# Аутентификация

## Теория и практика

### обеспечения безопасного доступа к информационным ресурсам

Рекомендовано Учебно-методическим объединением
по образованию в области информационной безопасности
и одобрено ФСТЭК России
в качестве учебного пособия для студентов высших
учебных заведений, обучающихся по специальностям
090102 — «Компьютерная безопасность»,
090105 — «Комплексное обеспечение информационной
безопасности автоматизированных систем»

*Под редакцией*
*А. А. Шелупанова, С. Л. Груздева, Ю. С. Нахаева*

**Москва**
**Горячая линия—Телеком**
**2009**

УДК 004.732.056(075.8)
ББК 32.973.2-018.2я73
А93

**Авторы:** А. А. Афанасьев, Л. Т. Веденьев, А. А. Воронцов, Э. Р. Газизова, А. Л. Додохов, А. В. Крячков, О. Ю. Полянская, А. Г. Сабанов, М. А. Скида, С. Н. Халяпин, А. А. Шелупанов

**А93** **Аутентификация.** Теория и практика обеспечения безопасного доступа к информационным ресурсам. Учебное пособие для вузов / А. А. Афанасьев, Л. Т. Веденьев, А. А. Воронцов и др.; Под ред. А. А. Шелупанова, С. Л. Груздева, Ю. С. Нахаева. — М.: Горячая линия—Телеком, 2009. — 552 с.: ил.

**ISBN 978-5-9912-0110-0**

Книга посвящена одному из аспектов проблемы управления доступом к информации в компьютерных системах — аутентификации.

Фактически защита информации начинается с аутентификации пользователей. Каждый пользователь современных компьютерных систем сталкивается с процедурами аутентификации неоднократно в течение рабочего дня. Книга описывает достоинства и недостатки практически всех существующих и используемых на настоящий момент способов аутентификации и ориентирована на широкий круг читателей.

Книга адресована студентам вузов и аспирантам, обучающимся по специальностям, связанным с защитой информации, ИТ-специалистам и специалистам по информационной безопасности; специалистам, получающим второе высшее образование в области защиты информации, и слушателям курсов переподготовки.

**ББК 32.973.2-018.2я73**

Адес издательства в Интернет WWW.TECHBOOK.RU

Учебное издание

**Аутентификация**
Теория и практика обеспечения безопасного доступа
к информационным ресурсам

Учебное пособие для вузов

Редактор *И. Н. Андреева*
Обложка художника *В. Г. Ситникова*
Компьютерная верстка *Е. В. Конова*, *Е. М. Патрушева*

Подписано в печать 15.08.09. Формат 70×100/16. Усл. печ. л. 45,75. Тираж 1000 экз. Изд. № 9110
ООО «Научно-техническое издательство «Горячая линия—Телеком»

**ISBN 978-5-9912-0110-0**

# ОГЛАВЛЕНИЕ

# ПРЕДИСЛОВИЕ

Судьба данного учебного пособия весьма необычна. На партнерской конференции по информационной безопасности компании Aladdin у нас появилась идея создать книгу по теоретическим и практическим вопросам аутентификации. Эту книгу можно было бы использовать не только при обучении студентов и аспирантов ВУЗов и СУЗов по учебным дисциплинам «Безопасность баз данных», «Программно-аппаратные средства обеспечения информационной безопасности», «Безопасность операционных систем», «Компьютерная безопасность» и т. д., но и в практической деятельности ИТ-специалистов, системных администраторов, администраторов безопасности различных сетей и систем.

Любая поисковая система в Интернет по запросу «аутентификация» предоставляет более 1 миллиона ссылок на различные информационные ресурсы. Этот очевидный факт подтверждает широкое распространение и использование механизмов аутентификации в практике обеспечения безопасности сетей, систем, различных приложений. Оценив интерес и востребованность данной технологии, мы решили взяться за дело.

Всю сложность воплощения нашей идеи мы поняли несколько позже, когда взялись за ее реализацию. Первая трудность состояла, главным образом, в том, чтобы объединить усилия специалистов зарубежных и российских компаний, признанных лидеров на рынке информационных технологий, таких, как компании Microsoft, Aladdin, Cisco Systems, Citrix, Oracle, Крипто-Про. При этом, помимо необходимых теоретических сведений, мы собирались предложить читателю различные решения, технологии и продукты для реализации задач по обеспечению безопасности доступа к данным и приложениям информационной системы организации, защищенных соединений. Речь идет как о представлении типовых решений, так и о возможной кастомизации продуктов под конкретные системы.

Другая сложность состояла в том, чтобы систематизировать, порой весьма противоречивые сведения, стили изложения, подходы к реализации решений в различных компаниях, и представить методически выверенные теоретические и практические материалы, в том числе и в виде готовых лабораторных работ для использования их в учебном процессе. Третья сложность состояла в том, чтобы преодолев препоны конкурентного противостояния компаний, создать полезную и, на наш взгляд, весьма своевременную и актуальную книгу без рекламы конкретных компаний. И наконец, любая работа, которая делается на общественных началах, зачастую страдает недостатком времени или возможности довести идею создания учебного пособия до логического завершения. Это обстоятельство явилось причиной отсутствия материалов в данной книге еще нескольких ведущих компаний — вендоров (IBM, Check Point Software Technologies, Сигнал-Ком, SUN и т. д.). Надеемся, что эти материалы войдут во второе издание данного учебного пособия. Потребовался весьма продолжительный период времени для решения организационных мероприятий, экспертизы и апробации материалов в учебных заведениях России, при проведении тренингов ИТ-специалистов, сотрудников служб информационной безопасности, студентов профильных ВУЗов и т. п.

К счастью, нам удалось преодолеть все эти трудности, и мы надеемся, что книга окажется полезной как в учебном процессе, так и в практической работе.

Учебное пособие состоит из теоретической и практической частей. Практическая часть содержит 9 лабораторных работ по типовым решениям с использованием продуктов различных компаний. Описание лабораторных работ можно найти по адресу в Интернет: http://www.aladdin.ru/book/

Согласно замыслу авторов, книга, которую Вы держите в руках, призвана открыть перед читателем суть и возможности технологии аутентификации, как базового элемента любой системы информационной безопасности современных компаний.

Специалистам, уже знакомым с данными технологиями, книга поможет систематизировать и расширить свои знания в части прикладного применения средств аутентификации и интеграции их с другими продуктами и решениями для защиты информации.

Развивать рынок аутентификации, способствовать повышению уровня и качества проектов в области ИТ-безопасности, а, главное, содействовать формированию четкого понимания ценности информации в современном мире — основная цель данной книги.

Мы искренне благодарим всех, кто поддерживал и продолжает поддерживать этот проект, помогает в его продвижении, а также распространении книги.

Особую благодарность выражаем Федеральной службе безопасности России (ФСБ России), Федеральной службе по техническому и экспортному контролю России (ФСТЭК России), Совету Безопасности Российской Федерации и Учебно-методическому объединению по образованию в области информационной безопасности за проявленный интерес, полезные замечания и конструктивную критику.

Отдельно хочется отметить вклад в работу при подготовке рукописи данной книги безвременно ушедшего из жизни руководителя аналитического отдела компании Aladdin, кандидата физико-математических наук Нахаева Ю.С.

Мы не планируем останавливаться на достигнутом результате и рассматриваем идею выпуска второго расширенного издания данной книги. Приглашаем к сотрудничеству всех заинтересованных специалистов, компании, ВУЗы.

Замечания, предложения и пожелания просьба направлять по адресу:

634050, Томск, пр-т Ленина, д.40
Институт системной интеграции и безопасности ТУСУР, Шелупанову А. А.
saa@udcs.ru
тел. 8 (3822) 413 426

129226 Москва, ул. Докукина, д. 16 корп. 1
ЗАО «Аладдин Р.Д.», генеральному директору Груздеву С. Л.
pr@aladdin.ru
тел. 8 (495) 223 0001

С уважением,

А. А. ШЕЛУПАНОВ,
Директор Института системной
интеграции и безопасности ТУСУР,
доктор технических наук, профессор

С. Л. ГРУЗДЕВ,
Генеральный директор компании Aladdin

# ЧАСТЬ I

# ТЕОРЕТИЧЕСКИЕ ОСНОВЫ

# Глава 1
# ОБЩИЕ СВЕДЕНИЯ ОБ АУТЕНТИФИКАЦИИ

## 1.1. Основные понятия и определения

Процесс регистрации пользователя в любой системе состоит из трех взаимосвязанных последовательно выполняемых процедур: идентификации, аутентификации и авторизации.

***Идентификация*** — процедура распознавания субъекта по его идентификатору. В процессе регистрации субъект предъявляет свой идентификатор системе, которая проверяет его наличие в своей базе данных. Субъекты с известными системе идентификаторами считаются легальными (законными), остальные относятся к нелегальным.

***Аутентификация*** — процедура проверки подлинности субъекта, которая позволяет достоверно убедиться в том, что субъект, предъявивший свой идентификатор, на самом деле является именно тем субъектом, идентификатор которого он использует. Для этого он должен подтвердить факт обладания некоторой информацией, которая может быть доступна только ему одному (пароль, ключ и т. п.).

***Авторизация*** — процедура предоставления субъекту определенных прав доступа к ресурсам системы после прохождения им процедуры аутентификации. Для каждого субъекта в системе определяется набор прав, которые он может использовать при обращении к ее ресурсам.

Для того чтобы обеспечить управление и контроль над данными процедурами, дополнительно используются процессы администрирования и аудита.

***Администрирование*** — процесс управления доступом субъектов к ресурсам системы. Данный процесс включает:

- создание идентификатора субъекта (учетной записи пользователя) в системе;
- управление данными субъекта, используемыми для его аутентификации (смена пароля, издание сертификата и т. п.);
- управление правами доступа субъекта к ресурсам системы.

***Аудит*** — процесс контроля (мониторинга) доступа субъектов к ресурсам системы, включающий протоколирование действий субъектов при их доступе к ресурсам системы в целях обнаружения несанкционированных действий.

Таким образом, в общем случае речь идет о пяти основных процедурах предоставления доступа к информации. При этом возможен различный подход к расстановке приоритетов при выполнении этих процедур.

## 1.2. Роль и задачи аутентификации. Место аутентификации в структуре основных направлений защиты информации

Независимо от типа системы аутентификации в ней всегда присутствуют пять элементов.

Первый элемент — конкретный человек или процесс, который должен проходить аутентификацию, — *субъект доступа*.

Второй элемент — опознавательный знак, *идентификатор*, который выделяет этого человека или этот процесс среди других.

Третий элемент — *отличительная характеристика* (аутентификатор), подтверждающая принадлежность идентификатора субъекту доступа.

Четвертый элемент — владелец системы (администратор), который несет ответственность за использование системы, и в разграничении авторизованных пользователей и остальных полагается на механизм аутентификации.

Пятый элемент — *механизм аутентификации*, который позволяет проверить присутствие отличительной характеристики.

При успешном прохождении аутентификации субъекту доступа должны быть выданы некоторые права (привилегии).

Для этого служит *механизм управления доступом*. С помощью этого же механизма субъект доступа лишается прав (привилегий), если аутентификация была неуспешной.

*Независимо от типа системы аутентификации в ней всегда присутствуют пять элементов.*

Примером аутентификации является вход физического лица в систему по паролю. *Физическое лицо* — это человек, которому разрешено пользоваться компьютером. Обычно в системе физическому лицу назначается символическое имя или идентификационный код пользователя, который мы будем называть *именем пользователя*. Например, если пользователь является авторизованным пользователем системы, то администратор присваивает ему имя пользователя «Пользователь». Отличительной характеристикой пользователя будет его секретный пароль, например, «qwerty». Данная процедура знакома, так как в процессе регистрации компьютер выдает запрос на ввод имени пользователя и пароля. Процесс включает в себя процедуру аутентификации, т.е. сравнение пароля, введенного с клавиатуры, с паролем, установленным либо самим пользователем, либо администратором системы. Процедура завершается успешно, если оба пароля совпадают. В этом случае механизм управления доступом разрешает пользователю продолжать работу на компьютере, и система использует имя пользователя каждый раз, когда ей требуется решение службы управления доступом к защищенному ресурсу.

Рассматривая проблемы защиты компьютеров, следует всегда проводить различие между тем, что мы хотим сделать, и тем, что мы в действительности делаем.

Первый вопрос «чего мы хотим» обычно озвучивают в виде *целей защиты*. Например, целью шайки сорока разбойников была защита добычи от воровства. В этом они полагались на механизм защиты — дверь пещеры. В вычислительной системе целью владельца системы является предоставление доступа только авторизованным (законным) пользователям.

На практике же всегда существует зазор между тем, что мы хотим, и что происходит на самом деле. Так, замок позволяет войти каждому, у кого есть экземпляр нужного ключа, однако посторонние смогут войти тоже, если мы не предотвратим попадание к ним ключа.

Это может оказаться трудным делом, особенно если те, кого мы стремимся не впустить с помощью замка, действительно хотят попасть внутрь. Более того, мы не всегда можем позволить себе поставить замки на все на свете. Часто имеется один большой замок на входной двери, и нам приходится доверять тем, кого мы впустили внутрь.

В компьютерных системах аутентификация и управление доступом обычно реализуются как две разные функции. Хотя иногда имеет смысл проводить различие между задвижкой, удерживающей дверь закрытой, и замком, который управляет задвижкой, задвижка и замок часто встроены в один механизм. В компьютерных системах процесс аутентификации подтверждает подлинность имени пользователя, а управление доступом осуществляется путем сравнения имени пользователя с правилами доступа, связанными с конкретным файлом или другим ресурсом. Если правила разрешают доступ пользователю с этим именем, то он получает возможность использовать ресурс.

*В компьютерных системах аутентификация и управление доступом обычно реализуются как две разные функции. Процесс аутентификации подтверждает подлинность имени пользователя. Управление доступом осуществляется путем сравнения имени пользователя с правилами доступа, связанными с конкретным файлом или другим ресурсом.*

Сорок разбойников стремились к тому, чтобы в пещеру имели доступ только члены шайки, но их механизм не мог предотвратить использование пароля другими людьми. Эта проблема свойственна как процессу аутентификации, так и механизму управления доступом.

Механизмы аутентификации несовершенны. Неавторизованные люди могут замаскироваться под легального пользователя.

Такая же проблема возникает и при управлении доступом: необходимо авторизовать на пользование системой только определенных людей и именно для этого устанавливается система управления доступом. В идеальном мире техники со средствами защиты доступ выдается по принципу «наименьшей привилегии», в соответствии с которым люди имеют ровно столько разрешений и привилегий, сколько им требуется: не больше и не меньше. Но в реальном мире система управления доступом не может дать людям ровно столько привилегий, сколько им требуется: мы вынуждены либо предоставлять им слишком много привилегий, либо отнимать некоторые из тех, которые действительно необходимы. На практике мера доверия авторизованным пользователям обычно расширяется, так что у них есть инструментарий для выполнения своей работы, даже если технически это позволяет им делать такие вещи, которые они делать не должны.

*В идеале доступ к информации выдается по принципу «наименьшей привилегии», в соответствии с которым люди имеют ровно столько разрешений и привилегий, сколько им требуется, но на практике мера доверия пользователям обычно расширяется.*

Управление доступом может быть очень сложным даже в отсутствие попытки добиться выполнения принципа наименьших привилегий. Современные вычислительные системы обеспечивают широкий диапазон подходов и механизмов управления доступом. Механизмы управления доступом даже в таких относительно простых системах, как Unix или Windows, позволяют пользователям и администраторам устанавливать весьма сложные наборы правил получения и лишения прав на использование различных ресурсов компьютера. Однако многие организации придерживаются относительно простого подхода, связывая управление доступом и аутентификацию, так что прошедшие аутентификацию пользователи имеют всего лишь небольшое количество ограничений доступа.

Хотя проблема аутентификации пользователей сама по себе является серьезной проблемой для компьютерных систем, пользователи не являются единственными субъектами, которые подлежат аутентификации.

*В настоящее время необходимо аутентифицировать и системы, действующие без вмешательства человека.*

В отличие от процесса аутентификации пользователя, здесь нет реального человека, стоящего рядом с сервером, чтобы выполнить аутентификацию. Мы же хотим иметь гарантию, что взаимодействуем с нужным оборудованием, которое находится под управлением нужных людей или предприятия. Никто не захочет заказывать туфли через компьютер, объявляющий себя «Shoes», если в конечном итоге он эти туфли не получит. Когда мы выполняем аутентификацию субъекта, представляющегося сервером компании Shoes, мы должны быть уверены, что именно он управляется и контролируется предприятием, принадлежащим компании Shoes. Обычно браузер предупреждает пользователя, если он не может аутентифицировать сервер и оставляет решение по управлению доступом за пользователем (Должен ли я оформлять заказ на туфли, с учетом того, что этот сервер, похоже, не является сервером компании Shoes? Полагаю, нет). В некотором смысле такой процесс переворачивает функцию автоматической аутентификации с ног на голову, но лежащие в основе концепции по-прежнему те же.

## 1.3. Факторы аутентификации

Для подтверждения свой подлинности субъект должен предоставить некоторую секретную информацию, которая должна быть доступна только ему одному. Он может предъявлять системе различные виды информации.

*Фактор аутентификации* — определенный вид информации, предоставляемый субъектом системе при его аутентификации.

### 1.3.1. Описание факторов аутентификации

Выделяют три фактора аутентификации, используемые в различных комбинациях: на основе знания чего-либо, обладания чем-либо, на основе биометрических характеристик (табл. 1.1).

*Таблица 1.1*

**Факторы аутентификации**

| *Фактор аутентификации* | *Классификация типов факторов аутентификации NCSC-TG-017*[1] | *Примеры факторов аутентификации* |
|---|---|---|
| На основе знания чего-либо (1-й) | Type 1: Authentication by Knowledge | · Пароль или парольная фраза<br>· PIN-код (Personal Identification Number) |
| На основе обладания чем-либо (2-й) | Type 2: Authentication by Ownership | · Физический ключ<br>· Карта с магнитной полосой<br>· OTP-токен, генерирующий одноразовый пароль |
| На основе биометрических характеристик (3-й) | Type 3: Authentication by Characteristic | · Отпечаток пальца<br>· Рисунок сетчатки глаза<br>· Голос |

[1] NCSC-TG-017 — документ «A Guide to Understanding Identification and Authentication in Trusted Systems», опубликованный U.S. National Computer Security Center. Руководство содержит комплект рекомендуемых инструкций по процедурам идентификации и аутентификации.

*Выделяют три фактора аутентификации, используемые в различных комбинациях: на основе знания чего-либо, обладания чем-либо, на основе биометрических характеристик.*

В некоторых компаниях организуется «строгий контроль» доступа в помещение, т. е. в определенные помещения доступ предоставляется только ограниченному числу лиц. Например, в серверную комнату может войти только администратор или в комнату финансового отдела компании могут иметь доступ только его сотрудники. Если при этом установить для компьютеров, находящихся в этих помещениях, строго определенные IP-адреса, то тогда появляется возможность более качественно выполнять аутентификацию при доступе сотрудников к ресурсам компьютерной сети. Им предоставляется доступ к определенным действиям или данным только в том случае, если они это делают в строго определенном помещении и соответственно с определенных компьютеров, имеющих определенные IP-адреса. В этом случае иногда говорят об использовании «четвертого» типа фактора аутентификации — ***на основе места проведения процедуры.*** Данный фактор не считается дополнительным, так как его нельзя использовать отдельно от других факторов для аутентификации субъекта. Например, нельзя обеспечить, чтобы только определенный сотрудник работал на строго определенном рабочем месте (компьютере).

В последнее время наметились тенденции интеграции логических средств аутентификации и средств контроля и управления доступом (СКУД). Смарт-карты, используемые для аутентификации пользователя при доступе к ресурсам компьютерной системы, интегрируются с RFID (радиочастотной идентификацией). В этом случае появляется возможность дополнительно использовать их для аутентификации человека при его доступе в различные помещения. По-прежнему в этом случае речь будет идти об использовании аутентификации «на основе обладания чем-либо». Это расширяет возможности использования смарт-карты, дает дополнительные удобства для пользователя, но не повышает качество аутентификации.

### *1.3.2. Многофакторная аутентификация*

Аутентификация может быть реализована с помощью одного из трех факторов аутентификации. Например, в процессе аутентификации у пользователя может быть запрошен пароль, либо потребуется представить отпечаток пальца.

Аутентификация, в процессе которой используется только один фактор аутентификации, называется *однофакторной.*

Аутентификация, в процессе которой используется несколько факторов аутентификации, называется *многофакторной.*

Например, в процессе аутентификации пользователь должен использовать смарт-карту и дополнительно пароль (или PIN-код). Также используются понятия двухфакторной и трехфакторной аутентификации при использовании комбинации двух и трех факторов аутентификации соответственно.

В документе NCSC-TG-017 вводятся термины для различных видов многофакторной аутентификации: типа 12, типа 23 и типа 123. Аутентификация типа 12 (произносится как «аутентификация типа один два»), например использует два фактора аутентификации: первый (на основе знания чего-либо) и второй (на основе обладания чем-либо).

*В документе NCSC-TG-017 вводятся термины для различных видов многофакторной аутентификации: типа 12, типа 23 и типа 123. Аутентификация типа 12 (произносится как «аутентификация типа один два»), например, использует два фактора аутентификации: первый (на основе знания чего-либо) и второй (на основе обладания чем-либо).*

Трехфакторная аутентификация использует комбинацию трех факторов аутентификации («на основе знания чего-либо», «на основе обладания чем-либо» и «на основе биометрии»). Эту аутентификацию называют *аутентификация типа 123*.

Если для аутентификации используется только один фактор аутентификации, она оказывается уязвимой. При многофакторной аутентификации используется несколько (два и более) факторов аутентификации, что обеспечивает большую безопасность.

*При многофакторной аутентификации используется несколько (два и более) факторов аутентификации, что обеспечивает большую безопасность.*

Наиболее распространено использование комбинации двух факторов при аутентификации пользователя в банкомате. Требуется одновременно использовать карту с магнитной полосой и PIN-код.

## Контрольные вопросы

1. Назовите процедуры, выполняемые при регистрации пользователя в системе.
2. Что такое аутентификация?
3. Что такое идентификация?
4. Что такое авторизация?
5. Что такое аудит?
6. Что такое администрирование?
7. Перечислите элементы аутентификации.
8. Для чего служит механизм управления доступом?
9. Перечислите факторы аутентификации.
10. Приведите примеры факторов аутентификации.

# Глава 2
# ПАРОЛЬНАЯ АУТЕНТИФИКАЦИЯ

## 2.1. Аутентификация с помощью запоминаемого пароля

Для проверки подлинности пользователей в информационных системах наиболее широко используется аутентификация по секретной информации, которая неизвестна непосвященным людям. При некомпьютерном использовании это может быть произносимый голосом пароль или запоминаемая комбинация для замка. В компьютерных системах — это пароль, вводимый с помощью клавиатуры.

***Парольная аутентификация*** — аутентификация на основе обладания неким секретным знанием («на основе знания чего-либо»).

В системах различной степени защищенности используют *постоянные*, *условно-постоянные* и *временные пароли*.

Чем длиннее пароль, тем он более стойкий (сложнее поддается подбору и другим типам атак). Не меньшее значение имеют алфавит пароля, предельное количество попыток его ввода, минимальное время, которое должно пройти между попытками, и другие параметры механизма аутентификации.

К сожалению, длинные и сложные пароли обладают и недостатками:

- их труднее запомнить;
- их медленнее набирают — соответственно, их проще подсмотреть.

Современные парольные политики (см. раздел 2.3) задают минимальную длину паролей (обычно 6—8 символов) и их рекомендуемую длину (10—12 символов). Максимальная длина пароля, как правило, ограничена особенностями реализации механизма аутентификации.

---

*Чем длиннее и сложнее пароль , тем он более стойкий.*

---

Компьютерная система для аутентификации вместо запроса пароля может использовать другой метод («на основе знания чего-либо») — метод секретных запросов и ответов.

---

*Парольная аутентификация является наиболее простым методом аутентификации с точки зрения реализации.*

---

## 2.2. Методы парольной аутентификации

### *2.2.1. Аутентификация на основе открытого пароля*

Самым старым и простым методом парольной аутентификации является аутентификация на основе открытого пароля (рис. 2.1).

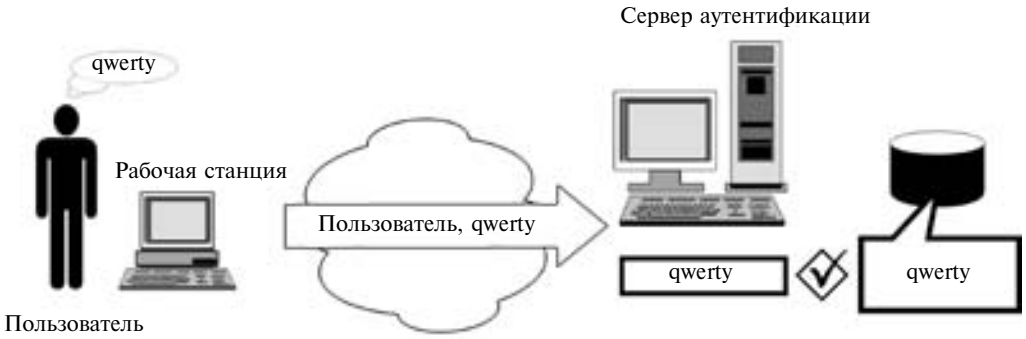



**Рис. 2.1. Аутентификация на основе открытого пароля**

Пример аутентификации пользователя на основе открытого пароля:

1. Пользователь вводит свои имя пользователя «Пользователь» и пароль «qwerty» на рабочей станции.

2. Имя пользователя и пароль передаются по сети в открытом виде.

3. Сервер аутентификации находит учетную запись пользователя в базе данных аутентификации и сравнивает введенные данные с ее содержимым.

В случае совпадения аутентификация признается успешной.

*Простым методом парольной аутентификации является аутентификация на основе открытого пароля*

## 2.2.2. Аутентификация на основе хэшированного пароля

В большинстве используемого в настоящее время программного обеспечения применяются пароли не в чистом виде, а их хэш-значения, получаемые с помощью вычисления криптографической хэш-функции.

***Однонаправленные хэш-функции*** (далее — хэш-функции) — это функции, которые принимают на входе строку переменной длины и преобразуют ее в выходную строку фиксированной (обычно меньшей) длины, называемую значением хэш-функции (хэш-значением).

Пример прохождения пользователем процедуры аутентификации на основе хэшированного пароля (рис. 2.2.):

1. Пользователь вводит свои имя «Пользователь», и пароль «qwerty» на рабочей станции.

2. Рабочая станция вычисляет хэш-значение N4a#@JD от введенного пароля. Имя пользователя и хэш-значение передаются по сети серверу аутентификации.

3. Сервер аутентификаци сравнивает результат вычисления хэш-значения (N4a#@JD) от введенного пользователем пароля с хэш-значением, хранящимся в учетной записи пользователя (N4a#@JD).

4. В случае совпадения аутентификация признается успешной.

Основным свойством однонаправленных хэш-функций является невозможность восстановления исходной информации при обладании полученным из нее хэш-значением.

Восстановить открытое значение пароля из файла паролей, где он хранится в виде хэш-значения, практически невозможно.

