Министерство образования и науки Российской Федерации

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ
(ТУСУР)

Кафедра физической электроники (ФЭ)

Н.В. Зариковская

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

Учебное пособие

2012

Зариковская Н.В.
Информационные технологии: учебное пособие. – Томск: Томский государственный университет систем управления и радиоэлектроники, 2012. – 97 с.

Содержание

«Кто владеет информацией, тот владеет миром»
Уинстон Черчилль

1. Введение

Термин «информация» происходит от латинского слова «information», что означает сведения, разъяснения, изложение.

Информация – это настолько общее и глубокое понятие, что его нельзя объяснить одной фразой. В это слово вкладывается различный смысл в технике, науке и в житейских ситуациях.

Под термином информация мы будем понимать сведения об объектах и явлениях окружающей среды, их параметрах, свойствах и состоянии, которые воспринимают информационные системы (живые организмы, управляющие машины и др.) в процессе жизнедеятельности и работы.

Информация есть характеристика не сообщения, а соотношения между сообщением и его потребителем. Без наличия потребителя, хотя бы потенциального, говорить об информации бессмысленно.

В случаях, когда говорят об автоматизированной работе с информацией посредством каких-либо технических устройств, обычно в первую очередь интересуются не содержанием сообщения, а тем, сколько символов это сообщение содержит.

Применительно к компьютерной обработке данных под информацией понимают некоторую последовательность символических обозначений (букв, цифр, закодированных графических образов и звуков и т. п.), несущую смысловую нагрузку и представленную в понятном компьютеру виде. Каждый новый символ в такой последовательности символов увеличивает информационный объём сообщения.

Информация может существовать в самых разнообразных формах:

- в виде текстов, рисунков, чертежей, фотографий;
- в виде световых или звуковых сигналов; в виде радиоволн;
- в виде электрических и нервных импульсов;
- в виде магнитных записей; в виде жестов и мимики;
- в виде запахов и вкусовых ощущений;
- в виде хромосом, посредством которых передаются по наследству признаки и свойства организмов и т. д.

Предметы, процессы, явления материального или нематериального свойства, рассматриваемые с точки зрения их информационных свойств, называются информационными объектами.

Информацию можно: создавать, передавать, воспринимать, использовать, запоминать, принимать, копировать, формализовать, распространять, преобразовывать, комбинировать, обрабатывать, делить на части, упрощать, собирать, хранить, искать, измерять, разрушать и др. Все эти процессы, связанные с определенными операциями над информацией, называются информационными процессами.

Информация обладает следующими свойствами, характеризующими ее качественные признаки: достоверность, полнота, ценность, своевременность, понятность, доступность, краткость и др.

Обработка информации – получение одних информационных объектов из других информационных объектов путем выполнения некоторых алгоритмов.

Обработка является одной из основных операций, выполняемых над информацией, и главным средством увеличения объёма и разнообразия информации.

Средства обработки информации – это всевозможные устройства и системы, созданные человечеством, и в первую очередь, компьютер – универсальная машина для обработки информации. Компьютеры обрабатывают информацию путем выполнения некоторых алгоритмов. В то время как живые организмы и растения обрабатывают информацию с помощью своих органов и систем.

2. Понятие информационной технологии

2.1. Что такое информационная технология

Под термином технология мы понимаем комплекс научных и инженерных знаний, реализованных в приемах труда, наборах материальных, технических, энергетических, трудовых факторов производства, способах их соединения для создания продукта или услуги, отвечающих определенным требованиям. Поэтому технология неразрывно связана с машинизацией производственного или непроизводственного, прежде всего управленческого, процесса. Управленческие технологии основываются на применении компьютеров и телекоммуникационной техники.

Информационная технология – это комплекс взаимосвязанных, научных, технологических, инженерных дисциплин, изучающих методы эффективной организации труда людей, занятых обработкой и хранением информации; вычислительную технику и методы организации и взаимодействия с людьми и производственным оборудованием, их практические приложения, а также связанные со всем этим социальные, экономические и культурные проблемы. Сами информационные технологии требуют сложной подготовки, больших первоначальных затрат и наукоемкой техники. Их введение должно начинаться с создания математического обеспечения, формирования информационных потоков в системах подготовки специалистов.

2.2. Этапы развития информационной технологии

Существует несколько точек зрения на развитие информационных технологий с использованием компьютеров, которые определяются различными признаками деления.

Общим для всех изложенных ниже подходов является то, что с появлением персонального компьютера начался новый этап развития информационной технологии. Основной целью становится удовлетворение персональных информационных потребностей человека, как для профессиональной сферы, так и для бытовой.

Признак деления – вид задач и процессов обработки информации

1-й этап (60–70-е гг.) – обработка данных в вычислительных центрах в режиме коллективного пользования. Основным направлением развития информационной технологии являлась автоматизация операционных рутинных действий человека.

2-й этап (с 80-х гг.) – создание информационных технологий, направленных на решение стратегических задач.

Признак деления – проблемы, стоящие на пути информатизации общества

1-й этап (до конца 60-х гг.) характеризуется проблемой обработки больших объемов данных в условиях ограниченных возможностей аппаратных средств.

2-й этап (до конца 70-х гг.) связывается с распространением ЭВМ серии IBM/360. Проблема этого этапа – отставание программного обеспечения от уровня развития аппаратных средств.

3-й этап (с начала 80-х гг.) – компьютер становится инструментом непрофессионального пользователя, а информационные

системы – средством поддержки принятия его решений. Проблемы – максимальное удовлетворение потребностей пользователя и создание соответствующего интерфейса работы в компьютерной среде.

4-й этап (с начала 90-х гг.) – создание современной технологии межорганизационных связей и информационных систем. Проблемы этого этапа весьма многочисленны. Наиболее существенными из них являются:

- выработка соглашений и установление стандартов, протоколов для компьютерной связи;
- организация доступа к стратегической информации;
- организация защиты и безопасности информации.

Признак деления – преимущество, которое приносит компьютерная технология

1-й этап (с начала 60-х гг.) характеризуется довольно эффективной обработкой информации при выполнении рутинных операций с ориентацией на централизованное коллективное использование ресурсов вычислительных центров. Основным критерием оценки эффективности создаваемых информационных систем была разница между затраченными на разработку и сэкономленными в результате внедрения средствами. Основной проблемой на этом этапе была психологическая – плохое взаимодействие пользователей, для которых создавались информационные системы, и разработчиков из-за различия их взглядов и понимания решаемых проблем. Как следствие этой проблемы, создавались системы, которые пользователи плохо воспринимали и, несмотря на их достаточно большие возможности, не использовали в полной мере.

2-й этап (с середины 70-х гг.) связан с появлением персональных компьютеров. Изменился подход к созданию информационных систем, ориентация смещается в сторону индивидуального пользователя для поддержки принимаемых им решений. Пользователь заинтересован в проводимой разработке, налаживается контакт с разработчиком, возникает взаимопонимание обеих групп специалистов. На этом этапе используется как централизованная обработка данных, характерная для первого этапа, так и децентрализованная, базирующаяся на решении локальных задач и работе с локальными базами данных на рабочем месте пользователя.

3-й этап (с начала 90-х гг.) связан с понятием анализа стратегических преимуществ в бизнесе и основан на достижениях телекоммуникационной технологии распределенной обработки информации. Информационные системы имеют своей целью не просто

увеличение эффективности обработки данных и помощь управленцу. Соответствующие информационные технологии должны помочь организации выстоять в конкурентной борьбе и получить преимущество.

Признак деления – виды инструментария технологии

1-й этап (до второй половины XIX в.) – «ручная» информационная технология, инструментарий которой составляли: перо, чернильница, книга. Коммуникации осуществлялись ручным способом путем переправки через почту писем, пакетов, депеш. Основная цель технологии – представление информации в нужной форме.

2-й этап (с конца XIX в.) – «механическая» технология, инструментарий которой составляли: пишущая машинка, телефон, диктофон, оснащенная более совершенными средствами доставки почта. Основная цель технологии – представление информации в нужной форме более удобными средствами.

3-й этап (40–60-е гг. XX в.) – «электрическая» технология, инструментарий которой составляли: большие ЭВМ и соответствующее программное обеспечение, электрические пишущие машинки, ксероксы, портативные диктофоны.

Изменяется цель технологии. Акцент в информационной технологии начинает перемещаться с формы представления информации на формирование ее содержания.

4-й этап (с начала 70-х гг.) – «электронная» технология, основным инструментарием которой становятся большие ЭВМ и создаваемые на их базе автоматизированные системы управления (АСУ) и информационно-поисковые системы (ИПС), оснащенные широким спектром базовых и специализированных программных комплексов. Центр тяжести технологии еще более смещается на формирование содержательной стороны информации для управленческой среды различных сфер общественной жизни, особенно на организацию аналитической работы. Множество объективных и субъективных факторов не позволили решить стоящие перед новой концепцией информационной технологии поставленные задачи. Однако был приобретен опыт формирования содержательной стороны управленческой информации и подготовлена профессиональная, психологическая и социальная база для перехода на новый этап развитии технологии.

5-й этап (с середины 80-х гг.) – «компьютерная» («новая») технология, основным инструментарием которой является персональный компьютер с широким спектром стандартных программ-

ных продуктов разного назначения. На этом этапе происходит процесс персонализации АСУ, который проявляется в создании систем поддержки принятия решений определенными специалистами. Подобные системы имеют встроенные элементы анализа и интеллекта для разных уровней управления, реализуются на персональном компьютере и используют телекоммуникации. В связи с переходом на микропроцессорную базу существенным изменениям подвергаются и технические средства бытового, культурного и прочего назначения.

Начинают широко использоваться в различных областях глобальные и локальные компьютерные сети.

2.3. Составляющие информационной технологии

Используемые в производственной сфере такие технологические понятия, как норма, норматив, технологический процесс, технологическая операция и т. п., могут применяться и в информационной технологии. Прежде чем разрабатывать эти понятия в любой технологии, в том числе и в информационной, всегда следует начинать с определения цели. Затем следует попытаться провести структурирование всех предполагаемых действий, приводящих к намеченной цели, и выбрать необходимый программный инструментарий.

Необходимо понимать, что освоение информационной технологии и дальнейшее ее использование должны свестись к тому, что нужно сначала хорошо овладеть набором элементарных операций, число которых ограничено. Из этого ограниченного числа элементарных операций в разных комбинациях составляется действие, а из действий, также в разных комбинациях, составляются операции, которые определяют тот или иной технологический этап. Совокупность технологических этапов образует технологический процесс (технологию). Он может начинаться с любого уровня и не включать, например, этапы или операции, а состоять только из действий. Для реализации этапов технологического процесса могут использоваться разные программные среды.

Информационная технология, как и любая другая, должна отвечать следующим требованиям:

- обеспечивать высокую степень расчленения всего процесса обработки информации на этапы (фазы), операции, действия;
- включать весь набор элементов, необходимых для достижения поставленной цели;

• иметь регулярный характер. Этапы, действия, операции технологического процесса могут быть стандартизированы и унифицированы, что позволит более эффективно осуществлять целенаправленное управление информационными процессами.

2.4. Инструментарий информационной технологии

Реализация технологического процесса материального производства осуществляется с помощью различных технических средств, к которым относятся: оборудование, станки, инструменты, конвейерные линии и т. п.

По аналогии и для информационной технологии должно быть нечто подобное. Такими техническими средствами производства информации будет являться аппаратное, программное и математическое обеспечение этого процесса. С их помощью производится переработка первичной информации в информацию нового качества. Выделим отдельно из этих средств программные продукты и назовем их инструментарием, а для большей четкости можно его конкретизировать, назвав программным инструментарием информационной технологии. Определим это понятие. В качестве инструментария можно использовать следующие распространенные виды программных продуктов для персонального компьютера: текстовый процессор (редактор), настольные издательские системы, электронные таблицы, системы управления базами данных, электронные записные книжки, электронные календари, информационные системы функционального назначения (технические, финансовые, бухгалтерские, для маркетинга и пр.) экспертные системы и т. д.

2.5. Устаревание информационной технологии

Для информационных технологий является вполне естественным то, что они устаревают и заменяются новыми. Так, например, на смену технологии пакетной обработки программ на большой ЭВМ в вычислительном центре пришла технология работы на персональном компьютере на рабочем месте пользователя. Телеграф передал все свои функции телефону. Телефон постепенно вытесняется службой экспресс-доставки. Телекс передал большинство своих функций факсу и электронной почте.

При внедрении новой информационной технологии в организации необходимо оценить риск отставания от конкурентов в

результате ее неизбежного устаревания со временем, так как информационные продукты, как никакие другие виды материальных товаров, имеют чрезвычайно высокую скорость сменяемости новыми видами или версиями.

Периоды сменяемости колеблются от нескольких месяцев до одного года. Если в процессе внедрения новой информационной технологии этому фактору не уделять должного внимания, возможно, что к моменту завершения перевода фирмы на новую информационную технологию она уже устареет и придется принимать меры к ее модернизации. Такие неудачи с внедрением информационной технологии обычно связывают с несовершенством технических средств, тогда как основной причиной неудач является отсутствие или слабая проработанность методологии использования информационной технологии.

2.6. Методология использования информационной технологии

Централизованная обработка информации на ЭВМ вычислительных центров была первой исторически сложившейся технологией. Создавались крупные вычислительные центры коллективного пользования, оснащенные большими ЭВМ (в нашей стране – ЭВМ ЕС). Применение таких ЭВМ позволяло обрабатывать большие массивы входной информации и получать на этой основе различные виды информационной продукции, которая затем передавалась пользователям. Такой технологический процесс был обусловлен недостаточным оснащением вычислительной техникой предприятий и организаций в 60–70-е гг.

Достоинства методологии централизованной технологии:

- возможность обращения пользователя к большим массивам информации в виде баз данных и к информационной продукции широкой номенклатуры;
- сравнительная легкость внедрения методологических решений по развитию и совершенствованию информационной технологии благодаря централизованному их принятию.

Недостатки такой методологии очевидны

- ограниченная ответственность низшего персонала, который не способствует оперативному получению информации пользователем, тем самым препятствуя правильности выработки управленческих решений;

• ограничение возможностей пользователя в процессе получения и использования информации.

Существуют следующие виды информационных технологий: информационная технология обработки данных; информационная технология управления; информационная технология поддержки принятия решения; информационная технология экспертных систем.

Целью нашего курса является изучение вопросов, связанных с информационной технологией обработки данных. Рассмотрим их более подробно.

3. Информационная технология обработки данных

3.1. Характеристика и назначение

Информационная технология обработки данных предназначена для решения хорошо структурированных задач, по которым имеются необходимые входные данные и известны алгоритмы и другие стандартные процедуры их обработки. Эта технология применяется на уровне операционной (исполнительской) деятельности персонала невысокой квалификации в целях автоматизации некоторых рутинных постоянно повторяющихся операций управленческого труда. Поэтому внедрение информационных технологий и систем на этом уровне существенно повысит производительность труда персонала, освободит его от рутинных операций, возможно, даже приведет к необходимости сокращения численности работников.

На уровне операционной деятельности решаются следующие задачи:

• обработка данных об операциях, производимых фирмой;

• создание периодических контрольных отчетов о состоянии дел в фирме;

• получение ответов на всевозможные текущие запросы и оформление их в виде бумажных документов или отчетов.

Примером может послужить ежедневный отчет о поступлениях и выдачах наличных средств банком, формируемый в целях контроля баланса наличных средств, или же запрос к базе данных по кадрам, который позволит получить данные о требованиях, предъявляемых к кандидатам на занятие определенной должности.

Существует несколько особенностей, связанных с обработкой данных, отличающих данную технологию от всех прочих:

• выполнение необходимых фирме задач по обработке данных. Каждой фирме предписано законом иметь и хранить данные о своей деятельности, которые можно использовать как средство обеспечения и поддержания контроля на фирме. Поэтому в любой фирме обязательно должна быть информационная система обработки данных и разработана соответствующая информационная технология;

• решение только хорошо структурированных задач, для которых можно разработать алгоритм;

• выполнение стандартных процедур обработки. Существующие стандарты определяют типовые процедуры обработки данных и предписывают их соблюдение организациями всех видов;

• выполнение основного объема работ в автоматическом режиме с минимальным участием человека;

• использование детализированных данных. Записи о деятельности фирмы имеют детальный (подробный) характер, допускающий проведение ревизий. В процессе ревизии деятельность фирмы проверяется хронологически от начала периода к его концу и от конца к началу;

• акцент на хронологию событий;

• требование минимальной помощи в решении проблем со стороны специалистов других уровней.

Хранение данных. Многие данные на уровне операционной деятельности необходимо сохранять для последующего использования либо здесь же, либо на другом уровне. Для их хранения создаются базы данных.

Создание отчетов (документов). В информационной технологии обработки данных необходимо создавать документы для руководства и работников фирмы, а также для внешних партнеров. При этом документы могут создаваться как по запросу или в связи с проведенной фирмой операцией, так и периодически в конце каждого месяца, квартала или года.

3.2. Сжатие, архивирование и хранение данных

Сжатие сокращает объем пространства, требуемого для хранения файлов в ЭВМ, и количество времени, необходимого для передачи информации по каналу установленной ширины пропускания. Это есть форма кодирования. Другими целями кодирования являются поиск и исправление ошибок, а также шифрование. Процесс поиска и исправления ошибок противоположен сжатию – он

увеличивает избыточность данных, когда их не нужно представлять в удобной для восприятия человеком форме. Удаляя из текста избыточность, сжатие способствует шифрованию, что затрудняет поиск шифра доступным для взломщика статистическим методом. Рассмотрим обратимое сжатие или сжатие без наличия помех, где первоначальный текст может быть в точности восстановлен из сжатого состояния. Необратимое или ущербное сжатие используется для цифровой записи аналоговых сигналов, таких как человеческая речь или рисунки.

Обратимое сжатие особенно важно для текстов, записанных на естественных и на искусственных языках, поскольку в этом случае ошибки обычно недопустимы. Хотя первоочередной областью применения рассматриваемых методов есть сжатие текстов, что отражает и наша терминология, однако, эта техника может найти применение и в других случаях, включая обратимое кодирование последовательностей дискретных данных.

Существует много веских причин выделять ресурсы ЭВМ в расчете на сжатое представление, т. к. более быстрая передача данных и сокращение пространства для их хранения позволяют сберечь значительные средства и зачастую улучшить показатели ЭВМ. Сжатие, вероятно, будет оставаться в сфере внимания из-за всё возрастающих объемов хранимых и передаваемых в ЭВМ данных, кроме того, его можно использовать для преодоления некоторых физических ограничений, например, сравнительно низкая ширина пропускания телефонных каналов.

3.2.1. Архивация данных в Windows

В системе Windows нет встроенных функций для работы с упакованными архивами, поэтому большинству пользователей приходится обзаводиться специальными утилитами и при этом решать проблему многообразия архивных форматов.

Емкость магнитных дисков и оперативной памяти постоянно растет, каналы передачи данных становятся все более мощными, и все же объем передаваемой и хранимой информации по-прежнему остается весьма значимым фактором, заставляющим нас использовать программные средства для работы с упакованными данными.

Для начала несколько слов о терминологии. Принято различать архивацию и упаковку (компрессию, сжатие) данных. В первом случае речь идет о слиянии нескольких файлов и даже каталогов в единый файл – архив (примером использования такой техно-

логии в чистом виде может служить формат TAR), во втором – о сокращении объема исходных файлов путем устранения избыточности (в данной работе рассматривается упаковка без потерь информации, т. е. с возможностью точного восстановления исходных файлов). Как правило, современные архиваторы обеспечивают также сжатие данных, являясь, таким образом, еще и упаковщиками, однако существуют и чисто «упаковочные» утилиты типа Gzip, сжимающие отдельные файлы, преобразуя их в формат Z или GZ.

При выборе инструмента для работы с упакованными файлами и архивами следует учитывать два фактора: эффективность, т. е. оптимальный баланс между экономией дисковой памяти и производительностью работы, и совместимость, т. е. возможность обмена данными с другими пользователями. Совместимость, пожалуй, сегодня более важна, так как по достигаемой степени сжатия конкурирующие форматы и инструменты различаются на небольшой процент в результирующем объёме файла, а вычислительная мощность современных компьютеров делает время обработки архивов не столь существенным показателем, как, скажем, десять лет назад. Поэтому при выборе инструмента для работы с архивами важнейшим критерием для большинства пользователей (во всяком случае тех, для кого обмен большими массивами данных – насущная проблема), вероятно, является способность программы «понимать» наиболее распространенные архивные форматы, даже если эти форматы не самые эффективные.

Действие большинства средств упаковки основано на использовании алгоритмов сжатия, предложенных в 80-х гг. Абрахамом Лемпелем и Якобом Зивом. Многие популярные архивные форматы (ZIP, LZH, ARJ, ARC, ICE и т. п.) появились в эпоху господства DOS. Для работы с ними использовались специализированные архиваторы-упаковщики (утилиты PKZIP/PKUNZIP, LHA, ARJ), которые позволяли архивировать целые каталоги и обеспечивали высокую степень сжатия для текстовых, графических и прочих типов файлов. Эти программы вызывались командной строкой с многочисленными параметрами, довольно громоздкими, хотя и обеспечивавшими богатые возможности. Вскоре стали появляться интегрирующие надстройки, с помощью которых можно было работать с различными форматами архивов не из командной строки, а с помощью меню.

По-настоящему прижились в мире персональных компьютеров, став сегодня фактическими стандартами, лишь немногие из старых архивных форматов – ZIP, ARJ и, пожалуй, еще LZH. По-

мимо этих традиционных форматов некоторые современные архиваторы позволяют работать с новым межплатформным форматом JAR (Java ARchive), который был создан специально для пересылки многокомпонентных Java-апплет, но может применяться и для работы с упакованными архивами общего назначения (в JAR применяются те же методы сжатия, что и в ZIP). Еще один формат, CAB, был предложен фирмой Microsoft, средства для работы с ним входят в состав Windows 9x; многие архиваторы, ориентированные на форматы ZIP и ARJ, позволяют также распаковывать CAB-архивы.

В настоящее время растет популярность формата RAR. Но, хотя технология RAR обеспечивает высокую степень сжатия, стандартом она так и не стала, отчасти из-за не очень гибкого механизма работы с большими архивами. Во многих случаях удачным решением проблемы совместимости является создание архивов в виде самораспаковывающихся программ (EXE-файлов). Многие программы, ориентированные на какой-либо из традиционных типов архивов, способны создавать и EXE-архивы на базе своего «родного» формата. Но это решение не всегда обеспечивает достаточную гибкость (например, не позволяет без специальных инструментов выборочно извлекать файлы из архива).

