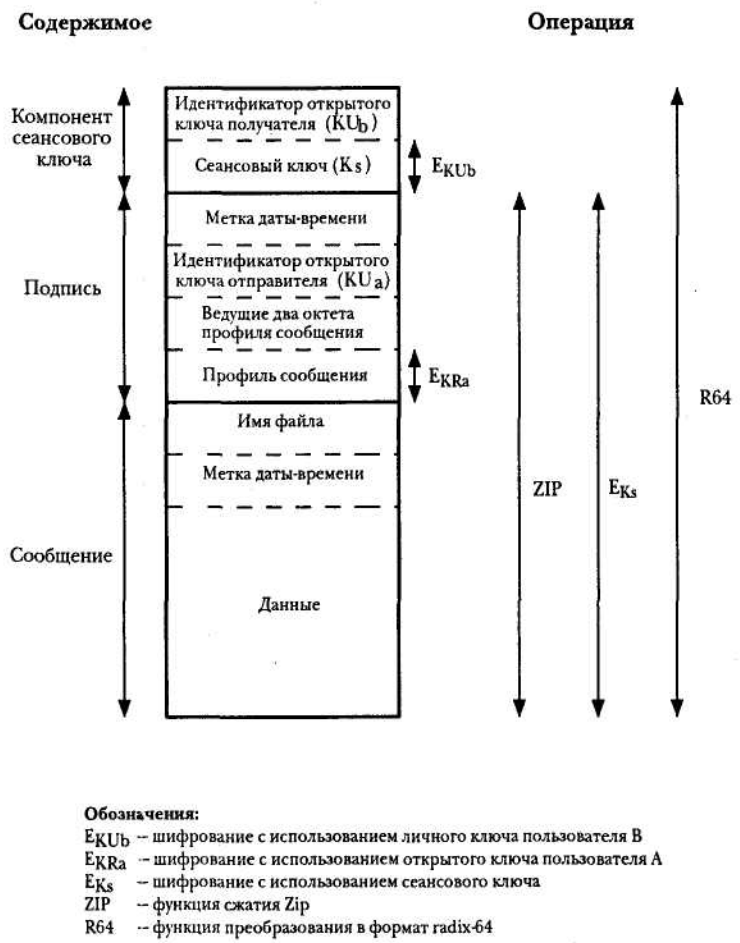


Рисунок 10.3 - Общий формат сообщения PGP (от A к B)

воспроизведения сообщения. Исключение имени файла и метки даты-времени компонента сообщения гарантирует, что отделенная подпись будет в точности совпадать **с** подписью, добавляемой в префикс сообщения. Отделенные подписи вычисляются для файла, в котором нет никаких полей заголовка (хедера) сообщения.

■ **Ведущие два октета профиля сообщения.** Чтобы обеспечить получателю возможность определить, соответствующий ли открытый ключ использовался для дешифрования профиля сообщения с целью аутентификации, проводится сравнение этих первых двух октетов открытого текста исходного профиля с первыми двумя октетами дешифрованного профиля. Эти октеты также служат 16-битовой последовательностью, используемой для проверки сообщения.

■ **Идентификатор открытого ключа отправителя.** Идентифицирует открытый ключ, который должен служить для дешифрования профиля сообщения и, следовательно, идентифицирует личный ключ, использовавшийся для шифрования профиля сообщения.

Компонент сообщения и необязательный компонент подписи могут быть сжаты с помощью ZIP и могут быть зашифрованы с использованием сеансового ключа.

Компонент сеансового ключа включает сеансовый ключ и идентификатор открытого ключа получателя, который использовался отправителем для шифрования данного сеансового ключа.

Весь блок обычно переводится в формат radix-64.

Связки ключей

Мы видели, что идентификаторы ключей в PGP очень важны и что два идентификатора ключей включаются в любое сообщение PGP, предполагающее конфиденциальность и аутентификацию. Эти ключи необходимо хранить и организовать некоторым стандартизованным образом для эффективного применения всеми участвующими в обмене данными сторонами. Схема, используемая в PGP, предполагает создание в каждом узле пары структур данных: одну для хранения пар открытых/секретных ключей данного узла, а другую — для хранения открытых ключей других пользователей, известных данному узлу. Эти структуры данных называются соответственно связкой личных ключей и связкой открытых ключей.

Общая структура **связки личных ключей** показана на рис. 4. Связку можно считать таблицей, в которой каждая строка представляет одну пару открытого/личного ключей, принадлежащих данному пользователю. Каждая строка содержит следующие поля.

■ **Метка даты-времени.** Дата и время создания данной пары ключей.

■ **Идентификатор ключа.** Младшие 64 разряда открытого ключа данной строки.

■ **Открытый ключ.** Открытый ключ данной пары.

■ **Личный ключ.** Личный ключ данной пары; это поле шифруется.

■ **Идентификатор пользователя.** Обычно здесь размещается адрес электронной почты пользователя (например, stallings@acm.org). Однако пользователь может указать для каждой пары ключей разные имена (например, Stallings, WStaliings, WilliamStallings и т.п.) или использовать один идентификатор пользователя несколько раз.

Кольцо личных ключей

Метка даты-времени	Идентификатор ключа*	Открытый ключ	Шифрованный личный ключ	Идентификатор пользователя*
•	•	•	•	•
•	•	•	•	•
•	•	•	•	•
T_i	$KU_i \bmod 2^{64}$	KU_i	$E_{h(P_i)}[KR_i]$	Пользователь *i*
•	•	•	•	•
•	•	•	•	•
•	•	•	•	•

Кольцо открытых ключей

Метка даты-времени	Идентификатор ключа*	Открытый ключ	Доверие владельцу
•	•	•	•
•	•	•	•
•	•	•	•
T_i	$KU_i \bmod 2^{64}$	KU_i	$trust\_flag_i$
•	•	•	•
•	•	•	•
•	•	•	•

Идентификатор пользователя*	Законность ключа	Подпись (подписи)	Доверие подписи (подписям)
•	•	•	•
•	•	•	•
•	•	•	•
Пользователь *i*	$trust\_flag_i$		
•	•	•	•
•	•	•	•
•	•	•	•

* — поле, используемое для индексации таблицы

Рисунок 10.4 - Общая структура связок личных и открытых ключей

Связка личных ключей может быть индексирована либо по полю Идентификатор пользователя, либо по полю Идентификатор ключа; цель такой индексации мы выясним позже.

Хотя предполагается, что связка личных ключей должна храниться только на машине пользователя, создавшего и владеющего соответствующими парами ключей и что она должна быть доступна только этому пользователю, имеет смысл сделать значения личных ключей защищенными настолько, насколько это возможно. Соответственно, сам личный ключ в открытом виде в связке ключей не хранится, а шифруется с помощью CAST-128 (или IDEA, или 3DES). При этом используется следующая процедура.

1. Пользователь выбирает фразу-пароль, которая будет служить для шифрования личных ключей.

2. Когда система с помощью RSA генерирует новую пару открытого/личного ключей, она требует от пользователя указать такую фразу-пароль. Из нее с помощью SHA-1 генерируется 160-битовый хэш-код, а затем пароль удаляется.

3. Система шифрует личный ключ с помощью CAST-128, используя 128 битов хэш-кода в качестве ключа. Хэш-код затем удаляется, а шифрованный личный ключ сохраняется в связке личных ключей.

Впоследствии, когда пользователь обращается к связке личных ключей, чтобы извлечь личный ключ, ему придется снова указать фразу-пароль. PGP извлечет шифрованный личный ключ, вычислит хэш-код пароля и дешифрует личный ключ с помощью CAST-128 с данным хэш-кодом.

Это очень компактная и эффективная схема. Как и в любой основанной на паролях системе, защищенность всей системы зависит от защищенности пароля. Чтобы не поддаться искушению записать пароль, пользователь должен использовать такую парольную фразу, которую угадать нелегко, а запомнить — просто.

На рис. 4.4 показана и общая структура **связки открытых ключей.** Эта структура данных позволяет хранить открытые ключи других пользователей,

известных данному. Пока что давайте проигнорируем некоторые поля, указанные в таблице, и опишем только часть из них.

■ **Метка** даты-времени. Дата и время создания данной записи.

■ **Идентификатор ключа.** Младшие 64 разряда открытого ключа данной записи.

■ **Открытый ключ.** Открытый ключ данной записи.

■ **Идентификатор пользователя.** Владелец данного ключа. С одним открытым ключом можно связать несколько идентификаторов пользователя.

Связка открытых ключей может быть индексирована либо по полю Идентификатор пользователя, либо по полю Идентификатор ключа; цель такой индексации мы выясним позже.

Теперь мы можем показать, как эти связки ключей применяются при передаче и приеме сообщений. Для простоты в следующем примере мы проигнорируем сжатие и преобразование в формат radix-64. Сначала рассмотрим передачу сообщения (рис. 10.5) и предположим, что сообщение должно быть и подписано, и зашифровано. Посылающий сообщение объект PGP выполняет следующие шаги.

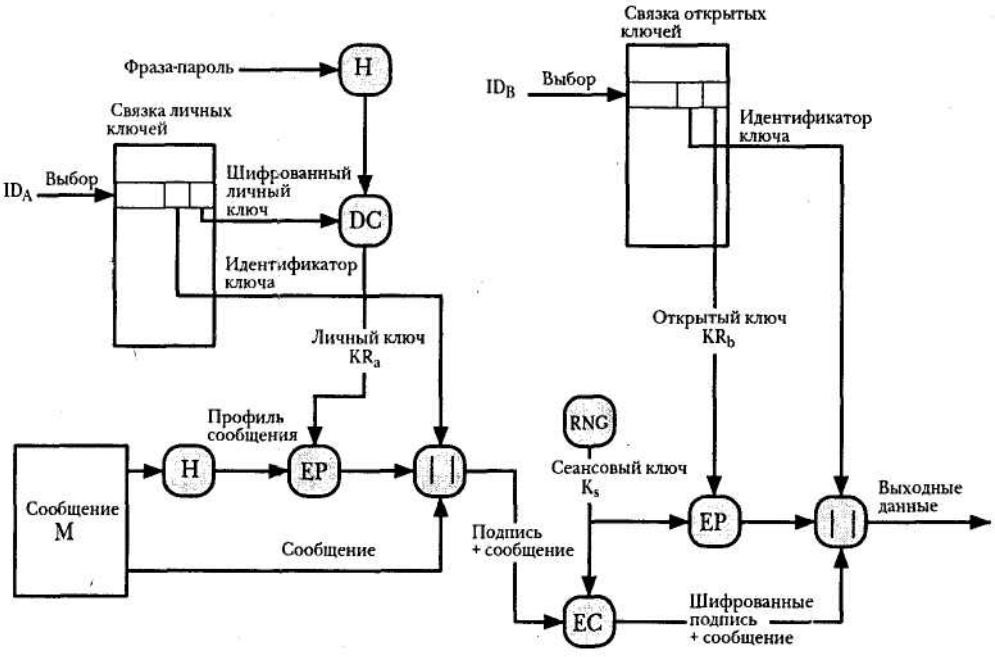


Рисунок 10.5 - Создание сообщения PGP (от А к Б, без сжатия и преобразования в формат radix-64)

1. Создание подписи сообщения.

• PGP извлекает личный ключ отправителя из связки личных ключей, используя введенное значение your_userid в качестве ключа поиска. Если соответствующая команда не предлагает значения your_userid, выбирается первый личный ключ в связке.

• PGP запрашивает у пользователя фразу-пароль, чтобы расшифровать личный ключ.

• Создается компонент подписи сообщения.

2. Шифрование сообщения.

• PGP генерирует сеансовый ключ и шифрует сообщение.

• PGP извлекает открытый ключ получателя из связки открытых ключей, используя значение her_userid в качестве ключа поиска.

• Создается компонент сеансового ключа сообщения.

Принимающий объект PGP выполняет следующие шаги (рис. 10.6).

1. Дешифрование сообщения.

• PGP извлекает личный ключ получателя из связки личных ключей, используя в качестве ключа поиска значение поля Идентификатор ключа компонента сеансового ключа сообщения.

• PGP запрашивает у пользователя фразу-пароль, чтобы расшифровать личный ключ.

• PGP открывает сеансовый ключ и дешифрует сообщение.