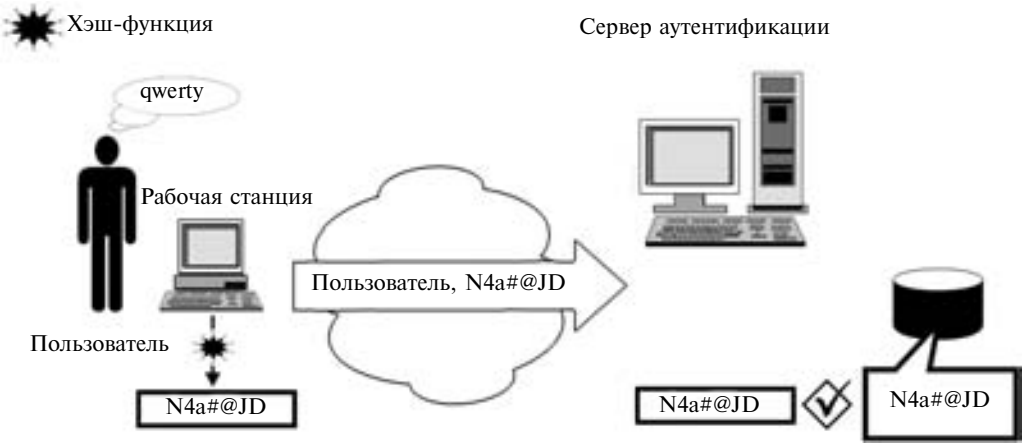



**Рис. 2.2. Аутентификация на основе хэшированного пароля**

Таким образом, если злоумышленник получит доступ к базе данных аутентификации, это ему ничего не даст, так как он не сможет восстановить пароль пользователя из хранящегося в базе хэш-значения.

*Однонаправленные хэш-функции не позволяют восстановить исходную информацию. Поэтому, если посторонний человек получит доступ к базе данных аутентификации, он не сможет восстановить пароль из хранящегося в базе хэш-значения.*

### 2.2.3. Аутентификация на основе PIN-кода

***PIN-код*** (Personal Identification Number) — это разновидность пароля, обычно используемого для аутентификации на локальном устройстве.

Несмотря на слова identification (идентификационное) и number (число), послужившие основой для аббревиатуры, PIN-код редко служит в качестве идентификатора пользователя, а символы, входящие в PIN-код, необязательно являются цифрами. В торговых автоматах и банкоматах применяется карта с магнитной полосой или смарт-карта. PIN-коды часто используются с другими видами устройств аутентификации, например смарт-картами.

Обычно PIN-код торгового автомата или банкомата состоит из четырех цифр. Таким образом, один из каждых 10000 клиентов имеют один и тот же PIN-код. PIN-код похож на простой «пароль».

Разница между PIN-кодом и паролем состоит в области и условиях их использования. Обычно для решений, в которых используется PIN-код, характерно следующее:

- в локальном устройстве, в котором осуществляется аутентификация с помощью PIN-кода, имеется интерфейс для пользователя, а не для программ. Никто не может ввести PIN-код, не используя клавиатуру данного устройства.
- PIN-код не передается по сети и не может быть перехвачен.

Иногда термин PIN-код используют неправильно, применяя его для обозначения коротких и простых паролей. Между этими терминами есть функциональная разница. Аутентификация по PIN-коду обычно используется в двухфакторной аутентификации типа 12.

*Неправильно использовать термин PIN-код для обозначения коротких и простых паролей. Между этими терминами есть функциональная разница.*

## 2.3. Парольные политики

В связи с тем что парольная аутентификация основана на запоминании некоторой информации, многие пользователи информационных систем с парольной аутентификацией выбирают в качестве секрета не произвольную и трудно угадываемую информацию, а легко запоминаемые выражения или свои личные данные. Это могут быть имена, имена членов семьи, названия компьютеров, даты рождения и другие очевидные комбинации.

Для повышения стойкости парольной защиты к перебору, во многих информационных системах реализуется проверка пароля на соответствие определенным требованиям и блокирование выбора простых паролей.

Обычно термины «правила формата пароля», «опции автоматического блокирования», «политика смены паролей» не различаются и называются одним общим термином — *парольные политики*. Парольные политики необходимы для повышения стойкости парольной защиты.

*Парольные политики необходимы для повышения стойкости парольной защиты.*

## 2.4. Недостатки методов аутентификации с запоминаемым паролем

Методы аутентификации с запоминаемым паролем обладают многими недостатками — пароль можно украсть, подсмотреть, подобрать (угадать) и т.д. Кроме того, довольно легко ввести в заблуждение пользователей и администраторов системы, заставив их открыть свой пароль, или же просто принудить их к открытию своего пароля.

Ниже в табл. 2.1 приведены известные атаки на системы, в которых используется аутентификация на основе пароля, а также способы защиты от подобных атак.

*Таблица 2.1*

**Атаки на пароли и защита от них**

| *Описание атаки* | *Защита от данной атаки* |
|---|---|
| ***Кража парольного файла*** | |
| Злоумышленник может прочитать пароли пользователя из парольного файла или резервной копии | *Хэширование пароля*<br>Каждая организация, разрабатывающая парольную аутентификацию, должна снабжать свои приложения этой защитой. |

*Продолжение таблицы 2.1*

| *Описание атаки* | *Защита от данной атаки* |
|---|---|
| ***Атака со словарем*** | |
| Злоумышленник, перебирая пароли, производит в файле паролей или его копии поиск, используя слова из большого заранее подготовленного им словаря. Злоумышленник вычисляет хэш-значение для каждого пробного пароля с помощью того же алгоритма, что и программа аутентификации. | *Безопасность файла*<br>Доступ на чтение к файлу паролей должен быть предоставлен лишь небольшому числу доверенных пользователей.<br>*Хэшированные с шумами (помехами) пароли*<br>Генерирование хэш-значения различным способом для каждого пользователя намного усложняет атаку со словарем: злоумышленник должен при подборе пароля каждого пользователя еще и подбирать способ хэширования пароля. Это достигается в системах с помощью использования меняющегося значения, называемого шумом.<br>*Правила формата пароля*<br>Такие правила могут требовать, чтобы пароль содержал как минимум одну цифру, как минимум один «специальный» символ, комбинации заглавных и строчных букв, и т.д. |
| ***Подбор пароля*** | |
| Исходя из знаний личных данных пользователя, злоумышленник пытается войти в систему с помощью имени пользователя и одного или нескольких паролей, которые он мог бы использовать (в том числе пароля, установленного по умолчанию). | *Правила формата пароля*<br>Как для «атаки со словарем» выше.<br>*Изменение пароля, установленного по умолчанию*<br>Пароль, установленный по умолчанию, должен изменяться сразу после первого использования. По возможности следует вовсе исключить практику использования общеизвестных паролей.<br>*Автоматическое блокирование*<br>После нескольких безуспешных попыток входа система или блокирует учетную запись пользователя на некоторое время, или вовсе аннулирует ее. |
| ***Социотехника*** | |
| *На пользователей:*<br>Злоумышленник представляется администратором и вынуждает пользователя или открыть свой пароль, или сменить его на указанный им пароль.<br>*На администраторов:*<br>Злоумышленник представляется законным пользователем и просит администратора заменить пароль для данного пользователя. | *Политика нераскрытия паролей*<br>В организации должны быть разработаны административные процедуры, запрещающие сообщать пароли другим лицам при любых обстоятельствах. Организация должна также извещать пользователей о том, что администратор никогда не обратится к пользователю с таким требованием.<br>*Политика смены паролей*<br>В организации должна действовать политика, согласно которой администратор меняет пароль пользователя только при условии, что он может установить его личность и передать новый пароль пользователю безопасным способом. Средства самостоятельного управления паролями могут удовлетворять обоим критериям. |
| ***Принуждение*** | |
| Для того чтобы заставить пользователя открыть свой пароль, злоумышленник использует угрозы или физическое принуждение. | *Сигнал о принуждении*<br>В некоторых системах предусматривается возможность для пользователя подавать сигнал о том, что вход осуществляется под принуждением. Обычно это реализуется с помощью специального пароля при входе в систему — пароль «вход под принуждением». |

*Продолжение таблицы 2.1*

| *Описание атаки* | *Защита от данной атаки* |
|---|---|
| ***Подглядывание из-за плеча*** | |
| Расположенный рядом злоумышленник или видеокамера следит за тем, как пользователь вводит свой пароль. | *Неотображение пароля*<br>В большинстве систем пароли либо не отображаются на экране, либо отображаются незначащими символами. В некоторых системах отображается количество таких символов, отличное от введенного. Вопреки этой технологии, злоумышленник может видеть, на какие непосредственно клавиши нажимает пользователь. Также применяются технологии, которые дают пользователю строго ограниченное время для ввода пароля, тем самым заставляя его вводить пароль максимально быстро. Таким образом, уменьшается вероятность его подсматривания, а также усложняется его подбор злоумышленником. |
| ***Троянский конь*** | |
| Злоумышленник скрытно устанавливает программное обеспечение, имитирующее обычный механизм аутентификации, но собирающее имена пользователей и пароли при попытках пользователей войти в систему. | *Особый режим интерактивного взаимодействия для механизма аутентификации*<br>В некоторых системах механизм аутентификации вызывается специально выделенным для этого сочетанием клавиш, недоступным для других программ. В ОС Microsoft Windows в качестве такого сочетания клавиш используется [Ctrl]—[Alt]—[Delete].<br>*Антивирусное программное обеспечение*<br>Организация может обнаруживать программы типа «троянский конь» с помощью антивирусного программного обеспечения.<br>*Средства обеспечения контроля целостности файлов*<br>В организации может использоваться система обнаружения вторжений (intrusion detection system) для определения модификации важных файлов, например, программы регистрации. |
| ***Аппаратный сниффер клавиатуры*** | |
| Злоумышленник скрыто устанавливает в компьютер пользователя аппаратное средство, собирающее информацию, которую вводит пользователь при входе в систему, например, Keykeriki для беспроводных клавиатур, KeyCarbon, KeyDevil или KeyGhost для проводных клавиатур. | *Безопасность рабочих помещений*<br>Служба безопасности компании должна предоставлять доступ в помещения, в которых располагаются компоненты информационной системы предприятия, только тем, кому он разрешен.<br>*Безопасность рабочих мест*<br>Служба безопасности компании должна обеспечить возможность контроля компонентов информационной системы предприятия для защиты от возможности установки в них незаконных аппаратных средств. Контроль над соответствующими компонентами информационной системы предприятия возлагается на сотрудников компании, службу ИТ или службу безопасности компании. |
| ***Трассировка памяти*** | |
| Злоумышленник использует программу для копирования пароля пользователя из буфера клавиатуры. | *Защита памяти*<br>Некоторые ОС используют аппаратную защиту буферов клавиатуры от возможности ее трассировки. |
| ***Отслеживание нажатия клавиш программными средствами*** | |
| Для предотвращения использования компьютеров не по назначению некоторые организации используют программное обеспечение, следящее за нажатием клавиш. Злоумышленник может для получения паролей просматривать журналы соответствующей программы. | *Безопасность файлов*<br>Доступ на чтение к журналам должен быть предоставлен лишь узкому кругу доверенных пользователей (администраторов) с помощью собственной или резидентной службы контроля доступа. |

*Окончание таблицы 2.1*

| *Описание атаки* | *Защита от данной атаки* |
|---|---|
| ***Регистрация излучения (перехват Ван Эка или фрикинг Ван Эка)*** | |
| Вим Ван Эк описал метод, которым злоумышленник может перехватывать информацию с монитора путем регистрации его излучения. Вин Швартау высказал идею приемников Ван Эка, регистрирующих не только видеосигналы. | *Неотображение пароля*<br>Как для «подглядывания из-за плеча» выше.<br>*Безопасность излучений*<br>Модернизация устройств для уменьшения излучения с помощью использования современных микрокомпонент, специально разработанных с учетом необходимости уменьшения излучения. Проектирование помещений и планирование расположения оборудования в нем с учетом предотвращения возможности утечки информации через паразитное излучение оборудования. |
| ***Анализ сетевого трафика*** | |
| Злоумышленник анализирует сетевой трафик, передаваемый от клиента к серверу, для восстановления из него имен пользователей и их паролей. | *Шифрование*<br>Весь сетевой трафик или только пароли могут шифроваться для передачи по сети (использование протокола SSL или VPN-соединений).<br>*Одноразовые пароли*<br>Использование методов аутентификации, в которых «пароли» пользователей изменяются каждый раз при входе в систему. |
| ***Атака на «золотой пароль»*** | |
| Злоумышленник ищет пароли пользователя, применяемые им в различных системах — домашняя почта, игровые серверы и т. п. Есть большая вероятность того, что пользователь применяет один и тот же пароль во всех системах. | *Шифрование*<br>Как для «анализа сетевого трафика» (см. выше).<br>*Одноразовые пароли*<br>Как для «анализа сетевого трафика» (см. выше). |
| ***Атака методом воспроизведения*** | |
| Злоумышленнк записывает последовательность передаваемых и получаемых субъектом доступа в процессе аутентификации данных. Позднее он осуществляет попытку аутентификации, передавая и получая записанные данные в той же последовательности. | *Использование надежных протоколов аутентификации*<br>Надежные протоколы аутентификации предполагают использование при обмене данными с субъъектом доступа криптографически защищенных меток времени.<br>*Одноразовые пароли*<br>Как для «анализа сетевого трафика» (см. выше). |

## Контрольные вопросы

1. Назовите методы парольной аутентификации.
2. Приведите пример аутентификации пользователя на основе открытого пароля.
3. Что такое однонаправленные хэш-функции?
4. Что такое PIN-код?
5. Назовите области и условия использования PIN-кода.
6. Для чего необходимы парольные политики?
7. Приведите примеры атак на системы, в которых используется аутентификация на основе пароля.

# Глава 3
# АУТЕНТИФИКАЦИЯ С ПОМОЩЬЮ БИОМЕТРИЧЕСКИХ ХАРАКТЕРИСТИК

Издавна люди узнавали друг друга по чертам лица, по голосу и другим приметам, характерным только для определенного человека. Даже сейчас лицо является основным признаком, по которому производят удостоверение личности человека, когда проверяют его паспорт, водительские права, пропуск для доступа в организацию — все они содержат фотографию человека, предъявляющего данные документы.

Современные технологии способны обеспечить удостоверение личности человека, используя характерные только ему одному характеристики. Данные технологии основаны на использовании знаний биометрики (или биометрии). Данная дисциплина занимается статистическим анализом биологических наблюдений и явлений.

***Биометрическая характеристика*** — это измеримая физиологическая или поведенческая черта живого человека, которую можно использовать для установления личности или проверки декларируемых личных данных.

Поскольку биометрический параметр уникален для данного человека, его можно использовать для однофакторной аутентификации пользователя. Его можно использовать совместно с паролем или с устройством аутентификации (например, таким, как смарт-карта) для обеспечения двухфакторной аутентификации.

Биометрическая аутентификация обычно является одним из наиболее простых методов для пользователей, которые должны проходить аутентификацию. В большинстве случаев хорошо спроектированная биометрическая система просто снимает показания с человека и правильно выполняет аутентификацию.

## 3.1. Биометрические характеристики

Биометрические характеристики делятся на физиологические и поведенческие.

*Биометрические характеристики делятся на физиологические и поведенческие.*

***Физиологические биометрические характеристики*** (физические биометрические характеристики, статические биометрические характеристики) — биометрические характеристики на основе данных, полученных путем измерения анатомических характеристик человека.

К физиологическим биометрическим характеристикам можно отнести:

- радужную оболочку глаза;
- отпечаток пальца;
- лицо;
- кисть;
- сетчатку.

***Поведенческие биометрические характеристики*** (динамические биометрические характеристики) — биометрические характеристики на основе данных, полученных путем измерения действий человека.

Характерной чертой для поведенческих параметров является их протяженность во времени — измеряемое действие имеет начало, середину и конец.

К поведенческим биометрическим характеристикам можно отнести:

- голос;
- подпись;
- ритм работы сердца.

Различия между поведенческими и физиологическими характеристиками являются достаточно искусственными.

Поведенческие биометрические параметры зависят от физиологии: голос зависит от формы голосовых связок, подпись — от ловкости кисти и пальцев. Некоторые физиологические биометрические характеристики (например, лицо) могут изменяться в зависимости от возраста или поведения человека. Поведение человека (например, то, как он кладет палец или смотрит в камеру) может влиять на эффективность работы системы аутентификации.

Физиологические биометрические характеристики обычно неизменны в течение жизни человека. Использование этих характеристик для аутентификации обычно воспринимается как насильственное воздействие, часто как вмешательство в частную жизнь человека. Поведенческие биометрические характеристики воспринимаются менее болезненно, но они менее стабильны, чем физиологические черты. Они могут изменяться под влиянием стресса и болезни и в целом обеспечивают, по сравнению с физиологическими параметрами, менее качественную аутентификацию.

*Поведенческие биометрические параметры достаточно зависимы от физиологии. Физиологические биометрические характеристики обычно неизменны в течение жизни человека и не могут быть изменены без существенного воздействия на человека.*

## 3.2. Как работают биометрические системы

Хотя биометрические технологии различаются объектами и способами измерений, все биометрические системы работают одинаково (рис. 3.1). Пользователь предоставляет образец (sample) — опознаваемое, необработанное изображение или запись физиологической или поведенческой характеристики. С помощью регистрирующего устройства (например, сканера или камеры), этот биометрический образец обрабатывается для получения информации об отличительных признаках, в результате чего получается контрольный шаблон (или шаблон для проверки). Шаблоны представляют собой достаточно большие числовые последовательности; сам образец невозможно восстановить из шаблона. Контрольный шаблон и есть «пароль» пользователя.

*Все биометрические системы работают одинаково: пользователь предоставляет образец, с помощью регистрирующего устройства этот биометрический образец обрабатывается, в результате чего получается контрольный шаблон.*

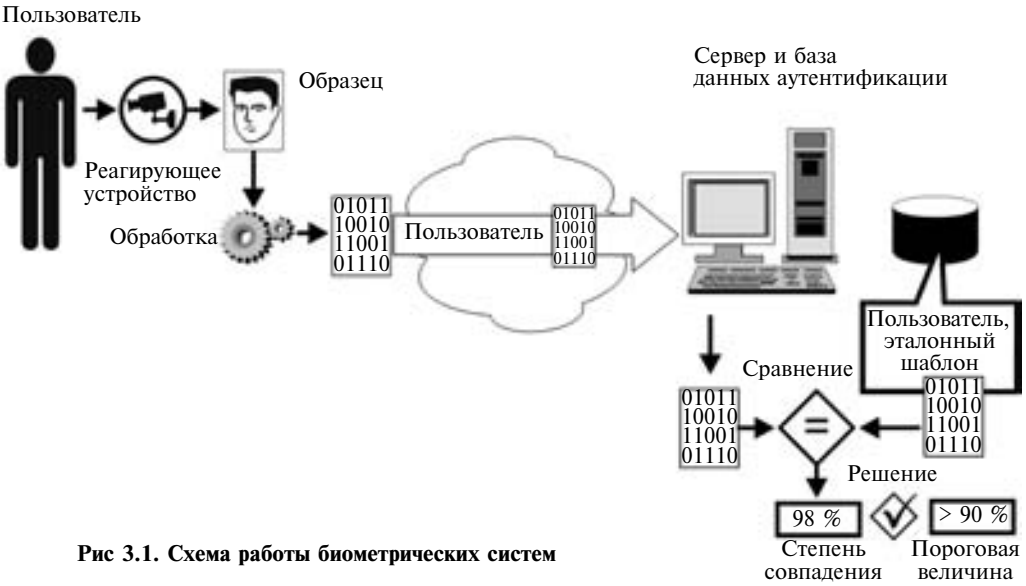



**Рис 3.1. Схема работы биометрических систем**

Контрольный шаблон сравнивается с *эталонным шаблоном* (или зарегистрированным шаблоном), созданным на основе нескольких образцов определенной физиологической или поведенческой характеристики пользователя, взятых при его регистрации в биометрической системе. Поскольку эти два параметра (контрольный и эталонный шаблон) полностью никогда не совпадают, то биометрической системе приходится принимать решение о том, «достаточно» ли они совпадают. Степень совпадения должна превышать определенную настраиваемую *пороговую величину*.

Биометрические системы могут ошибаться, контрольный шаблон может быть ошибочно признан:

- соответствующим эталонному шаблону другого лица;
- несоответствующим эталонному шаблону данного пользователя, несмотря на то что этот пользователь зарегистрирован в биометрической системе.

Точность биометрической системы измеряется двумя параметрами:

- *коэффициентом неверных совпадений* (FMR), также известным под названием ошибка типа I или *вероятность ложного допуска* (*FAR*);
- *коэффициентом неверных несовпадений* (FNMR), также известным под названием ошибка типа II или *вероятность ложного отказа в доступе* (*FRR*).

*Биометрические системы могут ошибаться, их точность измеряется коэффициентом неверных совпадений (FMR) и коэффициентом неверных несовпадений (FNMR)*

Оба коэффициента отражают способность системы предоставлять ограниченный вход авторизованным пользователям. Системы с низким значением FMR более защищены, а системы с низким значением FNMR более просты в использовании. В общем случае для

данных систем при задании пороговой величины действует правило: чем ниже FMR, тем выше FNMR. Таким образом, часто безопасность и простота использования конкурируют между собой.

Простота регистрации и качество «шаблонов» — важные факторы общей эффективности биометрической системы. Некачественный «шаблон» может осложнить работу пользователя, вынуждая его прибегнуть к повторной регистрации в биометрической системе.

## 3.3. Аутентификация и биометрическое распознавание

В режиме аутентификации биометрическая система проверяет заявленную личность, сверяя контрольный шаблон, сгенерированный из образца, с эталонным шаблоном (1:1 или сравнение один с одним). Для аутентификации необходимо, чтобы идентификатор личности был заявлен, например, вводом имени пользователя с клавиатуры, после чего контрольный шаблон данного лица сравнивается с эталонным шаблоном.

Некоторые системы аутентификации осуществляют очень ограниченный поиск среди многочисленного числа зарегистрированных записей. Например, пользователь с тремя эталонными шаблонами отпечатков пальцев может иметь возможность предоставить для проверки любой из трех пальцев, и система предпримет поиск совпадения 1:1 среди эталонных шаблонов данного пользователя.

***Биометрическое распознавание*** — это процесс определения личности пользователя, состоящий из одного шага. В режиме распознавания система определяет личность пользователя, осуществляя сравнение контрольного шаблона со многими биометрическими эталонными шаблонами (*1:N* или сравнение *один ко многим*). В случае нахождения совпадения одновременно определяется и удостоверяется личность пользователя.

Биометрическая идентификация широко распространена и нашла применение в таких областях, как судебная медицина и деятельность правоохранительных органов.

*В режиме аутентификации биометрическая система проверяет заявленную личность, сверяя контрольный шаблон, сгенерированный из образца, с эталонным шаблоном. Для аутентификации необходимо, чтобы идентификатор личности был заявлен.*

В биометрических системах, работающих только в режиме аутентификации, возможно использование *негативной идентификации* в процессе регистрации пользователя в биометрической системе, когда один контрольный шаблон сравнивается со многими, чтобы проверить, что данное лицо не зарегистрировано в базе данных, и таким образом, предотвратить двойную регистрацию в системе. Этот режим часто используется в крупных программах по предоставлению социальных пособий, в которых пользователи пытаются регистрироваться несколько раз для получения пособий под разными именами.

Существует нечто среднее между аутентификацией и распознаванием — «сравнение один к нескольким» (1:few). Этот тип приложений предполагает идентификацию пользователя по очень маленькой базе зарегистрированных пользователей. Четкого количественного разграничения между системами 1:N и 1:few нет, но любую систему, в которой поиск осуществляется среди более чем 500 записей, следует относить к типу 1:N.

# 3.4. Реализация биометрических систем

## 3.4.1. Физиологические биометрические характеристики

Основные физиологические биометрические характеристики, а также виды их реализации приведены в табл. 3.1.

*Таблица 3.1*

**Реализация физиологических биометрических характеристик**

| *Биометрическая характеристика* | *Регистрирующее устройство* | *Образец* | *Исследуемые черты* |
|---|---|---|---|
| Радужная оболочка глаза | Видеокамера, способная работать в инфракрасном диапазоне, камера для ПК | Черно-белое изображение радужной оболочки глаза | Полоски и бороздки в радужной оболочке глаза |
| Отпечаток пальца | Периферийное устройство настольного компьютера, карта стандарта PC card, мышь, микросхема или считыватель, встроенный в клавиатуру | Изображение отпечатка пальцев (оптическое, на кремниевом фотоприемнике, ультразвуковое, или бесконтактное) | Расположение и направление гребешковых выступов и разветвлений на отпечатке пальцев, мелкие детали |
| Лицо | Видеокамера, камера для ПК, цифровой фотоаппарат | Изображение лица (оптическое, двумерное (2D-фото) или трехмерное (3D-фото)) | Форма черепа, относительное расположение и форма носа, расположение скул |
| Кисть | Настенное устройство | Трехмерное изображение верха и боков кисти | Высота и ширина костей и суставов кисти и пальцев |
| Сетчатка | Настольное или настенное устройство | Изображение сетчатки | Расположение кровеносных сосудов на сетчатке |

В стадии разработки находятся новые биометрические технологии, связанные с другими физиологическими характеристиками:

- *Сравнение ДНК* — это самая совершенная биометрическая технология, дающая прямое доказательство идентичности личности (кроме одноядцевых близнецов, у которых одинаковый генотип). Этот метод иногда называется дактилоскопией ДНК, что сбивает с толку и вводит в заблуждение, поскольку отпечатки пальцев не «проникают до уровня генома». Биометрические системы, основанные на сравнении ДНК, могут быть введены в действие лишь через много лет.
- *Отпечаток ладони* — в этой системе используется расположение линий на ладони человека, также, как в биометрической технологии, использующей отпечатки пальцев.
- *Сосудистые рисунки* — расположение вен в различных частях тела человека, включая запястье и тыльную сторону ладони, а также лицо.
- *Сигналы, вырабатываемые сердцем* (мозгом, легкими), — в этой системе пользователь прикасается к датчику «биодинамической подписи» и остается с ним в контакте некоторое время (в зависимости от точности измерения — до 8 с). За это время датчик идентифицирует индивидуальные параметры человека.

*В стадии разработки находятся новые биометрические технологии: сравнение ДНК, отпечаток ладони, сосудистые рисунки, сигналы, вырабатываемые сердцем (мозгом, легкими).*

### 3.4.2. Поведенческие биометрические характеристики

Основные физиологические биометрические характеристики, а также виды их реализации приведены в таблице 3.2.

*Таблица 3.2*

**Реализация поведенческих биометрических характеристик**

| *Биометрическая характеристика* | *Регистрирующее устройство* | *Образец* | *Исследуемые черты* |
|---|---|---|---|
| Голос | Микрофон, телефон | Запись голоса | Частота, модуляция и продолжительность голосового образа |
| Подпись | Планшет для подписи, перо для ввода данных | Изображение подписи и показания соответствующих динамических измерений | Скорость, порядок линий, давление и внешний вид подписи |
| Динамика нажатия клавиш | Клавиатура | Ритм машинописи | Время задержки (промежуток времени, в течение которого пользователь удерживает конкретную клавишу) время «полета» (промежуток времени, который требуется пользователю для перехода с одной клавиши на другую) |

## 3.5. Недостатки аутентификации с помощью биометрических характеристик. Возможные атаки

В табл. 3.3 приведены известные методы атак на системы, использующие аутентификацию с помощью биометрических характеристик, а также способы защиты от подобных атак.

К недостаткам аутентификации с помощью биометрическсих характеристик можно отнести следующие:

**Вмешательство в частную жизнь.** Пользователям-клиентам в большей степени, чем пользователям-сотрудникам организаций, небезразличен факт хранения и распространения их биометрических данных. Если в организации устроено централизованное хранилище биометрических параметров, пользователи, не имея возможности контролировать распространение этих данных, опасаются:

- злоупотреблений (например, незаконного обмена с другими организациями);
- нецелевого использования («подмены функции»).

**Личные, культурные и религиозные аспекты.** Дактилоскопические системы вызывают неприятие у пользователей, которые считают, что их использование бросает на них тень преступного свойства, поскольку отпечатки пальцев, как известно, применяются в криминалистике.

Возникают также вопросы гигиены (будет ли прибор, регистрирующий геометрию руки, обрабатываться антисептическим раствором после каждого использования?) и травмоопасности (например, в системах сканирования сетчатки, в которых свет направляется в глаз), а также осознание того факта, что пользователи подвергаются риску причинения вреда со стороны преступников — от копирования или использования объектов биометрии под физическим принуждением до потери кисти или пальца.