С приходом Windows архиваторы обзавелись графическим интерфейсом. В некоторых случаях этот интерфейс лишь прикрывал собой ту или иную старую утилиту командной строки, но появились и полноценные, в том числе 32-разрядные, программы со встроенным механизмом для манипулирования архивами (как правило, какого-нибудь одного типа; самая известная ZIP-ориентированная программа такого рода – WinZip фирмы Nico Mak Computing).

В удобном виде манипуляции с командной строкой были «спрятаны» за интерфейсом популярных в России файловых оболочек типа DISCo Commander, FAR и особенно Windows Commander начиная с версий 4.xx. Эти оболочки позволяют путем настройки файлов конфигурации подключать любые внешние DOS-архиваторы командной строки (ARJ, PKZIP, ARC, LZH и т. п.) и организовывать прозрачное манипулирование архивами, представляя их в виде обычных каталогов. К сожалению, многие утилиты командной строки неспособны полноценно работать с длинными именами файлов (такая возможность появилась в программе PKZIP лишь начиная с версии 2.5 для Windows, в ARJ – начиная с версии 3.0), а организовать обмен файлами с архивом можно только в пределах окна оболочки. Кроме того, и сами программные

оболочки, и вызываемые из их среды архиваторы командной строки – коммерческие продукты (как правило, условно-бесплатные), каждую приходится приобретать отдельно. В отличие от утилит командной строки, рассчитанных на работу с одним типом архивов, Windows-программы более универсальны с точки зрения совместимости по форматам и к тому же используют такие преимущества новой ОС, как возможность давать объектам длинные имена и переносить файлы из одного приложения в другое.

Рассмотренные программы по большей части ориентированы на работу с архивами в формате ARJ или ZIP, но, как правило, содержат встроенные средства (или допускают подключение внешних модулей) для распаковки и просмотра архивов других типов. В общем, тесты показывают, что программы, ориентированные на формат ARJ (их, кстати, не так много), в среднем работают чуть быстрее аналогичных ZIP-архиваторов и к тому же обеспечивают больший коэффициент сжатия, однако архиватор, несовместимый с форматом ZIP, вряд ли можно сегодня считать полноценным инструментом. Все программы обладают удобными инсталляторами и стандартными средствами деинсталляции. Как правило, архиваторы могут выборочно регистрироваться в качестве средства для обработки распознаваемых ими типов файлов. Практически все архиваторы предусматривают работу с длинными именами объектов, однако если эти имена содержат русские буквы, то 16-разрядные программы их неузнаваемо искажают при упаковке. Наиболее удобные утилиты интегрируются в систему Windows 9x: позволяют упаковывать и распаковывать файлы с помощью перетаскивания, представлять архивы в виде обычных папок, вызывать контекстные меню для упакованных объектов, как для объектов «Рабочего стола» Windows. На архивирование 20-Мбайт массива данных программы тратили (в режиме с параметрами по умолчанию) от 1,5 (ArjFolder) до 4 мин (Q Cab). Наилучшую степень сжатия показала программа Q Cab: созданный ею EXE-архив оказался почти на 10% компактнее остальных архивов, которые, в свою очередь, различались по объему на 1–5%.

Лучшие из рассмотренных программ относятся к категории условно-бесплатных, некоммерческие разработки уступают им в разнообразии функций, совместимости и удобстве (хотя и не в эффективности сжатия). Лидером обзора являются Zip-ориентированные утилиты ZipMagic фирмы Mijenix, Zip Explorer Pro компании Aeco Systems и уже упомянутая WinZip фирмы Nico Mak Computing. Все они обеспечивают совместимость

с большим числом форматов, удобны в использовании. Первые две программы, правда, выгодно отличаются от WinZip возможностью работы с архивами как с папками. Практически не уступает лидерам по удобству и возможностям программа Е. Рошаля WinRAR, но она ориентирована прежде всего на не очень распространенный формат RAR, хотя и обеспечивает большинство необходимых функций для манипулирования Zip-архивами. Тем, кто предпочитает бесплатные утилиты, можно рекомендовать для работы с Zip-архивами программу Eazy Zip 98, а для работы с ARJ-архивами – ArjFolder.

3.2.2. Архивация данных в MS DOS

История развития MS DOS весьма похожа на историю развития всех информационных технологий. Более того, концепция MS DOS непосредственно заимствована из операционной системы UNIX. UNIX, в свою очередь, базируется на самых первых операционных системах типа OS/360 и даже IBM 704. Основные алгоритмы архивации данных вначале были опробованы на UNIX, а затем (иногда процесс развития тех или иных алгоритмов осуществлялся параллельно) в MS DOS. В принципе, большинство используемых в настоящее время утилит для архивации начали свою жизнь начиная с MS DOS 2.0.

Итак, в сороковых годах ученые, работающие в области информационных технологий, ясно поняли, что можно разработать такой способ хранения данных, при котором пространство будет расходоваться более экономно. Клод Шеннон, изучая нюансы различий между семантикой (semantics) (что некая сущность значит) и синтаксисом (syntax) (как некая сущность выражается), разработал большинство базовых понятий этой теории. Понимание того, что одно и то же значение (семантика) может быть реализовано различными способами (синтаксис), приводит к закономерному вопросу: «Какой способ выражения чего-либо является наиболее экономичным?». Поиск ответа на этот вопрос привел Шеннона к мысли об энтропии, которая, проще говоря, соотносится с количеством содержащейся в файле полезной информации. Методы сжатия пытаются увеличивать энтропию файла, то есть уменьшать длину файла, сохраняя при этом всю информацию.

Однако Шеннон не был первым, кто задумывался о сущности информации и определении ее количества. Первый шаг на этом пути сделал в 1928 г. Хартли. Основной полученный им результат

можно сформулировать примерно так: если в заданном множестве, содержащем N элементов, выделен некоторый элемент x, о котором известно лишь то, что он принадлежит этому множеству, то, чтобы найти x, необходимо получить количество информации, равное Iog2 N. Эту формулу обычно называют формулой Хартли. Формула Хартли является частным случаем более общей формулы Шеннона, позволяющей найти количество информации в случайном сообщении фиксированного алфавита. Пусть X_1, ..., X_n – символы этого алфавита, P_1, ..., P_n – вероятности их появления в тексте сообщения, тогда формула Шеннона принимает вид:

$$H = P_1 * \log 2\left(\frac{1}{P_1}\right) + ... + P_n * \log 2\left(\frac{1}{P_n}\right),$$

где H – количество бит информации в одном символе сообщения, или энтропия символа сообщения. Это число показывает минимальное среднее число бит, необходимых для представления одного символа алфавита данного сообщения.

В некоторых случаях алфавит сообщения может быть неизвестен, тогда выдвигаются гипотезы об алфавите сообщения. Имея разные алфавиты, можно достичь разных коэффициентов сжатия. Например, текстовый файл, если его рассматривать как последовательность битов, имеет энтропию порядка 0,7–0,9, если как последовательность байтов – 0,5–0,7, хотя популярные программы сжатия уменьшают размеры текстовых файлов до 0,3–0,4 от исходного размера.

Доказательство Шеннона не было конструктивным, т. е. не содержало способа построения этих оптимальных кодов, а лишь показывало их существование. До появления работы Шеннона, кодирование символов алфавита при передаче сообщения по каналам связи осуществлялось одинаковым количеством бит, получаемым по формуле Хартли. С появлением этой работы начали появляться способы, кодирующие символы разным числом бит в зависимости от вероятности появления их в тексте. Например, часто в файлах некоторые значения байта встречаются чаще других. Таким образом, за счет использования для каждого значения байта кода различной длины можно значительно уменьшить общий размер данных. Эта базовая идея лежит в основе алгоритмов сжатия Шеннона-Фано (Shannon-Fano) и Хаффмана (Huffman). Подобные алгоритмы выбирают более короткие коды для часто встречающихся значений, и более длинные для редко встречающихся значений. Обычно текстовые файлы (в которых одни значения байтов повторяются гораздо чаще других) они сжимают довольно хорошо.

Более тридцати лет алгоритм сжатия Хаффмана и его варианты оставались наиболее популярными методами. Однако в 1977 году два исследователя из Израиля предложили совершенно другой подход к этой проблеме. Абрахам Лемпел и Якоб Зив выдвинули идею формирования «словаря» общих последовательностей данных. При этом сжатие данных осуществляется за счет замены записей соответствующими кодами из словаря. Существуют два алгоритма, в настоящее время известные как LZ77 и LZ78. Они уже не требуют включения словаря данных в архив, так как если вы формируете ваш словарь определенным способом, программа декодирования может его восстанавливать непосредственно из ваших данных. К сожалению, LZ77 и LZ78 тратят много времени на создание эффективного словаря. Лемпел был приглашен фирмой Sperry для оказания им помощи в разработке способа наиболее эффективной упаковки данных на компьютерных лентах. В этой же фирме Терри Велч (Terry Welch) расширил алгоритм LZ78, создав новый вариант, широко известный как LZW.

Популярность алгоритма LZW в значительной степени связана с успехом программы compress. Исходный текст последней версии программы, осуществляющей как сжатие, так и декомпрессию, занимает всего 1200 строк. Ядро кода сжатия занимает не более сотни строк, а код декомпрессии не намного больше. Программисты считают, что это облегчает чтение и понимание алгоритма, а также позволяет адаптировать его для самых разных целей.

Алгоритмы LZ-стиля (включая LZW, LZ77, LZ78 и многие другие варианты) очень популярны везде, где требуется универсальное сжатие. LZW используется в стандарте модема V.42bis, протоколе передачи данных ZModem, форматах GIF, TIFF, ARC и других прикладных программах. Другие алгоритмы LZ используются в дисковых утилитах сжатия типа DoubleSpace и Stacker, графических форматах типа PNG, а также в универсальных утилитах архивирования и сжатия, включая ZIP, GZIP и LHA. Помимо пользующихся большим вниманием алгоритмов, базирующихся на словаре, существуют и другие подходы. Алгоритм сжатия Хаффмана (Huffman), основанный на статистических колебаниях распределения некоторых значений байтов, лег в основу нескольких очень эффективных методов сжатия, известных как арифметическое кодирование (arithmetic encoding), энтропийное кодирование (entropy coding) или Q-кодирование (Q-coding). Арифметическое кодирование улучшает сжатие Хаффмана двумя путями. Первое усовершенствование заключается в том, что оно не требует, чтобы выбранные

коды были целым числом бит. В то время как сжатие Хаффмана могло выбирать двух- и четырехбитовые коды, программа арифметического кодирования может использовать код длиной 6,23 бит. (Что такое 0,23 бит – чисто философский вопрос, если Вас это заинтересовало, то в отдельном разделе Вы найдете другое объяснение арифметического кодирования). Второе усовершенствование (которое может также использоваться в сжатии Хаффмана) заключается в том, что арифметическое кодирование использует более сложную статистику. Она не просто следит за частотой появления байта в файле, а оценивает частоту его появления в определенном контексте. Например, при использовании исходного алгоритма сжатия Хаффмана символ «r», встречающийся не слишком часто, мог бы получать довольно длинный код. Но в сложной программе арифметического кодирования символ «r», следующий за «q», будет закодирован очень компактно, так как высока вероятность того, что «r» следует сразу за «q». Комбинация этих двух усовершенствований приводит к очень эффективному сжатию.

Другие методы сжатия предназначены для данных определенного типа, а потому они плохо подходят для архивирования. Многие усовершенствованные методы, появлявшиеся в последнее время, основывались на синтезе этих трех методов (например, использование кодов Хаффмана для записей словаря) или выполнения сложной предварительной обработки данных, увеличивающей эффективность сжатия одним из этих методов.

Возможно, одним из наиболее существенных событий за последние несколько десятилетий в области алгоритмов сжатия стало появление патентов на программное обеспечение. С 1981 года United States Patent and Trademark Office (USPTO) начал принимать заявки на патентование алгоритмов программного обеспечения. Многие из представленных патентов были по методам сжатия. Наиболее известные из них – патенты фирмы Unisys на алгоритм сжатия LZW и патенты фирмы IBM на арифметическое кодирование. К сожалению, первоначально работа по обработке заявок в USPTO была поставлена неважно. В результате чего разным людям предоставлялись различные патенты на один и тот же алгоритм (причем иногда с почти идентичной формулировкой). Один положительный результат введения патентования вряд ли приходится оспаривать. Патентование программного обеспечения спровоцировало появление огромного количества работ по разработке новых алгоритмов сжатия (большая часть которых быстро патентуется их изобретателями). Однако другой эффект был абсолютно отрицательный. Мно-

гие из алгоритмов сжатия использовались специфическим образом, например, как часть международных стандартов (V.42bis и JPEG). Кроме того, отдельные компании и пользователи скопировали общедоступный код (так, реализация compress LZW широко копировалась для самых разных целей). Финансовые штрафы за использование этих алгоритмов (в форме авторских отчислений к владельцам патента) отвращали от поддержки этих стандартов авторов условно-бесплатного и бесплатного программного обеспечения или бесплатных библиотек. Некоторые компании публично объявили о том, что они не будут требовать авторских отчислений за использование их запатентованных алгоритмов в бесплатном программном обеспечении.

Однако так поступили далеко не все. Пока неясно, как этот конфликт отразится на индустрии бесплатного программного обеспечения и на патентном законодательстве. По крайней мере, одна организация, League for Programming Freedom, борется с патентами программного обеспечения и предпринимает активные шаги по их отмене.

В данной работе будут рассмотрены только современные способы архивации данных, а именно специализированные программы архиваторы. Устаревшие программы резервного копирования типа BACKUP-RESTORE рассматриваться не будут, ввиду их отсутствия в новых версиях MS DOS, начиная с MS DOS 6.22.

3.2.3. Архиваторы MS DOS типа RAR

С развитием компьютера стали увеличиваться и объемы информации хранимой в нем, что в свою очередь привело к развитию технологий по хранению этой информации в сжатом виде, то есть в архивах. Для этого было придумано множество программ, осуществляющих архивацию информации.

Однако в работе с этой информацией иногда нежелательно раскрывать полный архив, чтобы взять один или два требуемых файла или же просто посмотреть, что за информация в архиве.

Программы-архиваторы, за исключением единиц, не предоставляют удобных оболочек, позволяющих просто, быстро и в наглядной форме разобраться с содержимым архивов.

Архиватор RAR v2.50 для DOS – интегрированная программа управления архивами.

RAR – это очень мощное средство для создания архивов и управления ими.

Возможности RAR:

- полноэкранный интерактивный интерфейс (отключаемый);
- поддержка мыши и меню;
- поддержка не RAR архивов;
- стандартный интерфейс командной строки;
- оригинальный высокоэффективный алгоритм сжатия данных;
- специальный алгоритм для сжатия мультимедийных файлов;
- лучшая степень упаковки, чем у аналогичных продуктов, за счет использования режима «непрерывного» сжатия;
- информация об авторе архива (только в зарегистрированной версии);
- самораспаковывающиеся (SFX) обычные и многотомные архивы;
- восстановление физически поврежденных архивов;
- язык программирования для инсталляционных SFX-архивов;
- блокировка, шифрование, список порядка файлов, метки томов и др.

3.2.4. Сравнение архиваторов MS DOS и Windows

Несмотря на кажущуюся «моральную отсталость» MS DOS, в ряде случаев использование MS DOS архиваторов может быть гораздо более эффективным, по сравнению с аналогичными версиями для Windows.

1. Архиваторы, работающие в среде MS DOS, работают в реальном режиме процессора. Это обеспечивает не менее чем в 1,5 раза большую производительность процессора, по сравнению с защищенным режимом Windows.
2. Все версии архиваторов для MS DOS (по крайней мере, рассмотренные выше) весьма эффективно используют память XMS или EMS. С учетом реального режима работы процессора, это дает еще 10–15% выигрыша в производительности.
3. Как ни странно, версии архиваторов для MS DOS часто используют более совершенные алгоритмы сжатия. Причина проста: в большинстве случаев в версиях архиваторов для MS DOS опробуются возможные усовершенствования, переносимые далее в Windows. Причем, иногда применение некоторых новшеств в Win-

dows оказывается технически нецелесообразным. Пример – отказ от применения в архиваторе WinZip 8.0 Beta словаря переменной длины более 128 кБайт, из-за переполнения кэш-памяти современных процессоров.

И наконец, в случае повреждения загрузочных файлов Windows, единственным способом «спасти» архив является использование проверенных и надежных архиваторов для MS DOS.

4. Система управления базами данных (СУБД)

Современная жизнь немыслима без эффективного управления, важной категорией которого являются системы обработки информации, от которых во многом зависит эффективность работы любого предприятия или учреждения. Такая система должна:

- обеспечивать получение общих или детализированных отчетов по итогам работы;
- позволять легко определять тенденции изменения важнейших показателей;
- обеспечивать получение информации, критической по времени, без существенных задержек;
- выполнять точный и полный анализ данных.

Современные СУБД в основном являются приложениями Windows, так как данная среда позволяет более полно использовать возможности персональной ЭВМ, нежели среда DOS. Снижение стоимости высокопроизводительных ПК обусловило не только широкий переход к среде Windows, где разработчик программного обеспечения может в меньшей степени заботиться о распределении ресурсов, но также сделало программное обеспечение ПК в целом и СУБД, в частности, менее критичными к аппаратным ресурсам ЭВМ.

Среди наиболее ярких представителей систем управления базами данных можно отметить: Lotus Approach, Microsoft Access, Borland dBase, Borland Paradox, Microsoft Visual FoxPro, Microsoft Visual Basic, а также баз данных Microsoft SQL Server и Oracle, используемые в приложениях, построенных по технологии «клиент-сервер». Фактически, у любой современной СУБД существует аналог, выпускаемый другой компанией, имеющий аналогичную область применения и возможности, любое приложение способно работать со многими форматами представления данных, осуществлять экспорт и импорт данных благодаря наличию большого числа конвертеров. Общепринятыми также являются технологии, позво-

ляющие использовать возможности других приложений, например: текстовых процессоров, пакетов построения графиков и т. п., и встроенные версии языков высокого уровня (чаще - диалекты SQL и/или VBA) и средства визуального программирования интерфейсов разрабатываемых приложений. Поэтому уже не имеет существенного значения, на каком языке и на основе какого пакета написано конкретное приложение, и какой формат данных в нем используется. Более того, стандартом «де-факто» стала «быстрая разработка приложений» или RAD (от английского Rapid Application Development), основанная на широко декларируемом в литературе «открытом подходе», то есть необходимость и возможность использования различных прикладных программ и технологий для разработки более гибких и мощных систем обработки данных. Поэтому в одном ряду с «классическими» СУБД все чаще упоминаются языки программирования Visual Basic 4.0, Delphi и Visual C++, которые позволяют быстро создавать необходимые компоненты приложений, критичные по скорости работы, которые трудно, а иногда невозможно разработать средствами «классических» СУБД. Современный подход к управлению базами данных подразумевает также широкое использование технологии «клиент-сервер». Таким образом, на сегодняшний день разработчик не связан рамками какого-либо конкретного пакета, а в зависимости от поставленной задачи может использовать самые разные приложения. Поэтому более важным представляется общее направление развития СУБД и других средств разработки приложений в настоящее время.

4.1. Обзор и сравнительная характеристика программного обеспечения, используемого при создании СУБД

Рассмотрим более подробно программные продукты компании Microsoft, а именно: Visual FoxPro 3.0, Visual Basic 4.0, Visual C++, Access 7.0, SQL Server 6.5. Наиболее интересной чертой этих пакетов являются их большие возможности интеграции, совместной работы и использования данных, так как данные пакеты являются продуктами одного производителя, а также используют сходные технологии обмена данными.

Visual FoxPro отличается высокой скоростью, имеет встроенный объектно-ориентированный язык программирования с использованием xBase и SQL, диалекты которых встроены во многие СУБД. Имеет высокий уровень объектной модели. При использо-

вании в вычислительных сетях обеспечивает как монопольный, так и раздельный доступ пользователей к данным. Применяется для приложений масштаба предприятия для работы на различных платформах: Windows 3.x, Windows 95, Macintosh... Минимальные ресурсы ПК: для Visual FoxPro версии 3.0 - процессор 468DX, Windows 3.1, 95, NT, объем оперативной памяти - 8 (12) Мб, занимаемый объем на ЖМД - 15-80 Мб, а для Visual FoxPro версии 5.0 (выпущена в 1997 году) - Windows 95 или NT, 486 с тактовой частотой 50 МГц, 10 Мб ОЗУ, от 15 до 240 Мб на ЖМД.

Access входит в состав самого популярного пакета Microsoft Office. Основные преимущества: знаком многим конечным пользователям и обладает высокой устойчивостью данных, прост в освоении, может использоваться непрофессиональным программистом, позволяет готовить отчеты из баз данных различных форматов. Предназначен для создания отчетов произвольной формы на основании различных данных и разработки некоммерческих приложений. Минимальные ресурсы ПК: процессор 468DX, Windows 3.1, 95, NT, объем оперативной памяти - 12 (16) Мб, занимаемый объем на ЖМД - 10-40 Мб.

Visual Basic - это универсальный объектно-ориентированный язык программирования, диалекты которого встроены в Access, Visual FoxPro. Преимущества: универсальность, возможность создания компонентов OLE, невысокие требования к аппаратным ресурсам ЭВМ. Применяется для создания приложений средней мощности, не связанных с большой интенсивностью обработки данных, разработки компонентов OLE, интеграция компонентов Microsoft Office. Минимальные ресурсы ПК: процессор 368DX, Windows 3.1, 95, NT, объем оперативной памяти - 6 (16) Мб, занимаемый объем на ЖМД - 8-36 Мб.

Visual C++ - наиболее мощный объектно-ориентированный язык программирования, обладает неограниченной функциональностью. Предназначен для создания компонентов приложений для выполнения операций, критичных по скорости.

SQL Server - сервер баз данных, реализует подход «клиент-сервер» и взаимодействует с указанными пакетами. Главные достоинства: высоая степень защиты данных, мощные средства для обработки данных, высокая производительность. Область применения: хранение больших объемов данных, хранение высокоценных данных или данных, требующих соблюдения режима секретности. Минимальные ресурсы ПК: процессор 468DX-33MHi, Windows NT, объем оперативной памяти - 16 (32) Мб, занимаемый объем на

ЖМД - 80 Мб.

Указанные программные продукты имеют возможности визуального проектирования интерфейса пользователя, то есть разработчик из готовых фрагментов создает элементы интерфейса, программирует только их изменения в ответ на какие-либо события.

4.2. Принципы организации данных, лежащие в основе СУБД

Современные СУБД являются объектно-ориентированными и реляционными. Основной единицей является объект, имеющий свойства, и связи между объектами. СУБД используют несколько моделей данных: иерархическую и сетевую (с 60-х годов) и реляционную (с 70-х). Основное различие данных моделей - в представлении взаимосвязей между объектами.