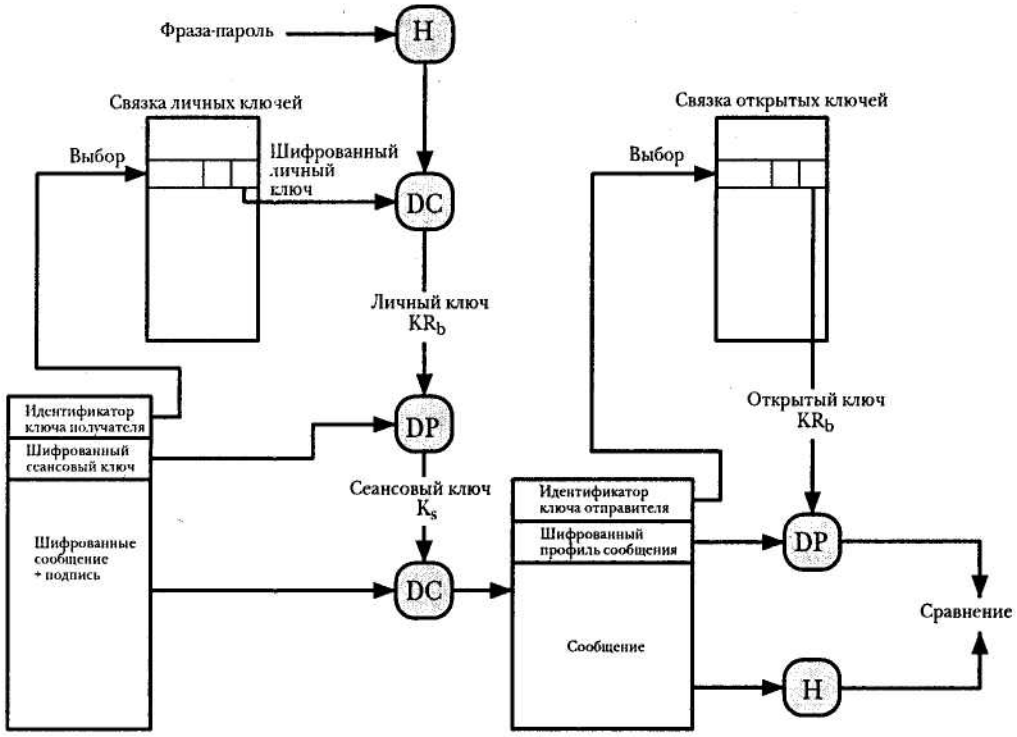


Рисунок 10.6 - Получение сообщения PGP (от A к B, без сжатия и преобразования в формат radix-64)

2. Аутентификация сообщения.

• PGP извлекает открытый ключ отправителя из связки открытых ключей, используя в качестве ключа поиска значение поля Идентификатор ключа компонента подписи сообщения.

• PGP восстанавливает полученный профиль сообщения.

• PGP вычисляет профиль сообщения для принятого сообщения и сравнивает его с профилем, пришедшим вместе с сообщением, чтобы убедиться в их идентичности.

Управление открытыми ключами

Как можно уже догадаться из приведенного выше описания, PGP содержит ясный и эффективный набор взаимосвязанных функций и форматов, обеспечивающих надежную конфиденциальность и средства аутентификации. Для завершенности системы необходимо решить еще одну проблему, а именно

проблему управления открытыми ключами. В документации PGP о важности этой проблемы говорится следующее.

Проблема защиты открытых ключей от несанкционированного вмешательства является отдельной и наиболее сложной практической проблемой приложений, использующих открытые ключи.

Это "ахиллесова пята" криптографии с открытым ключом, и в значительной мере сложность соответствующего программного обеспечения определяется сложностью решения именно этой задачи.

PGP предлагает структуру для решения этой проблемы и ряд опций, которые могут при этом использоваться. Ввиду того, что PGP предназначена для использования в самой разнообразной среде, не устанавливается никакой жесткой схемы управления открытыми ключами, как, например, это сделано в системе S/MIME, которую мы рассмотрим в этой же главе ниже.

Подходы к вопросу управления открытыми ключами

Суть проблемы заключается в следующем. Пользователь А должен построить связку открытых ключей, содержащую открытые ключи других пользователей, чтобы взаимодействовать с ними, используя PGP. Предположим, что связка ключей стороны А включает открытый ключ, приписанный стороне В, но на самом деле владельцем этого ключа является сторона С. Такая ситуация, в частности, может иметь место, если участник А взял ключ с электронной доски объявлений (BBS), которую участник В использовал для того, чтобы переслать открытый ключ, но ключ был скомпрометирован неким С. В результате возникла угроза по двум направлениям. Во-первых, С может посылать сообщения А, фальсифицируя подпись В, так что А будет считать сообщения прибывшими от В. Во-вторых, С сможет прочесть любое шифрованное сообщение от А к В.

Для минимизации риска того, что связка открытых ключей пользователя содержит ложные открытые ключи, можно предложить несколько вариантов действий. Предположим, что А требуется получить надежный открытый ключ В.

Вот несколько вариантов процедуры, которую при этом можно было бы использовать.

1. Получение ключа от B лично (физически). Пользователь B может сохранить свой открытый ключ (KU_b) на дискете и вручить эту дискету пользователю A. Пользователь A затем может загрузить такой ключ с дискеты в свою систему. Это действительно безопасный путь, но он имеет свои очевидные ограничения.

2. Проверка ключа по телефону. Если A может распознать B по телефону, то A может позвонить B и попросить продиктовать ключ в формате radix-64. Еще более удобный вариант выглядит так: B может передать свой ключ пользователю A в виде электронного сообщения. Пользователь A может с помощью PGP и с использованием SHA-1 сгенерировать 160-битовый про филь ключа и представить его в шестнадцатеричном формате; такой про филь называется "отпечатком" ключа. После этого A может позвонить B и попросить продиктовать строку, соответствующую отпечатку его ключа. Если два отпечатка совпадут, ключ считается проверенным.

3. Получение открытого ключа B от внушающего доверие обеим сторонам надежного посредника D. Для этого поставщик D создает подписанный сертификат. Такой сертификат должен включать открытый ключ B, время создания ключа и срок его действия. Сторона D генерирует профиль SHA-1 этого сертификата, шифрует его с помощью своего личного ключа и присоединяет полученную подпись к сертификату. Ввиду того что создать такую подпись в состоянии только D, никто другой не сможет фальсифицировать открытый ключ и заявить» что этот ключ был подписан стороной D. Подписанный сертификат может быть доставлен непосредственно стороне A стороной B или D либо выставлен на электронной доске объявлений.

4. Получение открытого ключа B от надежного уполномоченного узла сертификации. Опять же, сертификат открытого ключа создается и подписывается уполномоченным узлом. Пользователь A может затем получить

доступ к такому узлу, указав свое имя пользователя, и получить подписанный сертификат.

В случаях 3 и 4 пользователь А должен уже иметь экземпляр открытого ключа поставщика сертификатов и быть уверенным, что этот ключ надежен. В конечном счете А должен сам решить, насколько надежной для него является сторона, выступающая в роли поставщика.

Использование степеней доверия

Хотя в системе PGP не выдвигается никаких требований в отношении выбора уполномоченных центров сертификации и степеней доверия, PGP предлагает удобные средства использования степеней доверия, связывания степеней доверия с открытыми ключами и информацию об использовании степеней доверия.

Базовая схема выглядит следующим образом. Любой элемент в связке открытых ключей является сертификатом открытого ключа. С каждым таким элементом связывается поле соответствия ключа, которое задает степень доверия, с которой PGP будет считать, что истинным владельцем данного открытого ключа является указанный пользователь: чем выше степень доверия, тем сильнее привязка идентификатора пользователя к данному ключу. Это поле вычисляется PGP. С каждым элементом связывается также некоторое (возможно, нулевое) число подписей для данного сертификата, которые были собраны владельцем связки ключей. В свою очередь, с каждой подписью связывается поле доверия подписи, определяющее степень, в которой PGP доверяет данному объекту подписывать сертификаты открытых ключей. Значение поля соответствия ключа выводится из совокупности значений полей доверия подписи для данного элемента связки ключей. Наконец, каждый элемент определяет открытый ключ, связываемый с конкретным владельцем, а соответствующее поле доверия владельцу указывает степень доверия, с которой этот открытый ключ может использоваться для подписи других сертификатов открытых ключей; эта степень доверия определяется и присваивается пользователем. Значения полей доверия подписи можно рассматривать как

кэшированные копии значений полей доверия владельцу других элементов связки ключей.

Три поля, упоминаемые в предыдущем абзаце, содержатся в структуре, называемой байтом флага доверия. Содержимое этого флага доверия для каждого этих трех полей показано в табл. 10.2. Предположим, что мы имеем дело со связкой открытых ключей пользователя А. Операция определения степени доверия может быть описана следующим образом.

1. Когда А добавляет новый открытый ключ в связку открытых ключей, PGP должна присвоить значение флагу доверия, связанному с владельцем

Таблица 10.2

Содержимое байта флага доверия

(а) Степень доверия, приписываемая владельцу открытого ключа (размещается после пакета информации о ключе, определяется пользователем)	(б) Степень доверия, приписываемая паре "открытый ключ/идентификатор пользователя" (размещается после идентификатора пользователя, вычисляется PGP)	(в) Степень доверия, приписываемая подписи (размещается после пакета подписей, кэшированная копия значения поля OWNERTRUST для данного поставщика подписи)
Поле OWNERTRUST — неопределенное доверие — неизвестный пользователь — минимальный уровень доверия для подписи — средний уровень доверия для подписи — максимальный уровень доверия для подписи — данный ключ присутствует в связке секретных ключей (наивысшее доверие) Бит BUCKSTOP — устанавливается, если данный ключ присутствует в связке секретных ключей	Поле KEYLEGIT — неизвестное или неопределенное соответствие — ненадежное соответствие владельцу ключа — минимально надежное соответствие владельцу ключа — полное соответствие владельцу ключа Бит WARNONLY — устанавливается, если пользователь желает получить только предупреждение, когда для шифрования используется не вполне подтвержденный ключ	Поле SIGTRUST — неопределенное доверие — неизвестный пользователь — минимальный уровень доверия для подписи — средний уровень доверия для подписи — максимальный уровень доверия для подписи — данный ключ присутствует в связке секретных ключей (наивысшее доверие) Бит CONTIG — устанавливается, если подпись восходит по непрерывной цепочке надежных сертификатов к владельцу связки ключей с наивысшим доверием

этого открытого ключа. Если владельцем является А, и поэтому этот открытый ключ должен появиться также и в связке личных ключей, то полю доверия владельцу автоматически присваивается значение *наивысшее доверие (ultimate trust)*. Иначе PGP спрашивает пользователя А о том, какой уровень доверия следует приписать владельцу этого ключа и А должен ввести подходящее значение. Пользователь может указать, что этот владелец неизвестен, ненадежен, минимально надежен или вполне надежен.

2. Когда добавляется новый открытый ключ, к нему можно добавить одну или несколько подписей. Позже можно включить и другие подписи. Когда добавляется подпись, PGP выполняет поиск в связке открытых ключей, чтобы выяснить, значится ли имя автора этой подписи среди известных владельцев открытых ключей. Если да, то значение поля OWNERTRUST этого владельца присваивается полю SIGTRUST данной подписи. В противном случае соответствующему полю присваивается значение *неизвестный пользователь*.

3. Значение поля соответствия ключа вычисляется на базе значений полей доверия подписей данного элемента связки. Если по крайней мере одна подпись имеет значение *наивысшее (ultimate)* в поле доверия подписи, то в поле соответствия ключа устанавливается значение *полное (complete)*. Иначе PGP вычисляет взвешенную сумму значений полей доверия. Для подписей с максимальным уровнем доверия назначается вес 1/X, а подписям со средним уровнем доверия назначается вес 1/Y, где X и Y являются параметрами, задаваемыми пользователем. Если общая сумма весов поставщиков комбинаций "ключ/идентификатор пользователя" достигает 1, то считается, что соответствие надежно и для поля соответствия ключа устанавливается значение *полное (complete)*. Таким образом, при отсутствии наивысшего доверия для полного соответствия потребуется по крайней мере X подписей с максимальным уровнем доверия, или Y подписей со средним уровнем доверия, или некоторая подходящая их комбинация.