*Таблица 3.3*

**Атаки на биометрические системы и защита от них**

| *Описание атаки* | *Защита от данной атаки* |
|---|---|
| ***Подделка отличительной черты*** | |
| Злоумышленник изготавливает копию физической отличительной черты законного пользователя и предъявляет эту копию биометрическому датчику. | *Снятие показателей с высоким уровнем детализации*<br>При изготовлении эталонного шаблона с законного пользователя снимают дополнительные биометрические показатели, так что простая копия физической отличительной черты законного пользователя не будет отражать все ее параметры. |
| ***Воспроизведение поведения пользователя*** | |
| Злоумышленник записывает поведенческую отличительную черту пользователя и воспроизводит на биометрическом датчике. | *Изменяемое поведение*<br>При каждой попытке аутентификации система требует от пользователя различного проявления его поведенческой биометрической характеристики, так что просто ее запись и воспроизведение не будут приниматься. |
| ***Перехват биометрических показателей*** | |
| Злоумышленник перехватывает биометрические показатели законного пользователя в момент их передачи между устройствами. | *Шифрование биометрических данных*<br>Биометрические данные шифруются сразу после их получения от пользователя устройством считывания, их передача между устройствами осуществляется только в шифрованном виде. |
| ***Воспроизведение биометрической «подписи»*** | |
| Злоумышленник воспроизводит показатель биометрического датчика — «подпись», которая далее обрабатывается системой так, словно была получена от реального человека. | *Аутентификация биометрической «подписи»*<br>Меры аутентификации принимаются в отношении биометрических данных, чем гарантируется их поступление только из заслуживающих доверия источников. Использование ЭЦП для обеспечения целостности биометрической «подписи». |

**Непригодность для всех пользователей.** От 1 до 3% процентов людей не имеют частей тела, необходимых для внесения в систему хотя бы одного биометрического параметра. Немые пользователи не могут использовать голосовые системы. Пользователи, у которых по причине врожденной болезни, хирургического вмешательства или ранения не хватает пальцев или кистей, не могут использовать системы, регистрирующие отпечатки пальцев и параметры кисти.

## Контрольные вопросы

1. Перечислите физиологические биометрические характеристики.
2. Назовите поведенческие биометрические характеристики.
3. Опишите принцип работы биометрических систем.
4. Назовите параметры, определяющие точность биометрических систем.
5. Приведите примеры атак на системы, использующие аутентификацию с помощью биометрических характеристик, и способы защиты от подобных атак.

# Глава 4
# АУТЕНТИФИКАЦИЯ С ПОМОЩЬЮ ОДНОРАЗОВЫХ ПАРОЛЕЙ

В основе всех методов аутентификации на основе пароля лежит предположение о том, что только законный пользователь может пройти проверку, так как только он знает свой пароль.

Заметим, что используемый при этом идентификатор пользователя злоумышленник может легко узнать. Особенно, если в качестве идентификатора пользователя используется не назначаемый пользователю идентификационный номер — ID (случайная строка символов, состоящая из букв и цифр), а так называемое «*имя пользователя*».

Так как обычно «*имя пользователя*» — это различные варианты комбинации имени и фамилии пользователя, то определить его не составляет большого труда. Соответственно, если злоумышленнику удастся узнать и пароль пользователя, то ему будет легко представиться этим пользователем. Насколько бы ни был пароль засекреченным, узнать его иногда не слишком трудно. Злоумышленник может сделать это, используя различные способы атак (см. выше раздел «Недостатки методов аутентификации с запоминаемым паролем» в гл. 2).

Для каждой из этих атак есть методы защиты. Но большинство из этих вариантов защиты обладают различными недостатками. Некоторые виды защиты достаточно дороги (например, борьба с побочным электромагнитным излучением и наводками оборудования), другие создают неудобства для пользователей (правила формирования пароля, использование длинного пароля). Один из вариантов защиты от различных атак на аутентификацию на основе пароля — переход на аутентификацию с помощью одноразовых паролей.

Применение схем одноразовых паролей стало заметным шагом вперед по сравнению с использованием фиксированных паролей. Выше мы говорили, что злоумышленник, узнавший фиксированный пароль, может повторно его использовать с целью выдачи себя за легального пользователя. Частным решением этой проблемы как раз и является применение одноразовых паролей: каждый пароль в данном случае используется только один раз.

---

*Одним из вариантов защиты от различных атак на аутентификацию по паролю является аутентификация с использованием одноразовых паролей.*

---

***Одноразовые пароли*** (OTP, One-Time Passwords) — динамическая аутентификационная информация, генерируемая для единичного использования с помощью аутентификационных устройств (программных или аппаратных).

Одноразовый пароль (OTP) неуязвим для атаки методом анализа сетевого трафика, что является значительным преимуществом перед запоминаемыми паролями. Несмотря на то, что злоумышленник может перехватить пароль методом анализа сетевого трафика, поскольку пароль действителен лишь один раз и в течение ограниченного промежутка времени, у злоумышленника в лучшем случае есть весьма ограниченная возможность представиться пользователем с помощью перехваченной информации.

*Одноразовый пароль действителен один раз в течение ограниченного времени и при перехвате такого пароля злоумышленник имеет ограниченную возможность представиться пользователем.*

В качестве возможных устройств для генерации одноразовых паролей обычно используются OTP-токены.

***OTP-токен*** — мобильное персональное устройство, которое принадлежит определенному пользователю и генерирует одноразовые пароли, используемые для аутентификации данного пользователя.

Таким образом, аутентификация с помощью одноразовых паролей, по сравнению с аутентификацией на основе пароля, является аутентификацией с помощью другого фактора аутентификации — аутентификацией «на основе обладания чем-либо».

Другим важным преимуществом применения аутентификационных устройств является то, что многие из них требуют от пользователя введения PIN-кода:

- для активации OTP-токена;
- в качестве дополнительной информации, используемой при генерации OTP;
- для предъявления серверу аутентификации вместе с OTP.

Если дополнительно применяется еще и PIN-код, в методе аутентификации используются два фактора аутентификации, т. е. данный метод относится к двухфакторной аутентификации.

Простейшей схемой применения одноразовых паролей служит разделяемый список. В этом случае пользователь и проверяющий применяют последовательность секретных паролей, где каждый пароль используется только один раз.

Естественно данный список заранее распределяется между сторонами аутентификационного обмена.

Такая схема применяется в настоящее время в некоторых системах «Интернет-банк».

Модификацией этого метода является таблица вопросов и ответов, которая содержит вопросы и ответы, используемые сторонами для проведения аутентификации, причем каждая пара используется только один раз. Существенным недостатком этой схемы является необходимость предварительного распределения аутентифицирующей информации. После того как выданные пароли закончатся, пользователю необходимо получить новый список. Такое решение, во-первых, не удовлетворяет современным представлениям об информационной безопасности, поскольку злоумышленник может украсть или скопировать список паролей пользователя. Во-вторых, постоянно получать новые списки паролей вряд ли кому-нибудь понравится.

Вместе с тем, в настоящее время разработано несколько методов реализации технологии одноразовых паролей, исключающих указанные недостатки. В их основу легли различные криптографические алгоритмы.

*Простейшей схемой применения одноразовых паролей служит разделяемый список. Существенным недостатком этой схемы является необходимость предварительного распределения аутентифицирующей информации.*

## 4.1. Аппаратно-программные OTP-токены

OTP-токены имеют небольшой размер и выпускаются в виде:
- карманного калькулятора;
- брелока;
- смарт-карты;
- устройства, комбинированного с USB-ключом;
- специального программного обеспечения для карманных компьютеров, смартфонов, настольных компьютеров.

## 4.2. Как работают OTP-токены

Для генерации одноразовых паролей OTP-токены используют хэш-функции или криптографические алгоритмы:
- *симметричная криптография* (криптография с одним ключом) — в этом случае пользователь и сервер аутентификации используют один и тот же секретный ключ;
- *асимметричная криптография* (криптография с открытым ключом) — в этом случае в устройстве хранится закрытый ключ, а сервер аутентификации использует соответствующий открытый ключ.

*Для генерации одноразовых паролей OTP-токены используют хэш-функции или криптографические алгоритмы.*

Существуют различные комбинации использования данных криптографических алгоритмов в реализациях OTP-токенов.

Соответственно механизмы аутентификации, используемые OTP-токенами, можно разделить на две группы:
- аутентификация с одним секретным ключом,
- аутентификация с открытым ключом.

## 4.3. Методы аутентификации с помощью OTP-токенов

Обычно в OTP-токенах применяется симметричная криптография. Устройство каждого пользователя содержит уникальный персональный секретный ключ, используемый для шифрования некоторых данных (в зависимости от реализации метода) для генерации OTP. Этот же ключ хранится на сервере аутентификации, который выполняет аутентификацию данного пользователя. Сервер шифрует те же данные и сравнивает два результата шифрования: полученный им и присланный от клиента. Если результаты совпадают, то пользователь успешно проходит аутентификацию.

OTP-токены, использующие симметричную криптографию, могут работать в асинхронном или синхронном режиме. Соответственно методы, используемые OTP-токенами, можно разделить на две группы, работающие:
- в асинхронном режиме («запрос-ответ»);
- в синхронном режиме («только ответ», «синхронизация по времени», «синхронизация по событию»).

### 4.3.1. Метод «запрос—ответ» (Challenge—response)

В методе «запрос—ответ» OTP является ответом пользователя на случайный запрос от сервера аутентификации (рис. 4.1).

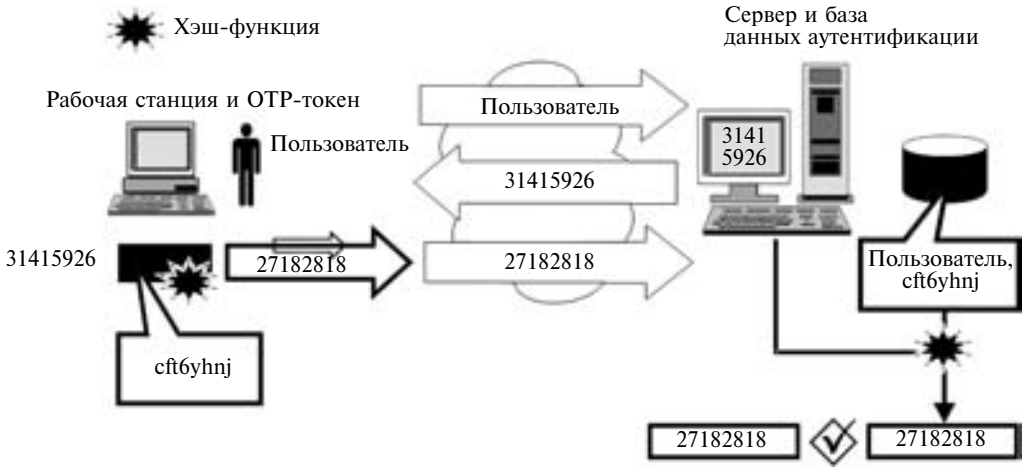



**Рис. 4.1. Метод «Запрос—ответ»**

Пример аутентификации пользователя при использовании OTP-токеном метода «запрос—ответ»:

1. Пользователь вводит свое имя пользователя на рабочей станции.
2. Имя пользователя передается по сети в открытом виде.
3. Сервер аутентификации генерирует случайный запрос («31415926»).
4. Запрос передается по сети в открытом виде.
5. Пользователь вводит запрос в свой OTP-токен.
6. OTP-токен шифрует запрос с помощью секретного ключа пользователя («cft6yhnj»), в результате получается ответ («27182818»), который отображается на экране OTP-токена.
7. Пользователь вводит этот ответ на рабочей станции.
8. Ответ передается по сети в открытом виде.
9. Аутентификационный сервер находит запись пользователя в аутентификационной базе данных и с помощью хранимого им секретного ключа пользователя зашифровывает тот же запрос.
10. Сервер сравнивает представленный ответ от пользователя («27182818») с вычисленным им самим ответом («27182818»).
11. При совпадении значений аутентификация считается успешной.

### 4.3.2. Метод «только ответ» (Response only)

В методе «только ответ» аутентификационное устройство и сервер аутентификации генерируют «скрытый» запрос, используя значения предыдущего запроса. Для начальной инициализации данного процесса используется уникальное случайное начальное значение, генерируемое при инициализации OTP-токена.

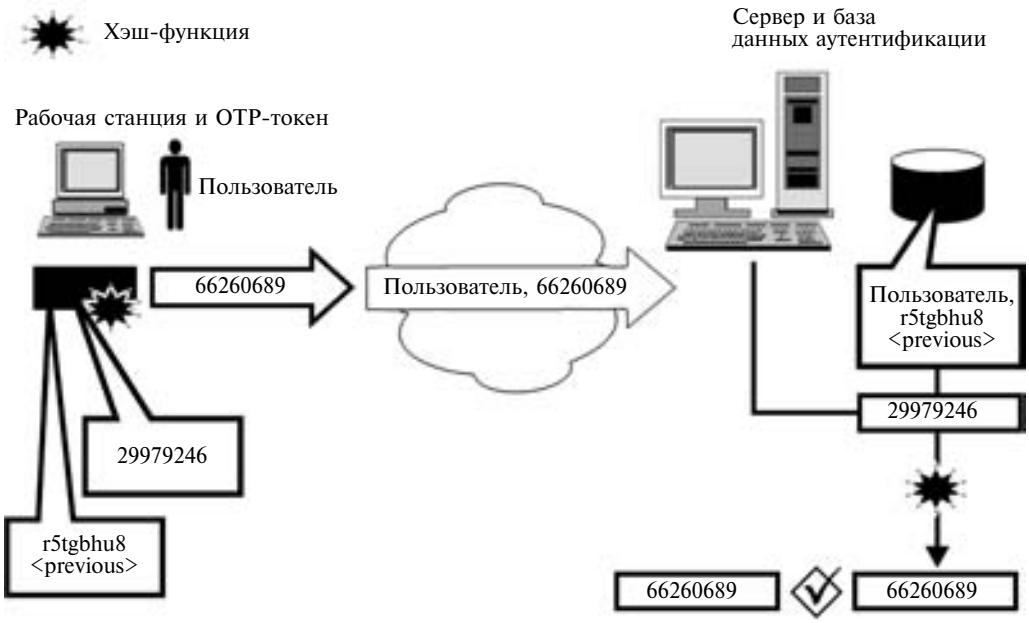



**Рис. 4.2. Метод «Только ответ»**

Пример аутентификации пользователя при использовании OTP-токеном метода «только ответ» (рис. 4.2):

1. Пользователь активизирует свой OTP-токен, который вычисляет и отображает ответ на «скрытый» запрос.

2. Пользователь вводит свое «имя пользователя» и этот ответ («66260689») на рабочей станции.

3. Имя пользователя и ответ («66260689») передаются по сети в открытом виде.

4. Сервер находит запись пользователя, генерирует такой же скрытый запрос и шифрует его с помощью секретного ключа пользователя, получая ответ на свой запрос.

5. Сервер сравнивает представленный ответ от пользователя («66260689») с вычисленным им самим ответом («66260689»).

6. При совпадении значений аутентификация считается успешной.

### *4.3.3. Метод «Синхронизация по времени» (Time synchronous)*

В режиме «синхронизация по времени» аутентификационное устройство и аутентификационный сервер генерируют OTP на основе значения внутренних часов. OTP-токен может использовать не стандартные интервалы времени, измеряемые в минутах, а специальные интервалы времени обычно равные 30 с.

Пример аутентификации пользователя при использовании OTP-токеном метода «синхронизация по времени» (рис. 4.3):

1. Пользователь активизирует свой OTP-токен, который генерирует OTP («96823030»), зашифровывая показания часов с помощью своего секретного ключа.

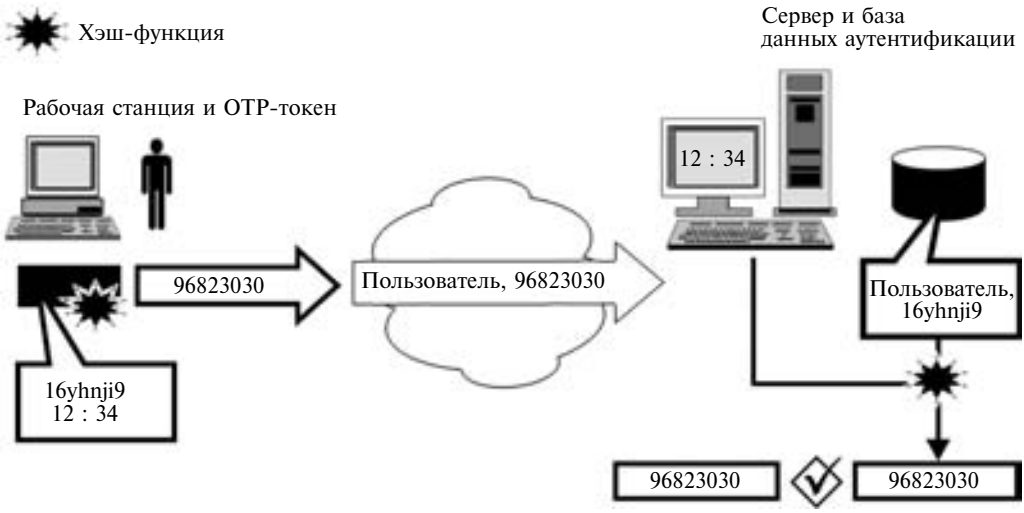



**Рис. 4.3. Метод «Синхронизация по времени»**

2. Пользователь вводит свое «имя пользователя» и этот OTP на рабочей станции.

3. Имя пользователя и OTP передаются по сети в открытом виде.

4. Аутентификационный сервер находит запись пользователя и шифрует показание своих часов с помощью хранимого им секретного ключа пользователя, получая в результате OTP.

5. Сервер сравнивает OTP, представленный пользователем, и OTP, вычисленный им самим.

6. При совпадении значений аутентификация считается успешной.

### *4.3.4. Метод «синхронизация по событию» (Event synchronous)*

В режиме «синхронизация по событию» OTP-токен и сервер аутентификации ведут количественный учет прохождения аутентификации данным пользователем, и на основе этого числа генерируют OTP.

Пример аутентификации пользователя при использовании OTP-токеном метода «синхронизация по событию» (рис. 4.4):

1. Пользователь активизирует свой OTP-токен, который генерирует OTP («59252459»), зашифровывая число раз прохождения аутентификации данного пользователя с помощью своего секретного ключа.

2. Пользователь вводит свое «имя пользователя» и этот OTP на рабочей станции.

3. Имя пользователя и OTP передаются по сети в открытом виде.

4. Аутентификационный сервер находит запись пользователя и шифрует значение числа раз прохождения аутентификации данного пользователя с помощью хранимого им секретного ключа пользователя, получая в результате OTP.

5. Сервер сравнивает OTP, представленный пользователем, и OTP, вычисленный им самим.

6. При совпадении значений аутентификация считается успешной.

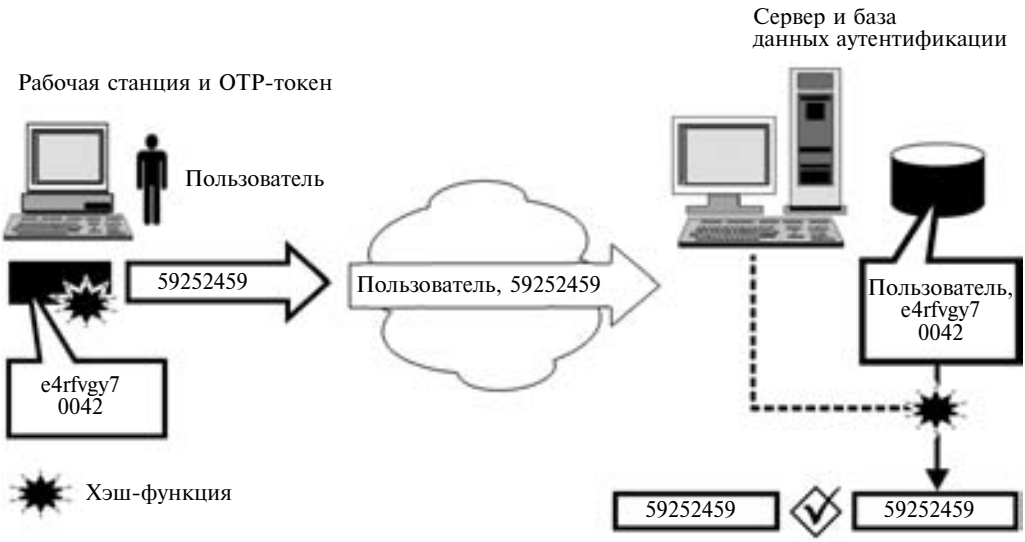



**Рис. 4.4. Метод «Синхронизация по событию»**

Некоторые OTP-токены могут использовать несколько различных методов реализации аутентификации с помощью OTP. Наиболее часто комбинируются методы «синхронизация по времени» и «синхронизация по событию».

## 4.4. Сравнение методов OTP-аутентификации

Метод «запрос—ответ», работающий в асинхронном режиме, предполагает большее количество шагов, совершаемых пользователем, чем любой из синхронных режимов.

Потенциальная проблема всех методов реализации аутентификации с помощью OTP, работающих в синхронном режиме, — возможность рассинхронизации OTP-токена и сервера, например:

- в режимах «только ответ» или «синхронизации по событию» сбой при аутентификации может привести к «отставанию» сервера от аутентификационного устройства;
- в режиме «синхронизации по времени» часы аутентификационного устройства могут уйти вперед или отстать от часов сервера.

*Потенциальная проблема всех методов реализации аутентификации с помощью OTP, работающих в синхронном режиме, — возможность рассинхронизации OTP-токена и сервера.*

При аутентификации с попощью OTP-токенов, как правило, предусматривается вариант решения проблемы рассинхронизации: сервер генерирует несколько возможных вариантов OTP — «*ответов*» от пользователя за некоторый короткий промежуток времени (для нескольких событий или единиц измерения времени).

# 4.5. Системы одноразовых паролей

## 4.5.1. Система S/Key

**Система S/Key**— система одноразовых паролей, разработанная в Беллкоре (Bell Communication Research Labs, Bellcore Labs) в начале 1990-х гг. в качестве метода регистрации для UNIX-систем.

Техническая концепция была впервые предложена Лесли Лэмпортом (Leslie Lamport) и опубликована в 1981 г. Основное отличие подхода Лэмпорта от других методик на основе принципа «запрос—ответ» состояло в том, что не было базы данных секретных ключей, поэтому взломщики не могли поставить под угрозу работу системы, украв эту базу данных.

В схеме Лэмпорта (рис. 4.5) используется последовательность значений односторонних хэш-функций, вычисляемых из базового секрета. Как и в случае традиционной парольной аутентификации в UNIX-системах, в схеме Лэмпорта использован тот факт, что вычисление хэшированного значения пароля не представляет сложности, а вот обратное получение пароля по значению хэша невозможно. В схеме Лэмпорта используется последовательность значений хэш-функции, каждое из которых вычисляется из предыдущего члена последовательности. Сервер хранит последнее значение хэш-функции в последовательности.

Схему Лэмпорта для трех актов аутентификации можно представить в виде последовательности следующих шагов:

1. Четыре раза последовательно вычисляется значение хэш-функции базового секрета пользователя. Конечный результат этих вычислений сохраняется в базе данных аутен-

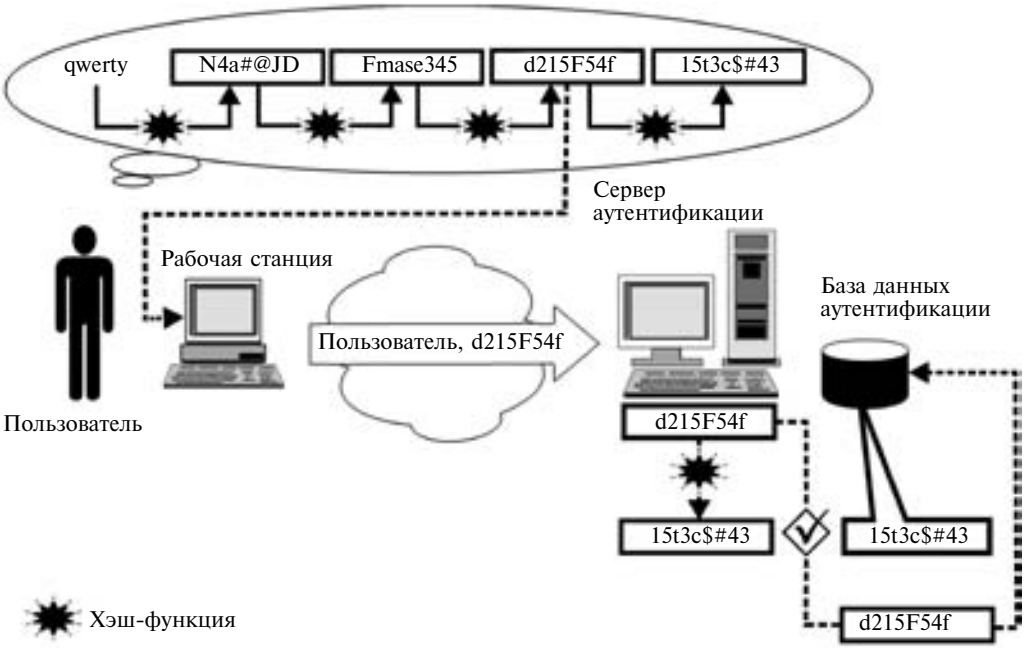



**Рис. 4.5. Схема Лэмпорта**

тификации, а промежуточные выдаются пользователю либо повторно вычисляются им при каждом акте аутентификации.

2. Пользователь в качестве одноразового пароля предоставляет предпоследнее в последовательности значение хэш-функции.

3. Сервер принимает одноразовый пароль, вычисляет значение хэш-функции и сравнивает со значением хэш-функции, хранящимся в базе данных аутентификации. Эти значения должны совпасть.

4. При совпадении (успешной аутентификации) сервер заменяет значение хэш-функции в учетной записи пользователя (*четвертое* значение хэш-функции) значением пароля, только что принятым от пользователя (*третьим* значением хэш-функции). При следующем входе пользователя в систему он должен предоставить *второе* значение хэш-функции, а при последнем входе — первое значение.

Схема Лэмпорта реализована в системе S/Key. В этой методике используются синхронные одноразовые пароли. В качестве одноразового пароля пользователь должен предоставлять предпоследнее значение хэш-функции. Это требует точного учета использованных паролей, а пользователи не очень сильны в подобной бухгалтерии. Но, как правило, S/Key-серверы подсказывают пользователю порядковый номер ожидаемого значения хэш-функции. Таким образом, в системе S/Key используется как бы «запрос—ответ», хотя в действительности эта информация опционна и предоставляется только для удобства пользователя.

В системе S/Key функция хэширования также включает в себя и случайное число, называемое «*примесью»,* которое объединяется с базовым секретом при генерации значений хэш-функции. «Примесь» не позволяет системе S/Key генерировать одинаковые последовательности значений хэш-функции, если пользователь попытается повторно воспользоваться базовым секретом или использовать один и тот же базовый секрет для разных компьютеров. Хотя на рисунке показано, что в файле паролей хранится только значение хэш-функции, система S/Key хранит также значение «примеси» и порядковый номер значения хэш-функции. Когда S/Key-сервер выдает запрос, содержащий текущий порядковый номер хэш-функции пользователя, он одновременно выводит и значение «примеси», используемое для генерации значений хэш-функции.

Как правило, пользователи системы S/Key используют для генерации одноразовых паролей программно реализованные OTP-токены. Чтобы воспользоваться программным OTP-токеном для данной системы, пользователь вводит базовый секрет (пароль), порядковый номер и значение «примеси». OTP-токен итеративно использует функцию хэширования для генерации правильного значения в последовательности и затем выводит результирующее значение хэш-функции. После этого пользователь копирует значение хэш-функции в ожидающее окно запроса пароля.