Иерархическая модель данных строится по принципу иерархии объектов, то есть один тип объекта является главным, все нижележащие - подчиненными. Устанавливается связь «один ко многим», то есть для некоторого главного типа существует несколько подчиненных типов объектов. Иначе, главный тип именуется исходным типом, а подчиненные - порожденными. У подчиненных типов могут быть в свою очередь свои собственные подчиненные типы. Наивысший в иерархии узел (совокупность атрибутов) называют корневым.

Сетевая модель данных строится по принципу «главный и подчиненный тип одновременно», то есть любой тип данных одновременно может одновременно порождать несколько подчиненных типов (быть владельцем набора) и быть подчиненным для нескольких главных (быть членом набора).

Реляционная модель данных: объекты и связи между ними представляются в виде таблиц, при этом связи тоже рассматриваются как объекты. Все строки, составляющие таблицу в реляционной базе данных должны иметь первичный ключ. Все современные средства СУБД поддерживают реляционную модель данных.

Объект (Сущность) - элемент какой-либо системы, информация о котором сохраняется. Объект может быть как реальным (например, человек), так и абстрактным (например, событие - поступление человека в стационар).

Атрибут - информационное отображение свойств объекта. Каждый объект характеризуется набором атрибутов.

Таблица - упорядоченная структура, состоящая из конечного

набора однотипных записей.

Первичный ключ - атрибут (или группа атрибутов), позволяющий однозначным образом определить каждую строку в таблице. Напротив, **альтернативный ключ** - атрибут (или группа атрибутов), не совпадающий с позволяющим первичным ключом и однозначным образом, определяющим каждую строку в таблице.

4.3. Современные технологии, используемые в работе с данными

Технология **«Клиент-сервер»** - технология, разделяющая приложение СУБД на две части: клиентскую (интерактивный графический интерфейс, расположенный на компьютере пользователя) и сервер, собственно осуществляющий управление данными, разделение информации, администрирование и безопасность, находящийся на выделенном компьютере. Взаимодействие «клиент-сервер» осуществляется следующим образом: клиентская часть приложения формирует запрос к серверу баз данных, на котором выполняются все команды, а результат исполнения запроса отправляется клиенту для просмотра и использования. Данная технология применяется, когда размеры баз данных велики, когда велики размеры вычислительной сети, и необходимо повысить производительность при обработке данных, хранящихся не на компьютере пользователя (в крупном учреждении обычно имеет место именно такая ситуация). Если технология «клиент-сервер» на применяется, то для обработки даже нескольких записей весь файл копируется на компьютер пользователя, а только затем обрабатывается. При этом резко возрастает загрузка сети и снижается производительность труда многих сотрудников.

Microsoft Access, Microsoft Visual FoxPro, Microsoft Visual Basic обеспечивают средства для создания клиентских частей в приложениях «клиент-сервер», которые сочетают в себе средства просмотра, графический интерфейс и средства построения запросов, а Microsoft SQL Server является на сегодняшний день одним из самых мощных серверов баз данных.

OLE 2.0 (Object Linking and Embedding - связывание и внедрение объектов) - стандарт, описывающий правила интеграции прикладных программ. Применяется для использования возможностей других приложений. OLE 2.0 используется для определения и совместного использования объектов несколькими приложениями, которые поддерживают данную технологию. Например, использо-

вание в среде Access таблиц Excel и его мощных средств построения диаграмм или использование данных, подготовленных Access, в отчетах, составленных в редакторе текстов Word (связывание или включение объекта).

OLE Automation (Автоматизация OLE.) — компонент OLE, позволяющий программным путем устанавливать свойства и задавать команды для объектов другого приложения. Позволяет без необходимости выхода или перехода в другое окно использовать возможности нужного приложения. Приложение, позволяющее другим прикладным программам использовать свои объекты, называется OLE сервером. Приложение, которое может управлять объектами OLE серверов, называется OLE контроллер или OLE клиент. Из рассмотренных программных средств в качестве OLE серверов могут выступать Microsoft Access, а также Microsoft Excel, Word и Graph. Microsoft Visual FoxPro 3.0 и 5.0 может выступать только в виде OLE клиента.

RAD (Rapid Application Development - Быстрая разработка приложений) — подход к разработке приложений, предусматривающий широкое использование готовых компонентов и/или приложений и пакетов (в том числе от разных производителей).

ODC (Open Database Connectivity - открытый доступ к базам данных) — технология, позволяющая использовать базы данных, созданные другим приложением при помощи SQL.

SQL (Structured Query Language - язык структурированных запросов) - универсальный язык, предназначенный для создания и выполнения запросов, обработки данных как в собственной базе данных приложения, так и с базами данных, созданных другими приложениями, поддерживающими SQL. Также SQL применяется для управления реляционными базами данных.

VBA (Visual Basic for Applications - Visual Basic для Приложений) - разновидность (диалект) объектно-ориентированного языка программирования Visual Basic, встраиваемая в программные пакеты.

5. Обработка экспериментальных результатов

Большое значение информационные технологии приобретают при обработке результатов экспериментов, при проведении научно-исследовательской работы и т. д.

При исследовании технических систем могут использоваться теоретические и эмпирические методы познания. Каждое из этих

направлений обладает относительной самостоятельностью, имеет свои достоинства и недостатки. В общем случае теоретические методы в виде математических моделей позволяют описывать и объяснять взаимосвязи элементов изучаемой системы или объекта в относительно широких диапазонах изменения переменных величин. Однако при построении теоретических моделей неизбежно введение каких-либо ограничений, допущений, гипотез и т. п. Поэтому возникает задача оценки достоверности (адекватности) полученной модели реальному процессу или объекту. Для этого проводится экспериментальная проверка разработанных теоретических моделей. Практика является решающей основой научного познания. В ряде случаев именно результаты экспериментальных исследований дают толчок к теоретическому обобщению изучаемого явления. Экспериментальное исследование дает более точное соответствие между изучаемыми параметрами. Но не следует и преувеличивать результаты экспериментальных исследований, которые справедливы только в пределах условий проведенного эксперимента. Таким образом, теоретические и экспериментальные исследования дополняют друг друга и являются составными элементами процесса познания окружающего нас мира. Как правило, результаты экспериментальных исследований нуждаются в определенной математической обработке. В настоящее время процедура обработки экспериментальных данных достаточно хорошо формализована, и исследователю необходимо только ее правильно использовать. Круг вопросов, решаемых при обработке результатов эксперимента, не так уж велик. Это - вопросы подбора эмпирических формул и оценка их параметров, вопросы оценки истинных значений измеряемых величин и точности измерений, вопросы исследования корреляционных зависимостей и некоторые другие.

Особое место при этом занимает оценка ошибок полученных в результате измерений и первичной обработки результатов.

5.1. Ошибки измерений

Основой всего естествознания является наблюдение и эксперимент.

Наблюдение - это систематическое, целенаправленное восприятие того или иного объекта или явления без воздействия на изучаемый объект или явление. Наблюдение позволяет получить первоначальную информацию по изучаемому объекту или явлению.

Эксперимент - метод изучения объекта, когда исследователь активно и целенаправленно воздействует на него путем создания искусственных условий или использует естественные условия, необходимые для выявления соответствующих свойств. Достоинствами эксперимента по сравнению с наблюдением реального явления или объекта является:

1) возможность изучения в «чистом виде», без влияния побочных факторов, затемняющих основной процесс;

2) в экспериментальных условиях можно получить результат более быстро и точно;

3) при эксперименте можно проводить испытания столько раз, сколько это необходимо.

Результат эксперимента или измерения всегда содержит некоторую погрешность. Если погрешность мала, то ею можно пренебречь. Однако при этом неизбежно возникают два вопроса: во-первых, что понимать под малой погрешностью и, во-вторых, как оценить величину погрешности. То есть и результаты эксперимента нуждаются в определенном теоретическом осмыслении.

5.2. Цели математической обработки результатов эксперимента

Целью любого эксперимента является определение качественной и количественной связи между исследуемыми параметрами либо оценка численного значения какого-либо параметра. В некоторых случаях вид зависимости между переменными величинами известен по результатам теоретических исследований. Как правило, формулы, выражающие эти зависимости, содержат некоторые постоянные, значения которых и необходимо определить из опыта. Другим типом задачи является определение неизвестной функциональной связи между переменными величинами на основе данных эксперимента. Такие зависимости называют эмпирическими. Однозначно определить неизвестную функциональную зависимость между переменными невозможно даже в том случае, если бы результаты эксперимента не имели ошибок. Тем более не следует этого ожидать, имея результаты эксперимента, содержащие различные ошибки измерения. Поэтому следует четко понимать, что целью математической обработки результатов эксперимента является не нахождение истинного характера зависимости между переменными или абсолютной величины какой-либо константы, а представление результатов наблюдений в виде

наиболее простой формулы с оценкой возможной погрешности ее использования.

5.3. Виды измерений и причины ошибок

Под измерением понимают сравнение измеряемой величины с другой величиной, принятой за единицу измерения. Различают два типа измерений: **прямые** и **косвенные**. При прямом измерении измеряемая величина сравнивается непосредственно со своей единицей меры. Например, измерение микрометром линейного размера, промежутка времени при помощи часовых механизмов, температуры - термометром, силы тока - амперметром и т. п. Значение измеряемой величины отсчитывается при этом по соответствующей шкале прибора. При косвенном измерении измеряемая величина определяется (вычисляется) по результатам измерений других величин, которые связаны с измеряемой величиной определенной функциональной зависимостью. Например, измерение скорости по пройденному пути и затраченному времени, измерение плотности тела по измерению массы и объема, температуры при резании - по электродвижущей силе, величины силы - по упругим деформациям и т. п. При измерении любой физической величины производят проверку и установку соответствующего прибора, наблюдение их показаний и отсчет. При этом никогда истинного значения измеряемой величины не получить. Это объясняется тем, что измерительные средства основаны на определенном методе измерения, точность которого конечна. При изготовлении прибора задается класс точности. Его погрешность определяется точностью деления шкалы прибора. Если шкала линейки нанесена через 1 мм, то точность отсчета ± 0,5 мм не изменить, если применим лупу для рассматривания шкалы. Аналогично происходит измерение и при использовании других измерительных средств.Кроме приборной погрешности на результат измерения влияет еще ряд объективных и субъективных причин, обуславливающих появление ошибки измерения - разности между результатом измерения и истинным значением измеряемой величины. Ошибка измерения обычно неизвестна, как неизвестно и истинное значение измеряемой величины. Исключение составляют измерения известных величин при определении точности измерительных приборов или их тарировке. Поэтому одной из важнейших задач математической обработки результатов эксперимента и является оценка истинного значения измеряемой величины по данным эксперимента с возможно меньшей ошибкой.

5.4. Типы ошибок измерения

Кроме приборной погрешности измерения (определяемой методом измерения) существуют и другие, которые можно разделить на три типа.

1. Систематические погрешности обуславливаются постоянно действующими факторами. Например, смещение начальной точки отсчета, влияние нагревания тел на их удлинение, износ режущего лезвия и т. п. Систематические ошибки выявляют при соответствующей тарировке приборов.

2. Случайные ошибки содержат в своей основе много различных причин, каждая из которых не проявляет себя отчетливо. Случайную ошибку можно рассматривать как суммарный эффект действия многих факторов. Поэтому случайные ошибки при многократных измерениях получаются различными как по величине, так и по знаку. Их невозможно учесть как систематические, но можно учесть их влияние на оценку истинного значения измеряемой величины. Анализ случайных ошибок является важнейшим разделом математической обработки экспериментальных данных.

3. Грубые ошибки (промахи) появляются вследствие неправильного отсчета по шкале, неправильной записи, неверной установки условий эксперимента и т. п. Они легко выявляются при повторном проведении опытов.

6. Современные методы защиты информации

Одной из важнейших проблем при работе с данными является их защита от повреждения и несанкционированного доступа к ним. В этой главе рассмотрим подробнее эти два вопроса. Начнем с того, что проблема защиты информации от постороннего доступа и нежелательных воздействий на нее возникла давно, с той поры, когда человеку по каким-либо причинам не хотелось делиться ею ни с кем или не с каждым человеком. С развитием человеческого общества, появлением частной собственности, государственного строя, борьбой за власть и в дальнейшем расширением масштабов человеческой деятельности информация приобретает цену. Ценной становится та информация, обладание которой позволит ее существующему и потенциальному владельцу получить какой-либо выигрыш: материальный, политический, военный и т. д.

В период существования примитивных носителей информации ее защита осуществлялась организационными методами, кото-

рые включали ограничение и разграничение доступа, определенные меры наказания за разглашение тайны. По свидетельству Геродота, уже в V веке до новой эры использовалось преобразование информации методом кодирования. Коды появились в глубокой древности в виде криптограмм (по-гречески – тайнопись). Спартанцы имели специальный механический прибор, при помощи которого важные сообщения можно было писать особым способом, обеспечивающим сохранение тайны. Собственная секретная азбука была у Юлия Цезаря. В средние века и эпоху Возрождения над изобретением тайных шифров трудились многие выдающиеся люди, в их числе известный философ Френсис Бэкон, крупные математики – Франсуа Виет, Джероламо Кардано, Джон Валлис.

С переходом на использование технических средств связи информация подвергается воздействию случайных процессов: неисправностям и сбоям оборудования, ошибкам операторов и т. д., которые могут привести к ее разрушению, изменениям на ложную, а также создать предпосылки к доступу к ней посторонних лиц. С дальнейшим усложнением и широким распространением технических средств связи возросли возможности для преднамеренного доступа к информации.

С появлением сложных автоматизированных систем управления, связанных с автоматизированным вводом, хранением, обработкой и выводом информации, проблема ее защиты приобретает еще большее значение. Этому способствовали:

- увеличение объемов информации, накапливаемой, хранимой и обрабатываемой с помощью ЭВМ и других средств вычислительной техники;
- сосредоточение в единых базах данных информации различного назначения и принадлежности;
- расширение круга пользователей, имеющих доступ к ресурсам вычислительной системы и находящимся в ней массивам данных;
- усложнение режимов функционирования технических средств вычислительной системы: широкое внедрение многопрограммного режима, режима разделения времени и реального времени;
- автоматизация межмашинного обмена информацией, в том числе и на больших расстояниях;
- увеличение количества технических средств и связей в автоматизированных системах управления и обработки данных;

• появление персональных ЭВМ, расширяющих возможности не только пользователя, но и нарушителя.

К настоящему времени и в самом человеческом обществе, и в технологии обработки данных произошли большие изменения, которые повлияли на саму суть проблемы защиты информации. Например, по данным зарубежной литературы, к концу 70-х годов деятельность в области сбора, обработки и использования информации достигла 46% валового национального продукта США, и на нее приходится 53% общей суммы заработной платы. Индустрия переработки информации достигла глобального уровня. Появилась возможность выхода в глобальную вычислительную сеть с домашнего компьютера. Появление «электронных» денег (кредитных карточек) создало предпосылки для хищений крупных сумм денег. В печати приведено множество конкретных примеров хищения информации из автоматизированных систем обработки данных, которые весьма убедительно иллюстрируют серьезность и актуальность проблемы. Парадоксально, но хорошо работающая система с качественными соединениями будет способствовать более успешной краже информации. Для предотвращения плачевного исхода следует не только эффективно реализовать защиту, но и установить для функций слежения и управления безопасностью такой же высокий приоритет, как и для управления компьютерными сетями. Хакеры создают свои клубы, например гамбургский клуб «Хаос-компьютер», распространяют свои бюллетени, обмениваются информацией через десятки «электронных почтовых ящиков». Коды, пароли, техническая информация, призывы и т.д. – все идет через «почтовые ящики». Такие клубы появляются и в России. Особая разновидность хакеров – крэкеры (cracker (англ.) – вор-взломщик). Крэкеры в отличие от хакеров воруют информацию с помощью компьютера, выкачивая целые информационные банки данных.

В последнее время широкое распространение получил новый вид компьютерного преступления – создание компьютерных вирусов, в качестве которых выступают специально разработанные программы, начинающие работать только по определенному сигналу. При этом вирус может размножаться, словно возбудитель болезни, когда соприкасается с другим программным обеспечением. Последствия от «заражения» программ подобными вирусами могут быть различными: от безобидных шуток в виде юмористических помех до разрушения программного обеспечения, восстановление которого может оказаться невозможным, а потери невосполнимыми.

При наличии простых средств хранения и передачи информации существовали и не потеряли значения до настоящего времени следующие методы ее защиты от преднамеренного доступа: ограничение доступа; разграничение доступа; разделение доступа (привилегий); криптографическое преобразование информации; контроль и учет доступа; законодательные меры.

Указанные методы осуществлялись чисто организационно или с помощью технических средств.

С появлением автоматизированной обработки информации изменился и дополнился новыми видами физический носитель информации и усложнились технические средства ее обработки.

С усложнением обработки, увеличением количества технических средств, участвующих в ней, увеличиваются количество и виды случайных воздействий, а также возможные каналы несанкционированного доступа. С увеличением объемов, сосредоточением информации, увеличением количества пользователей и другими указанными выше причинами увеличивается вероятность преднамеренного несанкционированного доступа к информации. В связи с этим развиваются старые и возникают новые дополнительные методы защиты информации в вычислительных системах:

- методы функционального контроля, обеспечивающие обнаружение и диагностику отказов, сбоев аппаратуры и ошибок человека, а также программные ошибки;
- методы повышения достоверности информации;
- методы защиты информации от аварийных ситуаций;
- методы контроля доступа к внутреннему монтажу аппаратуры, линиям связи и технологическим органам управления;
- методы разграничения и контроля доступа к информации;
- методы идентификации и аутентификации пользователей, технических средств, носителей информации и документов;
- методы защиты от побочного излучения и наводок информации.

Рассмотрим теперь некоторые вопросы подробнее.

6.1. Компьютерная вирусология и антивирусные программы

Идея компьютерных вирусов витала в воздухе несколько десятилетий. В 1986 году они стали реальностью и продолжают быть мощной силой в компьютерном мире. Термин компьютерный вирус используется в настоящее время очень широко, хотя большинство

людей не имеют представления о том, что это такое. Слово вирус заставляет нас думать о простуде и напоминает о лекарстве от насморка.

Компьютерный вирус - эта не особая форма жизни, а такая же программа, как и текстовый процессор. Поскольку вирус – это программа, компьютер запускает его как обычную программу, но то, что происходит потом, чаще всего бывает неожиданным и зависит от вида вируса.

Вирусы, черви, троянские кони, ошибки, а также логические и часовые бомбы – это все одинаково нежелательное, незваное, потенциально опасное программное обеспечение, но между этими видами программного обеспечения существуют принципиальные различия. Эти различия можно сформулировать в виде двух вопросов: требуется ли вирусу программа-носитель (host program) и способна ли она размножаться? Все четыре вышеупомянутые типа вирусов могут вызывать повреждения, но этот аспект не является основополагающим для их классификации. Определение каждого из этих вирусов приведено в таблице 6.1.

Таблица 6.1

Определение вируса, червя, троянского коня, ошибки и бомбы

Категория	Требуется ли носитель	Самокопирование
Вирус	Да. Вирусы нуждаются в программе-носителе. Вирусы, введенные в существующие исполняемые программы, обнаружить сложно, и время от времени они сами обеспечивают свой запуск	Да. Вирусы копируют сами себя, заражая при этом загрузочные секторы дискет, жестких дисков или программы
Червь	Нет. Программа-носитель не требуется, поскольку черви – это, как правило, проблема больших машин и они не скрыты от большинства пользователей	Да. Червь копирует сам себя при каждом удобном случае
Троянский конь	Нет. Хотя термин троянский конь иногда применяют к программам, содержащим в себе разру-	Нет. Большинство троянских коней активизируются при запуске и стремятся разрушить

	шающий код, этот термин чаще всего используется применительно к целому COM- или EXE-файлу	структуру текущего диска (FAT, каталога), удаляя себя во время этого процесса
Ошибка, логическая бомба, часовая бомба	Да. Программисты не могут сделать ошибку, не написав какого-либо кода, однако, честно говоря, большинство программистов делают ошибки неумышленно; логические и часовые бомбы включаются программистами намеренно в нормальный в прочих отношениях код	Нет. Этот код производит нечто большее, чем просто самокопирование. Логические и часовые бомбы скрыты, видны только их действия. Ошибки могут делать почти всё что угодно, кроме порождения новых ошибок

Вирусы могут заражать файлы и загрузочные секторы (в последних содержится небольшая программа начальной загрузки компьютера). На протяжении последних лет проанализированы 1074 разных вирусов, и результаты представлены на рис. 6.1. Отметим, что вирусов, заражающих файлы, намного больше, чем вирусов, заражающих загрузочные секторы дискет и жесткого диска (так называемых бутовых вирусов); некоторые бутовые вирусы не заражают жесткие диски, значительное количество поражает как COM-, так и EXE-файлы. Хотя бутовые вирусы заражают больше программ в организациях, число их разновидностей гораздо меньше числа разновидностей файловых вирусов. Не все бутовые вирусы заражают жесткие диски (например Brain), но большинство это делают.

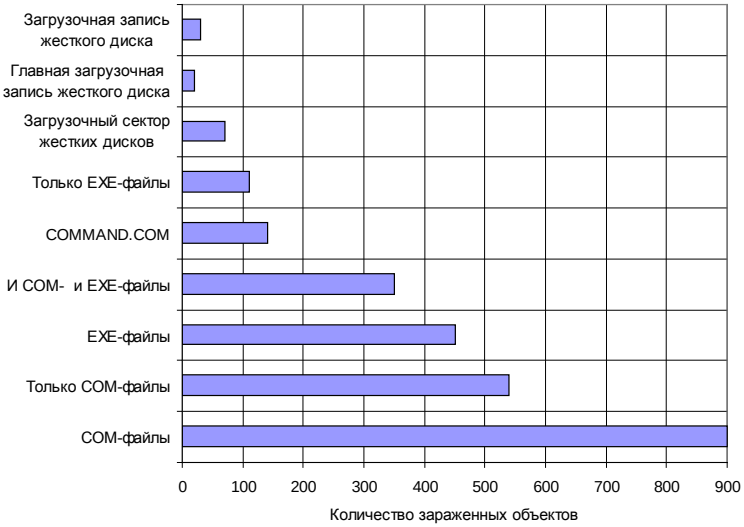

Рис. 6.1. - Анализ заражаемости файлов вирусами (результаты анализа по 1074 вирусам)

Был проведен анализ воздействия заражения вирусом на рабочую группу, данные приведены на рис.6.2.

Для снижения риска заражения вирусом необходимо использовать в работе антивирусное программное обеспечение.

Антивирусные продукты должны делать следующее.

1. Обнаруживать основные бутовые и файловые вирусы.