Периодически PGP выполняет проверку связки открытых ключей, чтобы поддерживать согласованность. По существу, это нисходящий процесс. Для каж-

дого поля OWNERTRUST при такой проверке PGP просматривает связку, находит все подписи, автором которых является данный владелец, и обновляет значения полей SIGTRUST, чтобы эти значения соответствовали значению поля OWNERTRUST. Весь процесс начинается с ключей, для которых указано наивысшее доверие. После этого значения всех полей KEYLEGIT пересчитываются на базе имеющихся подписей.

На рис. 7 показана примерная схема связывания доверия подписи и соответствия ключа. На рисунке отображена структура связки открытых ключей. В данном случае пользователь получил несколько открытых ключей, какие-то непосредственно от их владельцев, а какие-то от третьей стороны, например с сервера ключей.

Вершина, обозначенная на рисунке "Вы", представляет элемент связки открытых ключей, соответствующий данному пользователю. Этот ключ, очевидно, соответствует владельцу, поэтому значением поля OWNERTRUST является *наивысшее доверие.* Для любой другой вершины поле OWNERTRUST в связке ключей имеет значение *неопределенное доверие,* если только пользователем не задано некоторое другое значение. В данном примере пользователь указал, что он всегда доверяет подписывать другие ключи пользователям D, E, F и L. Частичное доверие подписывать другие ключи получили пользователи A и B.

Таким образом, закраска или отсутствие таковой для вершин на рис. 10.7 указывает уровень доверия, определенного для этих пользователей. Древовидная структура говорит о том, какими пользователями были подписаны соответствующие ключи. Если ключ был подписан пользователем, чей ключ также присутствует в данной связке ключей, от подписанного ключа к подписавшему данный ключ пользователю идет стрелка. Если ключ подписан пользователем, ключа которого в данной связке ключей нет, стрелка идет от подписанного ключа к знаку вопроса, означающему, что подписавшая ключ сторона данному пользователю неизвестна.

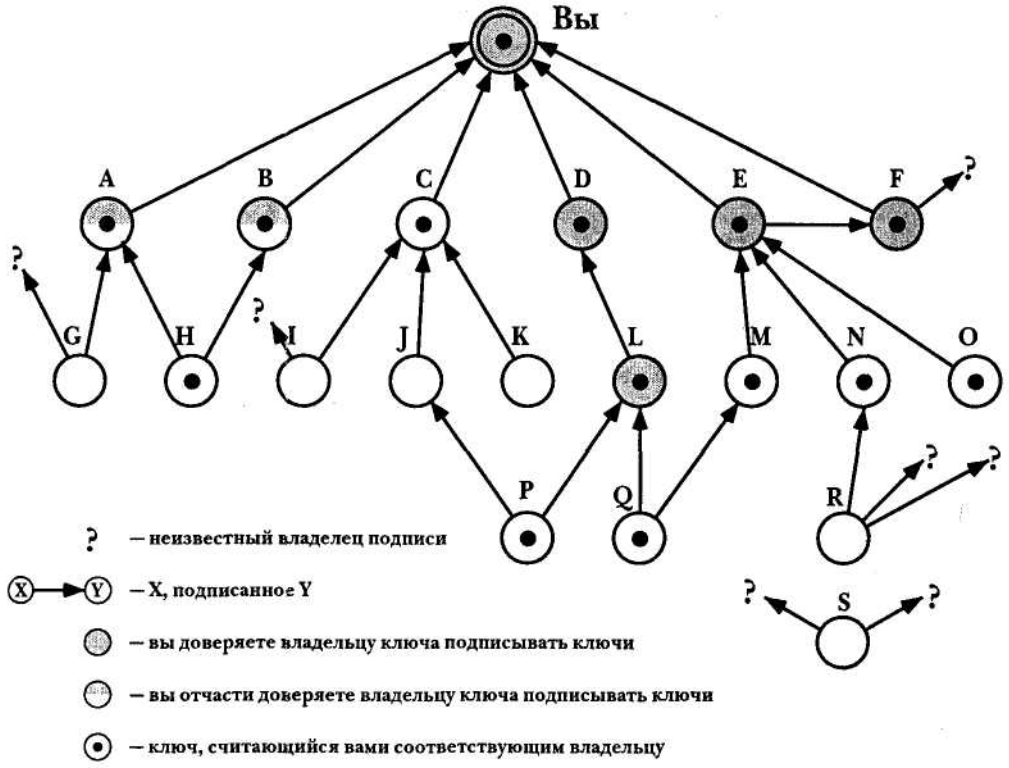


Рисунок 10.7 - Пример модели доверия PGP

На рис. 10.7 проиллюстрированы следующие моменты.

1. Обратите внимание на то, что все ключи, владельцам которых полностью или частично доверяет данный пользователь, были подписаны этим пользователем, за исключением вершины L. Такая подпись пользователя не всегда необходима, как здесь это имеет место для вершины L, но на практике большинство пользователей, скорее всего, подпишут ключи большинства владельцев, которым они доверяют. Поэтому, например, даже если ключ E уже был подписан надежным поставщиком F, пользователь решил подписать ключ E лично.

2. Мы предполагаем, что двух отчасти надежных подписей достаточно для того, чтобы сертифицировать ключ. Следовательно, ключ пользователя H расценивается системой PGP как надежный (т.е. соответствующий владельцу), ввиду того, что он подписан пользователями A и B, которым данный пользователь отчасти доверяет.

3. Ключ может быть определен как надежный, если он подписан одной вполне надежной или двумя частично надежными сторонами, но может оказаться, что пользователю этого ключа не доверяется подписывать другие ключи. Например, ключ N является надежным, поскольку он подписан стороной E, которой данный пользователь доверяет, но подписывать другие ключи стороне N не доверяется, поскольку данный пользователь не присвоил N соответствующее значение уровня доверия. Поэтому, хотя ключ R и подписан стороной N, система PGP не считает ключ R надежным. Такая ситуация имеет совершенно определенный смысл. Если вы хотите послать секретное сообщение некоторому адресату, совсем не обязательно, чтобы вы доверяли этому адресату хоть в какой-то степени. Необходимо только, чтобы вы были уверены в том, что имеете надежный открытый ключ соответствующего пользователя.

4. На рис. 7 показан также пример отдельной "вершины-сироты" S, с двумя неизвестными подписями. Такой ключ мог быть получен с сервера ключей. PGP не может считать этот ключ надежным только потому, что этот ключ пришел с имеющего хорошую репутацию сервера. Пользователь должен объявить этот ключ надежным, подписав его лично или сообщив PGP о том, что он готов полностью доверять одной из сторон, уже подписавших данный ключ.

В качестве резюме можно сказать следующее. Выше уже отмечалось, что с одним открытым ключом в связке открытых ключей можно связать несколько идентификаторов пользователей. Это может иметь место, например, в том случае, когда некоторая сторона изменила свое имя или выступает посредством подписей под многими именами, указывая для себя, например, разные адреса электронной почты. Так что открытый ключ можно рассматривать как корень некоторого дерева. Открытый ключ имеет некоторое число связанных с ним идентификаторов пользователей с рядом подписей, ассоциированных с каждым из этих идентификаторов. Привязка конкретного идентификатора пользователя к ключу зависит от подписей, связываемых с этим идентификатором пользователя, так что степень доверия данному ключу (для использования этого ключа в

целях подписания других ключей) оказывается функцией всех соответствующих подписей.

Отзыв открытых ключей

Пользователь может отменить действие своего текущего открытого ключа либо потому, что имеет подозрение в том, что ключ скомпрометирован, либо просто для того, чтобы избежать использования одного и того же ключа в течение слишком долгого времени. Обратите внимание на то, что для компрометации ключа требуется, чтобы противник каким-либо образом получил экземпляр вашего личного ключа в открытом виде или чтобы он получил как личный ключ из вашей связки личных ключей, так и вашу фразу-пароль.

Отзыв открытого ключа пользователя осуществляется в форме выпуска сертификата отмены ключа, подписанного владельцем данного ключа. Этот сертификат имеет такую же форму, как и обычный сертификат подписи, но включает индикатор, указывающий на то, что назначением данного сертификата является отмена соответствующего открытого ключа. Заметьте также, что для подписи отменяющего открытый ключ сертификата должен использоваться соответствующий личный ключ. Владелец должен попытаться распространить этот сертификат как можно шире и как можно быстрее, чтобы дать потенциальным корреспондентам возможность изменить их связки открытых ключей.

Следует учитывать, что противник, скомпрометировавший личный ключ владельца, может тоже выпустить такой сертификат. Однако это действие приведет к отрицанию принадлежности ключа противнику точно так же, как и законному владельцу открытого ключа, и поэтому эта угроза кажется намного меньшей, чем злонамеренное использование похищенного личного ключа.

Сжатие данных с помощью ZIP

В PGP используется пакет сжатия данных, называемый ZIP, авторами которого являются Жан-луп Галли (Jean-loup Gailly), Марк Адлер (Mark Adler) и Ричард Уэлз (Richard Wales). ZIP является свободно распространяемым пакетом, написанным на языке C, выполняемым как утилита на UNIX и в некоторых других системах. ZIP функционально равноценен PKZIP, широко доступному условно бесплатному пакету для систем под управлением Windows, разработанному PKWARE, Inc. Алгоритм ZIP обеспечивает, возможно, наиболее часто используемую технику сжатия данных, позволяя межплатформенный обмен данными: бесплатные и условно бесплатные версии ZIP доступны для Macintosh и других систем так же, как для Windows и UNIX.

Алгоритм ZIP и ему подобные появились в результате исследований Джейкоба Зива (Jacob Ziv) и Абрахама Лемпела (Abraham Lempel). В 1977 году они описали технологию, основанную на использовании буфера скользящего окна, содержащего текст, обработка которого выполнялась последней. Этот алгоритм обычно называют LZ77. Версия именно такого алгоритма используется в схеме сжатия ZIP (PKZIP, gzip, zipit и т.д.).

Алгоритм LZ77 и его варианты основаны на том факте, что слова и фразы внутри потока текста (или структуры изображения в случае GIF), вероятнее всего, повторяются. Когда это имеет место, повторная последовательность может быть заменена коротким кодом. Программа сжатия находит такие повторения и строит коды прямо по ходу выполнения, чтобы заменить повторную последовательность. В дальнейшем коды применяются повторно, чтобы обработать новые последовательности. Алгоритм должен быть определен таким образом, чтобы программа декомпрессии данных могла построить правильное отображение кодов в последовательности исходных данных.

Перед тем как приступить к детальному описанию LZ77, рассмотрим простой пример. Возьмем бессмысленную фразу

the brown fox jumped over the brown foxy jumping frog,

длина которой равна 53 октетам (байтам), или 424 битам. Алгоритм обрабатывает этот текст слева направо. Сначала каждый символ отображается в 9-битовый двоичный код, складывающийся из двоичной единицы, далее следует 8-битовый ASCII-код символа. В ходе дальнейшего выполнения алгоритм ищет повторяющиеся последовательности. Когда встречается повторение, алгоритм продолжает сканирование до конца повторяющейся последовательности. Другими словами, каждый раз, когда встречается повторение, алгоритм включает в повторяющуюся последовательность столько символов, сколько максимально возможно. Здесь первой найденной последовательностью является the brown fox. Эта последовательность заменяется указателем на предыдущую последовательность и данными о длине последовательности. В данном случае встретившаяся выше последовательность the brown fox находится на 26 символов раньше и длина этой последовательности равна 13 символам. Для данного примера выберем два варианта кодирования: 8-битовый указатель и 4-битовое значение длины или 12-битовый указатель и 6-битовое значение длины; 2-битовый заголовок указывает, какой вариант был выбран: значение 00 обозначает первый вариант, а 01 — второй. Таким образом, второе вхождение последовательности the brown fox кодируется в виде $<00_b x 26_d><13_d>$, или 00 00011010 1101.

Оставшаяся часть сжатого сообщения складывается из буквы y, последовательности $<00_b><27_d><5_d>$, которая заменяет последовательность из символа пробела и следующих за ним символов jump, а также последовательности символов ing frog.

Соответствующее отображение сжатия представлено на рис. 10.8. Сжатое сообщение состоит из 35 9-битовых символов и двух кодов, в сумме это 35x9 + 2 x 14 = 343 бита. В сравнении с 424 битами несжатого сообщения это дает коэффициент сжатия, равный 1,24.