*Как правило, пользователи системы S/Key используют для генерации одноразовых паролей программно реализованные OTP-токены.*

Программные реализации устройства аутентификации существуют для операционных систем типа UNIX, Microsoft и Macintosh. Программное устройство аутентификации обычно само обнаруживает запрос системы S/Key, так что оно способно автоматически вычислять правильное значение одноразового пароля. Когда это возможно, устройство аутентификации поддерживает функцию вырезания и вставки через буфер, что позволяет избегать ошибок набора при копировании запроса или ответа. Кроме того, для пользо-

вателей, которые не имеют возможности запускать на выполнение программу устройства аутентификации, имеется утилита, распечатывающая значения хэш-функции на бумаге. Хотя аппаратная реализация устройства аутентификации для системы S/Key технически несложна, на данный момент промышленных моделей не существует.

До тех пор пока сервер хранит только последнее значение хэш-функции из последовательности, а пользователь предоставляет в качестве пароля предпоследнее значение хэш-функции, злоумышленнику непросто получить действующее значение пароля. Он не может извлечь значение хэш-функции из файла паролей и произвести обратные вычисления предыдущего значения хэш-функции из последовательности или исходного значения базового секрета.

*Аппаратная реализация устройства аутентификации для системы S/Key технически несложна, на данный момент промышленных не существует.*

### 4.5.2. Группа OATH и система HOTP

Система HOTP (*HMAC-based One-Time Password System)* была разработана в 2005 г. в рамках инициативы группы открытой аутентификации OATH (Open AuTHentification) и описана в документе RFC 4226. Данная система основана на концепции OTP-аутентификации с синхронизацией по событию. Для генерации одноразового пароля используется алгоритм HMAC (Hashed Message Authentication Code).

*Система HOTP основана на концепции OTP-аутентификации с синхронизацией по событию.*

Этот алгоритм публичен и доступен для изучения любыми специалистами. Он обеспечивает возможность аутентификации в широком спектре программного обеспечения, в том числе и серверного.

Система HOTP предусматривает возможность задания «окна» попыток аутентификации и синхронизацию сервера аутентификации с OTP-токеном после успешного прохождения аутентификации.

Значение одноразового пароля вычисляется по формуле

$$HOTP(K, C) = Truncate(HMAC\text{-}SHA\text{-}1(K, C)),$$

где

- K — секретный ключ;
- C — счетчик числа раз прохождения аутентификации;
- HMAC-SHA-1 — процедура генерации HMAC, основанная на функции хэширования SHA-1;
- Truncate — процедура усечения 20-байтового значения HMAC-SHA-1 до 4 байт.

Пример аутентификации пользователя с помощью HOTP (рис. 4.6):

1. Пользователь генерирует значение HOTP с использованием хранимого на OTP-токене значения числа раз прохождения аутентификации и секретного ключа (592524594012).

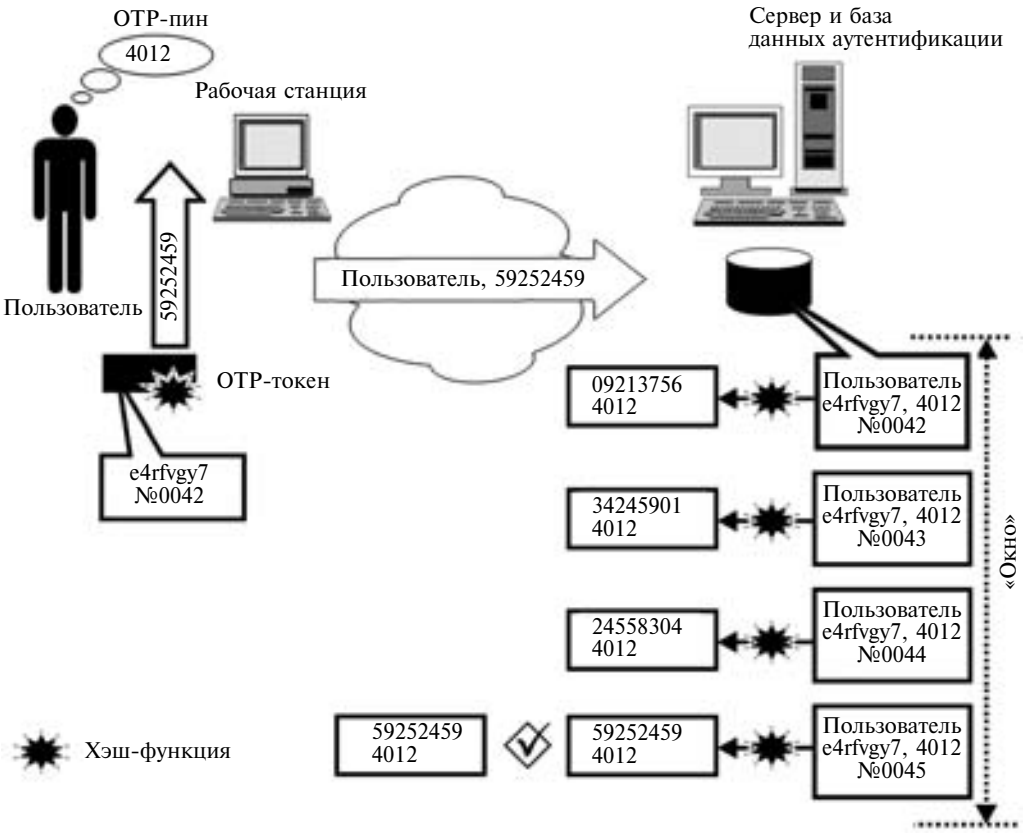



**Рис. 4.6. Система HOTP**

2. Пользователь вводит свое «имя пользователя» и OTP на рабочей станции.

3. Имя пользователя и OTP (592524594012) передаются по сети в открытом виде.

4. Аутентификационный сервер находит запись пользователя и генерирует OTP, используя хранимые на сервере значения числа раз прохождения аутентификации данного пользователя и секретного ключа пользователя и получая в результате OTP (592524594012).

5. Сервер сравнивает OTP, представленный от пользователя, и OTP, вычисленный им самим.

a. Если значения не совпадают, сервер увеличивает значение числа раз прохождения аутентификации пользователя на единицу и повторяет попытку.

b. Если значения совпадают — на сервере сохраняется новое значение числа раз прохождения аутентификации пользователя.

Аутентификация считается успешной.

c. Если достигнуто максимальное число неуспешных повторов процедуры аутентификации (задаваемое шириной «окна»), аутентификация считается неуспешной.

## 4.6. Недостатки методов аутентификации с помощью OTP. Возможные атаки

Ниже приведены известные атаки на системы, использующие аутентификацию с помощью OTP-токенов, и защиты от них.

*Таблица 4.1*

**Атаки на одноразовые пароли и защита от них**

| *Описание атаки* | *Защита от данной атаки* |
|---|---|
| ***Атака «Человек посередине»*** | |
| Злоумышленник перехватывает одноразовый пароль, посланный законным пользователем при аутентификации, блокирует законного пользователя и использует перехваченный пароль для входа в систему. | *Использование метода «запрос—ответ»*<br>Использование вместо синхронных одноразовых паролей, имеющих легитимность «в продолжительном» периоде времени, одноразовых паролей, работающих по принципу «запрос-ответ». Каждое новое соединение требует выполнения аутентификации заново. |
| ***Кража аутентификационного токена*** | |
| Злоумышленник похищает аутентификационный токен законного пользователя и использует его для входа в систему. | *PIN-коды в аутентификационных токенах*<br>Использование аутентификационных токенов, требующих от владельца ввода PIN-кода перед началом генерации OTP. |
| ***Подбор PIN-кода аутентификационного токена*** | |
| Злоумышленник вручную производит перебор всех возможных значений PIN-кода похищенного им аутентификационного токена законного пользователя. | *Блокирование после ввода неправильного PIN-кода*<br>Аутентификационный токен отключается после того, как пользователь вводит неправильное значение PIN-кода подряд более заданного количества раз.<br>*Увеличение задержки для каждого ввода неправильного PIN-кода*<br>Если вводится неправильное значение PIN-кода, то следующая попытка ввода PIN-кода возможна только через определенный промежуток времени, с каждым неправильным вводом эта задержка увеличивается. |
| ***Извлечение значения секретного ключа из программного аутентификационного токена*** | |
| Злоумышленник копирует программный аутентификационный токен (программное обеспечение), пытается найти в нем хранимый секретный ключ, чтобы потом его использовать для аутентификации под видом законного пользователя | *PIN-код является частью секретного ключа*<br>Частью секретного ключа аутентификационного токена является PIN-код, без его знания нельзя сгенерировать правильный OTP, даже зная часть секретного ключа, который хранится в программном аутентификационном токене. |
| ***Подбор PIN-кода аутентификационного токена с помощью известных OTP*** | |
| Злоумышленник перехватывает несколько правильных OTP, использованных для входа в систему, копирует программный аутентификационный токен (программное обеспечение), затем он пытается подобрать PIN-код путем перебора его возможных значений, для тестирования пробного значения PIN-кода используются перехваченные OTP. | *Использование «аппаратных» аутентификационных токенов*<br>В этом случае достаточно сложно произвести «в реальные сроки» перебор возможных значений PIN-кода до момента обнаружения владельцем пропажи токена и «информирования аутентификационного сервера» о том, что данный токен может быть использован злоумышленником. |

*Окончание таблицы 4.1*

| *Описание атаки* | *Защита от данной атаки* |
|---|---|
| ***Нечестный администратор аутентификационных токенов*** | |
| Злоумышленник является доверенным лицом либо является посредником доверенного лица, производящего инициализацию аутентификационного устройства до передачи его владельцу. Он может создать дубликат токена и, используя его, выдавать себя за владельца. | *Разделение ответственности при инициализации аутентификационных токенов*<br>В процессе программирования и активирования токена должны участвовать двое или более людей, каждый из которых выполняет строго ограниченный набор операций. |

### Примечание

При использовании программных аутентификационных токенов, строго говоря, эмулятор или отдельный закрытый ключ подтверждает подлинность только рабочей станции, а не пользователя. Даже при условии защиты с помощью PIN-кода, строгая двухфакторная аутентификация заменяется на однофакторную. Любому лицу, имеющему физический доступ к рабочей станции, чтобы представиться пользователем, остается только узнать его PIN-код. Этот подход может оказаться достаточно эффективным для сотрудников и клиентов, работающих дома, но он неприемлем для использования в офисе, его целесообразность для мобильных сотрудников сомнительна.

## Контрольные вопросы

1. Что такое одноразовые пароли?
2. Опишите принцип работы OTP-токенов.
3. Приведите пример аутентификации пользователя при использовании OTP-токеном метода «запрос—ответ».
4. Приведите пример аутентификации пользователя при использовании OTP-токеном метода «только ответ».
5. Приведите пример аутентификации пользователя при использовании OTP-токеном метода «синхронизация по времени».
6. Приведите пример аутентификации пользователя при использовании OTP-токеном метода «синхронизация по событию».

# Глава 5
# КРИПТОГРАФИЯ С ОТКРЫТЫМ КЛЮЧОМ

## 5.1. Общие сведения о криптографии с открытым ключом

### *5.1.1. Использование криптографии с открытым ключом*

В *криптографии с открытым ключом (асимметричная криптография)* алгоритмы используют связанные между собой пары ключей, состоящие из открытого и закрытого ключа. Для каждого человека или объекта генерируется ***ключевая пара:***

- ***открытый ключ***, доступный для всех;
- ***закрытый ключ***, известный только человеку, которому он выдан, и никому другому не раскрывается и никуда не передается.

Информация, зашифрованная с помощью одного ключа из ключевой пары, может быть расшифрована только с помощью другого ключа из этой же пары. Ключи математически связаны между собой так, что, зная открытый ключ, практически невозможно вычислить закрытый. Пользователь может повсеместно распространять свой открытый ключ, но он должен обязательно защищать свой закрытый ключ.

Криптографические методы защиты используют операцию преобразования информации, которая может выполняться одним или несколькими пользователями, обладающими некоторым секретом, без знания которого (с вероятностью, близкой к единице за разумное время) невозможно осуществить эту операцию.

К криптографическим методам защиты в общем случае относятся:

- шифрование информации (термин шифрование объединяет в себе два процесса: зашифровывание и расшифровывание информации);
- формирование и проверка цифровой подписи электронных документов.

***Электронная цифровая подпись (ЭЦП)*** — это реквизит электронного документа, который предназначен для защиты данного электронного документа от подделки, получен в результате криптографического преобразования информации с помощью закрытого ключа электронной цифровой подписи и позволяющий идентифицировать владельца сертификата ключа подписи, а также установить отсутствие искажения информации в электронном документе.

Применение электронной цифровой подписи позволяет:

- обеспечить подтверждение авторства (аутентичность) информации;
- обеспечить контроль целостности (в том числе истинности) информации;
- решать вопрос о юридическом статусе электронных документов, а значит, решать задачу разграничения ответственности между взаимодействующими абонентами (субъектами).

Таким образом, криптографию с открытым ключом можно использовать для:

- предотвращения возможности несанкционированного ознакомления с информацией при ее хранении в компьютере или на отчуждаемых носителях, а также при передаче по каналам связи;
- подтверждения подлинности электронного документа, доказательства авторства документа и факта его получения от соответствующего источника информации;
- обеспечения гарантий целостности (имитостойкости) — исключение возможности необнаружения несанкционированного изменения информации;
- аутентификации пользователей системы — владельцев секретных ключей.

Ниже приведен пример использования криптографии с открытым ключом для шифрования сообщения.

Для обеспечения конфиденциальности данных Автор может отправить личное сообщение Получателю, зашифровав его с помощью открытого ключа Получателя (находящегося в свободном доступе), потому что только Получатель обладает закрытым ключом и может расшифровать данное сообщение.

Поскольку ассиметричная криптография требует достаточно больших вычислительных ресурсов, обычно на практике она применяется в комбинации с симметричной криптографией. Само сообщение шифруется с помощью использования симметричного алгоритма, а секретный (сеансовый) ключ, использованный при шифровании данного сообщения, шифруется с помощью ассиметричного алгоритма для передачи сеансовых ключей по сети.

Пример использования криптографии с открытым ключом для электронной цифровой подписи сообщения приведен на рис. 5.1.

Сервер аутентификации хранит файл открытых ключей всех пользователей.

Для обеспечения удостоверения подлинности источника данных Автор может отправить Получателю сообщение, зашифровав его с помощью своего закрытого ключа. Получатель может быть уверен, что сообщение поступило именно от Автора, если он сможет расшифровать его с помощью использования его открытого ключа. Таким образом, личность отправителя может быть однозначно удостоверена, поскольку по определению только Автор имеет доступ к своему закрытому ключу.

Поскольку данные криптографические преобразования производятся над значением хэш-функции документа, любое изменение содержания документа приводит к уничтожению подписи автора документа. Таким путем можно гарантировать, что в документ, если он «содержит подпись» Автора, никто не вносил каких-либо изменений, кроме самого Автора данного документа.

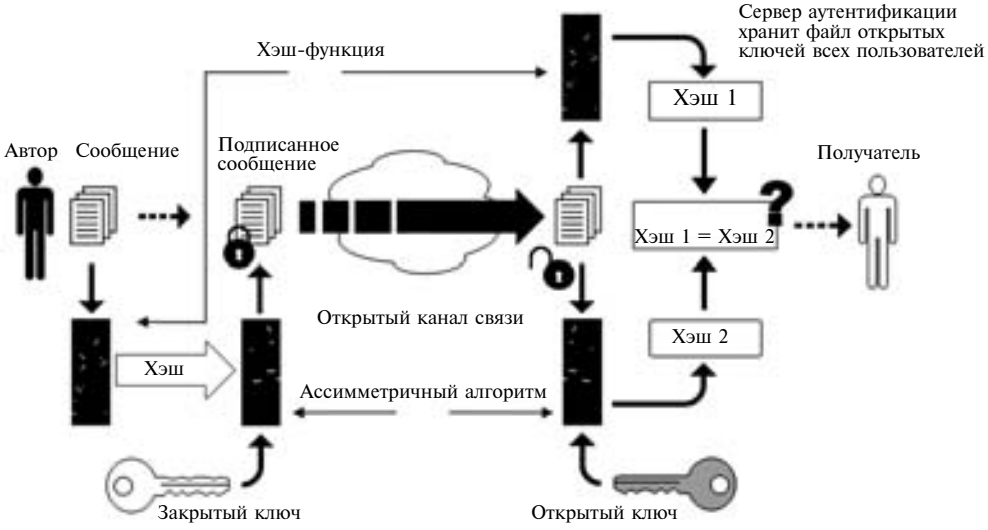



**Рис. 5.1. Пример использования криптографии с открытым ключом**

### 5.1.2. Аутентификация с помощью открытого ключа

Аутентификационный сервер хранит файл открытых ключей всех пользователей, а все пользователи хранят свои закрытые ключи. Пример аутентификации пользователя с помощью открытых ключей (упрощенный вариант) приведен на рис. 5.2:

1. Сервер посылает пользователю случайную строку, созданную генератором случайных чисел (ГСЧ).

2. Пользователь шифрует эту строку своим закрытым ключом и посылает ее обратно серверу вместе со своим именем.

3. Сервер находит в базе данных открытый ключ пользователя и расшифровывает сообщение, используя этот открытый ключ.

4. Если отправленная и расшифрованная строки совпадают, сервер предоставляет пользователю доступ к системе.

Никто другой не может воспользоваться закрытым ключом Пользователя, следовательно, никто не сможет выдать себя за него. Что более важно, Пользователь никогда не посылает на компьютер свой закрытый ключ. Злоумышленник, перехватывая сообщения, не получит никаких сведений, которые позволили бы ему вычислить закрытый ключ Пользователя и выдать себя за него.

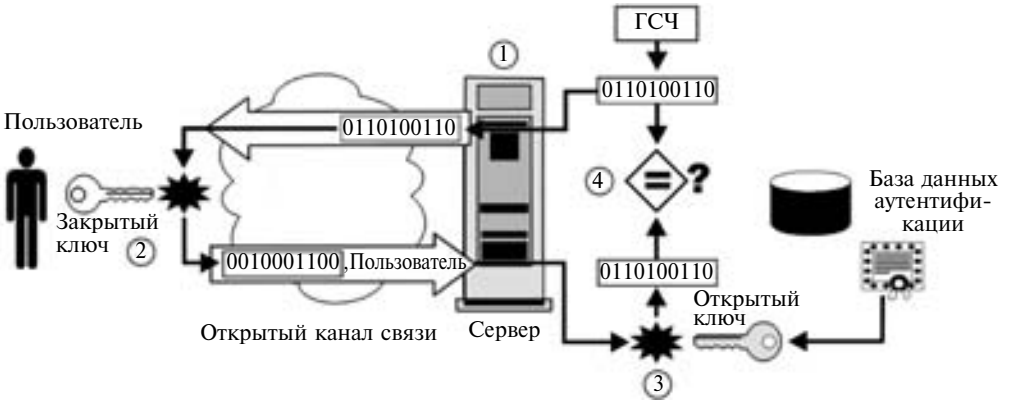



**Рис. 5.2. Аутентификация с помощью открытого ключа (упрощенный вариант)**

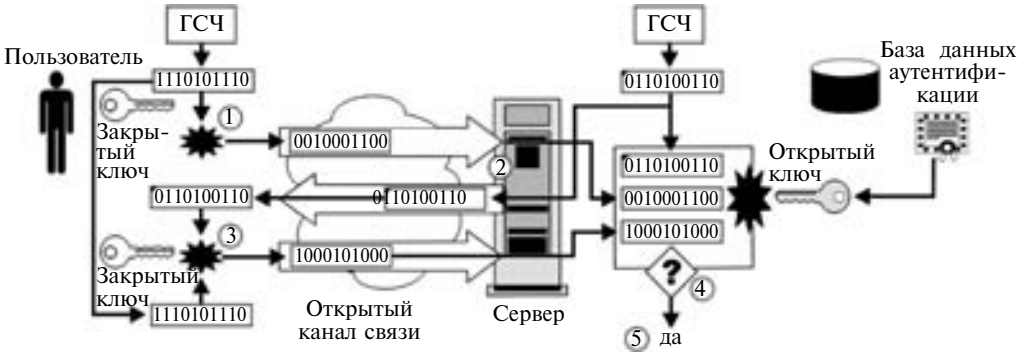



**Рис. 5.3. Аутентификация с помощью открытого ключа**

В данной процедуре на шаге 1 производится шифрование «случайной строки, присланной от сервера», что служит потенциальной уязвимостью процедуры, так как этим может воспользоваться злоумышленник для взлома данного протокола с помощью подобранного шифртекста. Безопасные идентификационные протоколы имеют более сложную форму (рис. 5.3):

1. Пользователь выполняет вычисление, основанное на некоторых случайных числах в своем закрытом ключе, и посылает результат серверу.

2. Сервер посылает другое случайное число.

3. Пользователь выполняет некоторое вычисление, основанное на случайных числах (как созданных им, так и полученных от сервера) в своем закрытом ключе, и посылает результат серверу.

4. Сервер выполняет некоторое вычисление для различных чисел, полученных от Пользователя, и его открытого ключа, проверяя, что Пользователю известен его закрытый ключ.

5. Если проверка завершается успешно, личность Пользователя подтверждается.

В этом случае шаг 1 позволяет защитить протокол от вскрытия с помощью подобранного шифртекста.

Данный протокол широко используется, если криптография с открытым ключом применяется в рамках одного небольшого предприятия, когда число пользователей невелико. Если же криптографию с открытым ключом используют для большого числа пользователей или нескольких предприятий, необходимо иметь инфраструктуру для управления ключами.

### *5.1.3. Аутентификация с помощью открытого ключа на основе российских криптографических алгоритмов*

В 2001 г. по инициативе российской компании «КРИПТО-ПРО» предложены «Рекомендации к средствам криптографической защиты информации на взаимодействие удостоверяющих центров, реестров сертификатов, сертификаты ключей формата X.509 и электронные документы формата CMS». Указанные Рекомендации составлены с учетом международных стандартов и рекомендаций, директивы Европейского парламента «Об электронной цифровой подписи» (1999/C 243/02) и рабочих документов Европейского института стандартов по телекоммуникациям ETSI (European Telecommunications Standards Institute).

Рекомендации описывают способ реализации и содержат требования на форматы открытых ключей и электронной цифровой подписи при использовании российских криптографических стандартов ГОСТ Р 34.10—94, ГОСТ Р 34.11—94. В рамках проходившего 57 заседания IETF (Internet Engineering Task Force) в Вене компания «КРИПТО-ПРО» представила три проекта информационных документов (Internet-Drafts), которые были приняты к дальнейшему рассмотрению. В проектах приводится описание использования криптографических алгоритмов ГОСТ 28147—89, ГОСТ Р 34.10—94, ГОСТ Р 34.10—2001, ГОСТ Р 34.11—94 в сертификатах открытых ключей, криптографических сообщениях и протоколе TLS. Впоследствии к существующим документам были добавлены еще два проекта, в которых приводятся описания ЭЦП в формате XML и криптографических алгоритмов шифрования и преобразования ключей.

В начале 2006 г. на официальном сайте IETF был опубликован первый в истории сообщества Интернет-стандарт для применения указанных российских криптографических алгоритмов — RFC 4357. В настоящее время комитет IETF утвердил и опубликовал новые

стандарты: RFC 4490 и RFC 4491, определяющие использование алгоритмов российских стандартов ГОСТ Р 34.11—94, ГОСТ Р 34.10—94, ГОСТ Р 34.10—2001, ГОСТ 28147—89 с включением их в документацию IETF (RFC) со статусом Internet-стандартов. RFC 4357 — INFORMATIONAL (стандартом не является, но на него нормативно ссылаются RFC 4490 и RFC 4491).

В настоящее время действуют следующие признанные на международном уровне документы, касающиеся использования российских криптографических алгоритмов:

- RFC4357 — описание дополнительных криптографических алгоритмов для использования совместно с ГОСТ 28147—89, ГОСТ Р 34.10—94, ГОСТ Р 34.10—2001 и ГОСТ Р 34.11—94;
- RFC4490 — описание использования в CMS (Cryptographic Message Syntax) алгоритмов ГОСТ 28147—89, ГОСТ Р 34.10—94, ГОСТ Р 34.10—2001 и ГОСТ Р 34.11—94;
- RFC4491 — описание использования алгоритмов ГОСТ 28147—89, ГОСТ Р 34.10—94, ГОСТ Р 34.10—2001 и ГОСТ Р 34.11—94 в сертификатах открытых ключей и в CRL;
- draft-chudov-cryptopro-cptls — описание использования алгоритмов ГОСТ 28147—89, ГОСТ Р 34.10—94, ГОСТ Р 34.10—2001 и ГОСТ Р 34.11—94 в протоколе TLS;
- draft-chudov-cryptopro-cpxmldsig — описание использования алгоритмов ГОСТ Р 34.10—94, ГОСТ Р 34.10—2001 и ГОСТ Р 34.11—94 в XML.

Эти документы разработаны в рамках соглашения о совместимости криптографических продуктов между ведущими российскими разработчиками средств криптографической защиты конфиденциальной информации.

Российские криптографические алгоритмы позволяют обеспечить реализацию следующих функций защиты информации:

- авторизация и обеспечение юридической значимости электронных документов;
- конфиденциальность и контроль целостности передаваемой информации;
- аутентификация связывающихся сторон;
- установление аутентичного защищенного соединения для обмена информацией.

## 5.2. Авторизация и обеспечение юридической значимости электронных документов

Авторизация и обеспечение юридической значимости электронных документов при обмене ими между пользователями осуществляется с помощью процедур формирования и проверки ЭЦП. Алгоритмы формирования и проверки ЭЦП определяются криптографическими стандартами:

- ***ГОСТ Р 34.10—94***. «Информационная технология. Криптографическая защита информации. Процедуры выработки и проверки электронной цифровой подписи на базе асимметричного криптографического алгоритма» (этот стандарт действителен до 31.12.2007). Данный стандарт определяет процедуры формирования и проверки ЭЦП с выполнением криптографических преобразований в экспоненциальной логике.
- ***ГОСТ Р 34.10—2001***. «Информационная технология. Криптографическая защита информации. Процессы формирования и проверки электронной цифровой подписи». Этот стандарт определяет процедуры формирования и проверки ЭЦП с выполнением криптографических преобразований на базе эллиптических кривых.
- ***ГОСТ Р 34.11—94***. «Информационная технология. Криптографическая защита информации. Функция хэширования». Этот стандарт определяет процедуру хэширования подписываемых данных.

## 5.3. Конфиденциальность и контроль целостности передаваемой информации

Конфиденциальность и контроль целостности информации обеспечивается путем ее шифрования и имитозащиты. Алгоритмы шифрования и имитозащиты определяются стандартом ***ГОСТ 28147—89***. Этот стандарт определяет следующие режимы шифрования данных:

- простая замена, тип шифрования — ECB (Electronic Codebook);
- гаммирование, тип шифрования — CCB (CipherCountBlock);
- гаммирование с обратной связью, тип шифрования — CFB (Cipher Feedback).

Алгоритмы, определяемые ГОСТ 28147—89, являются симметричными и используют ключи длиной 256 бит.

Алгоритм шифрования стандарта ГОСТ 28147—89 используется в условиях открытого распределения ключей с выработкой симметричного ключа парной связи, которая осуществляется с помощью алгоритма Диффи-Хеллмана на базе операций алгоритма ГОСТ Р 34.10—94 или ГОСТ Р 34.10—2001. Ключ Диффи-Хеллмана $K(x, y)$ вырабатывается на каждой стороне информационного обмена из собственного закрытого ключа и открытого ключа противоположной стороны, при использовании алгоритма:

- ГОСТ Р 34.10—94 ключ $K(x, y)$ вычисляется по формуле $K(x, y) = a^{xy} (\mathrm{mod} p)$;
- ГОСТ Р 34.10—2001 ключ $K(x, y)$ вычисляется по формуле $K(x, y) = ((\alpha \times x)(\mathrm{mod} q)) \circ (y \circ P)$, где $P$ — стартовая точка на цикле эллиптической кривой (две координаты по 256 бит), $\alpha$ — синхровектор (64 бит).