2. Обнаруживать самоизменяющиеся (self-mutating) вирусы. Одна из самых сложных задач для антивирусного сканера сегодня это обнаружение полиморфных вирусов (polymorphic viruses), принимающих новое обличие с каждой новой копией. Такие вирусы нельзя выявить методом простого сканирования кода программы. Поскольку количество полиморфных вирусов растет, сканеры, не способные бороться с такими вирусами, не могут справиться с таким количеством вирусов. За прошедший год появились десятка два полиморфных вирусов, написанных с помощью Mutation Engine и Trident Polymorphic Engine – подпольных инструментов, модифицирующих код каждой новой копии вируса. Обнаружение полиморфных вирусов не является существенной проблемой по сравнению с обнаружением обычных вирусов. Однако тот факт, что анти-

вирусный продукт обнаруживает сегодня много полиморфных вирусов, означает, что его разработчик хорошо владеет ситуацией.

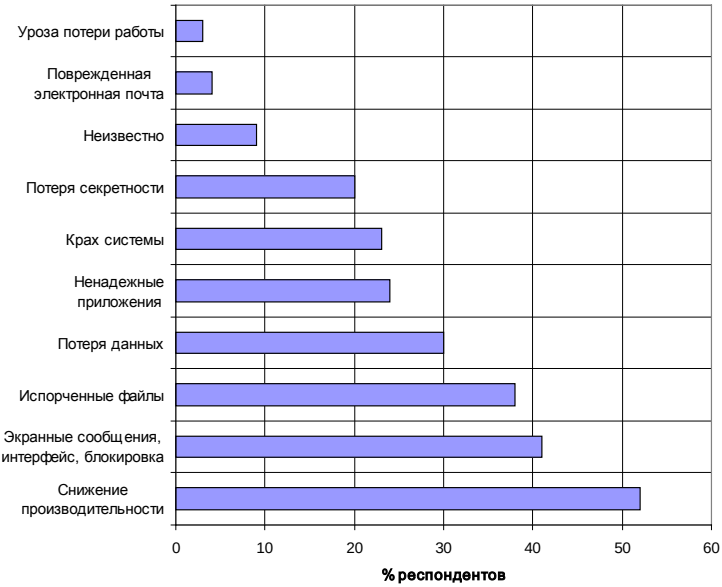

Рис. 6.2. - Анализ воздействия заражения вирусом на рабочую группу

3. Обнаруживать вирус, находящийся в памяти. Если машина заражена резидентным вирусом, то, вероятно, вирус находится резидентно в памяти. Сканер должен обнаружить и полностью обезвредить его, поскольку в противном случае он станет виновником новых проблем, а не помощником в их решении.

4. Обнаруживать заражения главной загрузочной записи. Обнаружить вирус в главной загрузочной записи несколько сложнее, чем в загрузочной записи дискеты. И это представляет серьезную проблему для пользователя, потому что программа DOS FORMAT не модифицирует главную загрузочную запись – это делают только DEBUG и FDISK. В результате пользователь, пытающийся уничтожить этот вирус с помощью команды FORMAT, благополучно уничтожит на жестком диске все, кроме вируса.

5. Обнаруживать вирусы в упакованных (сжатых) файлах. Существует полдюжины технологий сжатия, одну из которых применит разработчик программного обеспечения для упаковки вашей

любимой программы. Появляющаяся на компьютере программа, как правило, упакована и распаковывается при выполнении. Не существует продукта, способного распаковывать программы, запакованные по всем технологиям сжатия, так что будьте внимательны. Следует также помнить, что вирус может заражать упакованный файл как снаружи, так и изнутри. Практически любой сканер может обнаружить заражение файла снаружи, но далеко не все сканеры способны сканировать упакованные файлы (и распознавать различные методы сжатия). Это важно знать при выборе сканера, так как некоторые производители распространяют вирусы в упакованных файлах. Поскольку программные файлы распаковываются только один раз, сканировать внутри сжатого файла также нужно только раз. Некоторые программы-сканеры указывают, следует ли проверять упакованные файлы. Установите, какие форматы сжатия доступны выбираемому вами сканеру. Сможете ли вы потом отключить режим внутреннего сканирования упакованных файлов для экономии времени?

6. Обеспечивать адекватную информацию в сообщении. Вирусы не так часто встречаются, чтобы пользователь, который столкнулся с одним из них, знал, что ему делать в конкретной ситуации.

Таким образом, установленный на компьютере сканер должен выдавать информацию, касающуюся оценки риска, информацию о масштабе заражения, а также инструкцию по удалению.

Сканер должен выполнять следующие функции:

• Выводить информацию в файл или на принтер. Можете ли вы направить вывод сканера в файл, содержащий список зараженных файлов и описание характера заражений? Можете ли вы направить его на принтер? Это важно, если у вас много зараженных файлов, если вам удобнее пользоваться твердой копией в виде текстового или другого документа, или если ваш антивирусный продукт не удаляет вирус частично или полностью.

• Поддерживать сеть. Как продукт работает в сети? Может ли он сканировать сетевой диск (некоторые не могут)? Можно ли его запустить с сетевого диска? Имеет ли он режим извещения администратора сети при обнаружении вируса (в большинстве случаев нет)? Как продукт поступает с файлом на сетевом диске, открытым для записи или чтения? Будет ли он пропускать файл (и сообщать, что файл пропущен) или остановится на нем и можно ли это регулировать по желанию пользователя? Не зависает ли сканер в сети?

• Обладать достаточной скоростью сканирования. Сколько времени необходимо сканеру для сканирования программы? Желательно быстрое сканирование, но при условии, что это не повлияет на качество. Следует отметить, что некоторые сканеры работают быстрее при втором проходе жесткого диска, чем при первом (например, если они подсчитывают контрольную сумму).

• Не вызывать ложные тревоги. Обнаруживает ли сканер несуществующие вирусы? Проблемами, связанными с ложными тревогами, нельзя пренебрегать. Эти тревоги будоражат пользователей, служебный персонал, продавцов. В идеале сканер должен обнаруживать все возможные вирусы и не должен вызывать ложную тревогу.

• Пользовательские процедуры управления. Обычно сканеры используют в пакетном режиме посредством AUTOEXEC.BAT и с помощью меню для специального сканирования, если существует подозрение заражения или при копировании новой дискеты. Для запуска в пакетном режиме необходимо, чтобы сканер не требовал нажатия клавиш во время работы незараженного дисковода. Он должен обеспечивать вывод предупреждений или полезных кодов ошибок при обнаружении вируса или просто передавать управление на следующий файл, указанный в командном файле. При запуске из меню сканер не должен требовать ввода параметров в командной строке.

• Обеспечивать эффективность сканирования съемного носителя. После обнаружения вируса на жестком диске будет разумным просканировать все в пределах досягаемости, включая содержимое всех гибких дисков. Эффективны ли встроенные режимы для сканирования съемных носителей? Эффективность означает, что программа загружается только один раз, память сканируется только один раз, и команда сканирования другого диска требует одного или двух нажатий.

• Обеспечивать наращивание возможностей. Со временем вы, вероятно, захотите защититься от новых вирусов, которые появились после приобретения вашего сканера. Область действия многих сканеров можно расширить, добавляя сигнатуры, полученные от продавца. В некоторых случаях продавец передает эту информацию по телефону, факсу, BBS, CompuServe или авиапочтой. Добавляя сигнатуры, мы рассчитываем на дополнительные возможности.

• Получить сигнатуры, в которых один или более байтов могут использоваться в качестве шаблона (wildcards), что позволит

обнаружить шифрующиеся вирусы или их незначительно отличающиеся варианты.

• Добавить комментарии к файлу, поддерживающему сигнатуры.

• Иметь файл сигнатур, который контролировал бы появление незаконных изменений.

• Задавать адрес смещения для поиска (чтобы увеличить скорость сканирования).

• Показывать тип файла, в котором эта строка может быть найдена (опять-таки для увеличения скорости сканирования).

• Указывать, находится ли эта сигнатура в памяти, дисковом файле, загрузочной записи, таблице разделов и т. п.

• Делать все это с небольшими затратами и минимальными усилиями.

• Иметь режим удаления вируса. Имеет ли сканер режим для удаления вируса, обнаруженного в файле, который обеспечивает создание чистого файла? Обратите внимание на сканеры, которые способны удалять зараженный файл, переписывать его или переименовывать.

• Рационально использовать память. Какой объем памяти необходим продукту? Некоторые рабочие станции имеют менее 400 Кбайт свободной стандартной (conventional) памяти, не все сканеры смогут загружаться при таком объеме памяти. Уметь обнаруживать Stealth-вирусы. Обнаруживает ли сканер вирусы, применяющие новейшие Stealth-методы для маскировки, такие как несанкционированное использование прерываний (interrupt grafting) и создание туннелей (tunneling)? Обеспечивать совместимость. Может ли продукт работать в среде Windows? Если продавец утверждает, что может, сможет ли продукт обнаружить вирус при одновременной работе двух или более систем DOS? Может ли он восстанавливать инфицированные программы Windows? Может ли он работать в DR DOS при объеме стандартной памяти более 640 Кбайт? Тестирование на ложную тревогу очень важно; оно, вероятно, сможет обнаружить широкий спектр продуктов, способных вызывать ложные тревоги. Кроме того, многие сканеры создают ложную тревогу при сканировании их другим сканером. Сканеры не должны создавать ложную тревогу – ни при запуске, ни при сканировании их другими сканерами – коды сканера должны храниться в зашифрованном виде.

• Сканировать всю память. Выполняет ли продукт сканирование дополнительной (extended) памяти? Расширенной (expander

памяти)? Высшей (high) памяти? Блоков верхней (upper) памяти? Иметь автоматическое управление. Может ли антивирусный продукт обнаружить и удалить вирусы без вмешательства пользователя? Может ли продукт запускаться автоматически через определенные промежутки времени (ежечасно, ежедневно, еженедельно)? Если да – можно включить эту функцию в сценарий входа в сеть или в AUTOEXEC.BAT.

- Быть ориентированным на заказчика.
- Обеспечивать установку привилегий.
- Обеспечивать регистрацию событий. Может ли продукт вести главный журнал сети, обновлять его при каждом запуске и записывать идентификатор рабочей станции?
- Обеспечивать хранение сигнатур в файле, а не внутри кода сканера.
- Обеспечивать вывод на уровне ERRORLEVEL DOS. Это желательно для использования его в командных файлах.
- Обычно план предотвращения вирусов включает применение антивирусного программного обеспечения, чаще всего сканеров. Существует четыре вида программного обеспечения, используемого для обезвреживания вирусов.
- Интегрированные контроллеры (integrity checkers) или программы проверки контрольной суммы (checksum programs) могут определить, изменилась ли ваша программа с момента последнего запуска программы проверки контрольной суммы. Любое заражение вирусом изменяет зараженную программу, и если вируса нет в памяти, значит это изменение должна обнаружить программа контрольной суммы.
- Сканеры (детекторы) используются для идентификации вируса (если он есть) в зараженном файле или для того, чтобы проверить, не заражена ли программа. В отличие от подхода, применяемого программой проверки контрольной суммы, сканер устроен так, что может обнаружить вирус в новом программном обеспечении перед его запуском. Сканеры применяются гораздо чаще других антивирусных продуктов. Резидентные (TSR) программы и драйверы устройств могут быть установлены для наблюдения за вирусами или их действиями, а также для учета подозрительных действий (или предупреждения о них).

Программы очистки (фаги) могут удалять зараженные файлы при их обнаружении или вирус из этих файлов.

Если вы хотите достичь успеха при использовании сканера для обнаружения вируса – установите и используйте продукт пра-

вильно. После того как убедитесь, что сканер установлен правильно, выполните следующее.

1. Сделайте холодную перезагрузку с защищенной от записи дискеты, а не с жесткого диска. Поскольку вирус не может обойти защитный механизм дисковода для гибких дисков, будьте уверены, что его нет в памяти, откуда он смог бы поразить ваш сканер.

2. Запустите сканер с того диска, с которого вы загрузились, и сделайте это до запуска других программ, особенно если эти программы расположены не на том диске, с которого вы только что загрузились. Любой драйвер устройства, вызванный в CONFIG.SYS вашего жесткого диска, может содержать вирус. Просканируйте диск, с которого вы только что загрузились, и убедитесь, что он не заражен.

3. Смотрите внимательно на экран во время сканирования, чтобы заметить, не делает ли ваш сканер паузу во время обнаружения вируса.

4. Следуйте инструкциям вашего сканера. Отказ при вводе правильной командной строки может привести к тому, что вы непреднамеренно пропустите сканирование каталогов, содержащих вирус.

Рассмотрев вопрос о вирусах и мерах борьбы с ними, обратимся теперь к вопросам доступа к информации и регламентирования этого доступа.

6.2. Управление доступом и его реализация

Существует множество причин, по которым можно объяснить необходимость управления доступом к рабочим станциям сети.

• Обеспечение неприкосновенности информации личного характера, хранящейся на некоторых машинах. Все пользователи создают персональную информацию на своих персональных компьютерах. Никто бы из них не захотел, чтобы эта информация стала достоянием общественности.

• Защита конфиденциальности важной корпоративной информации. Используя средства управления доступом, вы можете хранить в секрете свои деловые планы, предписания, предложения, расчеты цен, платежные ведомости и другие секреты.

• Обеспечение целостности информации в машинах. Лишая несанкционированных пользователей доступа к вашему компьюте-

ру, вы снижаете риск нежелательных правок в платежной ведомости, получаемых счетах или других важных файлах.

• Снижение риска заражения вирусом, замедление распространения такого рода инфекции.

Управление доступом состоит не только в предотвращении доступа. Надлежащее управление доступом требует, чтобы легальные пользователи имели максимально простой доступ, а нелегальные – максимально сложный. В некоторых организациях информация в микрокомпьютерах имеет такое значение, что устранение возможности несанкционированного доступа гораздо важнее, чем предоставление возможности санкционированного. (В этом случае вы сталкиваетесь с проблемой). В обычных учреждениях легальные пользователи нуждаются в простом доступе и применении **прозрачной защиты** (защита, которая не видна пользователю, имеющему право доступа).

Одна из современных проблем защиты информации – это сложность в определении, кто и к какой информации должен иметь доступ. Отделы и службы создают информацию в поразительном темпе. Никакой офицер безопасности не в состоянии, сидя где-то в уголке офиса, эффективно принимать глобальные решения, касающиеся предоставления доступа на уровне микрокомпьютера к каждому фрагменту информации.

Решения по управлению и защите данных подразделяются по трем категориям.

• **Решения, ориентированные на сервер**. *Касаются* защиты данных, находящихся на **сервере.** Реализация этих решений (обычно в виде части файловой системы сетевой ОС) позволяет администратору системы указать определенные файлы, каталоги и тома в качестве защищенных сфер с определенными характеристиками, такими как **read-only** или **no access**. Решения, ориентированные на сервер, могут быть практичными в некоторых ситуациях (например, бездисковые рабочие станции), но, как правило, они сталкиваются с тремя проблемами. Во-первых, файлы должны физически храниться на сервере, чтобы их можно было защитить. Во-вторых, эта защита не применяется для портативных ПК или других средств мобильных вычислений. И, наконец, пользователи должны активно включиться в работу и изменить свои рабочие привычки, чтобы обеспечить эффективность этого решения.

• **Физические линии связи.** Продукты защиты физических линий связи используют шифровальные устройства на выходе из концентратора. Эти продукты разрабатываются для предотвраще-

ния перехвата данных и информации при передаче с рабочей станции на сервер. Эти решения, несомненно, обеспечивают защиту данных, пока они проходят между узлами сети, но ничего не делают для защиты данных, находящихся на рабочей станции или сервере.

• **Продукты для защиты рабочей станции.** Дополнительные продукты защиты рабочей станции защищают хранящиеся на ней данные. Эти продукты рассчитаны на определенный вид защиты файла и схему шифрования, они предоставляют или пресекают доступ к файлу, основываясь на отождествлении пользователя. Недостаток этих решений состоит в том, что большинство из таких продуктов проектируется в предположении, что защищенные данные остаются на рабочей станции, поэтому сами данные не защищаются при перемещении между пользователями с помощью таких инструментальных средств, как электронная почта.

Проблема, с которой сталкиваются системные администраторы при реализации одного или более взаимосвязанных (но архитектурно разобщенных) методов защиты, состоит в том, каким образом заставить эти решения работать совместно для создания обстановки, в которой будут защищены как постоянно хранящиеся, так и передаваемые данные, причем, не препятствуя обмену данными между пользователями.

В некоторых случаях безопасность на уровне файла или каталога реализуется путем предоставления пользователю защищаемой области (secure area) на сервере, где может размещаться важная или секретная информация. К сожалению, эта схема не согласуется со схемой работы пользователей. Анализ показывает, что некоторые, если не большинство пользователей, собирают, создают и хранят данные на локальных жестких дисках личных рабочих станций. Немногие помещают свою самую важную информацию в лоно сервера.

6.2.1. Открытая архитектура безопасности (OSA)

Как альтернатива централизованной на сервер безопасности была разработана открытая архитектура безопасности **OSA** *(Open Security Architecture).* OSA предлагает подход с централизацией на рабочей станции (workstation-centric) и разработана для реализации широкомасштабной открытой системы безопасности и защиты самых важных данных. Предполагается, что данные и их защита бе-

рут начало на рабочей станции и распространяются за ее пределы по всей организации.

В этом разделе описаны цели и средства проекта OSA, представлен список функций, обеспечиваемых OSA-совместимыми продуктами, и показано, каким образом эти функции касаются вопросов, поставленных ранее. OSA помогает определить, какие меры необходимо принять для улучшения управления и безопасности всей организации. Цели проекта OSA ориентированы на поддержку таких аспектов безопасности, как целостность данных, обеспечение доступности данных и организация санкционированного доступа к данным, и учитывают стоимостные ограничения. Реализация поставленных целей обеспечит безопасность по указанным аспектам смешанных сред персональных компьютеров, сетей и мобильных вычислений без существенного отрицательного влияния на производительность пользователя. Ниже описаны цели проекта OSA.

Цель № 1 проекта OSA. Информационный поток и мобильность OSA предполагает, что решения по управлению данными и безопасности не должны препятствовать потоку информации в вашей организации. Если пользователи привыкли посылать и получать несекретную информацию по таким каналам, как электронная почта, эта возможность должна быть предоставлена лицам, желающим передавать таким же образом и секретную информацию внутри организации. Решения по безопасности, рассчитанные на OSA, не должны препятствовать мобильности пользователей, полагающихся на портативные ПК. При принятии таких решений необходимо учитывать, каким образом мобильные системы могут вписываться в схему безопасности всей организации.

Цель № 2 проекта OSA: Период незащищенности. В соответствии с согласующимися с OSA решениями защищаемая сущность (файл) должна защищаться автоматически и незаметно при создании. Она никогда не существует в незащищенном состоянии. Этот критерий предписывает, чтобы защита обеспечивалась немедленно после того, как файл создан. Организационная политика безопасности определяет, каким образом устанавливается защита, и эта защита не должна вынуждать создателя файла предпринимать какие-либо действия. Защита сущностей во время их создания снимает ответственность с пользователя за перемещение элементов с незащищенной рабочей станции в хранилище, базирующееся на сервере.

Цель № 3 проекта OSA: Защита как неотъемлемый атрибут данных. Защита, применяемая к файлу, всегда остается с фай-

лом, независимо от того, где он размещен или как транспортируется. Защита обеспечивается в любом случае, невзирая на месторасположение файла. Например, если файл физически присоединяется к сообщению электронной почты и посылается другому пользователю, меры защиты, применяемые к файлу, не отменяются. Если файл копируется на гибкий диск и отправляется в региональный офис, защита остается с файлом. Не обеспечив защиту файла, вы рискуете потерять его.

Цель № 4 проекта OSA: Централизованное администрирование. Для того чтобы схема безопасности, реализованная OSA-совместимыми продуктами, была эффективнее по стоимости, необходимо администрирование из единого центра под управлением одного офицера безопасности. Этот офицер безопасности может реализовать правила обеспечения безопасности организации и взять под свою защиту все конечные точки (рабочие станции). Для этого необходимо принять меры по управлению конечными пунктами, являющимися локальными рабочими станциями, соединенными в сети, или удаленными машинами, не имеющими официальных средств коммуникации с центром. Для этого также необходимо принять меры по обеспечению централизованного администрирования в неоднородных средах рабочих станций и серверов. Например, из центрального пункта офицер безопасности может выполнять административные функции по отношению к рабочей станции, работающей в DOS или Windows, а также по отношению к рабочей станции, базирующейся на UNIX. Для реализации такого администрирования необходима централизованная база данных информации пользователя и рабочая станция, доступная для выполнения этих функций.

Цель № 5 проекта OSA: Интеграция с существующей инфраструктурой. Схема безопасности должна быть реализована без разрушения существующих программ, данных и сетевой инфраструктуры, имеющихся на местах. Она должна функционировать эффективно, независимо от типа сети или имеющихся транспортных возможностей.

Цель № 6 проекта OSA: Модульность. Модульность – самое эффективное по стоимости решение. Можно начать реализовывать безопасность и управление в любой точке кривой «эффективность/стоимость». Точку кривой вы выбираете в зависимости от уровня безопасности и количества имеющихся ресурсов. Очень важно помнить, что при получении дополнительной безопасности и, следовательно, привлечении дополнительных ресурсов подъем

кривой на более *высокие* уровни безопасности должен происходить плавно. Вы не должны аннулировать или исключать никаких существующих вложений или ресурсов.

6.2.2. Функции безопасности и управления, обеспечиваемые средствами OSA

Если вы построили модель перемещения данных и усвоили цели проекта OSA – можете приступить к изучению специальных функций и назначения продуктов, созданных для поддержки предлагаемой архитектуры. Основные функции OSA-совместимых и ориентированных на рабочие станции продуктов таковы.

Идентификация и аутентификация (I&A). Цель введения I&A состоит в обеспечении гарантий, что только легальные пользователи будут иметь доступ к защищаемым элементам (например, рабочим станциям, файлам и каталогам) сети. Процесс I&A происходит при включении рабочей станции, когда у пользователя запрашивается идентификатор и пароль. После того как пользователь прошел аутентификацию на рабочей станции, информация об этом в обязательном порядке передается другим компонентам системы безопасности и может поступать в любые уголки сети и подразделения организации, которые ее требуют (например, на серверы и большие ЭВМ). Таким образом, I&A должна выполняться на одном месте и по требованию передаваться по всей организации.

Для надлежащей реализации I&A система должна иметь различные уровни и режимы отождествления пользователя. На I&A уровне входа (entry level) продукты OSA проверяют пользовательский идентификатор и пароль. Это решение идеально для простых компьютерных систем и ПК, которые не могут изменить аппаратные средства.