Алгоритм сжатия

Алгоритм сжатия для схемы LZ77 и его варианты используют два буфера. **Скользящий буфер предыстории** содержит N символов источника, обработанных последними, а **буфер упреждающей выборки** содержит следующие L символов,

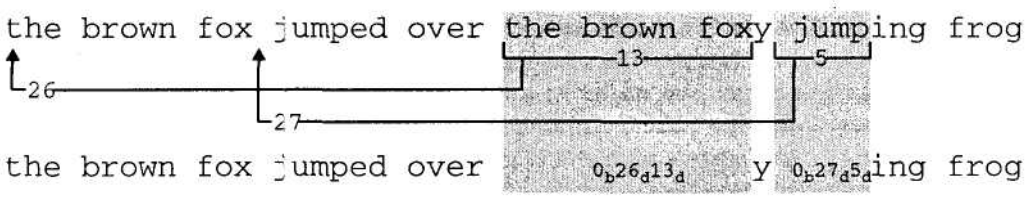


Рисунок 10.8 - Пример использования схемы LZ77

которые должны обрабатываться следующими (рис. 10.9(а)). Алгоритм пытается найти два или большее число символов из буфера упреждающей выборки в строке из скользящего буфера предыстории. Если совпадений не найдено, первый символ из буфера упреждающей выборки выводится как 9-битовый символ, сам этот символ перемещается в скользящее окно, а самый старый символ из этого окна выталкивается. Если совпадение обнаружено, алгоритм продолжает просматривать символы в поиске совпадающей последовательности наибольшей длины. Затем совпадающая строка выводится в виде трех значений (индикатор, указатель, длина). Для строки из K символов самые старые K символов из скользящего окна выталкиваются, а K символов кодированной строки сдвигаются в это окно.

На рис. 10.9(б) показано действие этой схемы на последовательности из нашего примера. На иллюстрации изображено 39-символьное скользящее окно и 13-символьный буфер упреждающей выборки. В верхней части иллюстрации уже обработано 40 первых символов и последние 39 из них в несжатом виде находятся в скользящем окне. Остальная часть данных источника находится в буфере упреждающей выборки. Алгоритм сжатия определяет следующее повторение символов, перемещает пять символов из буфера упреждающей выборки в скользящее окно и выводит код соответствующей строки. Состояние буфера после этих действий показано в нижней части иллюстрации.

Схема LZ77 является эффективной и адаптирующейся к природе вводимых данных, и, тем не менее, она имеет определенные недостатки. Алгоритм использует ограниченное окно для поиска совпадений в предыдущем тексте. Для очень длинных блоков текста в сравнении с размерами окна много потенциальных совпадений будет проигнорировано. Размер окна может быть увеличен, но за это придется платить следующим: (1) увеличением времени выполнения алгоритма ввиду того, что необходимо выполнять сравнения строк из буфера упреждающей выборки с каждой позицией в скользящем окне и (2) увеличением длины поля <указатель> ввиду необходимости указывать более длинные переходы.

Алгоритм декомпрессии

Распаковка сжатого по схеме LZ77 текста выполняется просто. Алгоритм декомпрессии должен сохранять последние *N* символов восстановленного вывода. Когда встречается закодированная строка, алгоритм декомпрессии использует значения полей <указатель> и <длина>, чтобы заменить код реальной строкой текста.

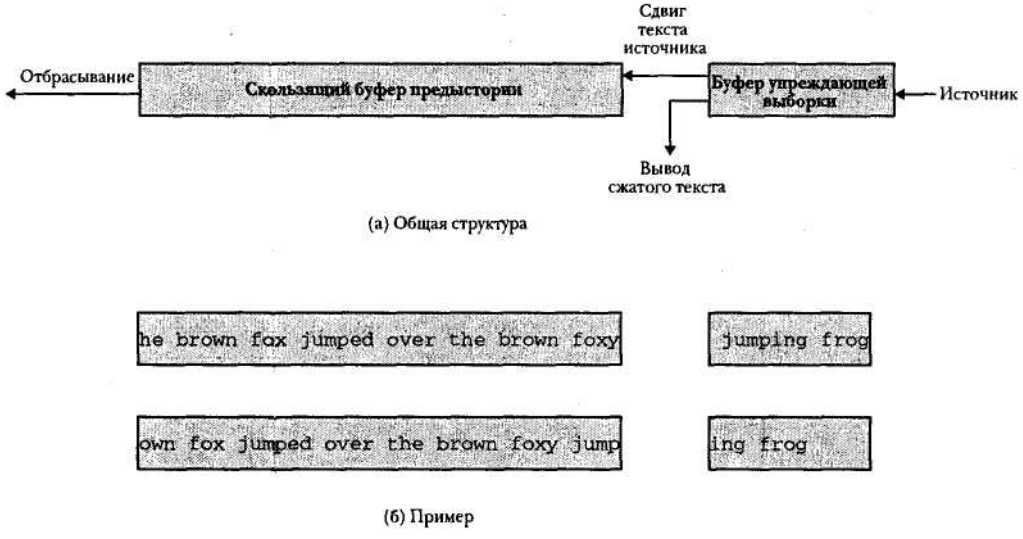


Рисунок 10.9 - Схема LZ77

Преобразование в формат radix-64

Как PGP, так и S/MIME применяется техника кодирования, называемая преобразованием radix-64. Эта техника позволяет отобразить вводимые произвольные двоичные данные в виде последовательности печатаемых символов. Данная форма кодирования имеет следующие характеристики.

1. Областью значений функции является набор символов, которые отличаются универсальной формой представления, а не специальная двоичная кодировка для этого набора символов. Таким образом, эти символы могут быть закодированы в любую форму, требуемую конкретной системой. На пример, символ "E" представляется в системе на базе ASCII как шестнадцатеричное 45, а в системе на базе EBCDIC — как шестнадцатеричное C5.

2. Этот набор символов складывается из 65 печатаемых символов, один из которых выступает в качестве заполнителя. С доступными при этом 2^6=64 символами каждый символ может использоваться для представления 6 битов данных ввода.

3. Никакие управляющие символы во множество не включаются. Таким образом, кодированное в формат radix-64 сообщение может беспрепятственно пройти системы почтовой обработки, просматривающие поток данных в поиске управляющих символов.

4. Символ дефиса ("-") не используется. Этот символ имеет особое значение в формате RFC 822, и поэтому его следует избегать.

В табл. 3 показано отображение 6-битовых вводных значений в символы. Набор символов складываются из буквенно-цифровых символов, а также символов "+" и "/". Символ "=" служит в качестве символа заполнителя.

Таблица 10.3

Кодирование Radix-64

6-битовое значение	Символ кодирования	6-битовое значение	Символ кодирования	6-битовое значение	Символ кодирования	6-битовое значение	Символ кодирования
0	A	16	Q	32	g	48	w
1	B	17	R	33	h	49	x
2	C	18	S	34	я	50	y
3	D	19	T	35	J	51	z
4	E	20	U	36	k	52	0
5	F	21	V	37	l	53	1
6	G	22	W	38	m	54	2
7	H	23	X	39	n	55	3
8	I	24	Y	40	o	56	4
9	J	25	Z	41	p	57	5
10	K	26	a	42	q	58	6
11	L	27	b	43	r	59	7
12	M	28	c	44	s	60	8
13	N	29	d	45	t	61	9
14	O	30	e	46	u	62	+
15	P	31	f	47	v	63	/
						(заполнитель)	=

На рис. 10.10 показана простая схема отображения. Двоичный ввод обрабатывается блоками по 3 октета, или 24 бита. Каждый набор из 6 битов в 24-битовом блоке отображается в символ. На рисунке символы представлены закодированными в виде 8-битовых величин. В таком типичном случае каждые 24 бита ввода расширяются до 32 битов вывода.

Для примера рассмотрим 24-битовую текстовую последовательность 00100011 01011100 10010001, которая может быть выражена в шестнадцатеричном формате как 235C91. Разобьем эту последовательность на блоки по 6 битов.

001000 110101 110010 010001

Выделенными 6-битовыми значениями в десятичном виде являются 8, 53, 50, 17. Находим кодировку этих значений в формате radix-64: IlyR. Если эти

символы сохранить в 8-битовом формате ASCII с разрядом четности, равным нулю, получим

01001001 00110001 01111001 01010010.

В шестнадцатеричном представлении это 49317952. Подводя итог, получаем следующее.

Входные данные	
Двоичное представление	00100011 01011100 10010001
Шестнадцатеричное представление	235C91
Входные данные в формате Radix-64	
Символьное представление	I1yR
Коды ASCII (8 битов, нулевой бит четности)	01001001 00110001 01111001 01010010
Шестнадцатеричное представление	49317952

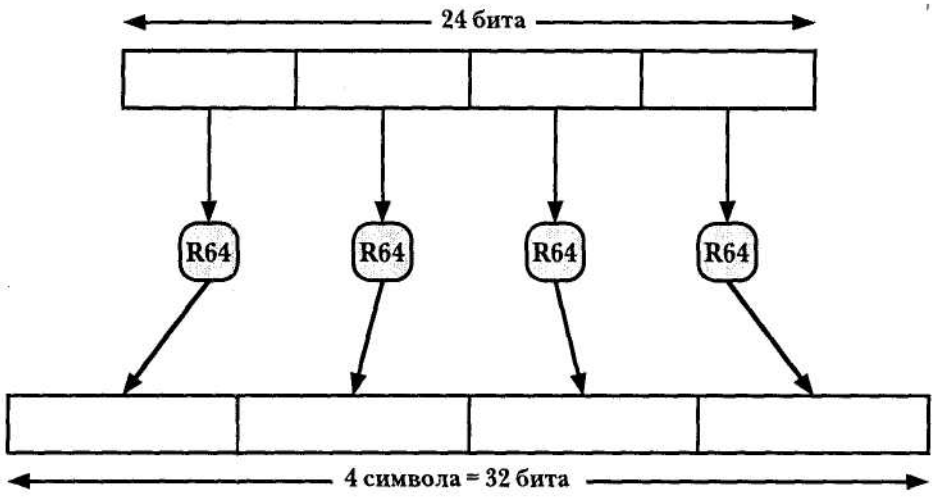


Рисунок 10.10 - Кодирование двоичных данных в виде печатаемых символов в формате radix 64

Генерирование случайных чисел PGP

В PGP используется сложная и мощная схема генерирования случайных и псевдослучайных чисел. PGP генерирует случайные числа на основе содержимого и на основе интервалов между нажатиями клавиш пользователем, а также псевдослучайные числа с помощью алгоритма, в основу которого положен алгоритм из документа ANSI X12.17. PGP использует эти числа в следующих целях.

- Истинно случайные числа

• применяются при создании пар ключей RSA,

• обеспечивают начальные значения для генератора псевдослучайных чисел,

• обеспечивают дополнительный ввод в процессе генерирования псевдослучайных чисел.

■ Псевдослучайные номера

• применяются при создании сеансовых ключей,

• служат для создания векторов инициализации (IV), используемых с сеансовыми ключами при шифровании в режиме CFB.

Истинно случайные числа

PGP поддерживает 256-байтовый буфер случайных битов. Все время PGP ожидает нажатия клавиш пользователем, отразив в 32-битовом формате момент, с которого началось ожидание. Когда нажимается клавиша, записывается время нажатия клавиши и 8-битовое значение нажатой клавиши. Информация о времени нажатия и клавише применяется при генерировании ключа, который, в свою очередь, служит для шифрования текущего значения из буфера случайных битов.

Псевдослучайные числа

При генерировании псевдослучайных чисел используется 24-байтовое начальное значение и создается 16-байтовый сеансовый ключ, 8-байтовый вектор инициализации и новое начальное значение, которые предполагается использовать для получения следующего псевдослучайного числа. Алгоритм строится на основе алгоритме X12.17, но использует для шифрования CAST-128 вместо "тройного" DES. Алгоритм задействует следующие структуры данных.

1. Ввод.

• randseed.bin (24 октета). Если этот файл пуст, он заполняется 24 истинно случайными октетами.

• Сообщение. Сеансовый ключ и IV, которые используются для шифрования сообщения, являются функциями этого сообщения. Это вносит до-

полнительную случайность для ключа и IV, но если противник уже знает содержимое сообщения в виде открытого текста, ему нет никакой необходимости выяснять значение сеансового ключа.

2. Вывод.