Ключ парной связи образуется хэшированием ключа $K(x, y)$ по алгоритму ГОСТ Р 34.11—94. Этот ключ имеет длину 256 бит и используется в установленной парной связи для шифрования случайных симметричных ключей сообщений по алгоритму ГОСТ 28147—89.

## 5.4. Аутентификация связывающихся сторон

Аутентификация связывающихся сторон осуществляется с помощью сертификатов открытых ключей сторон. Используются сертификаты стандарта X.509, выпущенные удостоверяющим центром или другим доверенным издателем. Для удостоверения сертификата используется ЭЦП его издателя в соответствии с алгоритмом ГОСТ Р 34.10—94 или ГОСТ Р 34.10—2001.

## 5.5. Установление аутентичного защищенного соединения

Для установления аутентичного защищенного соединения используется реализация протокола TLS (Transport Lauer Security, TLS 1.0, RFC 2246) на базе российских криптографических алгоритмов. Этим обеспечивается сетевая аутентификация клиент-сервер, конфиденциальность и целостность данных при работе клиент-серверных приложений, в частности, в Интернете.

Для установления аутентичного соединения между клиентом и сервером, ключи которых соответствуют алгоритмам ГОСТ Р 34.10—94 или ГОСТ Р 34.10—2001, в реализации протокола TLS используются российские криптографические алгоритмы ГОСТ 28147—89, хэширования ГОСТ Р 34.11—94 и обмена ключей по алгоритму Диффи-Хеллмана на базе алгоритма ГОСТ Р 34.10—94 или ГОСТ Р 34.10—2001.

## 5.6. Инфраструктура открытых ключей (PKI)

Для использования криптографии с открытым ключом необходимо гарантировать, что каждый закрытый и открытый ключ управляются корректным образом.

***Инфраструктура открытых ключей (Public Key Infrastructure — PKI)*** предназначена для управления открытыми ключами и сертификатами с целью поддержки услуг аутентификации, шифрования, целостности и неотказуемости.

Открытый ключ, связанный с определенным пользователем, должен быть удостоверен *сертификатом открытого ключа*. Более того, подлинность сертификата открытого ключа должна проверяться доверенным учреждением — *центром сертификации* (CA — certification autority).

***Сертификат открытого ключа*** — структура данных, состоящая их раздела данных и раздела подписи.

*Раздел данных* содержит открытые данные, которые включают, как минимум, открытый ключ и строку, идентифицирующую сторону (представляемый объект) для связывания с ним при условии, что открытый ключ подписывающего известен заранее.

*Раздел подписи* состоит из цифровой подписи органа сертификации под разделом данных. Тождественность представляемого объекта, таким образом, связывается с заданным открытым ключом. На практике наиболее часто используются сертификаты формата X.509 v3.

***Орган сертификации (CA — certification autority)*** — доверенная третья сторона, чья подпись под сертификатом подтверждает подлинность открытого ключа, связанного с представляемым объектом.

Простейшую модель PKI можно построить из одного компонента, который называется издателем (issuer). Он выполнял бы все необходимые функции. Пользователи использовали бы криптографию с открытым ключом в своих приложениях, получая и обрабатывая сертификаты и список аннулированных сертификатов (CRL). В рамках одного компонента трудно обеспечить необходимый уровень защищенности при выполнении всех необходимых задач, связанных с созданием и распространением сертификатов и CRL. Поэтому обычно PKI строится из различных компонентов, каждый из которых предназначен для специализированного выполнения нескольких задач.

## 5.7. Аутентификация с помощью открытого ключа на основе сертификата

Механизмы аутентификации на основе сертификатов обычно используют режим запрос—ответ. Пользователь, или точнее, программное обеспечение компьютера для генерирования ответа вырабатывает с помощью закрытого ключа пользователя цифровую подпись для случайного запроса от сервера аутентификации. Пользователь возвращает эту подпись серверу вместе с сертификатом открытого ключа. Сервер аутентификации проверяет подлинность сертификата открытого ключа, и, если она подтверждается, он проверяет подлинность цифровой подписи, используя открытый ключ пользователя из сертификата, таким образом, удостоверяя подлинность пользователя.

*Механизмы аутентификации на основе сертификатов обычно используют режим запрос—ответ.*

Общий процесс, с помощью которого аутентификационный сервер использует сертификат открытого ключа для получения подлинного открытого ключа пользователя, состоит из следующих этапов:

1. Получение подлинного открытого ключа CA (одноразовый процесс).
2. Получение идентификатора пользователя.
3. Получение по незащищенному каналу от этого пользователя его сертификата открытого ключа (согласующегося с его идентификатором).
4. Проверка текущей даты и времени относительно срока действия, указанного в сертификате (при проверке используются локальные доверенные часы);
5. Проверка текущей действительности открытого ключа CA.
6. Проверка подписи под сертификатом пользователя с помощью открытого ключа CA;
7. Проверка того, что сертификат не был отозван.
8. Если все проверки успешны, то сервер аутентификации принимает открытый ключ в сертификате как подлинный открытый ключ данного пользователя.

## 5.8. Организация хранения закрытого ключа

Несмотря на то, что криптография с открытым ключом может обеспечивать надежную аутентификацию пользователя, сам по себе закрытый ключ никак с ним не связан. Поэтому необходимо хранить закрытый ключ, обеспечивая его защиту от компрометации. Существует несколько способов хранения закрытого ключа.

### 5.8.1. Профиль пользователя/реестр

Самый простой вариант — хранить закрытые ключи внутри локального хранилища операционной системы, которое связано с учетной записью пользователя и защищено с помощью криптографических методов. Однако такое решение ассоциирует пользователя с его закрытым ключом только после авторизации в операционной системе и, следовательно, ключ нельзя использовать для начальной аутентификации.

*Простой вариант — ключи хранятся внутри локального хранилища операционной системы. Закрытый ключ связан с конкретным компьютером.*

Кроме того, закрытый ключ, хранящийся на жестком диске владельца компьютера, уязвим по отношению к прямым и сетевым атакам. Достаточно подготовленный злоумышленник может похитить закрытый ключ пользователя и с помощью этого ключа представляться этим пользователем.

Закрытый ключ в данном случае связан с конкретным компьютером.

### 5.8.2. Незащищенные носители

Для переноса закрытого ключа можно использовать любые сменные носители информации (дискеты, карты памяти, USB-флеш и пр.). В этом случае на носителе создается ключевой контейнер, содержащий зашифрованное значение закрытого ключа с помощью запоминаемого пароля.

Однако такая защита недостаточно эффективна — пароли уязвимы по отношению ко многим атакам, а зашифрованный контейнер беспрепятственно можно скопировать на любой другой носитель, обеспечивая создание дубликата для злоумышленников.

*Незащищенные носители — на сменном носителе информации создается ключевой контейнер, содержащий зашифрованное значение закрытого ключа.*

### 5.8.3. Touch Memory, Memory-карты

В качестве носителя информации можно использовать специализированные устройства аутентификации — touch memory (электронные ключи в виде так называемых «таблеток») и Memory-карты, выполненные в виде пластиковых карт примерно такого же размера, как и кредитные карты, но с встроенной микросхемой. И тот и другой тип устройств представляют собой сменный носитель информации с уникальным номером, прошиваемым при изготовлении, и памятью, в которой можно хранить данные пользователя.

Некоторые типы подобных устройств аутентификации предусматривают возможность двухфакторной аутентификации, требуя ввод PIN-кода для доступа к содержимому пользовательской памяти.

### 5.8.4. Смарт-карты и USB-ключи

Смарт-карты (как и Memory-карты) представляют собой пластиковые карты с встроенной микросхемой. Однако смарт-карты представляют собой более сложное устройство аутентификации, содержащее микропроцессор и операционную систему, контролирующую устройство и доступ к объектам в его памяти. Кроме того, смарт-карты, как правило, обладают возможностью проводить криптографические вычисления.

Смарт-карты находят все более широкое применение в различных областях, от систем накопительных скидок до кредитных и дебетовых карт, студенческих билетов и телефонов стандарта GSM.

Несмотря на название — устройства для чтения смарт-карт, как и большинство оконечных устройств или устройств сопряжения (IFD) способны как считывать, так и записывать информацию, если позволяют возможности смарт-карты и права доступа. Устройства для чтения смарт-карт могут подключаться к компьютеру с помощью:

- последовательного порта;
- слота PCMCIA;
- порта USB.

Устройства чтения смарт-карт могут быть интегрированы в клавиатуру.

Некоторые производители выпускают другие виды аппаратных устройств, в которых смарт-карты объединены с устройством чтения смарт-карты. По свойствам памяти и вычислительным возможностям они полностью аналогичны смарт-картам.

Наиболее популярны аппаратные «ключи», использующие порт USB. USB-ключи привлекательны для некоторых организаций, поскольку USB становится стандартом, находящим все большее распространение в новых компьютерах: организации не нужно приобретать для пользователей какие бы то ни было считыватели.

Смарт-карты и USB-ключи по своим возможностям являются интеллектуальными устройствами.

## 5.9. Интеллектуальные устройства и аутентификация с помощью открытого ключа

Смарт-карты и USB-ключи могут повысить надежность служб инфраструктуры открытых ключей PKI (Public Key Infrastructure): смарт-карта может использоваться для безопасного хранения закрытых ключей пользователя, а также для безопасного выполнения криптографических преобразований. Безусловно, интеллектуальные устройства аутентификации не обеспечивают абсолютную защиту, но их защита намного превосходит возможности обычного компьютера.

Для хранения и использования закрытого ключа разработчики используют различные подходы. Наиболее простой из них — использование интеллектуального устройства в качестве дискеты: при необходимости карта экспортирует закрытый ключ, и криптографические операции осуществляются на рабочей станции. Этот подход является не самым совершенным с точки зрения безопасности, зато относительно легко реализуемым и предъявляющим невысокие требования к интеллектуальному устройству.

Два других подхода более безопасны, поскольку предполагают выполнение интеллектуальным устройством криптографических операций. При первом пользователь генерирует ключи на рабочей станции и сохраняет их в памяти устройства. При втором пользователь генерирует ключи при помощи самого устройства и хранит их в его памяти. В обоих случаях, после того как закрытый ключ сохранен, его нельзя извлечь из устройства и получить любым другим способом.

*Пользователь генерирует ключи на рабочей станции и сохраняет их в памяти устройства.*

### 5.9.1. Генерация ключей вне устройства (рис. 5.4)

В этом случае пользователь может сделать резервную копию закрытого ключа. Если устройство выйдет из строя, будет потеряно, повреждено или уничтожено, пользователь сможет сохранить тот же закрытый ключ на новой карте. Это необходимо, если пользователю требуется расшифровать какие-либо данные, сообщения, и т. д., зашифрованные с помощью соответствующего открытого ключа. Однако закрытый ключ пользователя в этом случае может быть похищен.

### 5.9.2. Генерация ключей с помощью устройства (рис. 5.5)

В этом случае закрытый ключ не выходит из устройства, а также нет риска, что злоумышленник украдет его резервную копию. Способ использования закрытого ключа — обладание интеллектуальным устройством. Будучи безопасным, это решение выдвигает высокие требования к возможностям интеллектуального устройства: оно должно генерировать ключи и осуществлять криптографические преобразования. Также предполагается, что закрытый ключ не может быть восстановлен в случае выхода устройства из строя.

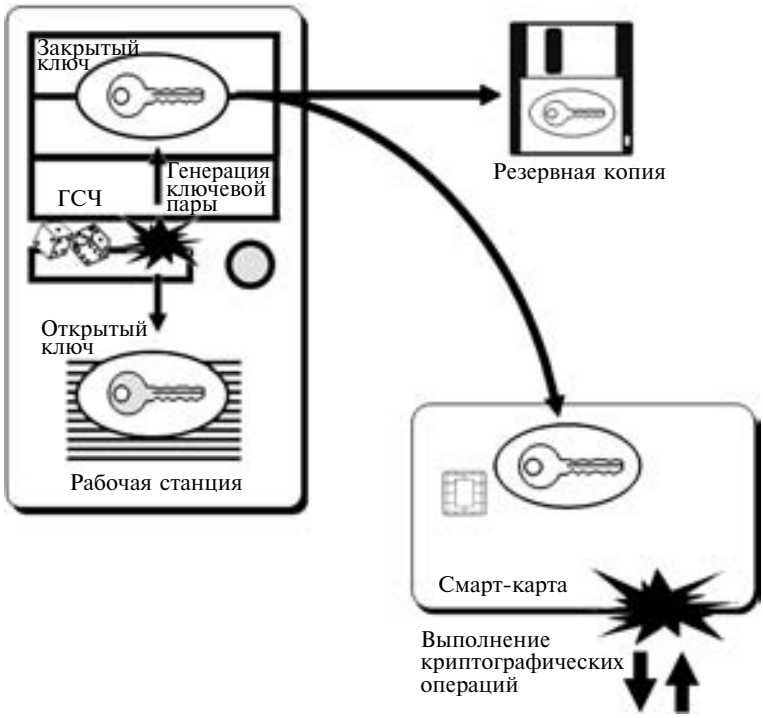



**Рис. 5.4. Генерация ключей вне устройства**

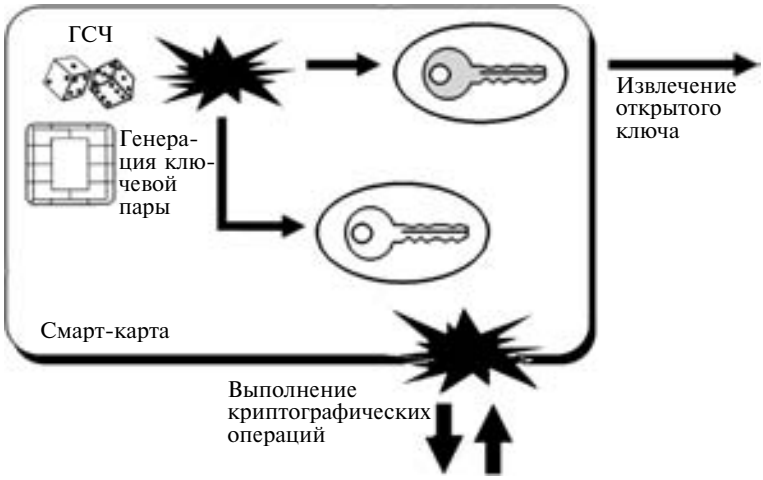



**Рис. 5.5. Генерация ключей с помощью устройства**
**Пользователь генерирует ключи с помощью самого устройства и хранит их в его памяти**

## 5.10. Недостатки аутентификации с помощью открытых ключей. Возможные атаки

Ниже приведены известные атаки на системы, использующие аутентификацию с помощью открытых ключей, и защита от них (табл. 5.1).

*Таблица 5.1*

**Методы атак на системы с открытым ключом и борьба с ними**

| *Описание атаки* | *Защита от данной атаки* |
|---|---|
| ***Факторизация ключа*** | |
| Злоумышленник может разложить ключ на множители и вывести значение личного ключа пользователя. | *Существенное увеличение размеров ключей* — на данный момент рекомендуется использовать ключи длиной не менее 512 бит. Для центров сертификации рекомендуется использовать ключи длиной 2048 бит. |
| ***Атака по известным сообщениям*** | |
| Злоумышленник подготавливает специальное сообщение и принуждает владельца подписать его личным ключом. Данный результат после преобразования может быть использован злоумышленником для формирования нового сообщения и выдачи его за сообщение от владельца личного ключа. | *Хэшированная цифровая подпись* — владелец формирует подпись для сообщения, подписывая своим личным ключом хэшированное значение сообщения. |
| ***Подделка открытого ключа*** | |
| Злоумышленник может подставить свой собственный открытый ключ, если получатель принимает неаутентифицированные ключи. | *Сертификаты открытого ключа* — публикуется назначение ключа владельца, и под этой публикацией ставится заслуживающая доверия подпись. |
| ***Атака «Человек посередине»*** | |
| Злоумышленник подставляет свой собственный открытый ключ вместо другого ключа и заново шифрует сообщения между объектами. | *Сертификаты открытого ключа* — как для «Подделки открытого ключа» (см. выше). |
| ***Фиктивное имя в сертификате*** | |
| Злоумышленник ставит имя жертвы в заявке на получение сертификата открытого ключа. | *Требование владения ключом* — Сертификаты выдаются только лицу, которое владеет запрошенным именем. |
| ***Подмена сертификата*** | |
| Злоумышленник использует законный сертификат для поддельного сервера с другим IP-адресом. | *Проверка достоверности имени хоста в сертификате* — имя в сертификате сравнивается с именем хост-машины, от который получен данный сертификат. |
| ***Фиктивный центр выдачи сертификатов*** | |
| Злоумышленник использует фиктивный орган выдачи сертификатов для создания поддельных сертификатов и затем заставляет браузеры принять его открытый ключ данного центра. | Эффективной защиты на данный момент нет. |

*Окончание таблицы 5.1*

| *Описание атаки* | *Защита от данной атаки* |
|---|---|
| ***Использование личного ключа*** | |
| Злоумышленник использует украденный личный ключ | *Список отозванных сертификатов* — периодически выпускается список сертификатов, которые были отозваны.<br>*Интерактивный отзыв сертификатов* — обеспечивается механизм выполнения запроса органу по выдаче сертификатов для подтверждения того, что сертификат не был отозван.<br>*Периодическая сертификация* — выдвигается требование, чтобы все сертификаты были «свежими» и обеспечивался механизм, который позволял бы динамически затребовать и получить от сертифицирующих органов «свежий» сертификат. |
| ***Взлом парольной фразы личного ключа*** | |
| Злоумышленник использует программу взлома для получения доступа к личному ключу законного пользователя, хранимого на локальном компьютере. | *Личный ключ на смарт-карте* — личный ключ хранится на смарт-карте, а не в зашифрованном файле. |
| ***Активная разведка личного ключа*** | |
| Злоумышленник внедряет в систему жертвы программу, которая перехватывает личный пароль жертвы в момент его использования. | *Реализация функции шифрования по личному ключу на смарт-карте* — личный ключ хранится на смарт-карте, которая реализует функцию шифрования, т. е. личный ключ никогда ее не покидает. |
| ***Кража резервной копии личного ключа*** | |
| Злоумышленник крадет резервную копию личного ключа, хранимого на дискете или в другом устройстве. | *Создание личного ключа на смарт-карте* — личный ключ генерируется на смарт-карте и никогда ее не покидает. |

## Примечания

- При аутентификации с помощью открытых ключей целесообразно использовать интеллектуальные устройства аутентификации. При этом весьма важным является исключительное владение пользователя закрытым ключом, т.е. обеспечение его защиты от компрометации. Это невозможно, если закрытый ключ хранится или криптографические преобразования осуществляются на компьютере пользователя.
- При аутентификации с помощью открытых ключей пользователь может допустить халатность 1-го типа: интеллектуальные устройства могут быть оставлены на рабочей станции. Если пользователь, отходя от своего рабочего места, оставляет смарт-карту в устройстве чтения или USB-ключ в порту, кто-то другой в офисе может легко представиться данным пользователем. Защита с помощью PIN-кода эффективна, если сеансы пользователей блокируются после определенного промежутка бездействия, а для разблокирования необходима повторная аутентификация. Но PIN-коды можно узнавать, например, «подглядывая из-за плеча». Организа-

ции должны призывать пользователей всегда носить интеллектуальные устройства с собой. Это обеспечивается автоматически, если устройства используются для контроля физического доступа в помещения.

- При аутентификации с помощью открытых ключей пользователь может допустить халатность 2-го типа: интеллектуальные устройства могут быть утеряны. Организация, в которой используются интеллектуальные устройства аутентификации, будет вынуждена каждому пользователю, потерявшему свое устройство, выдавать новое временное устройство или временно осуществлять аутентификацию с помощью альтернативного метода. При этом организация должна следить за тем, чтобы подобные мероприятия не ослабляли безопасность.
- Интеллектуальные устройства могут быть уязвимы по отношению к логическим и физическим атакам, а также атакам «троянских коней». Каждая организация, использующая интеллектуальные устройства с целью обеспечения безопасности, должна быть уверена в том, что производители устройств и разработчики программного обеспечения позаботились о принятии соответствующих логических и архитектурных контрмер. Правда, эти меры могут отрицательно повлиять на удобство использования интеллектуального устройства. Логические атаки осуществляются, когда интеллектуальное устройство работает в обычных физических условиях, а важная информация в виде байтов поступает на вход или снимается с выхода интеллектуального устройства. Физические атаки возможны, когда изменяются физические параметры, такие как: температура, частота, напряжение — с целью получения доступа к важной информации в памяти интеллектуального устройства. Атаки «троянских коней» предполагают размещение несанкционированного приложения на рабочей станции пользователя. Троянская программа ждет, пока пользователь введет действительный PIN-код в приложении, которому он доверяет, что сделает возможным использование закрытого ключа, а после этого предложит интеллектуальному устройству выработать цифровую подпись несанкционированных данных.

## Контрольные вопросы

1. Из каких элементов состоит ключевая пара и для чего предназначен каждый элемент?
2. Что такое электронная цифровая подпись? Приведите примеры использования.
3. В каких случаях можно использовать криптографию с открытым ключом?
4. Приведите пример использования криптографии с открытым ключом для шифрования сообщения.
5. Приведите пример аутентификации пользователя с помощью открытых ключей.
6. Для чего предназначена инфраструктура открытых ключей (PKI)?
7. Назовите способы хранения закрытого ключа.
8. Назовите недостатки аутентификации с помощью открытых ключей.
9. Приведите примеры атак на системы, использующие аутентификацию с помощью открытых ключей, и способы защиты от подобных атак.

# Глава 6
# ПРОТОКОЛЫ АУТЕНТИФИКАЦИИ В ЛОКАЛЬНОЙ СЕТИ

Существует несколько протоколов, описывающих процесс аутентификации субъектов в локальной сети. Например, в рамках операционных систем семейства Windows компании Microsoft использовались протоколы LAN Manager (LANMAN), NT LAN Manager (NTLM), NT LAN Manager версии 2 (NTLM v2) и Kerberos. В течение 1990-х гг. разработчики компании Microsoft придерживались в отношении аутентификации эволюционного подхода, так что новые продукты могли работать совместно с существующим установленным программным обеспечением:

- аутентификация в настольных системах в исполнении компании Microsoft начиналась с типовых методик блокирования экрана и введения менеджера локальной сети — LANMAN, использовавшего для аутентификации сетевых служб метод «запрос—ответ»;
- ОС Windows NT 4.0 привнесла в контроллеры доменов функцию непрямой аутентификации и протокол NTLM;
- в ОС Windows 2000 появилась возможность использовать протокол Kerberos.

## 6.1. Протоколы LAN Manager и NT LAN Manager

### *6.1.1. Архитектура, компоненты, участники, описание протоколов*

LAN- и NT-технология паролей компании Microsoft включала в себя две важные особенности. Во-первых, все базы данных паролей содержали значения хэш-функции паролей. Во-вторых, чтобы помешать активной разведке паролей, аутентификация основывалась на использовании технологии «запрос—ответ». Эти особенности, да еще после перехода от LANMAN к NTLM, создавали достаточно сложную паролевую среду.

Работающие под управлением Windows системы хранят значения хэш-функции паролей в системном файле и по возможности обеспечивают защиту файла с паролями от кражи. В современных Windows-системах имеется специальная область хранения, называемая *реестром*. Находящаяся в реестре база данных администратора учетных записей пользователей (Security Account Manager, SAM), содержит записи всех авторизованных пользователей, а также хранит значения хэш-функции от их паролей. В ОС Windows NT на доступ к записям в файле реестра накладываются ограничения на доступ пользователей, а доступ к записям в базе SAM ограничен особенно жестко. Это не предотвращает все попытки извлечь файл паролей Windows, однако увеличивает сложность подобных атак.

*Работающие под управлением Windows системы хранят значения хэш-функции паролей в системном файле и по возможности обеспечивают защиту файла с паролями от кражи.*

Оказалось, что аутентификация в ОС Windows уязвима к атакам двух типов: к атакам на базу данных SAM, выполняемым с помощью автономно работающих программ-взлом-

щиков, и к атакам программ-взломщиков, работающих с перехваченными парами «запрос—ответ».

Классические варианты атак на базу данных SAM появились как результат работ в рамках проекта Samba и привели к разработке бесплатно распространяемого пакета, позволявшего коллективно пользоваться файлами между UNIX-серверами и NT-клиентами. Для синхронизации паролей UNIX-серверу необходимо получение копий значений хэш-функции паролей NT-пользователей. Чтобы извлекать значения хэш-функции с целью выполнения над ними атаки угадывания методом проб и ошибок, взломщиками также использовалось инструментальное средство типа *pwdump* или его разновидности.

### 6.1.2. Хэширование в LANMAN

Рассмотрим, как функция хэширования в LANMAN преобразует принадлежащий пользователю длинный пароль (рис. 6.1).

Сначала функция изменяет пароль до вида 14-символьной цепочки, при необходимости добавляя или удаляя символы. Затем преобразует все символы в символы верхнего регистра. Это хороший ход с точки зрения удобства пользования, так как, несмотря на ошибки, сделанные пользователем при работе с клавишей <Shift>, система все равно распознает его пароль. Однако это снижает энтропию пароля.

После этого функция разделяет результат на два семибайтовых фрагмента и использует каждый из них в качестве 56-разрядного ключа для шифрования с помощью алгоритма DES. Каждый такой ключ используется для отдельного шифрования 64-разрядной константы. В ОС UNIX алгоритм DES используется так же, только LANMAN опускает добавку.

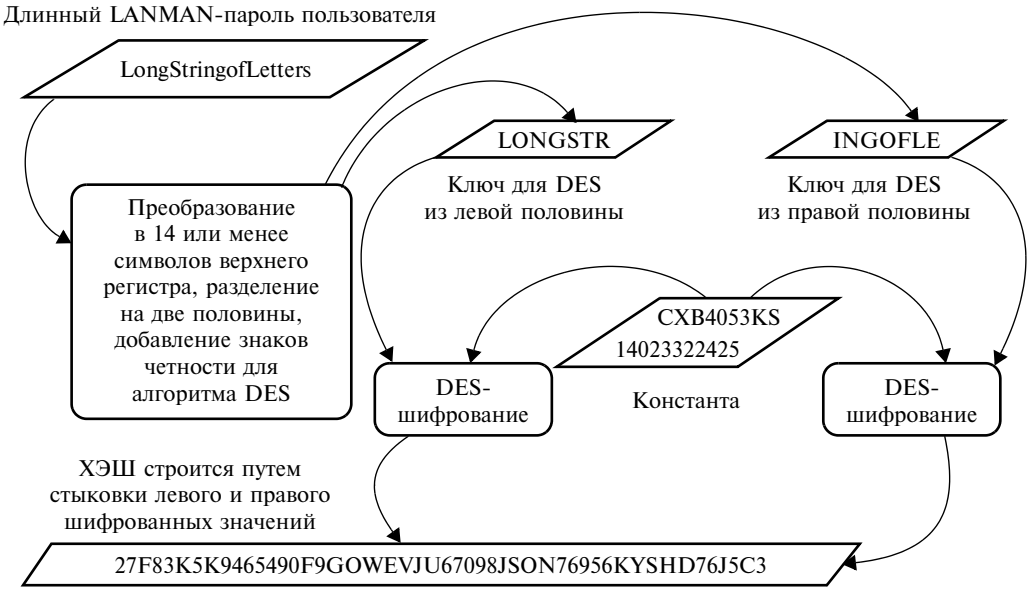



**Рис. 6.1. Хэширование в LANMAN**

Результаты шифрования соединяются в одну цепочку, образуя окончательный хэшированный результат.

При создании или изменении пароля пользователя система вычисляет значение хэш-функции и сохраняет его значение в базе данных SAM. Проверяя пароль, используемый для локального входа, система хэширует вводимый Пользователем пароль и сравнивает полученный результат с хэшированным значением в базе данных SAM.