Для компьютерных систем среднего уровня сложности (mid-range solution) I&A могут быть основаны на концепциях «о чем-то знаю, чем-то обладаю». Такие методы защиты обычно применяются в системах автоматических платежей (Automated Teller Machine). Пользователи снабжаются опознавательным знаком (token), который функционирует как безопасное хранилище пользовательской идентификации и другой информации, связанной с безопасностью, такой как ключи шифрования. Защищенная система не может быть активизирована, пока пользователь не представит опознавательный знак (нечто, чем обладает) и правильный пароль (нечто, о чем знает). При надлежащей реализации этот знак не только предоставляет

право доступа к защищенному ПК, но и используется для получения прав доступа к другим защищенным областям.

На верхнем уровне спектра I&A OSA обеспечивает метод, ориентированный на строгую I&A и использующий определенные интерфейсы прикладного программирования API (Application Programming Interfaces). Этот более сложный для реализации метод может потребовать интеграции биометрических приборов, таких как сканеры отпечатков пальцев.

Для того чтобы охватить схемой безопасности несметное множество систем, к которым может иметь доступ только один пользователь, OSA обеспечивает возможность стандартного *Ввода* (CSO – Common Sign-on). CSO может вызываться для безопасной и скрытой передачи информации об 1&A пользователя с рабочей станции на другие компоненты сети, например большую ЭВМ. В сочетании с остальными архитектурными средствами OSA эта возможность позволяет офицеру безопасности проектировать и управлять всеми процедурами подключения к центральным узлам и системам больших ЭВМ. CSO также гарантирует, что весь доступ к центральному узлу инициируется на безопасной рабочей станции и такой узел огражден от незащищенных и неконтролируемых *вхо*дов.

Защита файлов. Для управления (т. е. предоставления или лишения права) доступом защита файла использует большой объем информации о пользователе, которую мы получаем не только при I&A, но и из защищенной базы данных пользовательского профиля (profile).

Функции, обеспечивающие защиту файла, реализуются следующим образом.

- Время, в течение которого файл остается незащищенным, должно быть сведено к минимуму – как только файл создается, к нему применяется защита.
- Защита применяется к маршрутам передачи файла во время его перемещения по организации.
- Защита остается в силе неограниченное время, независимо от того, каким образом транспортируется защищенный файл.
- Целостность данных не подвергается риску. Защищенные данные доступны всем текущим приложениям, которые использовались до защиты. Например, файл электронной почты доступен приложению, независимо от применяемой защиты.
- Информация о безопасности, содержащаяся в (или при) файле должна перемещаться между системами различных типов.

Например, защита файла на DOS-ориентированной системе, интерпретированная DOS-ориентированным ядром безопасности, должна выглядеть так, чтобы при пересылке защищенного DOS файла системе UNIX защита файла оставалась неповрежденной.

Продукты, согласующиеся с OSA, для обеспечения безопасности применяют такие технические приемы, которые делают защиту и управление неотъемлемой частью самого файла. Эта защита реализуется с помощью **грифа** (label) на защищенном файле, который согласовывает идентификацию пользователя с действиями, которые пользователь может выполнять с файлом. Это похоже на конверт с указанием имени адресата, уточняющего, кто может его открыть и прочесть содержимое конверта. В следующем разделе описано, как функционирует защита файла с помощью грифа.

Когда пользователь или офицер безопасности предпринимает действия по защите файла, он должен определить три следующие элемента.

• **Владелец файла.** Владелец имеет право полностью распоряжаться файлом, ему предоставляются права полного доступа, и только он может изменить гриф файла.

• **Пользователь файла.** Пользователями являются любые лица в пользовательском сообществе, которые имеют доступ к этому файлу.

• **Привилегии.** Привилегии определяют, какие действия пользователь может выполнять с защищенным файлом. Привилегии связываются с каждым пользователем, имеющим доступ к файлу.

Информация о владельце, пользователях и привилегиях комбинируется с информацией управления, содержащейся в грифе, и присоединяется к файлу в виде заголовка. После присоединения грифа дополнительная защита может быть обеспечена с помощью прозрачного для пользователя шифрования, как самого грифа, так и полного содержимого файла.

Все доступы к защищенным файлам строго контролируются системой безопасности и согласуются с грифом. Всякий раз, когда осуществляется доступ к файлу, система безопасности проверяет, защищен ли файл. Если файл защищен, пользователю предоставляется доступ к нему только при наличии прав, соответствующих грифу.

Особенно важно отметить, что защита не изменяет общей структуры файла. Именно поэтому защищенные файлы могут обрабатываться так, как и незащищенные – в смысле сопровождения,

перемещения и расположения. Кроме того, поскольку гриф становится неотъемлемой частью файла, защита путешествует с файлом при его перемещении по организации.

Администрирование. Опыт показывает, что в тех случаях, когда административное управление слишком дорого или требует много ресурсов для обеспечения надлежащей эффективности принятого решения, – это решение, скорее всего, не будет реализовано. Эффективное администрирование безопасностью, как определено OSA, гарантирует следующее.

• **Обеспечение централизованного управления.** Самый эффективный способ реализации **безопасности** - это ее внедрение из единого центра OSA предписывает, что единственным центральным звеном безопасности (обычно это офицер организационной безопасности) должно быть лицо, имеющее возможность и обязанность реализовывать правила организационной безопасности. Обычно на это требование OSA ссылается как на централизованное администрирование (CSA – Central Site Administration).

• Помощь **доверенных пользователей офицеру безопасности.** Для того чтобы помочь офицеру безопасности при выполнении ежедневных обязанностей, OSA указывает, что офицер безопасности может окружить себя доверенными пользователями. Эти пользователи при необходимости могут выполнять определенные обязанности офицера безопасности. Используя эту схему, офицер безопасности имеет, например, возможность определить группу доверенных лиц для работы в качестве аудиторов. В обязанности аудитора входит сбор, обработка и анализ данных контрольной проверки, если в этом возникает необходимость. Эта схема помогает офицеру безопасности при выборе пользователей в подразделениях и возложении на них ежедневных обязанностей регистрации остальных пользователей. Эти доверенные пользователи могут действовать в соответствии с установленной схемой эксплуатации системы безопасности, но не могут ее изменять. Они освобождают офицера безопасности от повседневных организационных проблем и вопросов.

• **Обеспечение минимального вовлечения конечного пользователя**. Для того чтобы облегчить администрирование и реализацию схемы безопасности, как указывает OSA, необходимы процедуры для установки и изменения конфигурации безопасности и защиты. Пользователи должны выполнить простую процедуру для установки базового программного обеспечения безопасности,

офицер безопасности или доверенные пользователи должны создать конфигурацию безопасности.

• **Охват правилами организационной безопасности всех конечных пунктов.** При *увеличении* числа мобильных компьютерных систем, которые не всегда имеют доступ к локальной сети, важно охватить схемой организационной безопасности мобильные, дистанционно размещенные и локальные системы. Метод, применяемый OSA для достижения этой цели, – это распространение централизованного администрирования. Специфицируя этот метод, OSA утверждает, что конфигурации безопасности могут быть безопасным образом переданы пользователю по сети (возможно, через электронную почту или коллективно используемые файловые области) или через электронные средства передачи, использующие модем, либо через гибкий диск.

• **Обеспечение соблюдения правил обеспечения безопасности на многих платформах.** Для того чтобы охватить неоднородные сети правилами обеспечения безопасности, OSA должна применяться к рабочим станциям, имеющим различные операционные системы. Решение поставленной задачи требует реализации центрального *хранилища* (repository) информации для координации действий при регистрации пользователя и информации, связанной с безопасностью.

Ведение контроля. Функция ведения контроля, декларируемая архитектурой OSA, состоит в том, чтобы обеспечить постоянную оценку действенности схемы защиты. Ведение контроля дает офицеру безопасности набор контролируемых пара метров, которые анализируются для определения тех узких мест в схеме защиты, где возникли нарушения. На основании этой информации правила обеспечения безопасности могут быть откорректированы для устранения или предупреждения появления любых прорех в схеме защиты.

Ведение контроля предусматривает осуществление следующих функций.

• **Протоколирование критических событий.** Перечень событий, которые приводят к нарушению безопасности, должен быть определен в самом начале реализации правил обеспечения безопасности организации. Этот перечень событий используется офицером безопасности для определения диапазона информации, которую необходимо поместить в контрольный журнал. Качественный проект и реализация подсистемы ведения контроля, как указывает OSA, должны основываться на подходе, который заключается в

детализации информации. Этот подход позволяет офицеру безопасности контролировать только самые важные события и экономить пространство рабочей станции или диска сервера, используемое для хранения журнала безопасности. Он предотвращает также ситуацию перенасыщения информацией, когда ее объемы таковы, что времени, выделенного офицеру безопасности на анализ, не хватает для получения правильных результатов эффективности схемы безопасности.

• **Накопление информации для аудита.** Ведение контроля критических событий при установке требует, чтобы информация о происходящих событиях хранилась в центральном пункте управления безопасностью и затем объединялась для комплексного анализа. Архитектура OSA допускает вывод файлов контроля в стандартном, согласующемся с базами данных формате и их передачу электронной почтой или другими доступными средствами в центр. Это предоставляет офицеру безопасности возможность составить мнение об эффективности политики безопасности в масштабах всей организации, используя имеющиеся инструменты анализа и ведения баз данных.

• **Обеспечение простоты использования.** Поскольку просчеты в политике безопасности могут немедленно обнаруживаться непосредственно на рабочей станции, OSA утверждает, что офицер безопасности должен иметь в распоряжении множество простых базовых инструментов анализа. Эти инструменты ориентированы на быстрое обследование деятельности рабочей станции без разрушения или изменения каких-либо данных, которые позже могут быть объединены в едином аналитическом центре.

• **Предупреждение об опасности в реальном времени.** По сравнению с автономными компьютерами рабочие станции, объединенные в локальные сети, имеют дополнительные преимущества, обеспечиваемые постоянно действующей связью в реальном масштабе времени. Благодаря расширению возможностей существующих инструментов управления сетью и их сочетание со связанными по данным драйверами (data-link drivers) и возможностями ведения локального контроля, размещенный в центре администратор сети может получить извещение о появлении связанного с безопасностью события (такого, например, как отказ I&A на рабочей станции).

6.2.3. Создание приложений, оснащенных средствами безопасности

В связи с тем, что все больше и больше важных приложений переносится с больших ЭВМ на рабочие станции, необходимо обеспечить определенное выполнение этих приложений в контролируемой среде. OSA учитывает эту необходимость и предлагает метод для создания приложений со встроенными средствами поддержки системы обеспечения безопасности (security-aware). Такое приложение обеспечивает защиту в качестве одной из неотъемлемых составляющих своей функциональности и конструкции. При создании приложений со встроенными средствами поддержки безопасности могут предоставляться следующие возможности.

- **Использование информации I&A** Приложение выполняется только после подтверждения подлинности пользователя от ядра системы безопасности. Оно может заставить пользователя пройти процесс повторной аутентификации для гарантии того, что пользователь, выполняющий приложение, является тем же пользователем, который загружал систему и был допущен к работе.
- **Выполнение только в безопасной среде.** Приложение не запускается на незащищенной системе. Оно спроектировано так, что может выполняться только в безопасной системе либо в системе, которую офицер безопасности включил в состав определенной группы контролируемых им рабочих станций. Такое контролируемое выполнение приложения гарантирует, что при перемещении незаконным образом с одной системы на другую оно не будет в ней выполняться. Это имеет особо важное значение для критических приложений, ориентированных на рабочую станцию, в числе которых, например, приложения, манипулирующие большими суммами.

При создании приложений со встроенными средствами поддержки безопасности результаты аутентификации становятся известными системе безопасности, и такая информация, как идентификатор пользователя, его привилегии, права доступа к файлам, принадлежность к группе и т. д., может использоваться надлежащим образом для управления безопасностью всей организации.

6.3. Аутентификация: пароли, их современные разновидности

Личная ответственность – это ключ к управлению и защите любой системы, обрабатывающей информацию в интересах какого-либо лица или группы лиц. Такая ответственность требует **идентификации** (identification), производимой обычно с помощью регистрационного имени, регистрационного идентификатора, идентификатора пользователя, и **аутентификации** (authentication) пользователя (обычно выполняемой с помощью пароля). Если аутентификация завершается неудачно, идентификация недействительна. Пароль – самый популярный метод аутентификации. И, как следствие, – самое слабое место в сети для успешного несанкционированного проникновения в нее.

Фактически все компьютерные преступления в той или иной мере связаны с нарушением парольной защиты. Многие представители правосудия не считают пароль **обоснованной** и **достаточной мерой** компьютерной безопасности, поскольку проблемы безопасности, вызванные сбоем системы паролей, могут быть следствием ошибки, а не результатом вмешательства нарушителей.

Пароль (password) – это код, используемый для получения доступа к системам или файлам, оснащенным парольной защитой. Действенность защиты, обеспечиваемая системой паролей, в значительной степени зависит от сохранения паролей в тайне. Пароль подвержены опасности разглашения во время их использования, хранения, а также при сообщении их лицам, использующим эти пароли. Эта глава содержит рекомендации для предотвращения таких нарушений.

Несмотря на множество способов разрушения парольных систем, не следует делать вывод, что пароли бесполезны. Они достаточно эффективны при разумном использовании.

6.3.1. Некоторые общие решения по проблеме паролей

Парольная защита – вещь полезная, но многочисленные пути ее преодоления требуют, чтобы была создана не одна линия обороны. Этот раздел представляет некоторые варианты избыточных мер по защите – дополнительные средства аутентификации пользователей.

Парольная защита приложений. Многие системы управления доступом к ПК требуют ввода паролей для доступа. Поищите

такого рода программное и аппаратное обеспечение и попытайтесь преодолеть его парольную систему (например, на уровне операционной системы).

Синтезаторы речи. Применение речевого синтезатора допускает множество мощных приложений. Можно, например, установить синтезатор в службе управления и использовать последовательные кабели и распределитель для его соединения с множеством компьютеров. Когда нелегальный пользователь на любой машине попытается начать сеанс работы, синтезатор сообщит номер машины, ее расположение и характер проблемы.

Сигнатуры в паролях. Встраивайте сигнатуры в парольные системы. Все пользователи при печати оставляют отпечатки пальцев или **сигнатуру** (signature). Хотя система может исследовать многие аспекты этой сигнатуры, такие как использование заглавных букв, пунктуация и т. п., самый простой атрибут сигнатуры для анализа - это время ожидания при нажатии заданной клавиши. Скорость ввода различных последовательностей с клавиатуры уникальна для каждого пользователя, а профиль нажатия клавиш может быть сопоставлен с пользовательскими профилями, хранящимися в базе данных, для того, чтобы определить, является ли лицо, производящее ввод с клавиатуры, несанкционированным пользователем. Для сопоставления могут использоваться методы корреляции, и может быть установлен порог для принятия решения о том, разрешать или запрещать попытки пользователя войти в систему.

Модемы с обратным дозвоном. Модемы с обратным дозвоном (dial-back modems) представляют другое решение проблемы доступа. Эти модемы аутентифицируют пользователей, которые производят вызов из удаленных пунктов. Когда пользователь обращается к вашей сети через модем, модем с автоматическим обратным дозвоном фиксирует имя пользователя, кладет трубку, ищет в данных системы номер телефона, по которому должен звонить этот пользователь, и пытается соединиться с ним. Если номер пользователя на другом конце линии связи не отвечает, модем прерывает попытку несанкционированного доступа к системе.

Опознавательные знаки. Опознавательные знаки (token) - это физические ключи или магнитные карты, которые пользователь вставляет в считывающее устройство. Пользователь может вставить ключ до входа в систему, чтобы получить доступ к ПК, который обладает таким считывающим устройством, либо выполнить это во время входа в сеть для завершения процесса регистрации. Некоторые системы используют личные кредитные карточки поль-

зователя для аутентификации. Системы опознавательных знаков могут быть такими же строгими, как и генераторы паролей, если вы соедините их с паролем или с PIN. Если использовать одну лишь карточку, ее кража представляет ту же угрозу безопасности, что и кража пароля.

Пользователям не нужно знать свои пароли, если пароль и другие уникальные идентификаторы хранятся на **интеллектуальной карте** (smart card), снабженной собственным микроконтроллером. Микроконтроллер передает идентифицирующую информацию, которая проверяется считывающим устройством.

Применение интеллектуальных карт решает ряд традиционных проблем с паролями.

- Пользователю не нужно беспокоиться о том, что он может забыть свои пароли.
- Пароли могут быть длинными и трудно угадываемыми.
- Пользователи не могут совместно использовать пароли, если отдают во временное пользование свои интеллектуальные карты другим пользователям.

PC-Card – устройство для считывания карт, которое просматривает стандартные коммерческие банковские и кредитные карточки, применяя шифрование IATA Track-1 или ABA TRACK-2 на магнитной полоске. Кабель от клавиатуры ПК подсоединяется к блоку PC-Card, который крепится к клавиатуре. PC-Card самостоятельно присоединяется кабелем к разъему клавиатуры ПК с обратной стороны машины. В результате этого информация на магнитной полоске может считываться микрокомпьютером как ввод со стандартной клавиатуры и является частью схемы парольной защиты. Пользователи могут совместно использовать пароли или разглашать их, но вряд ли они оставят где-нибудь свои карточки American Express, чтобы ими могли воспользоваться другие. PC-Card стоит *$395*.

Интеллектуальные карты могут быть утеряны или украдены, но это не столь уж серьезная проблема. Банки решили эту проблему для банкоматов (ATM), требуя, кроме того, ввода запоминаемого пароля – персонального идентификационного номера (PIN – Personal Identification Number). Интеллектуальная карта не работает без PIN, а PIN не работает без интеллектуальной карты. Вы можете использовать тот же подход для защиты компьютера.

6.3.2. Персональные данные и устройства биометрического управления доступом

Термин **биометрия** происходит от греческих слов, означающих жизнь и измерение. **Биометрическое управление доступом** (BAC – Biometric Access Control) – это использование одной или более уникальных индивидуальных особенностей строения человеческого тела для того, чтобы убедиться, что вы – это вы, а не самозванец. Самый известный метод – дактилоскопия отпечатков пальцев. Но, кроме него, может использоваться геометрия руки, глазное давление, степень нажатия при написании имени, а также образцы голоса. Устройство BAC идентифицирует человека по уникальным характеристикам его тела и на основании этого разрешает или ограничивает доступ к системе.

Мы можем легко представить типы продуктов BAC, которые можно было бы изобрести. Предположим, что вы сидите за своим терминалом и ваша машина разрешает вам приступить к работе. Ваш стул, связанный с машиной, оценил ваш вес и отправил информацию устройству. Ваш сегодняшний вес был сопоставлен со вчерашним, и вы были отождествлены как «свой». Самая сложная модель стула могла бы измерять и сравнивать с образцом также давление на спинке стула, чтобы получить оценку степени вашей сутулости. Эта дополнительная информация подтверждает, что человек, сидящий на стуле – это действительно вы.

Вы можете использовать все виды устройства BAC для охраны комнаты или здания. Некоторые из них могут охранять отдельные рабочие станции. Поскольку такие устройства очень дороги, вам лучше было бы выбрать устройство, способное защитить сразу несколько пользователей. Фирма Recognition Systems предлагает считывающее устройство для геометрии руки, которое работает в пределах комнаты. После сканирования оно назначает пользователю временный, случайно сгенерированный пароль, который действует на машине пользователя всего несколько минут, предоставляя ему доступ к системе.

Использование устройства BAC – двухэтапный процесс.

- При приеме на работу устройство создает генерируемый машиной шаблон, который сравнивает с образцом человека, желающего получить доступ.
- При подключении к системе устройство анализирует образец пользователя и сравнивает его с шаблоном, который был создан при приеме на работу. Большинство устройств требуют, чтобы

пользователь ввел код идентификации до того, как устройство предоставит права доступа к системе. И код, и образец сопоставляются с информацией, которая хранится в устройстве по каждому пользователю.

Если система часто отказывает легальному пользователю, он может заново пройти процедуру допуска к работе. К сожалению, устройства BAC имеют следующие недостатки.

- Они очень дороги, обычная начальная цена – $1000.

- Устройства BAC несовершенны. Из-за высокого уровня ошибок эти устройства могут не допустить к работе хороших парней и в то же время могут беспрепятственно пропустить плохих. Теперь все устройства усовершенствованы, уровень ошибок находите в пределах 1%. Некоторые продукты, такие как VoiceKey фирм Ecco, стали значительно лучше после повторного применения. Однако если устройство допускает сбои в работе при допуске хотя бы одного легального пользователя из ста, а вы пропускаете через него по 500 пользователей каждое утро, кое-кому придется садиться и подниматься со своего стула по пять раз, чтобы определить, в чем проблема. Пользователи не в восторге от такого положения дел и вы, конечно, тоже, особенно если все обедают не на рабочем месте.

- Защиту устройства BAC легко обойти. В этом плане самый легкий путь – замаскироваться под респектабельного хорошо одетого посетителя, подходящего к двери с тяжелым пакетом в час пик, когда многие служащие проходят мимо. Большинство легальных пользователей придержат дверь перед незваным гостем. Можно бороться с такими злоумышленниками, используя вращающиеся двери, которые пропускают людей по одному, однако вращающиеся двери не позволяют применять ручные тележки, кресла на колесах и т. п. Другой довольно легкий путь обмануть любое устройство – засунуть жевательную резинку в устройство, отойти на несколько минут, а затем вернуться. Кто-то из входящих обнаружит проблему, вызовет помощь и дверь будет открыта, пока организация не дождется прихода ремонтника и чистки недремлющего ока. Бороться с этой хитростью можно с помощью нескольких скрытых камер, расположенных около двери, которые зафиксируют момент, когда злоумышленник засовывает резинку в устройство.

- На некоторых пользователей устройство BAC первое время наводит страх.

Несмотря на наши оговорки о недостатках этого оборудования, в следующих разделах мы рассмотрим некоторые устройства BAC.

Устройства считывания отпечатков пальцев. Устройства считывания отпечатков пальцев идентифицируют личность по форме и числу деталей – точек начала и конца линий на пальце. К такого рода продуктам относятся Thumbsoan AMS-PC (фирмы Access Management System) и Ridge Reader Mint (фирмы Fingermatrix).

Сканеры сетчатки глаза. Эти устройства сканируют образцы сетчатки глаза пользователя, сосредоточиваясь на уникальных кровеносных сосудах. С помощью инфракрасного излучения с яркостью лампочки рождественской елки берутся данные по 300 точкам в области сетчатки глаза, и собранная информация преобразуется в число. Пользователю с контактными линзами нет необходимости их снимать, но пользователи в очках должны их снять перед использованием сканирующего устройства. Все ведущие организации, производящие средства безопасности, считают такие продукты надежными, а большинство потребителей считают их слишком дорогими - зачастую по $6000 за штуку. Одним из продавцов является Eye-Denfity.