• К (24 октета). Первые 16 октетов, К[0..15], содержат сеансовый ключ, а последние восемь октетов, К[16 .. 23], включают значение IV.

• randseed.bin (24 октета). В этом файле размещается новое начальное значение для генератора псевдослучайных чисел.

3. Внутренние структуры данных.

• dtbuf (8 октетов). Первые четыре октета, dtbuf [0 .. 3], инициализируются с помощью текущего значения даты-времени. Этот буфер эквивалентен переменной DT из алгоритма X12.17.

• rkey (16 октетов). Ключ шифрования CAST-128, действующий на всех стадиях алгоритма.

• rseed (8 октетов). Эквивалент переменной V, из алгоритма XI2.17.

• rbuf (8 октетов). Псевдослучайное число, генерируемое алгоритмом. Это буфер эквивалентен переменной R, из алгоритма X12.17.

• К' (24 октета). Временный буфер для нового значения randseed.bin.

Алгоритм состоит из девяти шагов. Первый и последний шаги призваны уменьшить долю файла randseed.bin, которая может быть перехвачена противником. Остальные шаги, по существу, эквивалентны трем итерациям алгоритма X12.17 и иллюстрируются на рис. 10.11.

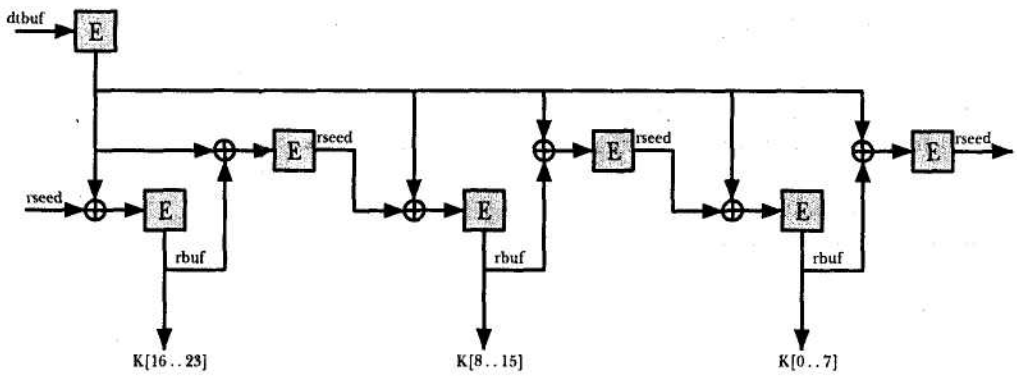


Рисунок 10.11 - Генерирование сеансового ключа и вектора инициализации PGP (шаги G2-G8 алгоритма)

Следующее пошаговое описание алгоритма соответствует описанию, предложенному Стефаном Ньюхаусом (Stephan Neuhaus).

1. [Дооперационная обработка начального значения.]

• randseed.bin копируется в К[0 .. 23].

• Хэш-код сообщения (он уже имеется, если сообщение подписано, иначе используется 4К первых октетов сообщения), служит в качестве ключа, вводится нулевое значение IV, и К шифруется в режиме CFB; результат сохраняется в К.

2. [Установка начального значения.]

• Для dtbuf[O..3] устанавливается значение, равное 32-битовому значению текущего локального времени. Значение dtbuf[4..7] обнуляется. Копируется rkey «- К [0.. 15]. Копируется rseed <*r*- К [16.. 23].

• 64-битовое значение dtbuf шифруется с использованием 128-битового значения rkey в режиме ECB; результат сохраняется в dtbuf.

3. [Подготовка к генерированию случайных октетов.] Устанавливается rcount <— О и **к** <— 23. Циклическое повторение шагов G4-G7 будет выполнено 24 раза (для **к** = 23 ... 0), и при каждом выполнении будет получен случайный октет, помещаемый в К. Переменная rcount представляет число еще неиспользованных случайных октетов в rbuf. Ее значение трижды уменьшается от 8 до 0, чтобы в результате было получено 24 октета.

4. [Доступны ли еще байты?] Если rcount = 0, следует перейти к шагу G5, в противном случае — к шагу G7. Шаги G5 и G6 представляют однократное выполнение алгоритма X12.17, порождающего новый набор из восьми случайных октетов.

5. **[Генерирование новых случайных октетов.]**

- rseed <— rseed ⊕ dtbuf .
- rbuf <- E_{rkey}, [rseed] в режиме ECB.

6. [Генерирование следующего начального значения.]

- rseed «— rseed ⊕ dtbuf .
- rbuf <*r*- E_{rkey}[rseed] в режиме ECB.
- Устанавливается rcount <— 8.

7. [Перенос по одному **байту из rbuf в К.]**

- Устанавливается rcount <— rcount -1.
- Генерируется истинно случайный байт b и устанавливается К[к] <- rbuf[rcount]0b.

8. [Готово?] Если **к** = 0, следует перейти к шагу G9, в противном случае установить **к** <- **к** -1 и перейти к шагу **G4.**

9. [Послеоперационная обработка начального значения и возвращение результата.]

- Генерируется еще 24 байта методом, представленным шагами G4-G7, но связывания с помощью операции XOR со случайным значением в G7 не производится. Результат помещается в буфер К'.
- К' шифруется в режиме CFB с ключом К [0.. 15] и вектором инициализации К [16 .. 23]; результат сохраняется в randseed.bin.
- Возвращается К.

Определить сеансовый ключ из 24 новых октетов, генерируемых на шаге G9a, должно быть невозможно. Однако чтобы гарантировать, что сохраненный файл randseed.bin не даст информации о последнем сеансовом ключе, шифруется 24 новых октета и результат сохраняется как новое начальное значение для генератора псевдослучайных чисел.

Этот тщательно разработанный алгоритм должен порождать криптографически надежные псевдослучайные числа. Анализ алгоритма показывает, что в нем нет внутренних зависимостей между битами сеансового ключа и что последовательные сеансовые ключи тоже являются независимыми.

Порядок выполнения работы

1. Ознакомиться с программным средством PGP для электронной почты и приложений хранения файлов.
2. Провести сравнительный анализ совместимости электронной почты и PGP.
3. Изучить криптографические ключи и связки ключей в PGP.
4. Решить предложенные преподавателем задачи.

Содержание отчета

В отчете необходимо привести:

1. Теоретические сведения.
2. Полное описание компонент системы PGP.
3. Подробное изложение решения задач приведенных в практической работе.
4. Выводы по работе.

Задачи:

1. В PGP используется режим шифрованной обратной связи (CFB) алгоритма CAST-128, тогда как большинство других приложений шифрования (отличных от приложений шифрования ключей) действует в режиме сцепления шифрованных блоков (CBC). Мы имеем CBC: $C, = E_K[C« \Phi P_(]$; $P, = C_w © D_K[C_(]$; CFB: $C_f = P, \Phi E_K[C_M]$; $P, = C, \Phi E\text{^}Cц]$. Оба варианта, кажется, обеспечивают одинаковую защиту. Предложите объяснение, почему в PGP используется режим CFB.

2. Какое ожидаемое число ключей в схеме PGP будет создано до того, как будет сгенерирован уже созданный ранее сеансовый ключ?

3. Чему равна вероятность того, что в схеме PGP у пользователя с *N* открытыми ключами идентификаторы по крайней мере двух ключей совпадут?

4. Первые 16 битов 128-битового профиля сообщения в подписи PGP пересылаются в открытом виде.

- В какой мере это компрометирует защиту алгоритма хэширования?
- В какой мере это в действительности выполняет свою функцию — а именно помогает определить, соответствующий ли ключ RSA использовался для того, чтобы дешифровать профиль сообщения?

5. На рис. 10.4 каждая запись в связке открытых ключей содержит поле доверия владельцу, значение которого указываемое степень доверия, оказываемого этому владельцу открытого ключа. Почему этого недостаточно? Иными словами, если этот владелец надежен и предполагается, что данный открытый ключ принадлежит этому владельцу, то почему этого не достаточно, чтобы сразу разрешить PGP использовать эту открытый ключ?

6. Насколько эффективным является алгоритм radix-64 с точки зрения криптоанализа при рассмотрении преобразования radix-64 как формы шифрования, когда нет никаких ключей, но противник знает только о том, что для шифрования английского текста применен некоторый алгоритм замены?

Литература

1. Соколов А.В., Шаньгин В.Ф. Защита информации в распределенных корпоративных сетях и системах. – М.: ДМК Пресс, 2002.
2. Вильям Столлингс Криптографическая защита сетей. – М.: Издательсткий дом “Вильямс”, 2001.
3. Молдовян А.А., Молдовян Н.А., Советов Б.Я. Криптография. – СПб.: Лань, 2000.

9 Жельников В. Криптография от папируса до компьютера. – М.: ABF, 1996.

10 Фомичев В.М. Симметричные криптосистемы. Краткий обзор основ криптологии для шифросистем с секретным ключом. – М.: Издательство МИФИ, 1995.

4. Молдовян А.А., Молдовян Н.А., Советов Б.Я. Криптография - СПб.: Издательство «Лань»,2000.-224 с.
5. Девянин П.Н., Михальский О.О. и др. Теоретические основы компьютерной безопасности. - М.: Радио и связь, 2000.-192 с.
6. Домашев А.В., Грунтович М.М., Попов В.О. Программирование алгоритмов защиты информации. – М.: Издательство “Нолидж”, 2002.

Лабораторная работа № 5
Разграничение доступа в ОС Novell Netware

Цель работы: На основе печатной и электронной литературы по Novell NetWare и с использованием базы знаний предыдущих дисциплин по информационной безопасности разобраться в принципе разграничения доступа в ОС Novell Netware и ответить на поставленные вопросы реферативно.

Основные темы:

1. Novell NetWare в корпоративной сети. Типовая корпоративная сеть. Основные требования к автоматизированным системам. Уровни информационной инфраструктуры корпоративной сети.

2. Уязвимости и атаки. Источники возникновения уязвимостей. Классификация уязвимостей по уровню в инфраструктуре информационной системы. Классификация уязвимостей по степени риска. Классификация атак по целям. Классификация атак по мотивации действий. Механизмы реализации атак. Источники информации об уязвимостях. CVE.

3. Защитные механизмы и средства. Идентификация и аутентификация пользователей, разграничение доступа пользователей к ресурсам автоматизированной системы, криптографические методы защиты информации, контроль целостности, защита периметра компьютерных сетей. Средства обеспечения информационной безопасности: межсетевые экраны, средства анализа защищенности, средства обнаружения атак, средства защиты информации от несанкционированного доступа.

4. Критерии оценки защищённости ОС. Уровень защиты С2. Common Criteria. British Standard (BS 7799). Государственная система защиты информации в России.

5. Установка и настройка Novell NetWare. Требования к аппаратному обеспечению сервера Novell NetWare 5.1. Установка сервера Novell NetWare

5.1. Установка клиентских частей на рабочие станции сети и утилиты NetWare Administrator на АРМ администратора безопасности.

6. Управление NDS. Понятие об NDS и Bindery. Схема NDS, классы и объекты. Свойства объектов и права на объекты NDS. Планирование дерева каталогов. Разделы NDS. Реплики. Утилиты управления разделами и диагностики NDS. Советы Novell по избежанию проблем с репликами. Утилиты для обеспечения надежности серверов.

7. Управление пользователями и группами в дереве NDS. Создание пользователей и групп. Шаблоны и организационные роли. Настройка требований к паролям пользователей. Разграничение прав на объекты NDS. Присвоение пользователям полномочий по доступу к объектам NDS и ресурсам файловой системы. Сценарии регистрации пользователя.

8. Аудит в системах NetWare. Регистрационные журналы на серверах. Аудит объектов NDS.

9. Настройка безопасности в сетях NetWare. Механизм защиты от взломщика. Защита от подделки пакетов. Настройки объекта Public. Уязвимые сервисные функции NetWare, которые следует отключить. Рекомендации независимых исследователей.

10. Защита серверов и рабочих станций. Физическая защита серверов и кабельной сети. Безопасность рабочих станций и межмашинных соединений. Дополнительные программно-аппаратные средства защиты информации от несанкционированного доступа, сертифицированные в Российской Федерации.

Порядок выполнения работы

1. Ознакомиться с литературой по сетям NetWare.
2. Разобраться с настройкой безопасности в сетях NetWare.
3. Ответы на контрольные вопросы.