### 6.1.3. Пароли в Windows в виде открытого текста

LANMAN не всегда использует пароли типа «запрос—ответ», так как другие сетевые продукты требуют применения паролей в виде открытого текста. Это особенно важно при совместной работе с другими продуктами. Например, некоторые версии пакета Samba для размещения Windows-совместимых сетевых служб на UNIX-системах использовали прямые текстовые пароли. Продукты семейства Windows, включая NT, имеют необходимые специальные программные модули, которые позволяют до некоторой степени автоматизировать процесс регистрации в службах, требующих пароли в виде открытого текста. Администраторы могут устанавливать специальный конфигурационный флаг в реестре ОС Windows, позволяющий использовать прямые текстовые пароли. В ОС Windows NT использование подобных паролей затруднено, так что администраторы предупреждаются о потенциальном риске перехвата паролей.

Поддержка прямых текстовых паролей служит причиной потенциальной проблемы, называемой «атакой под старину». В процессе этой атаки взломщик заставляет клиента поверить, что он должен предоставить серверу данные аутентификации в виде открытого текста. Клиент отвечает, автоматически предоставляя прямой текстовый пароль вместо его хэширования и использования.

*Поддержка прямых текстовых паролей служит причиной потенциальной проблемы, называемой «атакой под старину».*

### 6.1.4. Метод «запрос—ответ» в LANMAN, NTLM

В отличие от реализации метода «запрос—ответ» на основе устройств аутентификации, протокол регистрации в ОС Windows автоматически перехватывает запрос и генерирует ответ на основе пароля владельца. Когда пользователь запрашивает у сервера разрешение на вход, тот в ответ посылает случайное 8-байтовое число. Пользователь в ответ вводит «имя пользователя» и пароль, если этого еще не было сделано. Получив эти данные, рабочая станция вычисляет значение хэш-функции от пароля (поэтому его копия не хранится в виде открытого текста).

*Протокол регистрации в ОС Windows автоматически перехватывает запрос и генерирует ответ на основе пароля владельца.*

В общем случае рабочая станция хранит копию хэша пароля, и пользователю не придется вводить его снова при следующем входе в систему. Наконец, рабочая станция вы-

числяет ответ, трижды используя алгоритм DES для полученного случайного числа. Программное обеспечение ОС Windows вычисляет ответ на запрос аутентификации следующим образом (это реализовано как для LANMAN, так и для NTLM):

1. На первом этапе процедура использует 128-разрядное хэшированное значение пароля пользователя и получает из него три 56-разрядных фрагмента.

2. Затем она трижды выполняет шифрование специального одноразового числа, используя каждый из фрагментов в качестве ключа шифрования алгоритма DES.

3. На конечном этапе процедура объединяет результаты трех шифрований в 24-байтовый ответ.

### 6.1.5. Протокол NTLM

ОС Microsoft Windows NT 4.0 поддерживает три разных типа аутентификации — локальную, доменную и удаленную.

- Термин «локальная аутентификация» означает то, что физическое лицо регистрируется в устройстве непосредственно, не устанавливая удаленного соединения.
- Доменная аутентификация соответствует модели прямой аутентификации и отражает ситуацию, когда человек использует свой компьютер для входа в другой компьютер по сети. Примером является аутентификация, реализуемая менеджером локальной сети LANMAN.
- Удаленная аутентификация соответствует модели непрямой аутентификации, когда клиент регистрируется на сервере, который для верификации ответа пользователя обращается к другому серверу (*контроллеру домена* системы NT).

Под аутентификацией в NTLM обычно понимаются два последних типа аутентификации, которые представляют собой сетевую аутентификацию в системе NT.

Разработчики ОС Windows NT существенно улучшили механизм аутентификации по сравнению с тем, что был реализован в LANMAN. Ведь система NT поддерживала расширенный набор символов, что могло значительно увеличить количество возможных паролей. Появилось и несколько новых алгоритмов шифрования, которые уменьшали вычислительные накладные расходы и при этом сохраняли или даже увеличивали уровень защиты.

В результате в Windows NT используется новая процедура хэширования паролей. В NT сохраняется 14-символьное ограничение на длину пароля, но можно пользоваться любыми символами из набора символов Unicode. Пароли считываются и хранятся в виде последовательности из четырнадцати 16-битовых Unicode-символов. Для получения 128-разрядного хэшированного значения пароля в NT используется разработанный Роном Ривестом (Ron Rivest) коммерческий алгоритм хэширования Message Digest #4 (MD4) (более новый алгоритм MD5 широко используется в Internet-протоколах). Такой усовершенствованный хэш обычно называют *NTLM-хэшем*.

Хотя реализованный в NTLM механизм аутентификации использует для кодирования паролей улучшенную функцию хэширования, в нем по-прежнему применяется протокол «запрос—ответ». Однако для поддержки совместимости с LANMAN NTLM-аутентификация усложнена. NTLM-аутентификация требует вычислений как NTLM-хэша, так и LANMAN-хэша. Каждая парольная запись пользователя в базе данных SAM содержит два хэшированных значения пароля: вычисленные с помощью NTLM-хэширования и с помощью LANMAN-процедуры. При аутентификации по методу «запрос—ответ» NT-клиент вычисляет два ответа: с помощью NTLM-хэша и LANMAN-хэша. Такой подход позволяет NT-системам обеспечить преемственность со старым сетевым про-

граммным обеспечением. Но подобная совместимость часто сводит на нет повышение уровня защищенности, получаемое благодаря перепроектированной процедуре хэширования в NTLM.

### 6.1.6. Возможные атаки на LANMAN и NTLM

Ниже приведены известные атаки на системы, использующие аутентификацию с помощью протоколов LANMAN и NTLM, и защита от них (табл. 6.1).

*Таблица 6.1*

**Возможные атаки на LANMAN и NTLM и защита от них**

| *Описание атаки* | *Защита от данной атаки* |
|---|---|
| ***Маскировка под другого человека*** | |
| Осуществляется перехват нескольких одноразовых паролей пользователя, пользующегося протоколом X9.9. | *Использование более длинных ключей шифрования*<br>Существующие технические меры заменяются механизмами, в которых используются более длинные ключи шифрования, в результате чего увеличивается устойчивость к атакам методом проб и ошибок |
| ***Восстановление значения пароля пользователя*** | |
| Взлом паролей по частям. Пароль взламывается по частям, так что атака линейна для каждой части, но не топологическая. | *Взаимозависимое вычисление хэшированного значения*<br>Вид каждой части хэшированного значения пароля зависит от значения всех частей пароля. Возможность взлома части пароля отсутствует. |
| ***Восстановление значения пароля пользователя*** | |
| Использование хэшированного значения LANMAN-пароля для взлома NT-хэша.<br>Копирование хэшированных значений паролей из файлов восстановления ОС Windows NT. | *Шифрование базы данных*<br>Шифруется вся база данных паролей, в результате чего взломщики уже не могут атаковать хэшированные значения. |
| ***Принуждение к использованию пароля в виде открытого текста*** | |
| Взломщик заставляет сервер запросить у пользователя пароль в виде открытого текста, который может быть перехвачен в сети методами активной разведки. | *Блокирование работы более слабого механизма аутентификации*<br>Система конфигурируется таким образом, что использование слабых механизмов, введенных для обеспечения обратной совместимости, запрещается. |
| ***Подстановка хэшированного значения, прошедшего процедуру регистрации в системе*** | |
| Внедрение украденного значения хэша в базу данных SAM и такая модификация ОС Windows NT, в результате которой пользователь выглядит успешно прошедшим процедуру регистрации в системе. | *Исключение хранения базового секрета в ОС*<br>ОС Windows роль базового секрета играет хэш. Такая атака может быть сорвана, только если исключить хранение базового секрета внутри уязвимой операционной системы Windows. В рамках протоколов LANMAN и NTLM решения нет. Один из возможных подходов используется в протоколе Kerberos. |

## 6.2. Протокол Kerberos

Протокол Kerberos был специально разработан для того, чтобы обеспечить надежную аутентификацию пользователей.

Он может использовать централизованное хранение аутентификационных данных и является основой для построения механизмов Single Sign-On (возможность одноразовой аутентификации в нескольких приложениях). Протокол Kerberos предлагает механизм взаимной аутентификации клиента и сервера перед установлением связи между ними с учетом того, что начальный обмен информацией между клиентом и сервером может происходить в незащищенной среде, а передаваемые пакеты — перехвачены и модифицированы.

*Протокол Kerberos может использовать централизованное хранение аутентификационных данных и является основой для построения механизмов Single Sign-On.*

Протокол основан на понятии ticket (билет, удостоверение, мандат). Ticket является шифрованным пакетом данных, который выдан выделенным доверенным центром аутентификации. В терминах протокола Kerberos — Key Distribution Center (KDC, центр распределения ключей).

Когда пользователь выполняет первичную аутентификацию, после успешного подтверждения его подлинности KDC выдает первичное удостоверение пользователя для доступа к сетевым ресурсам — Ticket Granting Ticket (TGT). В дальнейшем, при обращении к отдельным сетевым ресурсам пользователь, предъявляя TGT, получает от KDC удостоверение для доступа к конкретному сетевому ресурсу — Service Ticket.

*Протокол основан на понятии ticket, который является зашифрованным пакетом данных, выданным центром аутентификации.*

Одним из преимуществ протокола Kerberos, обеспечивающим очень высокий уровень сетевой безопасности, является то, что во всех сетевых взаимодействиях не передаются ни пароли, ни значения хэша паролей в открытом виде. Все удостоверения являются шифрованными пакетами данных.

Примером реализации протокола Kerberos служит доменная аутентификация пользователей в операционных системах компании Microsoft, начиная с Windows 2000.

### 6.2.1. Архитектура, компоненты, участники протокола

В 1983 г. в МТИ (Массачусетском Технологическом институте) был начат проект «Афина», целью которого было создание модели предполагаемой среды распределенных вычислений следующего поколения для академических организаций. Группа участников проекта «Афина» решила задачу безопасности, связанную с KDC, на основе протокола Нидхэма—Шредера, но с учетом результатов работы Деннинга и Сакко. Чтобы обеспечить работу протокола Kerberos в крупномасштабных средах, проект «Афина» должен был предложить программное обеспечение для обработки процедуры регистрации на клиентских рабочих станциях и адаптировать серверы для работы с протоколом Kerberos. Программное обеспечение, адаптированное под работу с протоколом Kerberos, обычно называют *кербережированным*.

К 1989 г. Стив Миллер (Steve Miller) и Клиффорд Ньюмэнн (Clifford Neumann) с помощью других сотрудников МТИ сделали четыре версии протокола Kerberos. Версия 4 была первой выпущенной в общее пользование версией протокола, которая используется до сих пор. Однако стандартной версией для Internet-сообщества стала версия 5.

### *Сервер аутентификации*

Центр распределения ключей протокола Kerberos состоит из нескольких серверов, которые выполняют различные функции. Сервер аутентификации реализует протокол, сходный с протоколом Нидхэма—Шредера. Теоретически этот сервер может выдавать мандаты для обмена данными с любой керберезированной службой. На практике большинство рабочих станций использует сервер аутентификации только в целях выдачи мандатов для связи со службой выдачи разрешений на получение мандатов.

Чтобы получить мандат от сервера аутентификации, пользователь должен сконструировать сообщение KRB_AS_REQ. Сервер аутентификации отвечает сообщением KRB_AS_REP, которое содержит мандат и соответствующую статусную информацию. Запрашивая мандат, он предоставляет данные о своей личности, имя сервера и случайное одноразовое число. Протокол устанавливает временной период, в течение которого будет действовать коллективно используемый ключ, называемый в протоколе Kerberos *ключом сеанса*. Протокол также вводит в шифруемую по ключу часть идентификатор рабочей станции, с помощью которого он может контролировать, каким рабочим станциям разрешено использовать конкретный мандат. Это снижает вероятность несанкционированного использования мандата. Сервер аутентификации отвечает сообщением KRB_AS_REP, которое содержит мандат и соответствующую статусную информацию.

*Чтобы получить мандат от сервера аутентификации, пользователь должен сконструировать сообщение KRB_AS_REQ. Сервер аутентификации отвечает сообщением KRB_AS_REP, которое содержит мандат и соответствующую статусную информацию.*

Пользователь удостоверяется в правильности мандата, проверяя статусную информацию, предоставляемую сервером в сообщении KRB_AS_REP. В частности, ему необходимо проверить, что ответ содержит правильные имя сервера, случайное число и период действия. После этого он может спокойно пользоваться мандатом и ключом сеанса для связи с сервером.

### *Аутентификация для сервера*

Пользователь использует мандат протокола Kerberos и соответствующий ключ сеанса для создания сообщения KRB_AP_REQ, чтобы аутентифицировать себя, например, почтовому серверу. Кроме мандата, пользователь предоставляет еще и так называемый аутентификатор протокола Kerberos, который представляет собой шифрованный элемент данных, содержащий имя пользователя (Пользователь) и временною метку. Для шифрования аутентификатора пользователь применяет тот же ключ сеанса, который был в мандате (рис. 6.2).

Сервер дешифрует полученный мандат, извлекает из него ключ сеанса и использует его для дешифровки аутентификатора. Имя пользователя в аутентификаторе должно

совпадать с именем пользователя в мандате, а временная метка должна быть не очень старой — обычно в пределах последних пяти минут.

Если запрос проходит эти тесты, сервер посылает ответное сообщение KRB_AP_REP (если пользователь о нем просит). В этом ответе посылается временная метка из запроса, шифрованная с использованием ключа сеанса. Как и в случае протокола Нидхэма—Шредера, ответ имеет смысл, только если алгоритм шифрования не допускает выполнения атак с помощью переписывания или редактирования методом вырезки и копирования.

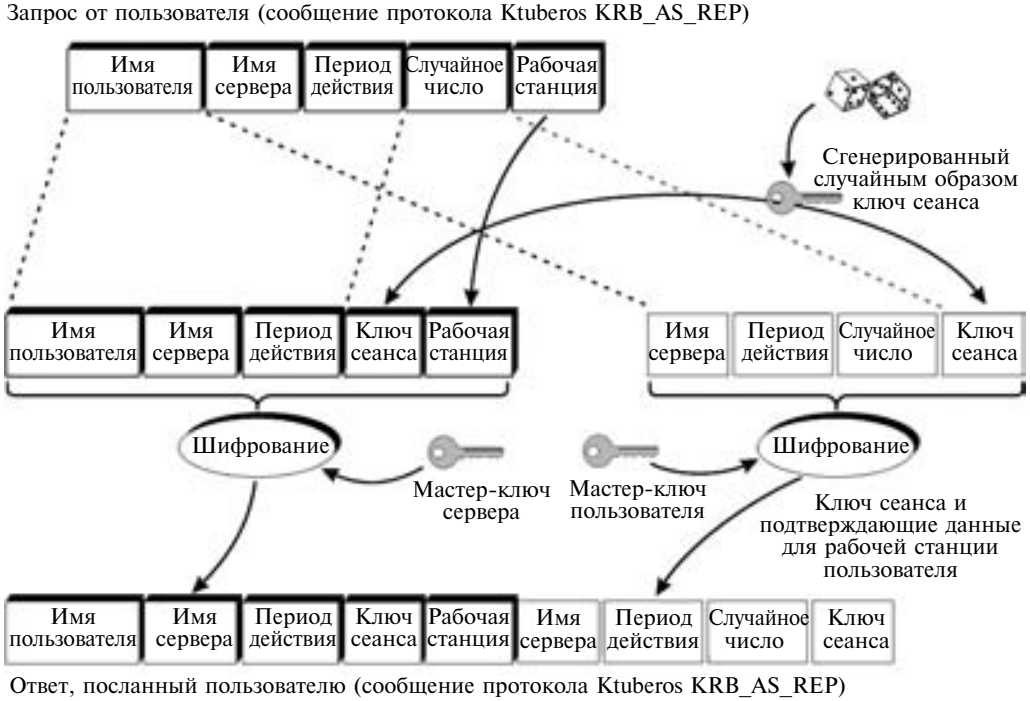



**Рис. 6.2. Аутентификация для сервера (табл. 6.2)**

*Таблица 6.2*

**Состав и наименование полей**

| *Поле* | *Назначение* |
|---|---|
| Имя пользователя | Клиент, который запрашивает мандат |
| Имя сервера | Нужная служба. Этот мандат шифруется с использованием мастер-ключа сервера |
| Период действия | Время, когда мандат и соответствующий ему ключ сеанса начинают действовать, и время, когда ключ и мандат теряют свою дееспособность |
| Ключ сеанса | Секретный ключ, коллективно используемый сервером и пользователем |
| Рабочая станция | Идентификатор компьютера (или компьютеров), на котором «может» работать пользователь |

### *Служба выдачи разрешений на получение мандата (рис. 6.3)*

Хотя сервер аутентификации протокола Kerberos можно использовать для генерации мандатов на работу с отдельной службой, однако проблема заключается в том, что протокол Kerberos требует использования мастер-ключа для обработки сообщений, которыми обменивается пользователь с сервером аутентификации, а большинство людей, работая на компьютере, обычно используют множество служб. Если мастер-ключ хранится на рабочей станции в то время, когда на ней кто-то работает, то существует риск его похищения.

Чтобы устранить эту проблему, мастер-ключом надо пользоваться как можно более короткий промежуток времени и убирать его из рабочей станции как можно быстрее. Но это приводит к другой проблеме: если мы стираем мастер-ключ после завершения подключения к одному серверу, то нам надо прочитать его снова при попытке подключиться к другому.

Традиционно в протоколе Kerberos в качестве мастер-ключей пользователей используются запоминаемые пароли, так что в этом случае придется мириться с бесконечными запросами на ввод пароля. Ни одна из альтернатив не является практичной.

*В протоколе Kerberos в качестве мастер-ключей пользователей используются запоминаемые пароли.*

Таким образом, возникает та же дилемма, которая привела к возникновению ключей сеанса: нужно вводить временный ключ, который можно использовать для выдачи других временных ключей. Действительно, проблема решается таким же образом: вместо того чтобы оставлять мастер-ключ в рабочей станции во время регистрации пользователя, вводится специальный ключ сеанса, который можно использовать для выпуска мандатов. Здесь используется хорошо известная в вычислительной технике практика: часто проблему можно решить, введя еще один уровень косвенности.

В протоколе Kerberos этот дополнительный временный ключ реализуется путем добавления к серверу KDC специального сервера, называемого *сервером выдачи разрешений на получение мандатов*. Этот сервер работает с мандатами, которые, конечно, называются

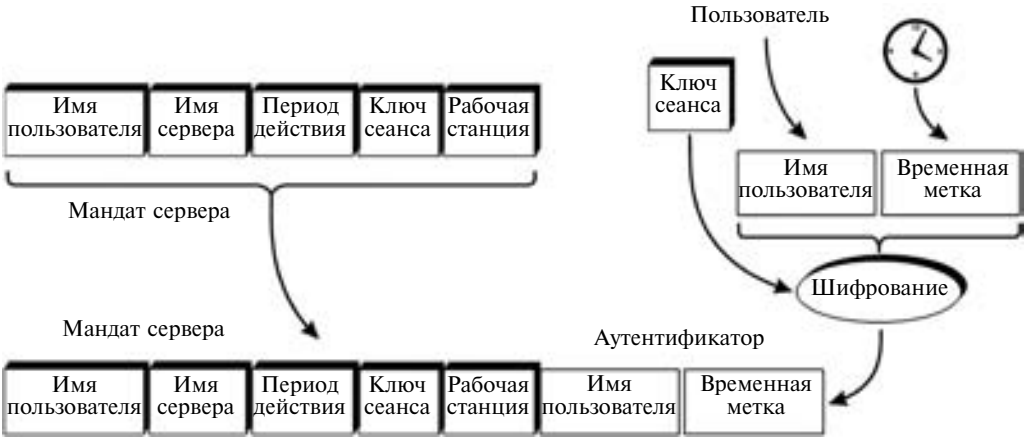



**Рис. 6.3. Запрос серверу от пользователя (Сообщение протокола Kerberos KRB_AP_REQ)**

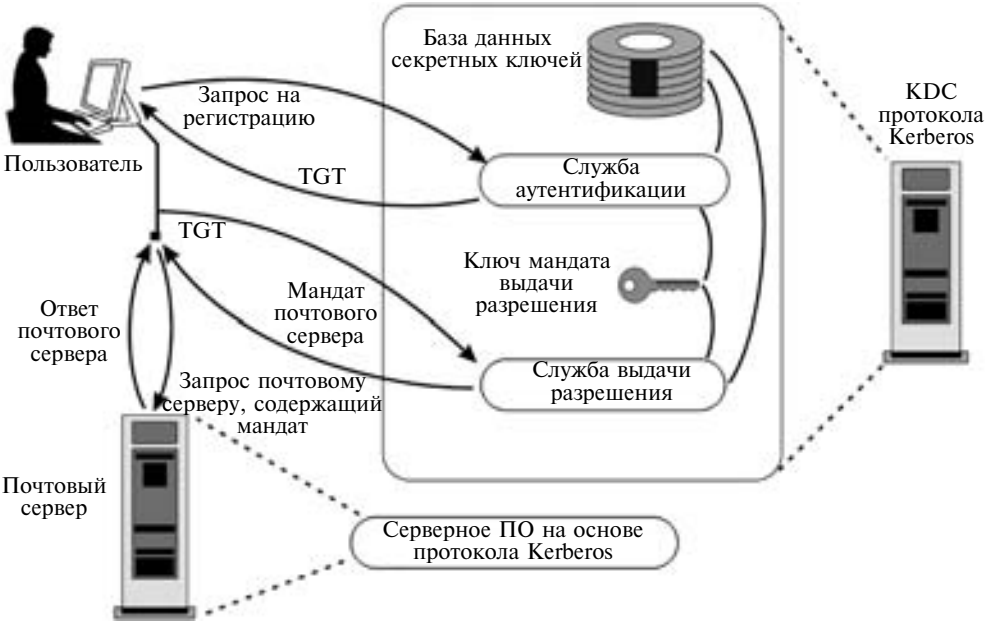



**Рис. 6.4. Взаимодействие рабочей станции KDC протокола Kerberos и почтового сервера**

*разрешениями на получение мандатов* (ticket-granting ticket, TGT). Пользователи могут посылать свои TGT этому специальному серверу, чтобы запрашивать мандаты для других служб.

В предыдущих примерах рассматривался случай установления Пользователем соединения с почтовым сервером через KDC (рис. 6.4). На практике Пользователь будет устанавливать связь не только с почтовым сервером. Скорее всего, он, как обычно, подключится к двум или более файл-серверам, одному или двум серверам печати, а также к другим службам. Несомненно, рабочая станция пользователя будет подключаться к некоторым из этих служб автоматически в момент его регистрации в системе. Но могут быть и другие службы, которые не входят в список регулярно используемых, или такие, к которым нет смысла подключаться без необходимости.

Благодаря серверу выдачи разрешений на получение мандатов обеспечивается относительно безопасный и дружественный к пользователю механизм однократной регистрации.

Когда пользователь регистрируется на своей рабочей станции, она немедленно связывается с сервером аутентификации KDC протокола Kerberos и получает TGT. После этого рабочая станция получает пароль пользователя (или другие данные аутентификации) и использует их для дешифровки ответа от сервера аутентификации.

Затем она пересылает TGT серверу выдачи разрешений на получение мандатов, чтобы получить мандаты на доступ к службам, немедленно необходимым пользователю: к обычно используемым им файл-серверам, почтовому серверу, серверу печати. Позднее рабочая станция может снова обратиться к серверу выдачи разрешений за дополнительными мандатами, если пользователю понадобятся дополнительные серверы. Рабочей станции больше не надо запрашивать у пользователя ввод пароля, так как для получения дополнительных мандатов она просто использует TGT и соответствующий ключ сеанса.

Сервер выдачи разрешений имеет собственный ключ, который используется протоколом Kerberos для шифрования его мандатов. Получив TGT, сервер дешифрует его и извлекает ключ сеанса, который используется при выдаче мандатов пользователю и называется *ключом для выдачи мандатов*. Сервер использует ключ для выдачи мандатов, чтобы дешифровать аутентификатор и проверить его достоверность. Сервер также сверяет период действия TGT с текущим временем. Кроме того, он проверяет период действия, запрошенный рабочей станцией пользователя для нового мандата. На следующем этапе сервер генерирует случайным образом ключ сеанса, который будет использоваться рабочей станцией пользователя и почтовым сервером.

*Сервер выдачи разрешений имеет собственный ключ, который используется протоколом Kerberos для шифрования его мандатов.*

Сервер выдачи разрешений берет данные, подлежащие возврату пользователю, и шифрует их для доставки. Сначала сервер создает мандат почтового сервера и шифрует его с помощью мастер-ключа почтового сервера. Затем группирует остальные данные, необходимые ему для ответа, и шифрует их с использованием ключа для выдачи разрешений пользователю. Оба блока шифрованных данных объединяются, формируя ответное сообщение, которое сервер посылает пользователю.

### *Аутентификация пользователей и рабочих станций*

Протоколы KDC, подобные протоколу Kerberos, в действительности уделяют основное внимание аутентификации пользователей сетевым службам. Классические протоколы слабо или вообще никак не обеспечивают установление личности пользователя для рабочей станции или даже KDC. Ничто не мешает кому угодно, включая взломщика, обратиться с запросом на получение мандатов для аутентификации конкретному серверу. Имея возможность устанавливать связь с KDC пользователя, любой может послать сообщение, заявляющее, что он — пользователь, и получить мандат для аутентификации почтовому серверу. Рассматриваемые далее протоколы гарантируют невозможность использования взломщиками полученных ими мандатов.

Данная ситуация имеет два интересных следствия. Во-первых, это означает, что протокол Kerberos не обязательно рассматривает сами рабочие станции как отдельные объекты, которые требуют аутентификации. Вместо этого подразумевается, что они являются взаимозаменяемыми устройствами, как это и имеет место в среде академических лабораторий. Во-вторых, у взломщиков есть возможность получить достаточное число мандатов, относящихся к конкретному человеку, чтобы провести атаку на его мастер-ключ. К счастью, эти следствия можно устранить, используя некоторые варианты реализации протокола Kerberos.

### *Аутентификация рабочих станций*

Особенность протоколов KDC, рассмотренных в предыдущих разделах, состоит в том, что для обеспечения безопасности действий рабочей станции они используют один ключ. В отличие от протокола ОС Windows NT, рабочая станция не имеет отдельного ключа для обмена данными с KDC или любой другой службой защиты. Поскольку в протоколе Kerberos реализована модель непрямой аутентификации, сама рабочая станция не может аутентифицировать отдельных пользователей. Она просто играет роль транс-

портного средства, с помощью которого авторизованные пользователи манипулируют ресурсами на серверах. Имея физический доступ, любой человек может воспользоваться рабочей станцией, так как протокол Kerberos не обеспечивает аутентификации, ориентированной на рабочую станцию.

*Поскольку в протоколе Kerberos реализована модель непрямой аутентификации, он не помогает самой рабочей станции аутентифицировать отдельных пользователей.*

Реализованный в протоколе Kerberos подход отражает идеологию распределенных вычислений с «тонким клиентом». Если клиентские рабочие станции просты и достаточно однородны в сети, то они могут быть взаимозаменяемыми. Не имеет значения, какая рабочая станция обслуживает пользователя; основной целью аутентификации является запрет использования не принадлежащих ему ресурсов.

Однако подобное видение не соответствует реальной практике, существующей в организациях, где преобладают персональные компьютеры на столах сотрудников. В каждом персональном компьютере имеются локальные файлы, принадлежащие конкретному владельцу или хранителю. Во многих случаях на нем установлено программное обеспечение, лицензированное на имя его владельца. Если протокол Kerberos будет аутентифицировать любого на каждой рабочей станции, то он не позволит контролировать доступ к таким персональным ресурсам. В подобных случаях рабочая станция должна иметь дополнительную процедуру аутентификации. Это может быть прямая аутентификация или основанная на протоколе Kerberos предаутентификация, которая описывается ниже в этом разделе. Компания Microsoft в ОС Windows 2000 использовала комбинацию этих методик.