Устройства верификации голоса. Устройства верификации голоса конструируют математическую модель вокального диапазона говорящего и используют ее для сличения с образцом голоса. Это гарантирует, что лица, находящиеся под домашним арестом, действительно находятся там, где им и надлежит быть, а пользователи компьютеров действительно те, за кого себя выдают. Одним из продуктов этого класса является VoiceKey фирмы Ecco. Цены на него колеблются от $1000 до 1500. Если вы решите приобрести такое устройство, убедитесь, что его нельзя обмануть с помощью магнитофона.

Устройства считывания геометрии руки. Устройства считывания геометрии руки используют свет для построения трехмерного изображения руки человека, проверяя такие характеристики, как длина и ширина пальцев и толщина руки. Эти устройства продает фирма Recognition System по цене $3500.

Устройства распознавания подписи. Эти устройства анализируют такие характеристики, как давление пера, очертание символа и скорость письма. Фирма AutoSi предлагает продукт Sign/On, который сравнивает координаты X-У с шаблоном, хранящимся на диске. Цена зависит от модели и колеблется от $700 до 1300.

Генераторы паролей. Другим инструментом для парольного управления доступом является **генератор паролей** – маленькое устройство, создающее одноразовые случайные пароли для пользователей. Для доступа к системе пользователю необходимо иметь

устройство и ввести пароль. Портативные генераторы паролей – это хорошая альтернатива модемам с обратным дозвоном. Каждый генератор использует уникальный ключ, связанный с PIN его владельца. Пользователь вводит PIN в ответ на запрос **устройства управления доступом к сети** (NACD – Network Access Control Device), а потом **получает пропуск** (Challenge) от NACD. Генератор паролей формирует уникальный пароль, который затем вводится пользователем. Если пароль совпадает с тем, чего ожидает NACD, NACD предоставляет пользователю доступ к центральному узлу или серверу.

Этот подход строже по сравнению с системой, основанной только на паролях, и не требует обратного автоматического вызова (или связанных с ним проблем, например, попытка войти в вашу систему из комнаты отеля).

Система SecurelD фирмы Security Dynamics из Кембриджа (Массачусетс) использует небольшой прибор размером с кредитную карточку с процессором, который генерирует новый пароль каждые 60 с. Его двойник в локальной сети занят генерацией тех же паролей и с той же периодичностью (это волшебство!). Таким образом, два пароля совпадают, если пользователь имеет достоверную карту. Эта система должна использоваться в сочетании с запоминанием паролей, чтобы кража карточки не привела к проникновению в вашу систему.

6.3.3. Основные принципы управления парольным доступом

Многие проблемы, связанные с паролями, происходят по одной причине – система паролей часто рассматривается как решение по безопасности, которое реализуется раз и навсегда. Вы принимаете решение по проблеме и пускаете ее дальше на самотек. Лучше всего запомнить закон энтропии: «Все в мире идет на убыль». Что применительно к проблемам защиты должно означать – если сегодня парольная система является превосходной мерой защиты, завтра она будет только адекватной и уж совсем никудышной в пятницу. Основные принципы, изложенные в этом разделе, основаны на документе Министерства обороны Password Management Guidelines.

Обязанности руководителя службы безопасности. Руководитель службы информационной безопасности имеет примерно

столько же обязанностей, сколько и пользователь при управлении паролями, но их ответственность различна.

Пароли в устанавливаемой системе. Многие сети устанавливаются с несколькими стандартными идентификаторами пользователя (SYSTEM, SUPERVISOR, GUEST, TEST, MASTER и т. д.), заранее включенными в системный список. Изменяйте пароли (или создавайте пароли, если их еще нет) для всех стандартных идентификаторов пользователя до того, как разрешите большинству пользователей доступ к системе. Вы можете гарантировать, что всегда поступаете именно так, если изначально идентифицируете стандартные пароли идентификатором пользователя, установив *срок их действия.*

Первое создание паролей. Администратор сети несет ответственность за создание и назначение начального пароля для каждого идентификатора пользователя. Затем администратор должен сообщить пользователю этот пароль. Однако в некоторых случаях необходимо скрыть пароль от администратора сети. (Администратор сети не должен знать, что находится в файлах «Проводки» или «Зарплата», важно только, чтобы они не подвергались опасности). В других случаях пользователь может легко предотвратить это разоблачение.

Предотвращение раскрытия. Для того чтобы предотвратить раскрытие пароля со стороны администратора сети, необходимо испробовать три надежных способа.

- Пароль пользователя должен определяться в опечатанной, многостраничной форме так, чтобы он не был виден на титульном листе формы. Администратор сети должен надлежащим образом защитить опечатанный пароль, пока тот не будет доставлен пользователю. В этом случае пароль генерируется системой случайно и неизвестен администратору сети. Пароль должен быть опечатан так, чтобы он не был виден и его нельзя было увидеть без разрушения печати. Для того чтобы доставить пароль данным способом, может понадобиться несколько дней.
- Присутствие пользователя при генерации пароля. Администратор сети инициирует процедуру, а затем пользователь защищает созданный пароль и удаляет его с экрана. Этот метод нельзя применять, если терминалы пользователей удалены.
- Применение такой системы, как NetWare, в которой администратор сети необходим только для установки минимальной длины пароля при создании пользовательского бюджета. При первом входе в систему пользователю сообщается, что истек срок дей-

ствия пароля и необходимо ввести новый. В этот момент пользователь вводит выбранный пароль в соответствии с любым принципом, который вы можете обеспечить.

Если пароли выбраны и доведены до пользователей, администратор сети должен требовать подтверждения того, что пароль получен в установленное время.

Аннулирование раскрытия. Если необходимо сообщить исходный пароль администратору сети, это разглашение можно аннулировать, немедленно изменив пароль с помощью обычной процедуры. Предполагается, что процедура изменения не сообщает новый пароль администратору сети.

Когда исходный пароль пользователя не защищен от раскрытия со стороны администратора сети, система должна идентифицировать идентификатор пользователя как *пароль, утративший силу.* Пользователь может изменить пароль обычной процедурой до получения права доступа к системе.

Допуск на изменение пароля. Изредка пользователь забывает пароль или вдруг возникает подозрение, что произошла утечка пароля пользователя. Во избежание этих проблем необходимо разрешить администратору сети изменить пароль любого пользователя путем генерации нового пароля. Администратор сети не должен знать пароль пользователя для генерации нового пароля, но он должен следовать тем же правилам для распространения нового пароля, что применяются при назначении исходного пароля. Администратор сети, несомненно, должен идентифицировать пользователя, когда нужно изменить забытый пароль. Не следует сообщать пароли по телефону, если вы не можете однозначно идентифицировать пользователя по голосу.

Идентификаторы групп. На весь срок существования сети каждый идентификатор пользователя должен быть назначен только одному человеку. Иными словами, два человека *не* могут иметь один и тот же идентификатор пользователя в одно и то же (и даже в разное) время. Вы должны расценивать как нарушение безопасности тот факт, что два или более пользователей знают пароль для определенного идентификатора пользователя (за исключением случая, когда этим вторым человеком является администратор сети, и идентификатор пользователя идентифицируется системой как пароль, утративший силу). Заметьте, здесь не запрещаются альтернативные формы идентификации пользователя (например, идентификатором группы или по должности) в целях, не связанных с аутентификацией (например, управления доступом к данным и почте).

Если использованы альтернативные идентификаторы, они должны основываться на идентификаторах пользователя.

Перепроверка идентификатора пользователя. Администраторы сети должны нести ответственность за разработку процедуры, с помощью которой они уведомляются, что идентификатор пользователя и пароль нужно удалить из сети (например, когда служащий покинет организацию). Они должны периодически перепроверять все идентификаторы пользователей и, при необходимости, обновлять номера телефонов и почтовые адреса. Эта перепроверка должна выполняться, по крайней мере, ежегодно.

Обязанности пользователя. Безопасность начинается с рабочего места. Если она там не начинается, значит, она там заканчивается. Убедитесь в том, что пользователи в одной с вами команде безопасности. Следующие разделы описывают минимальные требования для игроков такой команды.

Воспитание ответственности за безопасность. Пользователи должны осознавать свою личную ответственность за сохранность паролей и за протоколирование изменений в их пользовательских статусах любых предполагаемых нарушений безопасности и т. д. Для того чтобы гарантировать безопасность в пользовательской среде, требуйте, чтобы каждый пользователь подписывал свои обязанности, которые он понимает и признает.

Изменение паролей. Для того чтобы избежать утечки пароля, необходимо изменить его. Поэтому необходимо периодически изменять пароли, чтобы предотвратить возможность невыявленной утечки паролей. Изменяйте их достаточно часто, чтобы вероятность утечки была приемлемо малой. Пользователи должны изменять свои пароли без вмешательства администратора, чтобы избежать нежелательного разглашения паролей пользователей администратору сети.

Срок действия пароля. Самой очевидной угрозой безопасности со стороны парольной системы является утечка паролей. Чем больше срок действия, в течение которого пароль используется в целях аутентификации, тем вероятнее возможность утечки. В хорошей парольной системе вероятность утечки пароля возрастает при длительном применении пароля. Чем меньше срок действия, тем меньше вероятность утечки, но при более длительном использовании она непомерно высока. В этом случае вы должны с предубеждением относиться к использованию этого пароля и не полагаться на него безоглядно, доказывая вашу идентичность. Вы можете поддерживать приемлемый уровень уязвимости пароля,

ограничив время, в течение которого пароль можно использовать (срок действия пароля).

Устанавливайте максимальный срок действия для всех паролей. Для того чтобы защитить пароль от неизвестных угроз, установите максимальный срок службы пароля не более одного года. Опасность со стороны известных угроз может указывать на необходимость сокращения максимального срока действия – возможна до месяца или квартала. Кроме того, в зависимости от размера пароля и от того, как быстро нарушитель может войти в систему, вам придется изменять пароли еще чаще.

Аннулируйте пароль по истечении максимального срока. В установленный период времени до истечения срока действия пароля система должна сообщить пользователю, что пароль скоро утратит силу, затем потребовать у пользователя, который входит в систему с соответствующим идентификатором, изменить пароль перед тем, как ему будет разрешен дальнейший доступ к системе. Если пароль не изменен до истечения максимального срока службы, система должна заблокировать идентификатор пользователя. Заблокированный идентификатор пользователя не имеет права доступа к системе, но администратор сети может разблокировать идентификатор пользователя, изменив пароль для этого идентификатора, следуя тем же правилам, которые применяются при входе с исходным паролем. После того как пароль изменен, необходимо установить максимальный срок действия пароля, принятый системой.

Допуск на изменение. Для гарантии секретности пароля пользователям, не являющимся администраторами сети, должно быть разрешено, изменять только собственные пароли. Для того чтобы система признала пользователя, он должен ввести старый пароль, а затем идентификатор пользователя и пароль перед вводом нового пароля.

Процедура изменения. Процедура изменения пароля безопасным способом включает в себя следующее.

1. Вызов процедуры по запросу пользователя или при входе пользователя в систему с утратившим силу паролем. Если пароль пользователя утратил силу, система должна сообщить ему об этом.
2. Предоставление краткой информации о главных шагах по изменению пароля, включая предостережение от реализации любых шагов, пока не сделаны предыдущие.
3. Повторная аутентификация личности с помощью ввода пользователем текущего пароля. Это необязательно, если процедура

изменения является частью процедуры регистрации в системе (регистрации в системе с утратившим силу паролем).

4. Отображение нового пароля пользователю. Он должен отличаться от старого и должен быть сгенерирован по алгоритму. Пусть пользователь введет новый пароль дважды для того, чтобы система смогла убедиться, что пользователь может вводить правильный пароль.

5. Сокрытие нового пароля такими методами, как забивка символами или очистка экрана терминала. Большинство операционных систем сети позволяют это сделать автоматически.

6. Исправление базы данных паролей только в том случае, если два введенных пароля идентичны сгенерированному паролю. Затем удаление или отметка недействительного пароля и назначение данному идентификатору пользователя нового пароля. Вывод сообщения об этом действии на экран.

7. Если пользователь ошибается при вводе текущего пароля или при создании нового пароля, система должна сообщить пользователю об ошибке и завершить процедуру аварийно без изменения пароля. Если вы не можете изменить утративший силу пароль, старый пароль должен получить статус утратившего силу, и пользователю должна предоставляться возможность снова попытаться изменить пароль или выйти из системы. В контрольном журнале должна генерироваться запись, указывающая, изменен ли пароль.

6.3.3. Процедура регистрации через подключенную систему

Когда пользователи входят в вашу сеть из другой подключенной к ней сети, вы должны требовать от них аутентифицировать свою идентичность во время регистрации в системе, предоставив им пароль наряду с идентификатором пользователя. Используйте один из следующих методов:

- Метод, в основе которого лежит некоторая форма доверенной идентификации, пересылаемой между двумя центральными машинами.
- Метод, основанный на удаленной центральной машине, требующую идентификатор пользователя и пароль. Заметьте, что идентификаторы пользователя на разных узлах могут быть различны для одного и того же пользователя, и соответствующие сгенерированные машиной пароли, наверняка, отличаются. Заметьте также, что пароль, требуемый для удаленного центрального узла,

уязвим при перехвате со стороны локального или промежуточного узла.

Запоминание паролей. Поскольку пользователи должны передавать свои пароли сети во время аутентификации, им необходимо запомнить их. Пользователи должны помнить, а не записывать пароли на каком-то носителе. Если пароли нужно все же записать, их следует надежно защитить, чтобы не подвергать риску рассекречивания. Самым надежным местом записи пароля является обратная сторона кредитной карточки пользователя. Пользователи редко оставляют свои кредитные карточки без присмотра, как правило, всегда имеют их при себе. Если пароль записывается на кредитную карточку, он не должен сопровождаться информацией о системе, с которой работает. Пароль при записи на карточке можно записать в обратном порядке, чтобы предотвратить беспрепятственное использование потерянных карточек.

6.3.4. Механизмы аутентификации

Аутентификация может быть обеспечена различными механизмами, многие из которых могут использоваться одновременно.

Внутреннее хранение паролей. Обычно идентификаторы пользователей, пароли в том или ином виде, а также, при необходимости, и уровни доверия и допуска, связанные с каждым идентификатором пользователя, хранятся внутри самой сети без определенной формы управления доступом, основанной на использовании этой информации, нелегальные пользователи могут читать и модифицировать базу данных паролей. Вы должны постоянно помнить о возможности несанкционированного чтения и записи базы данных паролей. При чтении базы данных нелегальные пользователи могут узнать пароли. При записи в базу данных, например, пользователь *A* изменит пароль пользователя *B* так, что сможет войти в систему как пользователь *B*. Отметьте, что процедура входа в систему должна иметь возможность читать базу данных паролей, а процедура изменения паролей должна иметь возможность как чтения, так и записи в базу данных паролей.

Вы должны защищать сохраняемые пароли с помощью процедур управления доступом, предоставляемых сетью, путем шифрования паролей, или используя оба метода одновременно. Эти процедуры управления защищают базу данных паролей от несанкционированной модификации и утечки информации.

Использование шифрования. Вы должны шифровать сохраняемые пароли всякий раз, когда механизмы управления доступом, обеспечиваемые сетью, неадекватны для предотвращения разглашения паролей. Вы можете также использовать шифрование паролей даже тогда, когда считаете, что другие процедуры управления доступом адекватны. Шифрование защитит от возможного раскрытия пароля в том случае, если возможен обход процедуры управления доступом, например, при получении системных дампов. Зашифровывайте пароли немедленно после их ввода и немедленно после шифрования очищайте память, содержащую открытый текст пароля. Вы не должны расшифровывать пароли для сравнения. Система может сравнивать пароли путем шифрования пароля, введенного при входе в систему, и сравнения зашифрованной формы с зашифрованным паролем, хранящимся в базе данных паролей.

Ввод. Система должна требовать, ввод пароля после ввода идентификатора в систему. Если введенное значение правильно, система должна вывести на экран терминала дату и время последнего входа в систему этого пользователя.

Система не должна отображать вводимые пользователем пароли. Когда система не может этого предотвратить, для скрытия напечатанного пароля до или после ввода пароля на том месте, где он должен появиться, система должна вывести произвольную последовательность случайных символов.

Введенный пользователем пароль должен соответствовать паролю пользователя, который хранится в базе данных.

Передача. При передаче пароля из терминала пользователя на компьютер, на котором происходит аутентификация, вы должны защитить его одним из способов, адекватным ущербу, к которому может привести его разглашение. Пароли уязвимы не более чем данные, к которым они обеспечивают доступ. Таким образом, при передаче вы не должны обязательно защищать их как-то больше (например, путем шифрования), чем защищаете обычные данные.

Интенсивность попыток входа в систему. Контролируя интенсивность, с какой могут происходить попытки входа в систему (где каждая попытка состоит в угадывании пароля), вы ограничиваете количество угадываний, которое можно определить установленной верхней границей срока действия пароля. Для контроля нарушений, когда злоумышленник делает многократные попытки входа в систему через один и тот же порт, вы должны управлять интенсивностью угадывания пароля, на основе числа попыток на порт *доступа,* т. е. вы должны управлять каждым портом доступа

индивидуально для ограничения интенсивности, с какой злоумышленник может пытаться войти в систему с каждого порта. Когда нарушитель может свободно переключаться между разными портами доступа, вы должны контролировать интенсивность угадывания пароля на основе числа попыток на **идентификатор пользователя**.

Для интенсивности попыток входа в систему устанавливается диапазон от одной в секунду до одной в минуту. Этот диапазон обеспечивает достаточно дружелюбный интерфейс для пользователя, не допуская такого большого числа попыток входа в систему, при котором системе пришлось бы устанавливать слишком большие пароли, или слишком короткий срок службы пароля.

Отметьте, что никто не стремится замедлить вход в систему, и нет причины задерживать успешный вход в систему. Однако в случае неудачной попытки входа в систему допустимо использование внутреннего таймера, устанавливающего задержку до разрешения следующей попытки входа в систему. Пользователь не должен иметь возможности обойти эту процедуру.

Ведение контроля. Ведение контроля является не очень привлекательной процедурой и, возможно, не входит в «первую десятку списка самых любимых дел на работе». Но ведение контроля полезно, если нужно знать, кто и когда работает с вашей системой.

Контрольные журналы. Система должна создать контрольный журнал применения и изменения паролей. Такой журнал не должен содержать действительных паролей или строк символов, представляющих неправильные варианты паролей, поскольку эта информация может выдать пароль законного пользователя, который ввел с ошибкой свой идентификатор пользователя или пароль. В контрольный журнал попадают следующие события: успешный вход в систему, неуспешные попытки входа в систему, использование процедуры изменения паролей и блокирование идентификатора пользователя до конца срока действия его пароля. Каждое событие фиксируется с указанием даты и времени события, типа события, идентификатора пользователя для неуспешных входов в систему или действительного идентификатора пользователя для других событий и источника события (терминал или идентификатор порта доступа). Записи контрольного журнала изменений пароля должны указывать, успешны ли изменения.

Оперативное уведомление обслуживающего персонала системы. При пяти последовательных неудачных попытках входа в

систему с одного **порта** доступа или с одним идентификатором пользователя должно посылаться немедленное уведомление оператору или администратору сети. Хотя администратор или оператор сети не должен немедленно предпринимать какие-либо действия при получении уведомления, частота получения уведомлений может указывать, что попытка преодоления защиты развивается и может служить основанием для исследования и, возможно, корректирующего действия.

Уведомление пользователя. При успешном входе в систему пользователь должен быть уведомлен о следующем.

- Дате и времени последнего входа в систему пользователя.
- Расположении пользователя (определенном как можно точнее) при последнем входе в систему.
- Каждой неудачной попытке входа в систему, сделанной с идентификатором пользователя, после последнего успешного входа в систему.

Такое уведомление помогает пользователю определить, использовал или пытался кто-то угадать этот идентификатор пользователя или пароль.

Защита паролей. Вы можете использовать и другие средства защиты паролей, а не просто просить пользователей не писать свои пароли на мониторах. Можно достичь определенного уровня защиты с помощью вашего друга-компьютера, но иногда вы прибегаете к старомодному ручному управлению.

Вероятность одного угадывания. Вероятность того, что одна попытка угадывания пароля будет успешной, является одним из критических факторов парольной системы. Эта вероятность зависит от размера возможных паролей и статистического распределения реально используемых паролей. Поскольку многие пароли, созданные пользователем, очень легко угадать, все пароли должны генерироваться машиной.

Назначение паролей. При распределении паролей пользователям вы должны защищать пароли в той же степени, что и информацию, к которой они обеспечивают доступ. При изменении паролей система должна отображать сгенерированные машиной пароли на терминале пользователя вместе с соответствующими предупреждениями пользователю о защите пароля. Завершив процедуру изменения, система должна очистить или затереть отображаемый пароль, в зависимости от того, что больше подходит для данного типа рабочей станции или терминала. Когда администратор сети изменяет пароли, способ распределения должен быть со-

поставим с ущербом, который может быть причинен, если они будут разглашены.

Нападения на парольные системы принимают различные формы. Успешные нарушения могут быть ослаблены, поскольку пароли в основном предотвращают несанкционированный доступ к сетям. Поэтому вы должны использовать разнообразные меры для гарантии того, что защита, обеспечиваемая вашей парольной системой, настолько крепка, насколько это возможно. И, делая все, что в ваших силах, для гарантии совершенства системы, вы должны все время помнить, что она еще не совершенна. Более того, помните, что основным законом Вселенной является вырождение, и что закон вырождения применим и к паролям: *«Что кажется совершенным сегодня, будет адекватным завтра и абсолютно недостаточным в следующий понедельник».* Не закрывайте глаза на проблемы, которые могут возникнуть со временем, например: написанные на видном месте пароли, дополнительные администраторы, программное обеспечение для входа в систему с помощью троянского коня и др.

Обнаружение. Если вы предупреждаете проблему проникновения в парольные системы, используя данные в этой главе советы, вам не следует беспокоиться об их обнаружении. Поэтому можно сделать следующий вывод. Схема парольной защиты может не сработать, если пользователи, боясь забыть свои пароли, запишут их на столе или на компьютере, поместят записи в выдвижной ящик стола, вывесят пароли на шляпной ленточке, используют в качестве паролей свои инициалы или дату рождения или (как произошло в одной компании) поместят их на доске объявлений.

Но утечка пароля – это не всегда ошибка пользователя. Существуют как подпольные, так и коммерческие утилиты для нарушения парольной защиты.

Как ответственный за сохранность информации в организации, вы можете усилить защиту, усовершенствовав систему идентификации и аутентификации пользователей.

Начните с физической защиты сервера. Закройте сервер в отдельной комнате и убедитесь в том, что являетесь единственным обладателем ключа.

Просматривайте систему при поиске резидентных программ или троянских коней, предназначенных для перехвата паролей. Независимо от того, удастся вам или нет найти такие программы, примите меры для защиты системы от них. Установите требования к минимальной длине и множеству символов в паролях.