Содержание отчета

В отчете необходимо привести:

1. Теоретические сведения.
2. Подробное изложение тем, приведенных выше.
3. Выводы по работе.

Литература

1. Юджин Х., Спаффорд «Основы безопасности компьютерных систем», HackZone, 1999 г.
2. Громов В.В., Васильев В.А. «Энциклопедия компьютерной безопасности» (сборник). М.: 1999 г.
3. Соколов А.В., Шаньгин В.Ф. Защита информации в распределенных корпоративных сетях и системах.- ДМК Пресс, 2002.-656 с.
4. Романец Ю.Ф., Тимофеев П.А., Шаньгин В.Ф. Защита информации в компьютерных системах и сетях./ Под ред. В.Ф. Шаньгина.-М: Радио и связь, 2001 – 376 с.

Практические работы

Цель работы: Представить реферат (10-15 стр.) или перевод обзорной статьи по выбранной теме. Результаты подготовки сформировать в доклад продолжительностью 15-20 минут на одном из практических занятий. Хронология занятий и график выступлений соответствуют порядку, приведенному преподавателем.

Перечень тем для выполнения практических работ по курсу «Программно-аппаратная защита информации»

1. Уязвимость компьютерных систем:
   - Уязвимость "переполнение буфера" (buffer overflow). Общая характеристика, разновидности и причины возникновения.
   - Методы защиты от уязвимости "переполнение буфера".
   - Диверсификация (diversification) компьютерных систем для повышения их надежности и защищенности.
   - Защита путем внесения случайностей (рандомизация) в код, процесс выполнения программы, адреса памяти; использование случайного внутреннего кода.
   - Уязвимость типа "race condition" ("состояние гонок"). Характеристика, разновидности и причины возникновения, методы защиты.
   - Целенаправленное использование "случайных ошибок" (сбоев памяти, ошибок чтения и т.п.).
2. Идентификация пользователей КС — субъектов доступа к данным.
3. Основные подходы к защите данных от НСД.
4. Организация доступа к файлам.
5. Особенности защиты данных от изменения.
6. Построение программно-аппаратных комплексов шифрования.

7. Плата Криптон-3 (Криптон-4).
8. Защита программ от несанкционированного копирования.
9. Организация хранения ключей.
10. Защита программ от изучения.
11. Вирусы.
12. Устройства и системы технической разведки. Противодействие коммерческой разведке с помощью технических средств.
13. Примеры построения систем сетевой безопасности. Решения компании Cisco Systems по обеспечению безопасности корпоративных сетей. Продукты и решения компаний «Элвис плюс», НИП «Информзащита», «Анкад», «Инфотекс», «S-Terra CSP»
14. Вычислительная сеть как составная часть ЗКС. Сетевые уязвимости, угрозы и атаки:
    - Защищенная сетевая инфраструктура и ее основные элементы на разных уровнях.
    - Сетевые ОС как элемент ЗКС в ВС. Современные сетевые ОС (Windows, Unix/Linux, Netware) с точки зрения безопасности и защиты.
    - Основные уязвимости нижних уровней стека протоколов TCP/IP (уровень сетевого доступа и межсетевой уровень).
    - Основные уязвимости верхних уровней стека протоколов TCP/IP (транспортный уровень и прикладной уровень).
    - Атаки типа "Denial-of-Service" ("отказ в обслуживании"). Характеристика, разновидности и причины возникновения, методы защиты.
    - Интернет-вирусы и черви. Механизмы функционирования и распространения. Методы защиты (помимо антивирусного ПО).
15. Новые и особенные подходы к проектированию и разработке ЗКС:

- Защищенные платформы и ядра (NGSCS, PSOS), технология "доверенных вычислений" ("trusted computing").
- Защита систем электронной коммерции. Основные уязвимости, угрозы и варианты защиты.
- Методы защиты мобильного кода. Proof-carrying code ("код с внутренней гарантией).
- Аспекты безопасности и гарантии в готовых системах и системах с открытым исходным кодом (open source). Характеристика и сравнение.
- Социальный фактор в защите компьютерных систем.
- Основные подходы к защите многоагентных систем.

Структура и оформление практической работы

Реферат должен содержать следующее: титульный лист, содержание, списки условных обозначений и сокращений. Основные разделы и подразделы представленного материала, перечень использованной литературы.

Работа должна быть оформлена в соответствии с требованиями стандартов ЕСКД.

Литература

1. Коул Э. Руководство по защите от хакеров. Пер. с англ. - М. Изд. Дом «Вильямс», 2002.
2. Анин Б.Ю. Защита компьютерной информации. - СПб.: БХВ-Санкт-Петербург,2000.—384 с.
3. Молдовян А.А., Молдовян Н.А., Советов Б.Я. Криптография - СПб.: Издательство «Лань»,2000.-224 с.
4. Девянин П.Н., Михальский О.О. и др. Теоретические основы компьютерной безопасности. - М.: Радио и связь, 2000.-192 с.

5. Романец Ю.Ф., Тимофеев П.А., Шаньгин В.Ф. Защита информации в компьютерных системах и сетях./ Под ред. В.Ф. Шаньгина.-М: Радио и связь, 2001 – 376 с.
6. Соколов А.В., Шаньгин В.Ф. Защита информации в распределенных корпоративных сетях и системах.- ДМК Пресс, 2002.-656 с.
7. Галицкий А.В., Рябко С.Д., Шаньгин В.Ф. Защита информации в сети – - анализ технологий и синтез решений – М.: ДМК Пресс, 2004.-616 с.

11. КОНТРОЛЬНО-ИЗМЕРИТЕЛЬНЫЕ МАТЕРИАЛЫ

УЯЗВИМОСТЬ КОМПЬЮТЕРНЫХ СИСТЕМ

1) Под угрозой безопасности информации в компьютерной системе (КС) понимают:

a) возможность возникновения на каком-либо этапе жизненного цикла КС такого ее состояния, при котором создаются условия для реализации угроз безопасности информации.

b) событие или действие, которое может вызвать изменение функционирования КС, связанное с нарушением защищенности обрабатываемой в ней информации.

c) действие, предпринимаемое нарушителем, которое заключается в поиске и использовании той или иной уязвимости.

2) Уязвимость информации — это:

a) возможность возникновения на каком-либо этапе жизненного цикла КС такого ее состояния, при котором создаются условия для реализации угроз безопасности информации.

b) событие или действие, которое может вызвать изменение функционирования КС, связанное с нарушением защищенности обрабатываемой в ней информации.

c) это действие, предпринимаемое нарушителем, которое заключается в поиске и использовании той или иной уязвимости.

3) Атакой на КС называют:

a) возможность возникновения на каком-либо этапе жизненного цикла КС такого ее состояния, при котором создаются условия для реализации угроз безопасности информации.

b) событие или действие, которое может вызвать изменение функционирования КС, связанное с нарушением защищенности обрабатываемой в ней информации.

c) действие, предпринимаемое нарушителем, которое заключается в поиске и использовании той или иной уязвимости.

4) Искусственные угрозы исходя из их мотивов разделяются на:

a) непреднамеренные и преднамеренные

b) косвенные и непосредственные

c) несанкционированные и сонкцианированные

5) К непреднамеренным угрозам относятся:

a) ошибки в разработке программных средств КС

b) несанкционированный доступ к ресурсам КС со стороны пользователей КС и посторонних лиц, ущерб от которого определяется полученными нарушителем полномочиями.

c) угроза нарушения конфиденциальности, т.е. утечки информации ограниченного доступа, хранящейся в КС или передаваемой от одной КС к другой;

6) К умышленным угрозам относятся:

a) несанкционированные действия обслуживающего персонала КС (например, ослабление политики безопасности администратором, отвечающим за безопасность КС);

b) воздействие на аппаратные средства КС физических полей других электронных устройств (при несоблюдении условий их электромагнитной совместимости) и др.

c) ошибки пользователей КС;

7) Косвенными каналами утечки называют:

a) каналы, не связанные с физическим доступом к элементам КС
b) каналы, связанные с физическим доступом к элементам КС
c) каналы, связанные с изменение элементов КС и ее структуры.

8)К косвенным каналам утечки информации относятся:
а) использование подслушивающих (радиозакладных) устройств;
б) маскировка под других пользователей путем похищения их идентифицирующей информации (паролей, карт и т.п.);
в) злоумышленное изменение программ для выполнения ими несанкционированного копирования информации при ее обработке;

9)Непосредственными каналами утечки называют:
a) каналы, связанные с физическим доступом к элементам КС.
b) каналы, не связанные с физическим доступом к элементам КС
c) каналы, связанные с изменение элементов КС и ее структуры.

10)К непосредственным каналам утески информации относятся:
a) обход средств разграничения доступа к информационным ресурсам вследствие недостатков в их программном обеспечении и др.
b) перехват побочных электромагнитных излучений и наводок (ПЭМИН).
c) дистанционное видеонаблюдение;

11) Избирательная политика безопасности подразумевает, что:
a) права доступа субъекта к объекту системы определяются на основании некоторого внешнего (по отношению к системе) правила (свойство избирательности).
b) все субъекты и объекты системы должны быть однозначно идентифицированы;
c) каждому объекту системы присвоена метка критичности,

определяющая ценность содержащейся в нем информации;

12) Полномочная политика безопасности подразумевает, что:

a) каждому субъекту системы присвоен уровень прозрачности (security clearance), определяющий максимальное значение метки критичности объектов, к которым субъект имеет доступ.

b) все субъекты и объекты системы должны быть идентифицированы;

c) права доступа субъекта к объекту системы определяются на основании некоторого внешнего (по отношению к системе) правила (свойство избирательности).

13) Достоверная вычислительная база - это:

a) абстрактное понятие, обозначающее полностью защищенный механизм вычислительной системы (включая аппаратные и программные средства), отвечающий за поддержку реализации политики безопасности.

b) активный компонент системы, который может явиться причиной потока информации от объекта к объекту или изменения состояния системы.

c) пассивный компонент системы, хранящий, принимающий или передающий информацию.

14) Достоверная вычислительная база выполняет задачи:

a) поддерживает реализацию политики безопасности и является гарантом целостности механизмов защиты

b) функционирует на фоне избирательной политики, придавая ее требованиям иерархически упорядоченный характер (в соответствии с уровнями безопасности)

c) представляет собой некоторый набор требований, прошедших соответствующую проверку, реализуемых при помощи организационных мер

15) Уязвимость информации — это:

a) возможность возникновения на каком-либо этапе жизненного цикла КС такого ее состояния, при котором создаются условия для реализации угроз безопасности информации.
b) набор документированных норм, правил и практических приемов, регулирующих управление, защиту и распределение информации ограниченного доступа.
c) неизменность информации в условиях ее случайного и (или) преднамеренного искажения или разрушения.

ИДЕНТИФИКАЦИЯ ПОЛЬЗОВАТЕЛЕЙ КС-СУБЪЕКТОВ ДОСТУПА К ДАННЫМ

1) Идентификация объекта – это:

a) одна из функций подсистемы защиты.
b) взаимное установление подлинности объектов, связывающихся между собой по линиям связи.
c) сфера действий пользователя и доступные ему ресурсы КС

2) Процедуру установки сфер действия пользователя и доступные ему ресурсы КС называют:

a) авторизацией
b) аутентификацией
c) Идентификация

3) Авторизация – это:

a) предоставлением полномочий

b) подтверждение подлинности

c) цифровая подпись

4) Аутентификация – это:

a) подтверждение подлинности

b) предоставлением полномочий

c) цифровая подпись

5) Для проведения процедур идентификации и аутентификации пользователя необходимо:

a) наличие соответствующего субъекта (модуля) аутентификации;

b) наличие аутентифицирующего объекта, хранящего уникальную информацию

c) ответы a) и b)

6) Биометрическая идентификация и аутентификация пользователя это:

a) идентификация потенциального пользователя путем измерения физиологических параметров и характеристик человека, особенностей его поведения.

b) схема идентификации позволяющая увеличить число аккредитаций, выполняемых за один цикл, и тем самым уменьшить длительность процесса идентификации.

c) схема идентификации с нулевой передачей знаний.