В случае мандатов протокола Kerberos риск состоит в том, что взломщики могут получить достаточное число мандатов, предназначаемых какому-нибудь важному пользователю, чтобы провести успешную атаку на его мастер-ключ. В классической среде Kerberos это является серьезной угрозой, так как основой персональных мастер-ключей служат запоминаемые пароли.

### *Предаутентификация*

В протоколе Kerberos версии 5 была введена *предаутентификация,* поэтому серверы могли аутентифицировать запросы, посылаемые KDC, а не полагаться на аутентификацию запросов, выполняемую ими потом. Администраторы могут сконфигурировать Kerberos таким образом, что он будет требовать аутентификации при запросе мандатов или TGT. При этом KDC осуществляют рассылку шифрованных данных только тем, кто уже знает соответствующий ключ. Обычно процесс используется для аутентификации пользователя, получающего от KDC начальный TGT, хотя протокол способен поддерживать широкий набор альтернатив. Начальная предаутентификация должна быть доступна в виде опции в любой совместимой версии протокола Kerberos.

При традиционной Kerberos-аутентификации рабочая станция получает ключи от KDC до того, как ей нужен мастер-ключ пользователя. Так как в протоколе Kerberos в качестве мастер-ключа обычно используются пароли пользователей, то рабочая станция может отложить запрос пароля до момента получения отклика от KDC. В случае предаутентификации рабочая станция должна сначала получить мастер-ключ пользователя, что обычно означает получение пароля.

Рабочая станция посылает начальный запрос KDC, который обычно является запросом TGT. Для предаутентификации рабочая станция добавляет специально сформатированную шифрованную временную метку. Временная метка включает в себя текущее время суток и одностороннее хэшированное значение остальной части KDC-запроса, шифрованное с помощью мастер-ключа пользователя.

Получив запрос, KDC проверяет, требуется ли предаутентификация и если это так, отвергает запрос в случае отсутствия метки. Если предаутентификационная временная метка присутствует, то KDC ищет мастер-ключ пользователя и использует его для дешифрования временной метки. Если время суток во временной метке приемлемо, KDC вычисляет одностороннее хэшированное значение остальной части запроса и сравнивает его с хэшем из временной метки. Если они совпадают, KDC удовлетворяет запрос.

Заметим, что рабочая станция может рассматривать ответ KDC в качестве подтверждения личности пользователя. KDC пошлет законное ответное сообщение, а не сообщение об ошибке, только в том случае, если предаутентификация прошла успешно. Рабочая станция может дешифровать ответ KDC и проверить, содержит ли он правильное значение случайного числа, имени сервера и периода действия. В противном случае рабочая станция может сделать вывод, что пользователь пытается (безуспешно) выдать себя за кого-нибудь другого.

*Таблица 6.3*

**Возможные атаки на Kerberos и защита от них**

| *Описание атаки* | *Защита от данной атаки* |
|---|---|
| ***Повторное воспроизведение со старыми ключами*** | |
| Маскировка под другого человека<br>Взломщик посылает серверу ранее выпущенный мандат и воспроизводит посланные клиентом ранее сообщения, шифрованные с помощью этого ключа. | *Механизм «запрос—ответ» в работе протокола KDC с сервером*<br>Сервер посылает пользователю запрос, который требует ответа, зависящего от данных пользователя, шифрованных с помощью ключа сеанса. |
| ***Автономный взлом и воспроизведение*** | |
| Атакующая сторона в автономном режиме взламывает ключ сеанса и использует полученные сведения для повторного использования мандата, выпущенного с этим ключом. | *Временная метка в протоколе KDC*<br>Сообщения KDC включают информацию о времени суток, что позволят обнаруживать попытки повторного использования мандатов. |
| ***Автономный взлом мастер-ключа*** | |
| Атакующая сторона запрашивает мандаты от имени жертвы и использует их для взлома мастер-ключа жертвы грубой силой. | *Предаутентификация в KDC*<br>При запросе TGT пользователь должен предоставлять личную аутентификационную информацию. |
| ***Поддельное изменение времени*** | |
| Взломщик посылает серверу запрос с поддельным временем суток, так что недействительные мандаты становятся действующими. | *Аутентифицируемые сообщения о времени*<br>Сообщения, которые изменяют время на системных часах сервера, должны аутентифицироваться. |
| ***Восстановление или модификация закрытых данных*** | |
| Взломщик перехватывает запрос клиента KDC и возвращает другой набор ключей, которые ему неизвестны. | *Случайное разовое число, коллективно используемое с KDC*<br>Случайное число включается в запросы, посылаемые в KDC, и в ответы. |

### 6.2.2. Возможные атаки

Фундаментальной особенностью философии протокола Kerberos является четкое понимание, что находящиеся в сети компьютеры рано или поздно будут успешно атакованы (табл. 6.3). Это должно быть очевидным, так как в протоколе Kerberos в качестве базовых секретов служат пароли многократного использования. Конструкция Kerberos создавалась в попытке минимизировать общесистемные последствия вторжения на отдельные рабочие станции и серверы. Общая безопасность сервера, использующего протокол Kerberos, также основывается на предположении, что системные часы всех участвующих в обмене данными компьютеров хотя бы грубо, но синхронизированы. Протокол Kerberos будет выполнять свои защитные функции до тех пор, пока эти условия выполняются.

### 6.2.3. Реализация протокола Kerberos в ОС Windows 2000 и последующих ОС

Начиная с ОС Windows 2000, компания Microsoft заменила механизм доменной аутентификации ОС Windows NT, основанный на NTLM, на протокол Kerberos. Процедура доменной регистрации Windows превратилась в транзакцию, результатом которой является получение TGT. Отображение на файл-сервер теперь связано с обменом мандатами и ключами сеанса. Мастер-ключи и другая важная с точки зрения защиты информация хранятся в так называемой активной директории. Хотя реализация протокола Kerberos в ОС Windows имеет ряд отличительных элементов, компания Microsoft заявляет, что ее вариант будет удовлетворять всем требованиям совместимости со стандартным протоколом Kerberos. В частности, кербережированные Не-Windows-приложения будут способны обрабатывать мандаты Windows, обеспечивая возможность однократной регистрации между Windows- и Не-Windows-приложениями.

Благодаря протоколу Kerberos ОС Windows 2000 и последующие ОС по сравнению с более ранними продуктами семейства Windows имеют три существенных преимущества:

1. Обеспечивается более быстрая аутентификация на сервере, так как серверу не надо связываться с контроллером домена для непрямой аутентификации. Вместо этого сервер просто обрабатывает мандат.
2. Windows использует функции делегирования мандатов для передачи прав доступа пользователя к серверу посредством другого сервера.
3. Windows использует протоколы Kerberos, которые позволяют центрам распространения ключей, принадлежащим другим организациям, выдавать права доступа пользователям, работающим через другие центры распределения ключей контролируемым образом.

#### *Мастер-ключи и аутентификация рабочих станций*

Конечно, при вводе технологии Kerberos в продуктовую линию Windows неизбежны доработки. Компании Microsoft надо было придерживаться основной линии, чтобы сохранить совместимость с существующими продуктами и одновременно извлечь хотя бы некоторые преимущества из защитных функций протокола Kerberos. Это особенно заметно в том, как Microsoft адаптировала Kerberos к процессу регистрации в рабочей станции ОС Windows.

Как и предполагается, пользователь рабочей станции не видит ничего нового. Когда пользователь пытается зарегистрироваться, он нажимает на клавиатуре обычные клавиши и видит на экране паролевый диалог. Он вводит свое имя пользователя, выбирает домен и набирает на клавиатуре пароль. Но на системном уровне Windows преобразует все это в транзакции протокола Kerberos.

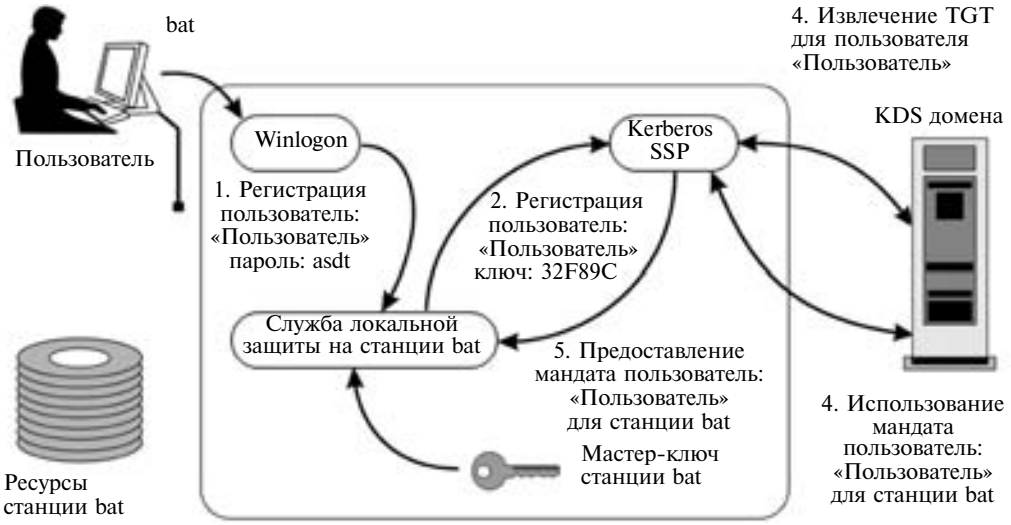



**Рис. 6.5. Процесс входа в систему**

Существенным отличием реализации протокола Kerberos в ОС Windows от традиционного варианта является то, что в ОС Windows рабочие станции рассматриваются как различные объекты. Каждая рабочая станция имеет в ареале свои особенности и собственный мастер-ключ, отличный от мастер-ключа регистрирующегося пользователя. Если, например, пользователь назвал свою рабочую станцию bat, то это будет ее имя внутри домена (рис. 6.5). Процесс регистрации в дополнение к другим мандатам, которые могут понадобиться, также организует получение мандата для аутентификации пользователя рабочей станции.

В ОС Windows 2000 и выше процесс входа в систему распределяется между тремя основными процессами: процессом Winlogon, который подсказывает пользователю о необходимости ввода его «имени пользователя» и пароля, процессом провайдера поддержки защиты Security Support Provider (SSP), который осуществляет связь с протоколом Kerberos, и процессом службы локальной защиты Local Security Authority (LSA), который защищает станцию.

Процессы протекают следующим образом:

- Получив аутентификационные данные пользователя, процесс Winlogon передает их процессу LSA.
- Процесс LSA путем хэширования преобразует пароль в мастер-ключ, который должен будет использоваться протоколом Kerberos.

*В ОС Windows 2000 процесс входа в систему распределяется между тремя основными процессами: процессом Winlogon, процессом провайдера поддержки защиты Security Support Provider (SSP), процессом службы локальной защиты Local Security Authority (LSA).*

- Затем LSA инициирует SSP-процесс протокола Kerberos и передает ему имя пользователя и мастер-ключ. В отличие от традиционного протокола Kerberos, в ОС Windows 2000 мастер-ключ оставляется в кэше. Это позволяет рабочей станции использовать механизм доменной аутентификации ОС Windows NT, если необходимо обмениваться данными с более старыми серверами. Очевидно, что держать такую информацию в кэше — это риск, причем совершенно ненужный, если во всей организации используется механизм аутентификации протокола Kerberos.
- Процесс SSP протокола Kerberos пытается связаться со своим контроллером домена и получить начальный TGT на имя пользователя. Запрос использует предаутентификацию, основанную на мастер-ключе пользователя.
- Если выбранный домен управляется более старым Windows NT-сервером и протокол Kerberos недоступен, то процесс LSA выполняет откат и использует NLTM-протокол.
- Если процесс SSP получает свой TGT протокола Kerberos, то он использует его для получения мандата для рабочей станции bat от имени пользователя. Получив мандат для рабочей станции, процесс SSP обеспечивает получение мандата для процесса LSA, который использует мастер-ключ станции bat для ее аутентификации. Если ключ аутентичен, то процесс LSA регистрирует пользователя в станции bat.

Избыточный шаг получения мандата для самой рабочей станции является необычной особенностью в сравнении с другими Kerberos-средами. Ключ сеанса в мандате не служит реальной цели. Этот подход в ОС Windows 2000 используется из-за того, что KDC хранит полномочия пользователя в каждом мандате, и это относительно ясный и непротиворечивый способ получения списка полномочий пользователя от KDC. Каждый мандат содержит авторизационную информацию и другие полномочия, необходимые для связывания пользователя с нужными ресурсами рабочей станции и правами доступа к ним. Процесс LSA на Windows-сервере работает таким же образом: он извлекает из мандата пользователя данные о полномочиях и использует их для запуска серверного процесса от имени и с разрешениями этого пользователя. На рабочей станции же процесс LSA обеспечивает работу приложений с именем этого пользователя.

### *Поддерживаемые службы и протоколы*

Службы и протоколы, которые в ОС Windows 2000 используют механизм аутентификации протокола Kerberos:

- файловые службы, включая службы доступа к файлам в Интернете и службы обмена с серверами (CIFS/SMB), а также службы системы управления распределенной файловой системой;
- службы вывода на печать;
- аутентификация Web-сервера информационному Интернет-серверу;
- службы аутентифицируемых вызовов удаленных процедур для удаленного управления серверами и рабочими станциями;
- запросы в активную директорию с использованием облегченного протокола доступа к директории (Lightweight Directory Access Protocol, LDAP);
- аутентификация для конфигурирования шифрованного канала связи хост-хост с использованием протокола IPSEC;
- аутентификация запросов об уровнях качества обслуживания.

# 6.3. Протокол Kerberos + PKINIT

## 6.3.1. Архитектура, компоненты, участники протокола

### Сертификаты в протоколе Kerberos

Существуют способы интеграции шифрования на основе открытого ключа в протокол Kerberos. Хотя эти методы не обязательно преобразовывают среду Kerberos в архитектуру с истинно автономной аутентификацией (процесс всегда будет зависеть от присутствия KDC протокола Kerberos), они исключают необходимость в коллективно используемом секрете, основанном на пароле многократного применения.

Рассмотрим методику, называемую PKINIT (public key initialization — инициализация открытого ключа). PKINIT (рис. 6.6) использует пару закрытого и открытого ключа пользователя в специальной версии процесса предаутентификации. Пользователь регистрируется в своей рабочей станции и предоставляет личный ключ. Рабочая станция связывается с KDC, посылая предаутентификационный запрос на получение TGT-мандата. В запросе содержится обычная информация и копия сертификата открытого ключа пользователя. Запрос подписывается цифровой подписью, получаемой с помощью его личного ключа.

*PKINIT использует пару закрытого и открытого ключа пользователя в специальной версии процесса предаутентификации.*

Получив запрос, KDC сначала пытается проверить достоверность сертификата пользователя. Он должен быть выпущен органом, известным KDC, иначе он будет отвергнут. Затем KDC генерирует TGT для пользователя и шифрует соответствующий ключ сеанса с использованием открытого ключа пользователя. После этого ответ подписывается с помощью собственного ключа KDC, так что пользователь имеет возможность после получения проверить его целостность. После того как пользователь дешифрует ключ сеанса, он может использовать его вместе с TGT для аутентификации себя другим серверам. Личный ключ ему не понадобится до следующей регистрации в системе.

Это не единственный способ использования методики PKINIT. Также ее можно использовать для генерации коллективно используемого секрета временными ключами алгоритма Диффи—Хеллмана. В этом случае предаутентификационный запрос пользователя содержит временный ключ Диффи—Хеллмана. По-прежнему пользователь должен под-


**Рис. 6.6. Методика PKINIT (Public Key Initialization)**

писать запрос отдельным ключом и предоставить копию сертификата ключа его подписи. Перед генерацией TGT с помощью коллективного секрета Диффи—Хеллмана KDC верифицирует сертификат и подпись пользователя на основе данных предаутентификации.

Метод PKINIT обладает важным свойством: он может исключить KDC из процесса аутентификации, полностью полагаясь на сертификаты пользователей. Ответственность за верификацию личности пользователя несет тот орган, который выпустил сертификат. Приняв сертификат, протокол KDC фактически аутентифицировал соответствующего пользователя. По этой причине KDC должен скрупулезно проверять сертификаты и выдавать мандаты только в том случае, если сертификат признается им без оговорок.

В ОС Windows метод PKINIT используется для интеграции открытых ключей в свою среду аутентификации на основе протокола Kerberos. Запрос TGT содержит копию сертификата пользователя и подписывается с помощью его личного ключа. Протокол KDC ОС Windows 2000 подтверждает достоверность сертификата и факт его выдачи органом, который ему известен. После этого KDC верифицирует временную метку процесса предаутентификации и цифровую подпись.

*В ОС Windows метод PKINIT используется для интеграции открытых ключей в свою среду аутентификации на основе протокола Kerberos.*

Проверив достоверность предаутентификационных данных, KDC строит TGT, включающий специфические для ОС Windows 2000 авторизационные данные — идентификаторы пользователя и группы, к которой он относится. После этого KDC шифрует ответ с помощью открытого ключа из сертификата пользователя и подписывает его собственным ключом. Получив ответ, система пользователя дешифрует и верифицирует его и затем, как того требует протокол Kerberos, использует TGT для запроса доступа к другим серверам.

### *Использование смарт-карт и USB-ключей*

Для получения доступа к сертификатам пользователя на этапе аутентификации в систему и для хранения закрытых ключей пользователей для связки Kerberos+PKINIT в ОС Windows 2000 используются смарт-карты и USB-ключи. Служба управления ресурсами смарт-карт была интегрирована в операционную систему, и для настройки аутентификации по сертификатам достаточно активизировать данную службу, установив драйверы считывателей смарт-карт. При нахождении в домене графический интерфейс ОС Windows 2000 (GINA) заменяется вариантом с поддержкой работы со смарт-картами.

На смарт-карту записывается сертификат и связанный с ним закрытый ключ, выписанные на доменном центре сертификации с использованием политик, предусматривающих возможность его использования для интерактивной аутентификации пользователя в системе.

При подключении смарт-карты к рабочей станции для аутентификации пользователя, хранящийся на ней сертификат используется для запроса TGT, а операция с закрытым ключом, возможная после ввода PIN-кода, используется для подписания этого запроса.

*На смарт-карту записывается сертификат и связанный с ним закрытый ключ.*

### 6.3.2. Возможности использования российских криптографических алгоритмов

Процесс аутентификации в протоколе Kerberos состоит из двух этапов: начальная аутентификация субъекта и последующая аутентификация сервисов. Для решения задачи начальной аутентификации широко используется технология PKINIT, поскольку она позволяет использовать сертификат открытого ключа вместо пароля.

*Для решения задачи начальной аутентификации используется технология PKINIT.*

Начальная аутентификация осуществляется путем отправки CMS (Cryptographic Message Syntax)-сообщения. При этом используются следующие криптографические алгоритмы:

- формирования/проверки подписи (алгоритм ГОСТ Р 34.10—94 или ГОСТ Р 34.10—2001);
- шифрования/дешифрования информации (алгоритм ГОСТ 28147—89);
- контроля целостности передаваемой информации (ключевой хэш на базе алгоритма хэширования ГОСТ Р 34.11-94);
- обмена ключей (с использованием алгоритма Диффи—Хеллмана на базе алгоритмов ГОСТ Р 34.10—94 и ГОСТ Р 34.10—2001).

Подробное описание специфики формирования CMS сообщения на базе российских криптографических алгоритмов приводится в утвержденном и опубликованном комитетом IETF стандарте RFC 4490 [CPCMS].

Связь документа с международными стандартами показаны на рис. 6.7.

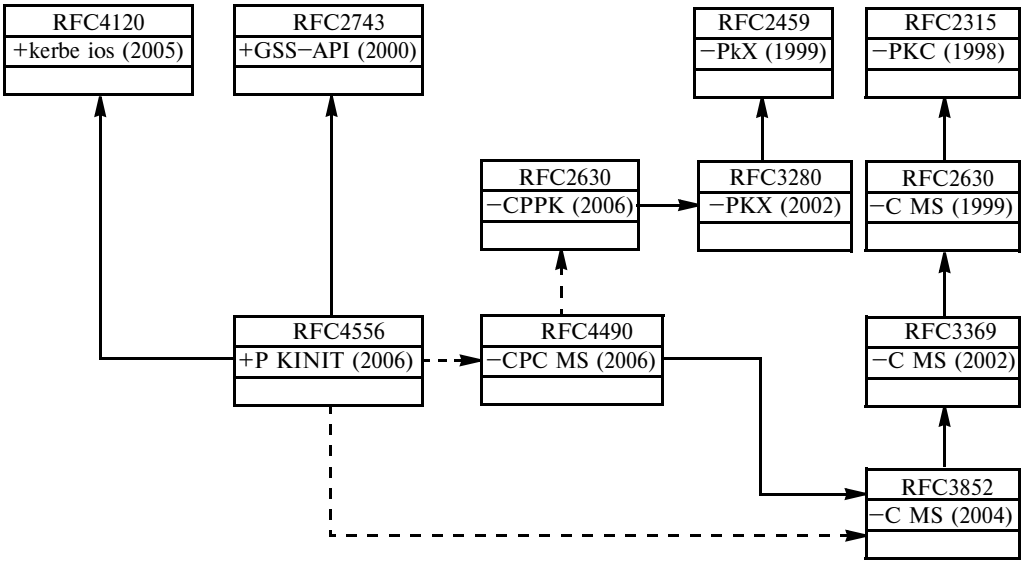



**Рис. 6.7. RFC 4490 и международные стандарты**

### 6.3.3. Возможные атаки (табл. 6.4)

*Таблица 6.4*

**Возможные атаки на Kerberos + PKINIT и методы защиты от них**

| *Описание атаки* | *Защита от данной атаки* |
|---|---|
| ***Подделка открытого ключа*** | |
| Если получатель принимает неаутентифицированный ключ, то взломщик просто подставляет свой собственный ключ вместо правильного ключа. | *Сертификаты открытого ключа*<br>Публикуется назначение открытого ключа владельцу, и под этой публикацией ставится заслуживающая доверия цифровая подпись. |
| ***Человек посередине*** | |
| Взломщик представляет собственный открытый ключ вместо другого ключа и заново шифрует сообщения между двумя объектами. | *Сертификаты открытого ключа*<br>Публикуется назначение открытого ключа владельцу, и под этой публикацией ставится заслуживающая доверия цифровая подпись. |
| ***Фиктивное имя в сертификате*** | |
| Взломщик ставит имя жертвы в заявке на получение сертификата открытого ключа. | *Требование владения ключом*<br>Сертификаты выдаются только лицу, которое владеет запрошенным именем. |
| ***Подмена сертификата*** | |
| Взломщик использует законный сертификат для реализации протокола SSL на поддельном сервере, который выдает себя за другой сервер. | *Проверка достоверности имени хоста в сертификате*<br>Имя в сертификате сравнивается с именем хост-машины, участвующей в SSL-соединении. |
| ***Использование личного ключа*** | |
| Взломщик полагается на автономную аутентификацию и использует украденный личный ключ. | *Список аннулированных сертификатов*<br>Периодически выпускается список всех сертификатов, которые были аннулированы.<br>*Интерактивный отзыв сертификатов*<br>Обеспечивается механизм выполнения запроса органа по выдаче сертификатов для подтверждения того, что сертификат не был аннулирован.<br>*Периодическая выдача сертификатов*<br>Выдвигается требование, чтобы все сертификаты были выданы недавно, и обеспечивается механизм, который позволял бы сертифицирующим органам выпускать такие сертификаты, если сертификат не был аннулирован. |

## Контрольные вопросы

1. Назовите основные особенности протоколов LAN Manager и NT LAN Manager.
2. Назовите типы аутентификации в NTLM.
3. Приведите примеры атак на системы, использующие аутентификацию с помощью протоколов LANMAN и NTLM, и защиты от них.
4. Перечислите преимущества протокола Kerberos.
5. Опишите функции сервера аутентификации, входящего в состав центра распределения ключей протокола Kerberos.
6. Приведите примеры атак на Kerberos и способы защиты от них.
7. Перечислите преимущества реализации протокола Kerberos в ОС Windows 2000 и последующих ОС в сравнении с более ранними продуктами семейства Windows.
8. Приведите пример способа интеграции шифрования в протокол Kerberos.
9. Возможные атаки на Kerberos + PKINIT и методы защиты от них?

# Глава 7
# МЕХАНИЗМЫ АУТЕНТИФИКАЦИИ ПРИ ОСУЩЕСТВЛЕНИИ ПОДКЛЮЧЕНИЙ

В настоящее время методы использования паролей по-прежнему применяются широко. Механизмы аутентификации по протоколу Point-to-Point Protocol (PPP) часто используются в среде модемного доступа и включают протоколы Password Authentication Protocol (PAP), Challenge Handshake Protocol (CHAP) и Extensible Authentication Protocol (EAP). Разработка протокола EAP продолжается, но уже сейчас он использует существующие и только появляющиеся технологии аутентификации в каналах PPP. Протоколы TACACS+ и Remote Access Dial-In User Service (RADIUS) поддерживают масштабируемые решения в области аутентификации.

Все протоколы рассмотрены ниже на примерах, в которых при аутентификации взаимодействуют два или более типовых устройств. Инициатором процесса, как правило, выступает устройство маршрутизатор отделения (назовем его Twiggi), которое обращается к серверу сетевого доступа (или NAS). Последний выполняет аутентификацию маршрутизатора Twiggi и принимает решение. Мы будем называть его аутентификатором, а маршрутизатор Twiggi — аутентифицируемым устройством.

## 7.1. Протокол PPP PAP

Аутентификация с помощью протокола PAP выполняется следующим образом (рис. 7.1).

1. Инициатор аутентификации, в нашем примере маршрутизатор отделения Twiggi, обращается к серверу сетевого доступа и устанавливает с ним связь.

2. После установления связи маршрутизатор передает пару «имя устройства—пароль» серверу NAS до тех пор, пока аутентификация не будет завершена или пока связь не прервется.

3. После успешной аутентификации маршрутизатор отделения Twiggi получает подтверждение.

Протокол PAP не является сильным аутентификационным методом. Он аутентифицирует только вызывающего оператора, а пароли пересылаются по каналу, который считается уже «защищенным». Таким образом, этот метод не дает защиты от использования чужих паролей и неоднократных попыток подбора пароля. Частота и количество неудачных попыток входа в сеть контролируются на уровне вызывающего оператора.

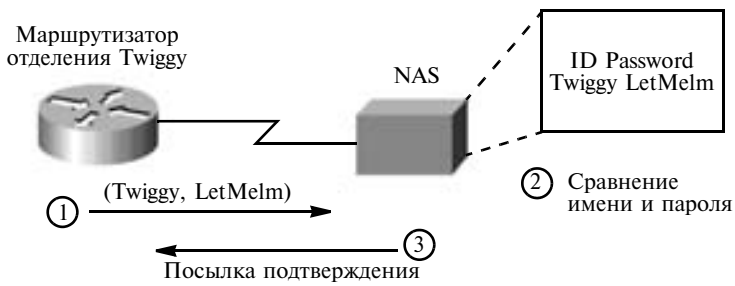



**Рис. 7.1. Аутентификация с использованием протокола PAP**

## 7.2. Протокол PPP CHAP

CHAP используется для периодической аутентификации центрального компьютера или конечного пользователя с помощью согласования по трем параметрам. Аутентификация происходит в момент установления связи, но может быть повторена и после ее установления.

Аутентификация с помощью протокола CHAP проходит следующим образом (рис. 7.2):

1. Маршрутизатор отделения Twiggi устанавливает связь с сервером сетевого доступа (NAS). CHAP обеспечивает безопасность сети, требуя от операторов обмена «текстовым секретом». Этот секретный ключ никогда не передается по каналу связи.