Если ваша фирма допускает удаленную обработку данных, используйте модемы с обратным дозвоном.

Если фирма, в которой Вы будете работать, имеет средства, предложите вложить их в интеллектуальные карты, опознавательные знаки, биометрические устройства управления доступом.

Активно включайтесь в управление паролями. Убедитесь, что сеть установлена корректно - в смысле периодических изменений и сроков действия паролей. Если вы вносите какие-либо изменения, проконтролируйте пользователей, бюджеты которых были активными перед изменениями, поскольку дополнительные ограничения, которые вы вносите в систему, будут влиять только на последующее определение параметров новых пользователей.

Установите процедуру для периодического изменения паролей и научите пользователей ею пользоваться. Пользователи должны отвечать за защиту своих паролей и за их изменение при подозрении утечки пароля. Найдите журнал ведения контроля под кучей бумаг на столе и обязательно просмотрите его.

Несмотря на сказанное, пароли являются только первым эшелоном обороны. Они не защищают передаваемые данные, потому что для их перехвата достаточно только пары зажимов типа "крокодил" и кассетного магнитофона.

7. Шифрование и цифровая подпись

Шифрование *(encryption)* – это мощная алгоритмическая техника кодирования, используемая для создания компьютерных книг. Шифрование защищает файлы вашего компьютера и передаваемую по сети информацию от глаз пользователей, не имеющих права доступа к такой информации. Выполняется шифрование с помощью преобразования данных к такой форме, в которой они могут быть прочитаны только с помощью специального ключа дешифрования.

Любая мало-мальски, приличная система защиты данных должна иметь в своем составе подсистему шифрования (даже некоторые сетевые операционные системы приятно удивляют нас тем, что используют шифрование данных). Компания Computer Intelligence, специализирующаяся на маркетинговых исследованиях, изучив работу 6000 различных предприятий США с ноября 1986 г. по ноябрь 1988 г., установила, что 29% всех компаний и банков, работающих в основных отраслях экономики, используют системы защиты данных. Самыми многочисленными пользователями систем

защиты являются правительство (16%), а также медицинские и учебные учреждения (11%).

Шифрование снижает опасность несанкционированного доступа, обеспечивая следующие свойства системы управления данными.

• **Конфиденциальность** (Confidentiality). Это означает, что каждый пользователь может быть уверен в сохранении секретности данных. Хотя право шифрования данных может иметь целая группа пользователей, в то же время только лицо, имеющее ключ дешифрования, может просмотреть закодированный файл. Таким образом, шифрование обеспечивает сохранение секретности при передаче информации по обычным каналам.

• **Аутентификация** (Authentication) – механизм проверки права доступа для ввода данных и предотвращение утечки при их передаче.

• **Целостность** (Integrity). Метод проверки «монолитности» потока данных. В соответствии с основами компьютерной гигиены шифрование защищает данные от вирусной инфекции.

• **Управление доступом** (Access control). Это означает ограничения доступа пользователей к системным ресурсам.

Пользователи, не видящие опасности потери данных, могут сказать, что в идее шифрования нет рационального зерна. Их мало радует то, что шифрование замедляет работу, заставляет их запоминать длинные ключи и вообще навязывает дополнительные правила поведения. Для того, чтобы успешно продать систему с шифрованием данных, вы должны предупредить пользователя об опасностях незащищенной системы и в то же время предложить ему простую в применении и не замедляющую работу систему.

7.1. Основные сведения о шифровании данных

Проблема. В наше время злоумышленник может беспрепятственно подключиться к любому сетевому кабелю. Более того, будучи зарегистрированным пользователем на легально подключенной к сети рабочей станции, он может, запустив на выполнение программу – сетевой анализатор, перехватить любые данные, передаваемые по сети. Таким образом, очень легко получить копии паролей или другие конфиденциальные данные.

Решение. Если вся информация, передаваемая по сети, автоматически шифруется еще до начала передачи, злоумышленник потерпит фиаско. «Жучки» и сетевые анализаторы выдадут ему

зашифрованные денные, но без ключей дешифрования эту информацию нельзя интерпретировать. При этом шифрование и дешифрование данных может выполняться либо аппаратными средствами, такими как электронные ключи защиты данных или модемы, либо специальными программами, которые одновременно работают и на компьютере-источнике, и на компьютере-приемнике информации.

Проблема. Многие пользователи работают на ПК, подключенных к сети. В этом случае для злоумышленников жесткие диски их компьютеров являются кладезями информации. Учитывая, что IBM-совместимые персональные компьютеры не имеют встроенной системы защиты, данные, записанные на их дисках, практически общедоступны.

Решение. Шифруйте все особо важные файлы, записанные на жестком диске или на дискетах.

Шифрование не лишено проблем. Если все процессы шифрования и дешифрования выполняются с одним и тем же ключом, то право доступа к этому ключу означает получение доступа и ко всем данным. Если используется несколько ключей, то пользователям трудно их запомнить, особенно если они очень длинные. К тому же, забыв ключ, вы рискуете потерять всю информацию. Если в организации есть люди, которые могут помочь пользователю, позабывшему ключ, значит, в этой организации есть люди, которые могут помочь самим себе в получении доступа к секретной и зашифрованной информации.

Как выполняется шифрование.

При шифровании исходное сообщение или файл (plaintext – исходный текст) и ключ модифицируются (шифруются) с помощью алгоритма кодирования (encoding algorithm), в результате чего мы получаем зашифрованный текст (cipher text). Для восстановления исходного текста необходимо выполнить обратную операцию, используя ключ и алгоритм декодирования (decoding algorithm).

Рассмотрим пример. Я хочу отправить сообщение Другу, которое никто, кроме Друга, не смог бы прочесть. Я шифрую сообщение (исходный текст) с помощью ключа, в результате чего получаю зашифрованный текст. Друг декодирует зашифрованный текст, используя ключ дешифрования, и читает сообщение. Злоумышленник может попытаться либо раздобыть секретный ключ, либо восстановить исходный текст без ключа, прибегнув к другим методам. Однако система защиты данных, обладающая высокими криптографическими свойствами, не позволит восстановить исходный текст из зашифрованного без ключа дешифрования.

Шифрование может применяться и одним пользователем, например шифрование файлов на жестком диске для предотвращения попыток незваных гостей ознакомиться с их содержимым. В этом случае говорят о массовом шифровании (bulk encryption) – шифровании большого объема данных.

При двойном шифровании (double encryption) применяется два различных метода зашифровывания. В этом случае передача информации осуществляется в следующей последовательности:

1. получение исходного текста;
2. предварительное шифрование;
3. вторичное шифрование;
4. передача или запись данных;
5. предварительная дешифрация;
6. окончательная дешифрация;
7. получение исходного текста.

Применение шифрования. В работе сети шифрование может выполняться в любой комбинации на одном из следующих уровней модели ISO OSI (International Standards Organization Open System Interconnection):

- Управления линией передачи данных;
- Транспортном;
- Приложений.

Шифрование на уровне управления линией передачи данных. Отправитель шифрует информацию только один раз на уровне управления линией передачи данных (уровень 2), а затем передает по линии связи. При переходе с одной линии связи на другую данные расшифровываются, а затем снова зашифровываются. Поскольку каждый этап шифрования и дешифрования может потребовать много времени, скорость передачи информации и производительность сети снижаются. Кроме того, в каждом узле текст некоторое время находится в незашифрованном виде.

Шифрование на транспортном уровне. Информация шифруется на транспортном уровне (уровень 4) и передается по сети в зашифрованном виде, что исключает опасность ее утечки, а также снижение производительности. Этот подход эффективнее, когда протокол поддержки уровня 4 выполнен в виде аппаратного обеспечения, а не реализуется программой, работающей на главном узле, с которого можно получить ключ и метод шифрования.

Шифрование на уровне приложений. Шифрование на уровне приложений (уровень 7) мало зависит от нижележащих

уровней и совсем не зависит от их протоколов. При таком подходе необходимо обеспечить одновременную работу соответствующего программного обеспечения.
Несколько общих вопросов о шифровании

7.2. Аутентификация

Аутентификация (authentication) применительно к обработке данных в цифровой форме – это процесс, при котором может выполняться проверка, предназначено ли данное цифровое сообщение или содержимое такого сообщения его получателю. Протоколы аутентификации основываются либо на использовании криптосистем с обычным секретным ключом, реализующих стандарт шифрования данных DES (Data Encryption Standard), например система Kerberos, либо на использовании систем с открытым ключом, в которых применяется цифровая подпись (digital signature), называемая также цифровыми отпечатками пальцев, подобных системе RSA (Rivest-Shamir-Adleman).

В основном под аутентификацией подразумевается использование цифровой подписи, имеющей вид кода, которая сообщает получателю, что отправитель является именно тем, за кого себя выдает. Такое название выбрано потому, что при передаче документов в цифровой форме этот код выполняет ту же функцию, что и собственноручная подпись на печатном документе. Цифровая подпись, подобно обычной, сообщает, что указанное лицо (организация) написало или, по крайней мере, согласно с содержимым документа, под которым стоит такая подпись. Получатель, как и третья сторона, может проверить, действительно ли документ исходит от лица, подпись которого стоит под ним, и что документ после его подписания не был прочитан посторонними. Таким образом, системы обеспечения аутентификации использует два метода: метод получения подписи под документом, гарантирующим невозможность подделки, и метод проверки того, подпись была действительно сделана тем лицом, которому она принадлежит. Кроме того, секретную цифровую подпись нельзя изменить (лицо, подписавшее документ, уже не может отказаться от нее, утверждая, что она подделана).

В отличие от шифрования, цифровая подпись является сравнительно новым изобретением, поскольку необходимость в ней возникла с распространением систем цифровых коммуникаций.

7.3. Криптография с открытым ключом

Традиционно криптография основывалась на том, что отправитель и получатель сообщения знали и использовали один и тот же секретный ключ. Отправитель с его помощью шифровал сообщение, а получатель – расшифровывал. Этот метод называется *криптографией с секретным ключом* (secret-key cryptography) или *симметричная криптография* (symmetric cryptography). Главная проблема такого метода заключается в том, чтобы получатель и отправитель использовали один и тот же ключ, который должен быть неизвестен всем остальным. Если они находятся на большом расстоянии друг от друга, им приходится доверять курьеру, телефону или другой системе передачи сообщений, скрывая то, что передается секретный ключ. Любой, кто подслушает или перехватит ключ во время передачи, сможет затем расшифровать все сообщения, используя этот ключ. Создание, передача и хранение ключей называется *распределением ключей* (key management). Все криптосистемы должны выполнять распределение ключей, однако при использовании метода криптографии с секретным ключом этот процесс имеет весьма сложный характер.

В 1976 г. был разработан метод криптографии с открытым ключом для распределения ключей. Этот метод предполагает наличие двух ключей – открытого и личного. Открытый ключ можно разглашать, а личный необходимо хранить в тайне. При этом необязательно, чтобы и отправитель, и получатель имели доступ к одной и той же секретной информации. При обмене сообщениями пересылается только открытый ключ. Таким образом, пользуясь данным методом, можно не беспокоиться о надежности каналов передачи информации. Любой человек может отправить конфиденциальное послание точно так же, как и открытую информацию, поскольку для его расшифровки необходим секретный ключ, единственным обладателем которого является получатель данного сообщения.

Более того, криптография с открытым ключом, в отличие от криптографии с секретным ключом, может применяться не только для секретности (шифрование), но и для аутентификации (цифровая подпись).

Несколько слов о том, как выполняется шифрование при использовании открытого ключа. Если я хочу послать сообщение Другу, я нахожу в справочнике открытый ключ Друга, использую его для кодирования сообщения и отправляю письмо. Получив мое

послание, Другс помощью своего личного ключа декодирует и читает мое сообщение. Таким образом, любой может послать Другу закодированное сообщение, но только Друг может прочесть его. Естественно, единственное требование к такому методу – исключение возможности получения личного ключа из соответствующего открытого ключа.

При аутентификации криптография с открытым ключом применяется следующим образом. Для того чтобы подписать сообщение, я выполняю определенные вычисления, применив секретный ключ и само сообщение. В результате я получаю подпись, которая дополняет отправляемое сообщение. Если Друг хочет убедиться подлинности подписи, он также выполняет некоторые вычисления, используя полученный текст, подпись и мой открытый ключ. Если после решения несложных математических уравнений получается правильный результат, подпись подлинная. В противном случае можно сделать вывод, что подпись была подделана либо сообщение изменено.

7.3.1. Достоинства и недостатки метода криптографии с открытым ключом

Главным достоинством криптографии с открытым ключом является повышенная безопасность: нет необходимости ни передавать или сообщать кому бы то ни было секретные ключи, ни убеждаться в их подлинности. В системах с секретным ключом существует опасность, что противник сможет раскрыть секретный ключ во время его передачи.

Системы с открытым ключом пригодны к применению для цифровой подписи. Аутентификация с секретным ключом требует ознакомления с некоторой частью секретной информации и иногда сопряжена с третьей стороной, пользующейся доверием. Отправитель может отказаться от ранее отправленного послания, мотивируя это тем, что кто-то перехватил секретную информацию, предназначенную другой стороне. При аутентификации, проводимой с использованием открытого ключа, отказ от подписи невозможен, а сообщение, подписанное цифровой подписью, может однозначно идентифицировать его автора и для третьей стороны, например судьи. Таким образом, аутентификация с открытым ключом гарантирует подлинность подписи, подтверждающей документ, тогда как аутентификация с секретным ключом не может предоставить таких гарантий.

Главным же недостатком криптографии с открытым ключом является скорость. Некоторые широко распространенные методы шифрования с секретным ключом работают значительно быстрее, чем имеющиеся в настоящее время методы с открытым ключом. В этой главе будет рассмотрен метод шифрования с открытым ключом RSA с описанием его скоростных характеристик.

При шифровании лучше всего сочетать два метода, что позволяет использовать преимущества высокой секретности, предоставляемые системами с открытым ключом, вместе с преимуществами высокой скорости работы, присущих системам с секретным ключом. В этом случае система с открытым ключом используется только для декодирования секретного ключа, с помощью которого затем расшифровывается основная часть файла или сообщения. Таким образом, криптография с открытым ключом не заменяет криптографию с секретным ключом, а дополняет ее, позволяя повысить секретность. Первоначально техника использования открытого ключа применялась для обмена секретными ключами между разными системами, в которых эти ключи использовались. Сегодня эта функция метода шифрования с открытым ключом остается основной.

Криптография с секретным ключом является исключительно важным методом и остается предметом изучения и исследований. Позже мы рассмотрим некоторые системы с секретным ключом.

Стандарт шифрования данных (DES)

Один из подходов к шифрованию данных, предложенный IBM, отражен в стандарте шифрования данных (DES – Data Encryption Standard). Этот стандарт был разработан NIST, одобрен правительством США и без особого труда «взломан» NSA. До недавнего времени DES был самым доступным стандартом при шифровании. В нем используется единый 56-битовый секретный ключ. (Вы можете прочитать о 64-битовых ключах, но дополнительные биты используются не для повышения секретности зашифрованного текста, а для размещения битов проверки или других целей). Благодаря DES, созданному в 1977 г., на его основе было разработано много различных продуктов. Kerberos, версия стандарта DES, разработанная MIT, может стать частью спецификации OSF DCE (Open Software Foundation Distributed Computing Environment).

Стандарт OES был *опубликован* NIST в *Federal Information Processing Standard Publication* (FIPS POB 46) в 1977 г. и рекомендован к использованию в системах, которые защищают конфиденциальность и целостность важной негрифованной информации,

обрабатываемой федеральными правительственными учреждениями. Работа алгоритма, изложенного в FIPS PUB 46, была проверена на компьютерах IBM. Позже этот стандарт был принят в качестве национального американского стандарта X3.92-1981/R1987. Стандарт DES дважды пересматривался, последний раз – в 1988 г. Стандарт (FIPS PUB 46-1) был действителен до 1993 г. Сейчас NIST рассматривает вопрос о том, оставить ли DES стандартом на последующие 5 лет. Ресертификация вполне возможна, но NIST предлагает свой собственный стандарт – DSS (он рассматривается в этой главе несколько позже). American Bankers Association также одобрила DES для защиты фондов и выполнения секретных операций с помощью обычных линий связи. Однако NSA сертифицировало DES только как стандарт для шифрования информации с грифом, не превышающий «секретно». При этом NSA далеко не в восторге от DES. Несколько лет тому назад представители NSA уверяли, что агентство не собирается сертифицировать DES после 1987 г. NIST и некоторые частные организации, такие как American Bankers Association, настаивали на сертификации. В конце 1987 г. NIST одержал победу OES был ресертифицирован еще на пять лет.

7.4. Использование стандарта DES

Вы можете шифровать данные, применив только программные и аппаратные средства или комбинируя их. Зашифрованный файл остается на диске в таком виде до тех пор, пока вы не пожелаете его использовать. Кроме того, с высокой степенью секретности вы можете передавать зашифрованные по стандарту DES-файлы, используя модем с протоколом Xmodem. DES описывает криптографический алгоритм, который преобразовывает исходный текст в зашифрованный, используя ключ (сложные математические вычисления). При дешифрации применяются тот же алгоритм и ключ. DES состоит из 16 операций (проходов), в которых данные и ключ объединяются в описанной выше форме, основываясь на операциях перестановки и инвертирования *(скремблирования)*. Цель этих операций - достижение такого состояния, при котором данные и ключ перемешаны так, что каждый бит зашифрованного текста зависит от каждого бита данных и каждого бита ключа (ключ в DES имеет размер 56 бит). Хороший алгоритм после определенного количества проходов порождает зашифрованный текст, в котором нет никакой корреляции с исходными данными или ключом.

DES использует 16 проходов по нескольким соображениям. Во-первых, чтобы получить достаточную степень скремблирования данных и ключа, необходимо выполнить как минимум 12 проходов. Дополнительные проходы выполняются из соображений безопасности. Во-вторых, в электронике 16 выполнений операций преобразуют ключ к его исходному виду, пригодному для дальнейшего использования. В-третьих, большое количество проходов необходимо для того, чтобы отпугнуть аналитика или исследователя, который при изучении работы алгоритма с начального или конечного состояния столкнется с задачей дешифрации.

7.4.1. Уровень секретности, обеспечиваемый DES

Уровень секретности, который может обеспечить DES, зависит от нескольких факторов: совершенства математического аппарата, длины ключа, распределения ключей, формата входного потока данных, режима работы, реализации, прикладной программы и важности.

DES был разработан для защиты негрифованных компьютерных данных, обрабатывающихся в федеральных компьютерных системах, от многочисленных пассивных и активных агрессивных воздействий при обмене данными или их хранении. При этом подразумевается, что лицо, имеющее достаточную подготовку, может найти уязвимое место в системе защиты с помощью средств, соответствующих значению данных, которые необходимо получить. Такая защита применяется в системах электронных платежей, защиты личной информации, аутентификации личности, парольной защиты, управления доступом и т. д.

Несколько организаций, изучив DES, пришли к выводу, что его исследование является достаточно сложной математической задачей. Но используемая в нем длина ключа данных (56 бит), тем не менее, подвергается критике, так как многие считают ее недостаточной для обеспечения высокой степени секретности. Другие исследователи, рассмотрев алгоритм, отмечают, что он достаточно устойчив, но при снижении количества проходов с 16 до 6–8 его устойчивость падает. Тем не менее, если любые два алгоритма были «взломаны» на персональном компьютере за 0,3 или 3 секунды, то это означает, что они не соответствуют DES. Существует только один DES, и любое его изменение приводит к тому, что алгоритм уже не соответствует требованиям DES. С точки зрения криптографии любой алгоритм, полученный путем какого-то изменения ис-

ходного алгоритма, может значительно отличаться по степени предоставляемой защиты данных. Поэтому DES представляет достаточно надежный алгоритм, а многие другие, хотя они очень похожи на него, таковыми не являются.

Специалисты NIST определили, что, по крайней мере, в тех областях, где применяется DES, он будет обеспечивать достаточную степень защиты. Из всех различных методов федеральными правительственными учреждениями для защиты разного рода компьютерных данных (за исключением информации, определенной федеральным документом 10 U.S.C. Section 2315) применяется именно DES. Однако NIST планирует усовершенствовать DES с помощью одного из стандартных криптографических алгоритмов, которые предоставляют другие типы защиты в специальных областях (например, цифровая подпись, обмен ключами, экспорт технологии защиты). NIST не планирует в дальнейшем использовать DES в правительственных системах обеспечения безопасности данных и фактически намерен разработать другой стандарт шифрования. Хотя DES обладает достаточной степенью надежности, однако существует вероятность того, что NSA, CIA и IBM смогут декодировать его.

7.4.2. Криптографические ключи стандарта DES

Пользователи, работающие в правительственных учреждениях с продукцией, разрешенной к применению NSA, могут платно получить криптографические ключи DES для этой продукции, обратившись в NSA. Если вы принадлежите к такого рода пользователям, обратитесь за более подробной информацией к служащему ближайшего представительства службы Communications Security (COMSEC). Кроме того, пользователи DES, в том числе и федеральные организации, могут создавать свои собственные криптографические ключи. Ключи DES должны генерироваться и распределяться таким образом, чтобы обеспечить высокую степень защиты компьютерных данных. Электронное распределение ключей включает в себя использование автоматизированного процесса генерации, доставки, хранения и уничтожения криптографических ключей. Задача освещения специфичных вопросов генерации ключей выходит за рамки данной главы.

Ключи, используемые для защиты электронных платежей, не должны быть постоянными, а время от времени должны меняться, по крайней мере, ежегодно. Значительные электронные платежи

необходимо защищать индивидуальными ключами, а входные данные для достижения высокой степени секретности должны соответствующим образом форматироваться. Поскольку алгоритмы шифрования широко известны, для достижения успеха необходимо соблюдать особую осторожность при передаче ключей. На практике для передачи ключей DES применяется алгоритм RSA, который обсуждается в этой главе.

7.4.3. Порядок допуска к применению продукции, использующей стандарт DES

На сегодня NSA уже не занимается выдачей допусков к применению продукции, использующей DES, которая предназначается для работы в телекоммуникационном оборудовании и системах и соответствует требованиям FIPS POB 140 (бывший Federal Standard 1027). Руководители федеральных министерств и ведомств были уведомлены NIST о том, что они могут самостоятельно принимать решение о приобретении оборудования, которое не соответствует всем критериям стандарта FIPS POB 140. Это решение позволяет организациям закупать дешевое оборудование, которое удовлетворяет их требованиям и в то же время не нуждается в получении разрешения от NSA.