7)Для чего используется процедура “рукопожатия”:

a) для взаимной проверки подлинности

b) для распределения ключей между подлинными партнерами

c) для безопасного использования интеллектуальных карт

8) Параллельная схема идентификации позволяет увеличить:

a) число аккредитаций, выполняемых за один цикл, и тем самым уменьшить длительность процесса идентификации.
b) регистрацию времени для каждого сообщения
c) объект-эталон для идентификации и аутентификации пользователей

9)Какие существуют формы представления объектов, аутентифицирующих пользователя:
a) внешний аутентифицирующий объект, не принадлежащий системе;
b) внутренний объект, принадлежащий системе, в который переносится информация из внешнего объекта.
c) варианты a) и b)

10) Внешняя и внутренняя формы представления аутентифицирующего объекта должны быть:
a) семантически тождественны
b) модифицированы
c) структурированы

11) Внешние объекты могут быть технически реализованы на различных носителях информации?
a) да
b) нет
c) Не знаю

12) Для чего были разработаны протоколы идентификации с нулевой передачей знаний:
a) для безопасного использования интеллектуальных карт
b) для взаимной проверки подлинности
c) для распределения ключей между подлинными партнерами

13) Механизм запроса-ответа используется для:

a) проверки подлинности

b) шифрования

c) регистрации времени для каждого сообщения

14) Кто разработал алгоритм идентификации с нулевой передачей знания:

a) Гиллоу и Ж. Куискуотером

b) У. Фейге

c) А. Фиат и А. Шамир

15) Схему идентификации с нулевой передачей знаний предложили:

a) У. Фейге, А. Фиат и А. Шамир

b) Гиллоу и Ж. Куискуотером

c) А. Фиат и А. Шамир

ЗАЩИТА ИНФОРМАЦИИ В КС ОТ НЕСАНКЦИОНИРОВАННОГО ДОСТУПА

1) Для чего создается система разграничения доступа к информации:

a) для защиты информации от НСД

b) для осуществления НСДИ

c) определения максимального уровня конфиденциальности документа

2) Сбои, отказы технических и программных средств могут быть использованы для НСД?

a) да

b) нет

c) незнаю

3)Какие методы организации разграничения доступа используются в КС:

a) матричный

b) структурированный
c) метод Гиллоу-Куискуотера

4)Мандатный метод основывается на:
a) многоуровневой модели защиты
b) использование матриц доступа
c) криптографическом преобразовании

5)Какой из функциональных блоков должна содержать система разграничения доступа к информации:
a) блок криптографического преобразования информации при ее хранении и передаче;
b) блок контроля среды размещения
c) блок контроля среды выполнения.

6) Диспетчер доступа реализуется в виде:
a) аппаратно-программных механизмов
b) аппаратных механизмов
c) программных механизмов

7) Под ядром безопасности понимают:
a) локализованную, минимизированную, четко ограниченную и надежно изолированную совокупность программно-аппаратных механизмов, доказательно правильно реализующих функции диспетчера доступа.
b) сведения о лицах, предметах, фактах, событиях, явлениях и процессах независимо от формы их представления.
c) событие или действие, которое может вызвать изменение функционирования КС, связанное с нарушением защищенности обрабатываемой в ней информации.

8) Главным условием создания ядра безопасности является:

a) обеспечение многоуровневого режима выполнения команд

b) мандатное управление

c) Матричная структура

9) Под организацией доступа к ресурсам понимается

a) весь комплекс мер, который выполняется в процессе эксплуатации КС для предотвращения несанкционированного воздействия на технические и программные средства, а также на информацию.

b) хранения атрибутов системы защиты, поддержки криптографического закрытия информации, обработки сбоев и отказов и некоторые другие.

c) предотвращение несанкционированного перехода пользовательских процессов в привилегированное состояние

10) При эксплуатации механизмов аутентификации основными задачами являются:

a) генерация или изготовление идентификаторов, их учет и хранение, передача идентификаторов пользователю и контроль над правильностью выполнения процедур аутентификации в КС.

b) разграничение прав пользователей и обслуживающего персонала по доступу к ресурсам КС в соответствии с функциональными обязанностями должностных лиц;

c) реализация механизма виртуальной памяти с разделением адресных пространств;

11)В чем заключается правило разграничения доступа

a) лицо допускается к работе с документом только в том случае, если уровень допуска субъекта доступа равен или выше уровня конфиденциальности документа, а в наборе категорий, присвоенных

данному субъекту доступа, содержатся все категории, определенные для данного документа.

b) лицо допускается к работе с документом только в том случае, если уровень допуска субъекта доступа ниже уровня конфиденциальности документа, а в наборе категорий, присвоенных данному субъекту доступа, содержатся все категории, определенные для данного документа.

c) лицо допускается к работе с документом только в том случае, если уровень допуска субъекта доступа ниже уровня конфиденциальности документа, а в наборе категорий, присвоенных данному субъекту доступа,не содержатся все категории, определенные для данного документа.

12) Правильность функционирования ядра безопасности доказывается путем:

a) полной формальной верификации его программ и пошаговым доказательством их соответствия выбранной математической модели защиты.

b) использования дополнительных программных или аппаратно-программных средств.

c) использования строго определенного множества программ.

13) Мандатное управление позволяет упростить процесс регулирования доступа?

a) Да

b) Нет

c) Не знаю

14) Матричное управление доступом предполагает использование:

a) матриц доступа

b) аппаратно-программных механизмов

c) субъекта допуска

15) Основной проблемой создания высокоэффективной защиты от НСД является

a) предотвращение несанкционированного перехода пользовательских процессов в привилегированное состояние.
b) использования дополнительных программных или аппаратно-программных средств.
c) разграничение прав пользователей и обслуживающего персонала по доступу к ресурсам КС в соответствии с функциональными обязанностями должностных лиц

АППАРАТНО-ПРОГРАММНЫЕ СРЕДСТВА КРИПТОГРАФИЧЕСКОЙ ЗАЩИТЫ ИНФОРМАЦИИ

1)Аппаратно-программмные средства криптографической защиты информации выполняют функции:

a) аутентификацию пользователя, разграничение доступа к информации, обеспечение целостности информации и ее защиты от уничтожения, шифрование и электронную цифровую подпись.
b) организовывают реализацию политики безопасности информации на этапе эксплуатации КС.
c) проверяют на отсутствие закладок приборов, устройств.

2) Надежность защиты информации в компьютерной системе определяется:

a) конкретным перечнем и свойствами функций КС;
b) используемыми в функциях КС методами;
c) варианты a) и b)

3) Использование аппаратных средств снимает проблему:

a) обеспечения целостности системы.

b) разграничение прав пользователей и обслуживающего персонала по доступу к ресурсам КС в соответствии с функциональными обязанностями должностных лиц

c) использования строго определенного множества программ.

4)Криптографические функции плат КРИПТОН образующие ядро системы безопасности реализуются

a) аппаратно

b) программно

c) аппаратно и программно

5)К частично контролируемым компьютерным системам можно отнести современные КС, использующие

a) ОС Windows 95/98, Windows NT, различные версии UNIX

b) Windows NT, Windows XP

c) различные версии UNIX

6) Безопасность в частично контролируемым компьютерных системамах может быть обеспечена

a) изоляцией от злоумышленника ненадежной компьютерной среды, отдельного ее компонента или отдельного процесса с помощью полностью контролируемых средств.

b) схемой идентификации позволяющая увеличить число аккредитаций, выполняемых за один цикл, и тем самым уменьшить длительность процесса идентификации.

c) внешней аутентификацией объекта, не принадлежащего системе;

7) Платы серии КРИПТОН, обеспечивают защиту:

a) ключей шифрования и электронной цифровой подписи (ЭЦП), так и неизменность их алгоритмов.

b) аппаратно-программных механизмов

c) реализации механизма виртуальной памяти с разделением адресных пространств;

8) К основным компонентам сети относятся:

a) центры коммутации пакетов, маршрутизаторы, шлюзы и сетевые экраны;

b) субъекты доступа

c) платы серии КРИПТОН

9) В качестве ключевых носителей устройств криптографической защиты данных серии КРИПТОН используются:

a) дискеты, смарт-карты и Touch-Memory.

b) смарт-карты, Touch-Memory

c) дискеты, смарт-карты

10)Средства серии КРИПТОН независимо от операционной среды обеспечивают:

a) защиту ключей шифрования и электронной цифровой подписи (ЭЦП) и неизменность алгоритма шифрования и ЭЦП.

b) криптомаршрутизацию

c) функции шифрования и электронной цифровой подписи.

11) В системе Secret Disk используется:

a) смешанная программно-аппаратная схема защиты с возможностью выбора

b) реализация механизма виртуальной памяти с разделением адресных пространств;

c) механизм RUN-файлов позволяет в процессе работы запускать любые программы с предварительной проверкой их целостности.

12) В чем заключается особенность системы Secret Disk:

a) для доступа к защищенной информации необходим не только вводимый пользователем пароль, но и электронный идентификатор.

b) для доступа к защищенной информации необходим только вводимый пользователем пароль.

c) для доступа к защищенной информации необходим только электронный идентификатор.

13) Мастер-ключ в Устройствах криптографической защиты данных серии КРИПТОН загружается:

a) до загрузки операционной системы

b) после загрузки операционной системы

c) вообще не загружается

14) Криптографических функций в устройствах криптографической защиты данных серии КРИПТОН выполняются:

a) внутри платы

b) в операционной системе

c) в блоке загрузки операционной системы

15) Абонентские места, персональные компьютеры или терминалы клиента являются основными компонентами сети?

a) Да

b) Нет

c) Не знаю

МЕТОДЫ И СРЕДСТВА ОГРАНИЧЕНИЯ ДОСТУПА К КОМПОНЕНТАМ ЭВМ

1) Под защитой информации понимается

a) совокупность мероприятий, методов и средств, обеспечивающих решение следующих задач по проверке целостности информации и исключении несанкционированного доступа к ресурсам ПЭВМ и хранящимся в ней программам и данным.

b) совокупность мероприятий, методов и средств, обеспечивающих решение следующих задач по реализации механизма виртуальной памяти с разделением адресных пространств;

c) совокупность мероприятий, методов и средств, обеспечивающих решение следующих задач по разграничению прав пользователей и обслуживающего персонала.

2) Возможные каналы утечки информации по классификации разделяют:

a) человек, аппаратура, программа

b) человек, линия связи

c) коммутационное оборудование, человек

3)К группе каналов утечки информации в которой основным средством является человек, относятся следующие утечки:

a) расшифровка программой зашифрованной информации;

b) несанкционированный доступ программы к информации;

c) копирование программой информации с носителей.

4) К группе каналов утечки информации в которой основным средством является аппаратура, относятся следующие утечки:

a) подключение к ПЭВМ специально разработанных аппаратных средств, обеспечивающих доступ к информации;

b) хищение носителей информации (магнитных дисков, дискет, лент)

c) копирование программой информации с носителей

5) К группе каналов утечки информации в которой основным средством является программа, относятся следующие утечки:

a) несанкционированный доступ программы к информации
b) хищение носителей информации (магнитных дисков, дискет, лент)
c) использование специальных технических средств для перехвата электромагнитных излучений технических средств ПЭВМ.

6) К средствам активной защиты относятся:

a) искаженные программы (программы вирусы, искажение функций)
b) заказное проектирование
c) специальная аппаратура

7) К средствам пассивной защиты относятся:

a) частотный анализ
b) авторская эстетика
c) аппаратура защиты (ПЗУ, преобразователи)

8) К средствам собственной защиты относятся:

a) машинный код
b) сигнатура
c) корреляционный анализ

9) Может ли информативный сигнал в сети электропитания быть каналом утечки информации?

a) Да
b) Нет
c) Не знаю

10) Мероприятия по инженерно-технической защите информации от утечки по электромагнитному каналу подразделяются на:

a) организационные и технические
b) технические и коммутационные
c) организационные и объективные

11) Технические мероприятия направлены :
a) на недопущение выхода информативного сигнала за пределы контролируемой территории с помощью сертифицированных технических средств защиты.
b) на использование специальных технических средств для перехвата электромагнитных излучений технических средств ПЭВМ.
c) на защиту ключей шифрования и электронной цифровой подписи (ЭЦП) и неизменность алгоритма шифрования и ЭЦП.