Требования FIPS PUB 140 в настоящее время пересматриваются и будут переизданы в виде документа FIPS PUB140-1. В него будут включены все ранее издававшиеся дополнительные выпуски, не вошедшие в ныне действующий стандарт. Кроме того, NIST проверяет различные методы тестирования на предмет их соответствия требованиям FIPS PUB 140-1. Федеральные организации до утверждения FIPS PUB 140-1 могут затребовать у поставщиков письменное подтверждение о соответствии их продукции требованиям FIPS PUB 140 для того, чтобы убедиться в качестве своего оборудования.

Что такое RSA?

RSA – это криптосистема с открытым ключом, применяемая как для шифрования, так и для аутентификации. Авторы RSA – Ривест, Шамир и Адлеман (Ron Rivest, Adi Shamir и Leonard Adieman) – разработали эту систему в 1977 г. Рассказывая о RSA, мы не будем вдаваться в подробности математических лабиринтов этого стандарта, так как многим нашим читателям, даже обладающим быстродействующим компьютером, все равно будет очень

трудно пробраться через чащу чисел, коэффициентов и вычислений.

И шифрование, и аутентификация, основанные на RSA, выполняются без какого-то ни было совместного использования личных ключей. Любой человек работает только с открытым ключом другого человека и своим собственным. Таким образом, каждый может отправить зашифрованное сообщение или проверить подпись, применив только открытый ключ. Расшифровать же сообщение или подписать его может только тот человек, который обладает соответствующим личным ключом. Надежность систем с RSA зиждется на высокой сложности вычислений – простая методика вычислений могла бы привести к «взлому» кода RSA.

7.5. Преимущества стандарта RSA по сравнению со стандартом DES

RSA не является альтернативным или конкурирующим с DES стандартом. Он предназначен для поддержки DES или другого шифра. Поэтому в системах коммуникаций с повышенной защитой данных RSA обычно применяется вместе с DES.

RSA обладает двумя важными функциями, отсутствующими у DES, – возможностью обмена секретными ключами без какой-либо угрозы утечки секретной информации и поддержкой цифровой подписи. На практике RSA и DES обычно объединяются следующим образом: сначала послание шифруется с помощью получаемого случайным образом ключа DES, а затем, перед отправкой этого послания по несекретным коммуникационным каналам, секретный ключ шифруется по методу RSA. Затем зашифрованное в соответствии с DES сообщение и зашифрованный в соответствии с RS ключ отправляются вместе. Согласно этому протоколу, получившему название *цифровой конверт* RSA (RSA digital envelope), секретная информация пересылается по открытым каналам только в зашифрованном виде.

Вас удивляет тот факт, что RSA не используется для шифрования всего сообщения вместо DES? Для коротких сообщений метод RSA идеален, но при шифровании посланий большего объема пальма первенства принадлежит все же DES (или какому-нибудь другому шифру), так как он значительно превосходит по скорости RSA.

В некоторых случаях вполне достаточно алгоритма DES без использования RSA. Это относится к тем небольшим многопользо-

вательским системам, в которых можно согласовать секретный ключ DES, например, если две стороны договариваются о секретной встрече. Кроме того, в однопользовательских системах нет необходимости использовать RSA. Если вы хотите зашифровать ваши личные файлы, вы можете сделать это с помощью алгоритма DES, применив в качестве ключа ваш личный пароль. Таким образом, использование RSA оправдано только в распределенных многопользовательских системах. Кроме того, RSA и другие криптосистемы с открытым ключом применяются в системах цифровой подписи.

Метод RSA абсолютно надежен. Даже при использовании MailSafe (продукт, основанный на технологии RSA) в режиме самого нижнего уровня секретности для «взламывания» шифра понадобится около 10 лет работы суперкомпьютера Cray-1. Если необходимо достичь высокого уровня секретности, MailSafe может применить вместо 400-битового, 7004-битовый ключ. Это замедлит «процесс шифрования в пять раз, а дешифрования на Cray (по методу проб и ошибок) – до 36 миллионов лет.

В настоящее время, на рынке существует множество средств, в которых алгоритм RSA выполнен аппаратно, и их ассортимент постоянно пополняется более новыми и быстрыми моделями микросхем. На сегодня самые быстрые микросхемы RSA имеют пропускную способность около 64 килобит в секунду. Ожидается, что в ближайшие годы будет преодолен барьер в 1 мегабит в секунду.

При шифровании по методу RSA размер данных увеличивается незначительно. При шифровании сообщение может увеличиться до размера, кратного размеру блока, который, в свою очередь, определяется длиной модуля (в большинстве приложений - 512 бит). Процесс аутентификации не требует шифрования текста сообщения, поэтому размер данных не изменяется. Однако к сообщению в этом случае присоединяется цифровая подпись. Для метода RSA подпись обычно имеет размер, равный размеру одного блока. Кроме того, иногда к сообщению присоединяется и *сертификат* (certificate). Сертификат - это подписанный документ, удостоверяющий личность и открытый ключ человека, подписывающего сообщение. Он предназначен для предотвращения ситуаций, когда один человек выдает себя за другого и сообщает при этом фальшивый ключ. Сертификация применяется во всех методах цифровой подписи. Типовой сертификат RSA при размере модуля 512 бит имеет длину 300 байт, сообщения большого объема могут содержать два сертификата.

7.5.1. Применение на практике RSA для шифрования

Обычно на практике RSA объединяется с криптографическими системами, в которых используется метод шифрования с секретным ключом, такой как DES (цифровой конверт). Предположим, что вы хотите послать зашифрованное сообщение Джиму. Для этого вы сначала шифруете исходный текст по методу DES, используя случайным образом сгенерированный ключ. После этого находите открытый ключ Джима в справочнике и с его помощью шифруете секретный ключ. Закодированное по алгоритму DES сообщение, закодированный по алгоритму RSA и секретный ключ DES отправляются Джиму, объединяясь, таким образом, в цифровой конверт *RSM*. Получив цифровой конверт, Джим с помощью своего личного ключа расшифровывает ключ DES, а затем посредством последнего само сообщение.

7.5.2. Применение на практике RSA для аутентификации

Предположим, вы хотите отправить Джиму подписанное сообщение. Вы подсчитываете длину каждой строки сообщения (так называемое хэширование), получая в результате индексное описание (дайджест) исходного текста, которое впоследствии используется для получения цифровой подписи. С помощью вашего личного ключа RSA вы шифруете дайджест, получая цифровую подпись, которую вы посылаете Джиму с сообщением. Джим, получив ваше послание и подпись, раскодирует подпись с помощью вашего открытого ключа, чтобы получить дайджест сообщения. После этого ему остается только выполнить хэширование, используя ту же процедуру, которой пользовались вы, и сравнить с результатом декодирования дайджеста из подписи. Совпадение означает подлинность подписи, и Джим может быть уверен, что сообщение исходит именно от вас. В противном случае Джим не примет во внимание полученное письмо, поскольку оно либо исходит от кого-то другого, либо было изменено при передаче.

Обычно открытый ключ гораздо меньше, чем личный, – это означает, что верификация подписи происходит гораздо быстрее, чем ее создание. Это достаточно важно, поскольку сообщение или документ подписывается только один раз, а процедура проверки подлинности подписи может выполняться значительно чаще.

Процедура хэширования должна исключать всякую возможность подбора другого сообщения, которое хэшируется к такому же значению, или существования двух разных сообщений с одинаковой величиной хэша. Если это возможно, то злоумышленник может поставить вашу подпись под фальшивым письмом. Для того чтобы исключить вероятность совпадения, были разработаны специальные функции хэширования (например, MD4 или MD5, генерирующие 128-й дайджест), которые с успехом применяются в криптографии.

Как мы уже упоминали, сертификат, присоединяемый к посланию, используется для предотвращения ситуаций, когда один человек выдает себя за другого и сообщает при этом фальшивый ключ. Имя, указанное в сертификате, и серийный номер самого сертификата дополняется цифровой подписью. Это позволяет получателю (или третьей стороне) удостовериться в аутентичности открытого ключа. В подписанном сообщении может находиться один или несколько (для большей безопасности) сертификатов.

7.5.3. Устранение ошибок при передаче

Цифровая подпись RSA, в отличие от обычной подписи, является самой надежной, поскольку она не только идентифицирует личность, но и гарантирует подлинность самого послания. Применив процедуру хэширования, никто не сможет воспользоваться подписью другого человека или изменить текст послания.

Поэтому RSA позволяет получателю обнаружить любую ошибку, возникшую при передаче сообщения. При верификации обнаруживается любое изменение текста сообщения. Однако RSA не может дать ответ на вопрос, является ли данное изменение ошибкой, возникшей по техническим причинам при передаче послания, или преднамеренной подделкой.

7.5.4. Защита от компьютерных вирусов

Алгоритм RSA позволяет не только определить ошибку, возникшую при передаче сообщения, но и обнаружить любое изменение файла, хранящегося на диске. Поскольку действие любого вируса отражается на содержимом файлов, с помощью RSA можно выявить изменения в файлах, возникшие по вине вируса.

Один из методов применения RSA для обнаружения вирусов заключается в том, что вначале с помощью алгоритма RSA для

каждого файла создается подпись, которая впоследствии используется для проверки его целостности. Если подпись не совпадает, значит файл мог быть заражен вирусом. Естественно, изменение файла может произойти и по другим причинам, таким как перекомпиляция исходного кода или физический сбой жесткого диска. Но если неожиданное изменение вызывает у вас подозрение, воспользуйтесь одной из специализированных антивирусных программ.

Другим методом защиты от вирусов, в котором применяется криптография, является коммерческое программное обеспечение, скрепляемое цифровой подписью. В этом случае при установке или в процессе эксплуатации программы пользователь, применив открытый ключ поставщика, может в любой момент проверить ее целостность. Конечно, хороший вирус может изменить открытый ключ, что приведет к прерыванию процесса проверки. Кроме того, для вирусов сама антивирусная программа является достаточно «лакомым кусочком». Самым надежным методом является встраивание антивирусных возможностей в операционную систему, но даже в этом случае операционная система должна обладать надежным механизмом самозащиты.

7.5.5. Альтернативы RSA

Кроме RSA, существуют и другие криптосистемы с открытым ключом. Некоторые системы, например, основаны на *математической задаче сверхдлинных последовательностей (knapsack problem).* Однако из-за того, что многие из них были раскрыты, такие системы не популярны. Тем не менее, одна из них, реализованная в системе EIGamal, впоследствии послужила основой для разработки нескольких методов цифровой подписи. Один из таких методов, автором которого является Шнорр (Schnorr), в свою очередь, был использован NIST в качестве исходного для разработки стандарта цифровой подписи. Поскольку столь уважаемая организация предложила этот метод в качестве стандарта, самое пристальное внимание было уделено изучению его преимуществ по сравнению с методом RSA. Несмотря на то, что при шифровании и верификации она более медлительна, а цифровая подпись больше, чем подпись RSA, тем не менее, система EIGamal с успехом применяется во многих областях.

Кроме того, для некоторых областей были разработаны криптосистемы, основанные на методе дискретного типизирования. Преимуществом таких систем является то, что они эффективнее

RSA при аппаратной реализации. Однако защищенность таких систем вызывает некоторые сомнения, поскольку создание алгоритма их работы теоретически является самой простой задачей.

При *Вероятностном шифровании (probabilistic encryption)* исходный текст дважды кодируется одним и тем же ключом, в результате чего получаем два разных зашифрованных текста. Такой тип шифрования привлекательнее, поскольку он более устойчив к раскрытию, хотя при этом увеличивается объем данных. Профессор MIT Рахин (M.O. Rahin) предложил систему цифровой подписи, которая по эффективности близка методу RSA. Эта система выгодно отличается от RSA тем, что системы, использующие RSA, менее устойчивы при попытках нарушения защиты, не связанных с вычислениями. Метод Рабина не очень надежен при попытках подделки сообщений, но только в тех случаях, когда злоумышленник пытается заставить кого-то подписать определенное сообщение. (Более подробно метод описан в техническом отчете).

Еще одна схема получения цифровой подписи основывается на адаптации интерактивного самообучающегося протокола (Более подробно описан в Fiat A. and Shamir A. «How to prove yourself: Practical solutions to identification and signature problems», *Advanced Cryptonym – Crypto'S6,* pp 186-194, Springer-Verlag, New York, 1987). Эта схема работает быстрее эквивалентных алгоритмов, использующих вычисления, однако размер подписи значительно превышает размер подписи, получаемой при методе RSA. В то же время существует несколько вариантов, при которых подпись имеет меньший размер. Применение таких систем, основанных на идентификации, целесообразнее в тех областях, где используются интеллектуальные карты (smart card), чем для сетевых коммуникаций. В последние годы завоевывают популярность криптосистемы, основанные на выполнении математических операций над эллиптическими кривыми.

Главным преимуществом RSA перед другими криптосистемами с открытым ключом является тот факт, что данный метод можно применить как для шифрования, так и для аутентификации. Кроме того, метод RSA получил за последние годы широкое распространение и с честью выдержал многочисленные испытания. RSA привлекает к себе гораздо больше внимания, чем другие криптосистемы, он глубже изучается и применяется, поэтому само собой разумеется, что он надежнее недавно разработанных и малоизученных систем. Ни для кого не секрет, что многие криптосистемы с открытым ключом, которые вначале предоставляли высокую

степень защиты, впоследствии все равно были «взломаны». До настоящего времени еще ни одна из систем с открытым ключом, кроме RSA, не была столь устойчива к многочисленным попыткам нарушения защиты.

7.5.6. Применение RSA в настоящее время

RSA используется во всем мире большим количеством программных продуктов, на разных платформах и в разных областях промышленности. RSA применяется также и во многих коммерческих программных продуктах, например пакетах создания электронных форм Lotus Notes и Derlina's PerForm Pro, предоставляющих возможность создания электронной подписи по алгоритму RSA. Microsoft, IBM, Apple, Sun, Digital и Novell встраивают RSA в операционные системы. Среди аппаратного обеспечения, в котором применяется RSA, можно назвать такие средства, как Secun Telephone Units (Motorola и AT&T), платы Ethernet (Xerox) и т. д. Многие поставщики, такие как WordPerfect Corporation, уже объявили о планах внедрения RSA в свои продукты.

RSA применяется во многих правительственных учреждениях США, например в CIA, Министерстве обороны, Госдепартаменте, Министерстве труда, а также в национальных лабораториях, таких как Lawrence Livermore и Sandia National Lab. Однозначно ответить на вопрос, как широко применяется RSA в этих учреждениях, невозможно, поскольку правительство может использовать RSA без лицензии. Кроме правительственных организаций, многие ведущие корпорации, такие как Boeing, Shell Oil, DuPont, Raytheon и Citicoif, выбрали RSA для внутреннего использования. Исследовательские организации, например: University of California, Bellcore и Nation Science Foundation, также остановили свой выбор на RSA.

В Европе RSA еще более распространен, чем в США. Например, Европейским финансовым сообществом RSA был выбран в качестве стандарта для выполнения электронных платежей, в то время как в банковском деле США использование RSA еще не вошло в повседневную практику (хотя это может произойти, когда RSA станет официальным стандартом).

Применение RSA переживает период бурного роста и через несколько лет может стать повсеместным. Для этого необходимо, чтобы RSA обладал более высокими скоростными характеристиками при аутентификации, чем при шифровании, поскольку на про-

дукцию, в которой используется аутентификация, легче получить экспортную лицензию.

7.5.7. Официальный стандарт

RSA используется в качестве одной из составных частей многих стандартов. Стандарты ISO 9796 и CCITT X.509 рассматривают RSA как приемлемый криптографический алгоритм. Стандарты SWIFT (Society for Worldwide Interbank Financial Telecommunication) и ETEBAC 5 (применяется во Франции в банковском деле) также используют RSA. Этот алгоритм задействован и в австралийском стандарте цифровой подписи AS2805.6.5.3, и в стандарте PKCS, используемом в индустрии программного обеспечения, и в стандарте PEM (Privacy Enhanced Mail), применяемом в сетях Internet. Согласно соглашениям о реализации систем такого типа, описанных в OSI Implementor's Workshop (OIW), все они должны содержать RSA, как это сделано в PKCS и PEM. В настоящее время разрабатывается много различных стандартов, которые будут анонсированы в ближайшие годы. Во многих из них, по-видимому, будет применен RSA для систем цифровой подписи или защиты данных.

7.5.8. Стандарт де-факто

На сегодня RSA является одной из самых распространенных систем с открытым ключом, которую часто называют стандартом де-факто. RSA уже применяется во многих отраслях, и область его использования постоянно увеличивается. Более того, по распространенности RSA значительно превосходит все остальные системы с открытым ключом.

Не уменьшая значение официальных стандартов, необходимо отметить, что в развитии цифровой технологии чрезвычайно важен учет не только официальных, но и общепринятых стандартов, которые уже стали стандартами де-факто. Если во всем мире используется какая-то общепризнанная система аутентификации с открытым ключом, то все пользователи этой системы, реализуемой разным программным обеспечением на разных аппаратных платформах, могут без проблем обмениваться документами, скрепленными цифровой подписью. Возможность реализации такого взаимодействия является неотъемлемой частью стратегии развития цифровой технологии.

Главное препятствие на пути повсеместного внедрения безбумажной технологии - недостаточная степень секретности при аутентификации. В подавляющем большинстве случаев для контрактов, чеков, официальных писем, а также документов, удостоверяющих личность, в настоящее время используется бумага. До тех пор, пока сохраняется необходимость применения бумаги, будет затягиваться полный переход к обществу, основанному на использовании безбумажной технологии. Цифровая подпись с ее свойством верификации - вот то средство, которое необходимо, чтобы перевести самые важные документы с бумажных носителей на электронные. Цифровая подпись делает возможным использование в электронной форме договоров найма, завещаний, аттестатов, чеков и избирательных регистрационных форм. При этом любой из этих документов в бумажном виде является лишь копией электронного оригинала. Широкое внедрение в повседневную практику общественной жизни таких документов основывается на использовании общепринятого стандарта аутентификации.

DSS – цифровая подпись.

DSS (Digital Signature Standard) – это протокол цифровой подписи с открытым ключом, предложенный в качестве стандарта NIST. Специалисты NIST (с помощью NSA) разработали DSS для использования его в качестве общедоступного стандарта. Однако при этом возникли некоторые проблемы: размер ключа (недавно он был увеличен с 512 до 1024 бит), надежность алгоритма, заявление о нарушении патентов и другие критические высказывания, невзирая на участие в разработке этого стандарта специалистов NSA-J. Пересмотрев свои предложения, NIST разослал их в правительственные учреждения для ознакомления. Если кампания NIST по принятию DSS в качестве официального стандарта увенчается успехом, правительственным учреждениям и частному капиталу придется приложить немало усилий, чтобы выполнить требования нового стандарта.

Kerberos.

Разработан в MIT (Massachusetts Institute of Technology). Kerberos – это общедоступный (public domain) протокол аутентификации, предназначенный для использования в сетях. Концепция аутентификации пользователей сети основывается на использовании понятий *мандат* (Ticket), *аутентификатор* (Authenticator) и *центр генерации ключей (КОС* – Key Distribution Center). KDC – это компьютер, находящийся в безопасном месте, который предоставляет каждому пользователю ключ DES. Когда пользователи хотят

получить доступ к прикладной программе, KDC снабжает их мандатами временного действия. При этом генерируется аутентификатор, подтверждающий мандат. Затем пользователи предъявляют мандат и аутентификатор программе, которая проверяет право доступа и допускает их к работе. Исходный код системы Kerberos Version 5 можно получить непосредственно в MIT.

Производительность.

Некоторые скептики, которые все еще не могут сказать хотя бы несколько одобрительных слов о микрокомпьютерах, могут подвергнуть сомнению возможность программного обеспечения микрокомпьютеров выполнять задачи RSA в режиме реального времени.

Независимо от того, выполняется ли шифрование или дешифрование, предназначена ли данная система для получения цифровой подписи или ее проверки, операции RSA сводятся, по существу, к работе со степенными функциями, в которых выполняется ряд умножений. На практике для открытых ключей часто выбирается небольшой показатель степени. Фактически целые группы пользователей могут работать с одним и тем же показателем степени открытого ключа. Это позволяет выполнять шифрование и верификацию быстрее дешифрования и получения подписи.

RSA работает *медленнее,* чем DES. DES выполняет шифрование файла со скоростью около 1,5 мегабита в секунду, т. е. около 187500 байт в секунду. В то же время программа MailSafe, в которой использован алгоритм RSA, на компьютере типа AT шифрует данные со скоростью 3500 байт в секунду, что примерно в 50 раз медленнее, чем DES. Такое различие в значительной степени объясняется тем, что в DES применяется относительно простой алгоритм и ключи небольшого размера. При реализации в программных продуктах DES работает примерно в 100 раз быстрее RSA. Если же алгоритм реализуется аппаратно, то в зависимости от выполнения скорость работы DES может превосходить скорость работы RSA в 1000–10000 раз. Рынок сбыта средств RSA в настоящее время постоянно расширяется, поэтому в ближайшие годы RSA возможно немного сократит этот разрыв. Однако достичь производительности DES алгоритму RSA никогда не удастся. В самых высокопроизводительных системах, существующих на сегодня, в которых RSA реализован аппаратно, достигается пропускная способность более 64 килобит в секунду при использовании 512-битовых модулей (подразумевается, что при этом выполняется, как минимум, 128 операций с открытыми ключами RSA в секунду).

8. Список использованных источников

1. Гриценко В.И., Паншин Б.Н. Информационные технологии: вопросы развития и применения. – Киев: Наука. Думка, 1988 – 272 с.

2. Громов Г.Р. Очерки Информационной технологии. – М.: ИнфоАрт, 1992. – 336 с.

3. Быков Р.Е., Гуревич С.Б. Анализ и обработка цветных и объемных изображений. – М.: Радио и связь, 1984. – 248 с., ил.

4. Попов А.А. Создание приложений для FoxPro 2.5/2.6 в DOS и WINDOWS. – М.: Издательство «ДЕСС КОМ», 2000. – 672 с.

5. Математическая теория планирования эксперимента. / Под ред. С. М. Ермакова. - М.: Наука, 1983. – 392 с.

6. Современный эксперимент: подготовка, проведение анализ результатов. / Под ред. В.Г. Блохин, О.П. Глудкин, А.И. Гуров, М.А. Ханин. – М.: Радио и связь, 1997. – 232 с. ил.

7. Планирование эксперимента и статическая обработка данных: учеб. пособие. – Томск: Гос. Ун-т систем управления и радиоэлектроники, 2000. – 231 с.

8. Фигурнов В.Э. «IBM PC для пользователя. Краткий курс» – М.: ИНФРА-М, 1998. – 480 с.: ил.

9. Защита информации в персональных ЭВМ. Спесивцев А.В., Вегнер В.А., Крутяков А.Ю. и др. – М.: Радио и связь, МП «Веста», 1993. – 192 с.: ил.

10. Защита компьютерной информации от несанкционированного доступа. Щеглов А. Ю. – СПб: Наука и Техника, 2004. – 384 с.: ил.