12) Организационными мероприятиями предусматривается
a) исключение нахождения в местах наличия информативного сигнала злоумышленника и контроль за его действиями и передвижением
b) исключение значительной части загрузочных модулей из сферы их досягаемости.
c) исключение несанкционированного доступа к ресурсам ПЭВМ и хранящимся в ней программам и данным

13) Активные способы защиты информации при ее утечке через сеть электропитания направленные на:
a) создание маскирующего шума
b) перехвата информации
c) минимизацию паразитных связей внутри ПЭВМ

14) Пассивные способы защиты информации при ее утечке через сеть электропитания направленные на
a) минимизацию паразитных связей внутри ПЭВМ

b) создание маскирующего шума

c) перехвата информации

15) Для минимизации паразитных связей внутри ПЭВМ используются

a) радиоэкранирующие и радиопоглощающие материалы

b) двигатели-генераторы

c) разомкнутые линии

ЗАЩИТА ПРОГРАММ ОТ НЕСАНКЦИОНИРОВАННОГО КОПИРОВАНИЯ

1) Под системой защиты от несанкционированного использования и копирования понимается

a) комплекс программных или программно-аппаратных средств, предназначенных для усложнения или запрещения нелегального распространения, использования и (или) изменения программных продуктов и иных информационных ресурсов.

b) комплекс аппаратных и программных средств, предназначенных для автоматизированного сбора, хранения, обработки, передачи и получения информации.

c) комплекс правовых норм, организационных мер, технических, программных и криптографических средств, обеспечивающий защищенность информации в КС в соответствии с принятой политикой безопасности.

2) Под надежностью системы защиты от несанкционированного копирования понимается:

a) способность противостоять попыткам изучения алгоритма ее работы и обхода реализованных в нем методов защиты.

b) способность систем с открытыми ключами генерировать цифровые подписи, обеспечивающие различные функции защиты, компенсирует избыточность требуемых вычислений.

c) способность к самостоятельному внедрению в тела других программ и последующему самовоспроизведению и самораспространению в информационно-вычислительных сетях и отдельных ЭВМ

3) Методы, затрудняющие считывание скопированной информации основываются на

a) придании особенностей процессу записи информации, которые не позволяют считывать полученную копию на других накопителях, не входящих в защищаемую КС

b) разграничении прав пользователей и обслуживающего персонала по доступу к ресурсам КС в соответствии с функциональными обязанностями должностных лиц

c) использования дополнительных программных или аппаратно-программных средств.

4) Для защиты от несанкционированного использования программ могут применяться электронные ключи?

a) да

b) нет

c) не знаю

5) Мероприятия по инженерно-технической защите информации от утечки по электромагнитному каналу подразделяются на:

a) организационные и технические

b) технические и коммутационные

c) организационные и объективные

УПРАВЛЕНИЕ КРИПТОГРАФИЧЕСКИМИ КЛЮЧАМИ

1) Любая криптографическая система основана на использовании:

a) криптографических ключей
b) разомкнутых линии
c) односторонних функций

2) В симметричной криптосистеме отправитель и получатель сообщения используют

a) один и тот же секретный ключ
b) разные секретных ключи
c) вообще не используют секретных ключей

3) Асимметричная криптосистема предполагает использование

a) двух ключей открытого и личного (секретного)
b) системы разграничения доступа
c) переносных носителей для хранения секретной информации

4) Под ключевой информацией понимают:

a) совокупность всех действующих в АСОИ ключей
b) совокупность документов и массивов документов и информационных технологий, реализующих информационные процессы.
c) совокупность свойств, обусловливающих пригодность информации удовлетворять определенные потребности ее пользователей в соответствии с назначением информации.

5)Какая из функций не входит в процесс управления ключами?

a) переадресация ключей
b) генерация ключей
c) распределение ключей

6) Модификация ключа – это

a) генерирование нового ключа из предыдущего значения ключа с помощью односторонней (однонаправленной) функции.

b) генерирование нового ключа из последующего значения ключа с помощью односторонней (однонаправленной) функции.

c) генерирование нового ключа из предыдущего значения ключа с помощью двусторонней (двунаправленной) функции.

7) Под функцией хранения ключей понимают

a) организацию их безопасного хранения, учета и удаления.

b) организацию их генерации, учета и удаления.

c) организацию их безопасного хранения, учета и сопостовления.

8) Механизм отметки времени позволяет каждому субъекту сети определить:

a) насколько старо пришедшее сообщение, и отвергнуть его, если появится сомнение в его подлинности.

b) были ли внесены изменения в файл.

c) какие информационные потоки в системе являются "легальными", то есть не ведут к утечке информации

9) Модель рукопожатия применяется для:

a) проверки подлинности партнеров

b) для симметричных криптосистем с секретными ключами

c) для асимметричных криптосистем с открытыми ключами

10)Каким из перечисленных способов не реализуется Распределение ключей между пользователями компьютерной сети:

a) документирование алгоритмов обеспечения защиты информации

b) использованием одного или нескольких центров распределения ключей

c) прямым обменом сеансовыми ключами между пользователями сети

11) Задача распределения ключей сводится к

a) построению протокола распределения ключей

b) взаимному подтверждению подлинности участников сеанса

c) использование минимального числа сообщений при обмене ключами

12) Протокол Kerberos основывается на

a) симметричной криптографии

b) ассиметричной криптографии

c) нескольких центров распределения ключей

13) Первым алгоритмом с открытыми ключами был алгоритм:

a) Диффи-Хеллмана

b) А. Фиата

c) А. Шамира

14) SKIP Протокол управления:

a) криптоключами

b) защищенного канала

c) симметричной криптосистемой

15) Метод Диффи-Хеллмана дает возможность шифровать данные при каждом сеансе связи на новых ключах?

a) Да

b) Нет

c) Не знаю

ЗАЩИТА ПРОГРАММНЫХ СРЕДСТВ ОТ ИССЛЕДОВАНИЯ

1) В каких режимах может выполняться изучение логики работы программы:

a) статическом

b) динамическом

c) и в статическом и в динамическом

2) Сущность статического режима заключается

a) в изучении исходного текста программы

b) в выполнение трассировки программы

c) в использование самогенерирующих кодов

3) Динамический режим изучения алгоритма программы предполагает

a) выполнение трассировки программы

b) изучении исходного текста программы

c) использование самогенерирующих кодов

4) Средства противодействия дизассемблированию могут защитить программу от трассировки?

a) Нет

b) Да

c) Не знаю

5) Какой метод может противодействовать дизассемблированию

a) шифрование

b) хэширование

c) изучение

6) Сущность метода, основанного на использовании самогенерируемых кодов, заключается в том что

a) исполняемые коды программы получаются самой программой в процессе ее выполнения.

b) исполняемые коды программы получаются самой программой после процесса ее выполнения.

c) исполняемые коды программы получаются самой программой до процесса ее выполнения.

7) Трассировка программ обычно осуществляется с помощью:

a) программных продуктов, называемых отладчиками
b) шифрования
c) самогенерируемых кодов

8) Под компьютерным вирусом понимается:

a) автономно функционирующая программа, обладающая способностью к самостоятельному внедрению в тела других программ и последующему самовоспроизведению и самораспространению в информационно-вычислительных сетях и отдельных ЭВМ.
b) программа имеющая доступ к файлам системы, и имеющая возможность работать с процессами системы.
c) программа не имеющая доступ к файлам системы, и не имеющая возможность работать с процессами системы.

9) Резидентные вирусы это:

a) вирусы, которые после активизации постоянно находятся в оперативной памяти компьютера и контролируют доступ к его ресурсам;
b) вирусы, которые выполняются только в момент запуска зараженной программы.
c) вирусы, заражающие программы, хранящиеся в системных областях дисков.

10) Транзитные вирусы это:

a) вирусы, которые выполняются только в момент запуска зараженной программы.

b) вирусы, которые после активизации постоянно находятся в оперативной памяти компьютера и контролируют доступ к его ресурсам;

c) вирусы, заражающие программы, хранящиеся в системных областях дисков.

11) Вирусы-мутанты (MtE-вирусы) это

a) вирусы, содержащие в себе алгоритмы шифрования, обеспечивающие различие разных копий вируса.

b) вирусы, пытающиеся быть невидимыми на основе контроля доступа к зараженным элементам данных;

c) вирусы, заражающие программы, хранящиеся в системных областях дисков.

12) Stealth-вирусы это

a) вирусы, пытающиеся быть невидимыми на основе контроля доступа к зараженным элементам данных:

b) вирусы, содержащие в себе алгоритмы шифрования, обеспечивающие различие разных копий вируса.

c) вирусы, которые после активизации постоянно находятся в оперативной памяти компьютера и контролируют доступ к его ресурсам;

13) Загрузочные (бутовые) вирусы это:

a) вирусы, заражающие программы, хранящиеся в системных областях дисков.

b) вирусы, которые после активизации постоянно находятся в оперативной памяти компьютера и контролируют доступ к его ресурсам;

c) вирусы, содержащие в себе алгоритмы шифрования, обеспечивающие различие разных копий вируса.

14) Трянские программы это:

a) программы которые содержат скрытые последовательности команд (модули), выполняющие действия, наносящие вред пользователям.

b) программы , содержащие в себе алгоритмы шифрования, обеспечивающие различие разных копий вируса.

c) программы которые после активизации постоянно находятся в оперативной памяти компьютера и контролируют доступ к его ресурсам;

15) Файловые вирусы это:

a) вирусы, заражающие файлы с программами

b) вирусы, заражающие программы, хранящиеся в системных областях дисков.

c) вирусы, которые после активизации постоянно находятся в оперативной памяти компьютера и контролируют доступ к его ресурсам.

СПИСОК ЛИТЕРАТУРЫ

1. Герасименко В.А., Малюк А.А. Основы защиты информации. М.: МОПО РФ, МИФИ, 1997, 537 с.
2. Петраков А.В. Основы практической защиты информации. М.: Радио и связь, 1999, 368с.
3. Петраков А.В. Утечка и защита информации в телефонных каналах. М.: Энергоатомиздат. 1996. 320 с.
4. Орлов В.А., Петров В.И. Приборы наблюдения ночью и при ограниченной видимости. М.: Военное издательство, 1989.
5. Гавриш В. Практическое пособие по защите коммерческой тайны. Симферополь: “Таврида”, 1994.
6. Демин В.П., Куприянов А.И., Сахаров А.В. Радиоэлектронная разведка и радиомаскировка. М.: Изд-во МАИ, 1997, 156 с.
7. Никулин О.Ю., Петрушин А.Н. Системы телевизионного наблюдения. –М.: Оберг-РБ, 1996.
8. Поздняков Е.Н. Защита объектов. – М.: Концерн “Банковский Деловой Центр”, 1997 г. – 224 с.
9. Ярочкин В.И. Информационная безопасность. Учебной пособие для студентов непрофильных вузов. – М.: Междунар. отношения, 2000 г. – 400 с.
10. Андрианов В.И. и др. “Шпионские штучки” и устройства для защиты объектов и информации: Справочное пособие. – Лань, СПб., 1996 г. – 272 с.
11. Лагутин В.С., Петраков А.В. Утечка и защита информации в телефонных каналах. – М.: Энергоатомиздат, 1996 г. – 304 с.
12. Торокин А.А. Основы инженерно-технической защиты информации. М: “Ось-89”, 1998, 334 с.

13. Хорев А.А. Защита информации от утечки по техническим каналам. Часть 1. Технические каналы утечки информации. Учебное пособие. М.: Гостехкомиссия России, 1998, 320 с.
14. Петраков А.В., Дорошенко П.С., Савлуков Н.В. Охрана и защита современного предприятия. М: Энергоатомиздат, 1999, 568 с.
15. Абалмазов Э.И. Методы и инженерно-технические средства противодействия информационным угрозам. – М.: Изд-во “Компания “Гротек”, 1997 г. – 246 с.
16. Каторин Ю.Ф. и др. Большая энциклопедия промышленного шпионажа. – СПб.: ООО “Изд-во “Полигон”, 2000. – 896 с.
17. Демин В.П. и др. Радиоэлектронная разведка и радиомаскировка. – М.: Изд-во МАИ, 1997. – 156 с.
18. Домашев А.В., Грунтович М.М., Попов В.О. Программирование алгоритмов защиты информации. – М.: Издательство “Нолидж”, 2002.
19. Соколов А.В., Шаньгин В.Ф. Защита информации в распределенных корпоративных сетях и системах. – М.: ДМК Пресс, 2002.
20. Вильям Столлингс Криптографическая защита сетей. – М.: Издательсткий дом “Вильямс”, 2001.