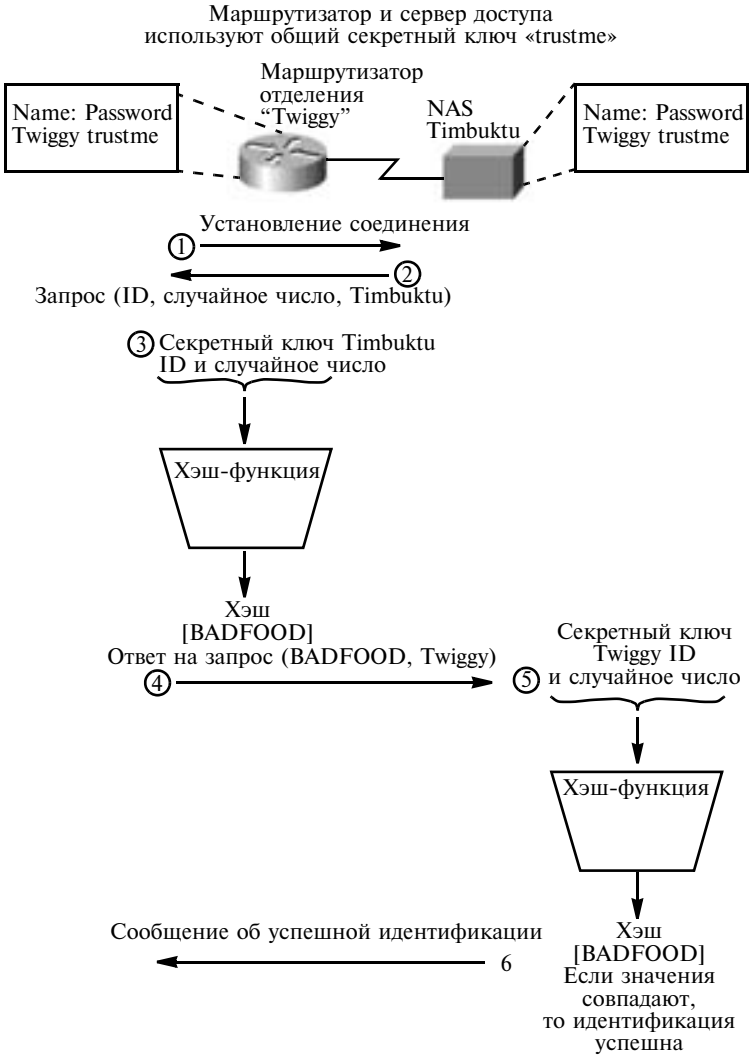



**Рис. 7.2. Аутентификация с помощью протокола CHAP**

2. После установления связи аутентификатор передает вызывающему устройству запрос, который состоит из имени устройства (или его ID), случайного числа и дополнительных атрибутов, например, имени центрального сервера (при аутентификации в локальной сети) или имени пользователя (при удаленной аутентификации).

3. Вызывающее устройство проводит вычисления с помощью хэш-функции. Имя устройства, случайное число и общий «текстовый секрет» один за другим подаются на вход хэш-функции. После этого вызывающее устройство отправляет серверу ответ, который состоит из значения хэш-функции и имени центрального сервера (при аутентификации в локальной сети) или имени пользователя (при удаленной аутентификации).

4. При получении ответа аутентификатор проверяет поставленное в ответе имя и выполняет те же вычисления.

5. Затем результат этих вычислений сравнивается с величиной, поставленной в ответе. Если эти величины совпадают, аутентификация считается успешной, система выдает соответствующее уведомление и устанавливает связь.

Секретные пароли на локальном и удаленном устройстве должны быть идентичны. Поскольку «текстовый секрет» никогда не передается по каналам связи, никто не может подслушать его с помощью каких-либо устройств и использовать для нелегального входа в систему. Пока сервер не получит адекватный ответ, удаленное устройство не сможет подключиться к местному устройству.

CHAP обеспечивает защиту от использования чужих паролей за счет пошаговых изменений аутентификатора и применения переменной величины запроса. Повторяющиеся запросы предназначены для ограничения времени, в течение которого система теоретически остается подверженной любой отдельной хакерской атаке. Частоту и количество неудачных попыток входа в систему контролирует аутентификатор.

*CHAP обеспечивает защиту от использования чужих паролей за счет пошаговых изменений аутентификатора и применения переменной величины запроса.*

### Примечание

Обычно в качестве односторонней хэш-функции CHAP используется MD5, а общий секрет хранится в текстовой форме. У компании Microsoft есть свой вариант протокола CHAP (MS-CHAP), где пароль (на вызывающей машине и на аутентификаторе) хранится в шифрованном виде. Это дает протоколу MS-CHAP некоторое преимущество. В отличие от стандартного протокола CHAP он может использовать доступные базы данных шифрованных паролей.

## 7.3. Протокол PPP EAP

Этот общий протокол аутентификации PPP поддерживает множество аутентификационных механизмов. EAP не выбирает конкретный аутентификационный механизм на этапе контроля соединения и откладывает этот выбор до аутентификации. Такой сценарий позволяет аутентификатору запросить больше информации до определения конкретного аутентификационного механизма. Кроме того, это дает возможность использовать

«внутренний» сервер, который реально запускает различные механизмы, тогда как аутентификатор PPP служит лишь для обмена аутентификационными данными.

Аутентификация с помощью протокола EAP проходит следующим образом (рис. 7.3):

1. Маршрутизатор отделения Twiggi начинает аутентификацию, устанавливая связь с сервером сетевого доступа NAS (далее аутентификатором).

2. После установления связи аутентификатор отправляет один или несколько запросов для аутентификации вызывающего его устройства (маршрутизатора Twiggi). В запросе имеется поле, где указано, что именно запрашивается. Так, например, здесь можно указать такие типы запросов, как аутентификация MD5, аутентификация с помощью сертификатов X.509, одноразовых паролей и т. д. При этом запрос типа MD5 сходен с протоколом аутентификации CHAP.

3. Как правило, аутентификатор отправляет первоначальный аутентификационный запрос, за которым следуют один или несколько дополнительных запросов о предоставлении аутентификационной информации. При этом первоначальный запрос не является обязательным и может опускаться в случаях, когда аутентификация обеспечивается иными способами (при связи по выделенным каналам, выделенным номерам и т. д.). В этих случаях вызывающая сторона отправляет пакет ответных данных на каждый запрос. Как и пакет запроса, пакет ответных данных содержит поле, соответствующее полю запроса.

4. Аутентификатор завершает процесс отправлением пакета, который свидетельствует об успешной или неуспешной аутентификации.

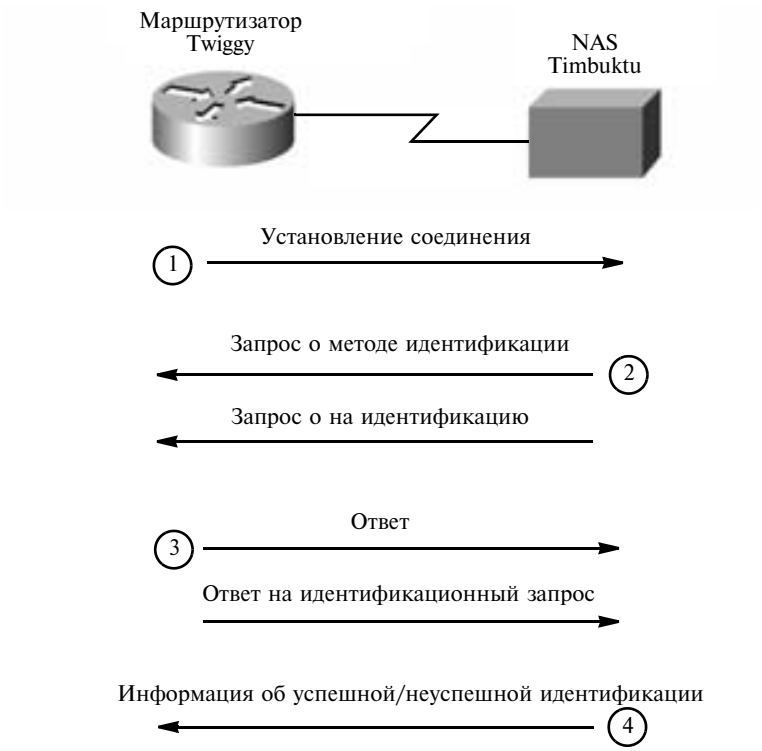



**Рис. 7.3. Аутентификация с помощью протокола EAP**

## 7.4. Протокол TACACS+

TACACS+ является протоколом последнего поколения из серии протоколов TACACS. TACACS — это простой протокол управления доступом, основанный на стандартах User Datagram Protocol (UDP), разработанных Bolt, Beranek, and Newman, Inc. (BBN) для Military Network (MILNET). Компания Cisco несколько раз совершенствовала и расширяла протокол TACACS, и в результате появилась ее собственная версия TACACS, известная как TACACS+.

*TACACS — это протокол управления доступом, основанный на стандартах User Datagram Protocol (UDP).*

TACACS+ пользуется транспортным протоколом TCP. Модуль-демон (процесс, запускаемый на машине UNIX или NT) сервера «слушает» порт 49, который является портом протокола IP, выделенным для протокола TACACS. Этот порт зарезервирован для выделенных номеров RFC в протоколах UDP и TCP. Все текущие версии TACACS и расширенные варианты этого протокола используют порт 49.

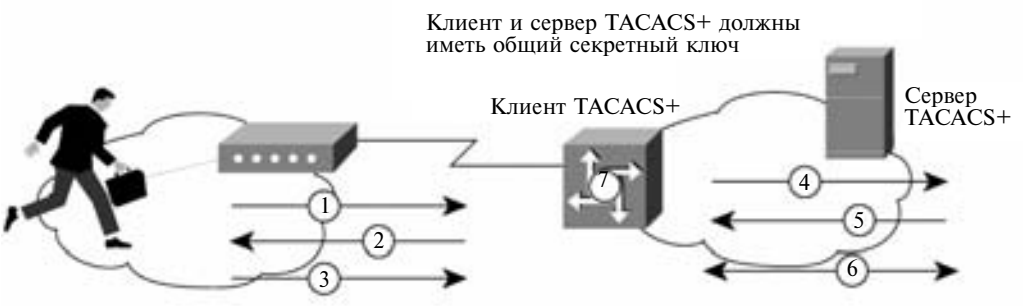
Протокол TACACS+ работает по технологии клиент/сервер, где клиентом TACACS+ обычно является NAS, а сервером TACACS+, как правило, считается «демон» (т. е. процесс, запускаемый на машине UNIX или NT). Фундаментальным структурным компонентом протокола TACACS+ является разделение аутентификации, авторизации и учета (AAA — Authentication, Authorization, Accounting). Это позволяет обмениваться аутентификационными сообщениями любой длины и содержания, и, следовательно, использовать для клиентов TACACS+ любой аутентификационный механизм, в том числе PPP PAP, PPP CHAP и Kerberos. Аутентификация не является обязательной. Она рассматривается как опция, которая конфигурируется на месте. В некоторых местах она вообще не требуется, в других местах она может применяться лишь для ограниченного набора услуг.

Обычно аутентификация предшествует авторизации, однако это не обязательно. В запросе на авторизацию можно указать, что аутентификация пользователя не проведена (личность пользователя не подтверждена). В этом случае лицо, отвечающее за авторизацию, должно самостоятельно решить, допускать такого пользователя к запрашиваемым услугам или нет. Протокол TACACS+ допускает только успешную или неуспешную авторизацию, однако этот результат допускает настройку на потребности конкретного заказчика. Авторизация может проводиться на разных этапах, например, когда пользователь впервые входит в сеть и хочет открыть графический интерфейс или когда пользователь запускает PPP и пытается использовать поверх PPP протокол IP с конкретным адресом IP. В этих случаях «демон» сервера TACACS+ может разрешить предоставление услуг, но наложить ограничения по времени или потребовать список доступа IP для канала PPP.

Учет представляет собой запись действий пользователя и обычно следует за аутентификацией и авторизацией. В системе TACACS+ учет может выполнять две задачи:

1) учитывать использованные услуги (например, выставление счетов);

2) обеспечивать безопасность. Для этого TACACS+ поддерживает три типа учетных записей. Записи «старт» указывают, что услуга должна быть запущена. Записи «стоп» говорят о том, что услуга только что окончилась. Записи «обновление» (update) являются промежуточными и указывают на то, что услуга все еще предоставляется. Учетные записи




1. Пользователь инициирует соединение PPP с сервером доступа

2. Сервер доступа запрашивает у пользователя имя и пароль

3. Пользователь отвечает на запрос

4. Клиент TACACS+ посылает зашифрованный пакет серверу TACACS+

5. Сервер TACACS+ сообщает результаты идентификации

6. Клиент и сервер обмениваются авторизационной информацией

7. Клиент TACACS+ обрабатывает параметры, полученные во время авторизации

**Рис. 7.4. Взаимодействие пользователя, клиента и сервера TACACS+**

TACACS+ содержат всю информацию, которая используется в ходе авторизации, а также другие данные, например, время начала и окончания (если это необходимо) и данные об использовании ресурсов.

Транзакции между клиентом TACACS+ и сервером TACACS+ идентифицируются с помощью общего «секрета», который никогда не передается по каналам связи. Обычно этот секретный ключ вручную устанавливается на сервере и на клиенте. TACACS+ можно настроить на шифрование всего трафика, который передается между клиентом TACACS+ и демоном сервера TACACS+.

Взаимодействие между пользователем, с одной стороны, и клиентом и сервером TACACS+, с другой, происходит так, как показано на рис. 7.4.

В ходе аутентификации TACACS+ используются пакеты трех типов: START, CONTINUE и REPLY. START и CONTINUE всегда отправляются клиентом, а REPLY — сервером.

Аутентификация начинается, когда клиент отправляет серверу сообщение START, которая описывает тип будущей аутентификации и может содержать имя пользователя и некоторые аутентификационные данные. Пакет START отправляется только в качестве первого сообщения аутентификационной сессии TACACS+ или сразу же после повторного запуска этой сессии. (Повторный запуск может проводиться по просьбе сервера, которая содержится в пакете REPLY). Пакет START всегда имеет порядковый номер, равный единице.

В ответ на пакет START сервер отправляет пакет REPLY. Сообщение REPLY указывает, завершилась ли аутентификация или ее следует продолжить. Если пакет REPLY требует продолжения аутентификации, он также указывает, какую дополнительную ин-

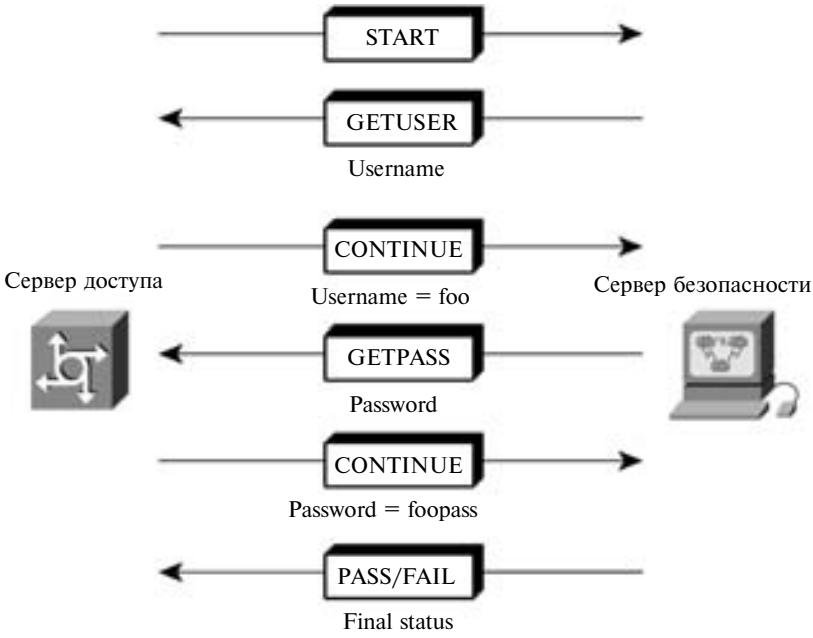



**Рис. 7.5. Аутентификация по протоколу TACACS+**

формацию ему нужно предоставить. Клиент собирает эту информацию и отправляет ее серверу в сообщении CONTINUE.

После аутентификации клиент может начать процесс авторизации (если она требуется). Сессия авторизации состоит из двух сообщений: сообщения REQUEST (запрос) и следующего за ним сообщения RESPONSE (ответ). Сообщение REQUEST содержит фиксированное число полей, которые описывают пользователя или процесс, и переменный набор аргументов, которые описывают услуги и опции, требующие авторизации.

Аутентификация по протоколу TACACS+ происходит так, как показано на рис. 7.5.

Авторизация TACACS+ показана на рис. 7.6.

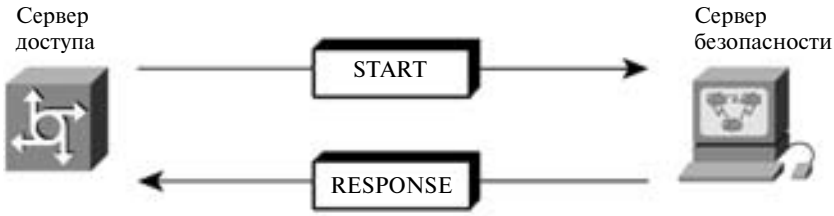



**Рис. 7.6. Авторизация по протоколу TACACS+**

## 7.5. Протокол RADIUS

Протокол RADIUS был разработан компанией Livingston Enterprises, Inc., в качестве протокола аутентификации серверного доступа и учета. В июне 1996 года, пятый проектный вариант протокола RADIUS был представлен на рассмотрение IETF. В настоящее время спецификация RADIUS (RFC 2058) и стандарт учета RADIUS (RFC 2059) предложены для утверждения в качестве общепринятых стандартов.

*Протокол RADIUS разработан в качестве протокола аутентификации серверного доступа и учета.*

Связь между NAS и сервером RADIUS основана на UDP. В целом считается, что протокол RADIUS не имеет отношения к подключению. Все вопросы, связанные с доступностью сервера, повторной передачей данных и отключениями по истечении времени ожидания, контролируются устройствами, работающими под управлением протокола RADIUS, но не самим протоколом передачи.

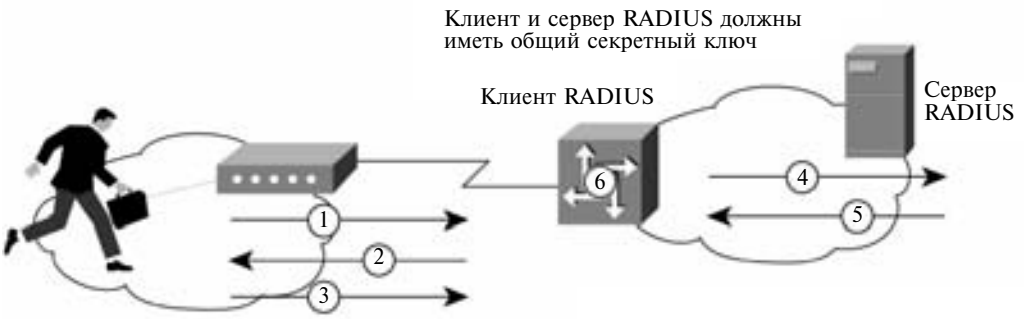
Протокол RADIUS основан на технологии клиент/сервер. Клиентом RADIUS обычно является NAS, а сервером RADIUS считается «демон», работающий на машине UNIX или NT. Клиент передает пользовательскую информацию на определенные серверы RADIUS, а затем действует в соответствии с полученными от сервера инструкциями. Серверы RADIUS принимают запросы пользователей на подключение, проводят аутентификацию пользователей, а затем отправляют всю конфигурационную информацию, которая необходима клиенту для обслуживания пользователя. Для других серверов RADIUS или аутентификационных серверов других типов сервер RADIUS может выступать в роли клиента-посредника (proxy).

Взаимодействие между пользователем, с одной стороны, и клиентом и сервером RADIUS, с другой, происходит так, как показано на рис. 7.7.

Сервер RADIUS может поддерживать разные методы аутентификации пользователя. Если пользователь предоставит ему свое имя и оригинальный пароль, этот сервер может поддержать PPP PAP или CHAP, UNIX login и другие механизмы аутентификации. Обычно регистрация пользователя состоит из запроса (Access Request), который поступает из NAS на сервер RADIUS, и соответствующего ответа (положительного или отрицательного), который выдает сервер. Пакет Access Request содержит имя пользователя, шифрованный пароль, IP-адрес системы NAS и номер порта. Формат запроса дает возможность пользователю запросить определенный тип сессии. Например, если запрос производится в алфавитно-цифровом режиме, из этого следует, что запрашивается услуга одного типа ("Service-Type = Exec-User"), но если запрос делается в пакетном режиме PPP, значит услуга должна быть другой ("Service Type = Framed User" или "Framed Type = PPP").

*Сервер RADIUS может поддерживать разные методы аутентификации пользователя: PPP PAP или CHAP, UNIX login и другие механизмы аутентификации.*

Когда сервер RADIUS получает от NAS запрос Access Request, он проводит поиск указанного имени пользователя в базе данных. Если в базе данных такого имени нет, то сервер загружает стандартный профиль, используемый по умолчанию, или отправляет




1. Пользователь инициирует соединение PPP с сервером доступа

2. Сервер доступа запрашивает у пользователя имя и пароль

3. Пользователь отвечает на запрос

4. Клиент RADIUS посылает имя пользователя и зашифрованный пароль серверу RADIUS

5. Сервер RADIUS отвечает сообщениями Accept, Reject или Challenge

6. Клиент RADIUS обрабатывает параметры, полученные от сервера вместе с сообщениями Accept или Reject

**Рис. 7.7. Взаимодействие между пользователем, клиентом и сервером RADIUS**

пользователю отрицательный ответ. Этот отрицательный ответ может при необходимости сопровождаться текстом, поясняющим причины отказа.

В системе RADIUS функции аутентификации и авторизации совмещены. Если имя пользователя найдено в базе данных и пароль указан верно, сервер RADIUS выдает положительный ответ, в котором приводится список пар атрибутов для данной сессии. Типичными параметрами являются тип услуги (shell или framed), тип протокола, адрес IP, присваиваемый пользователю (статический или динамический), список объектов доступа или статический маршрут, который необходимо добавить в таблицу маршрутизации NAS. Конфигурационная информация на сервере RADIUS определяет, какие средства следует установить на машине NAS.

*В системе RADIUS функции аутентификации и авторизации совмещены.*

Аутентификация и авторизация RADIUS показаны на рис. 7.8.

Учетные функции протокола RADIUS могут использоваться независимо от функций аутентификации и авторизации. Они позволяют в начале и в конце каждой сессии отправлять данные о количестве ресурсов (т. е. времени, пакетов, байтов и т. д.), использованных в ходе этой сессии. Провайдер услуг Интернет (ISP) может использовать программные средства контроля доступа и учета RADIUS для удовлетворения специальных требований безопасности и биллинга.

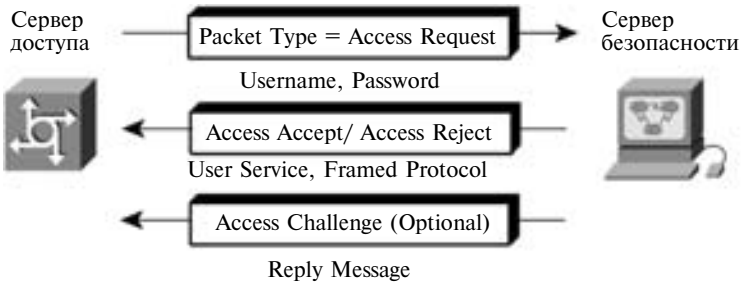



**Рис. 7.8. Аутентификация и авторизация по протоколу RADIUS**

Транзакции между клиентом и сервером RADIUS аутентифицируются с помощью общего «секрета», который никогда не передается по сетевым каналам. Кроме того, обмен любыми пользовательскими паролями между клиентом и сервером RADIUS идет только в шифрованном виде, что исключает подслушивание чужих паролей и последующее злоупотребление ими.

## 7.6. Стандарт IEEE 802.1x и протокол EAPOL

Стандарт сетевой аутентификации IEEE 802.1x нашел широкую поддержку у производителей сетевого оборудования и ПО как для беспроводных, так и для проводных сетей. Говоря о технологии сетевой аутентификации пользователей, стоит упомянуть протокол PPP, который наиболее часто используется для подключения клиентов по коммутируемым линиям к Интернет-провайдерам. Протокол PPP также используется некоторыми сервис-провайдерами для аутентификации пользователей, применяющих xDSL или кабельные модемы. Кроме того, PPP является частью протокола L2TP, на котором основан безопасный удаленный доступ к системам на базе Windows 2000 и выше.

Итак, протокол PPP изначально использовался для подключения удаленных пользователей, и поэтому он должен был иметь механизмы аутентификации пользователей. Первоначально поддерживалась только передача имени пользователя или пароля в нешифрованном виде, что не соответствует современным требованиям сетевой безопасности.

В последнее время для протокола были разработаны новые механизмы аутентификации под общим названием EAP (Extensible Authentication Protocol). Принципы работы данного протокола были рассмотрены выше. Протокол EAP был создан с целью упразднения частных механизмов аутентификации и распространения стандартизированных подходов — схем типа «запрос—ответ» (challenge-response) и инфраструктуры, основанной на публичных ключах и пользовательских сертификатах. Стандартизация механизмов EAP позволила сделать аутентификацию прозрачной для серверов доступа различных производителей. Например, при подключении пользователя к серверу удаленного доступа и использовании механизма EAP протокола PPP для аутентификации сам сервер доступа не должен знать или поддерживать конкретные механизмы или алгоритмы аутентификации, его задача в этом случае — лишь передать пакеты EAP-сообщений RADIUS-серверу, на котором фактически производится аутентификация. В этом случае сервер доступа исполняет роль посредника между клиентом и RADIUS-сервером, в задачи которого входит передача EAP-сообщений между ними. Модель контроля доступа к портам стандарта 802.1x показана на рис. 7.9.

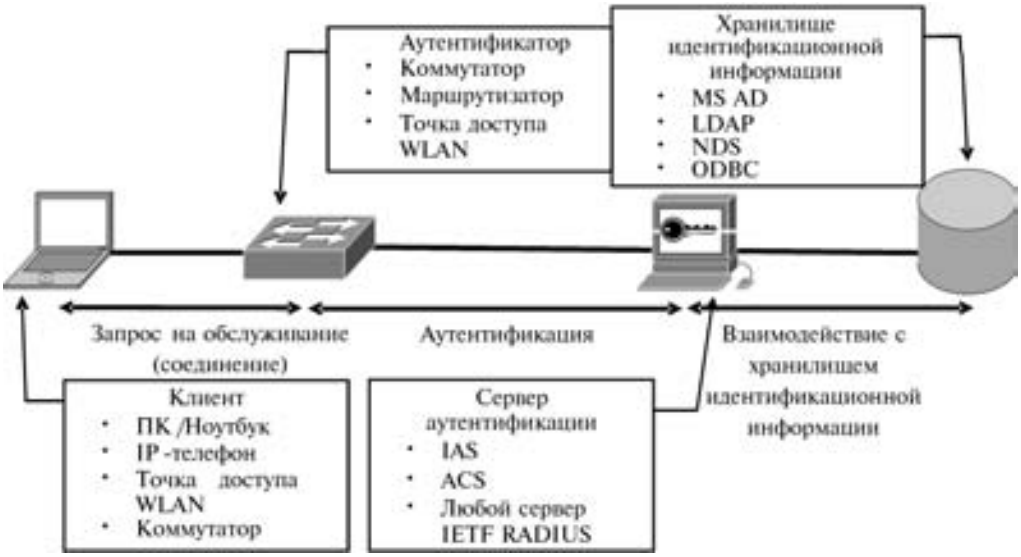



**Рис. 7.9. Модель контроля доступа к портам стандарта 802.1x. Основные элементы**

Стандарт 802.1x описывает процедуру передачи EAP-сообщений сервером доступа (например, коммутатором или беспроводной точкой доступа) в проводных или беспроводных Ethernet-сетях. При этом стандарт 802.1x напрямую упаковывает EAP-сообщения в Ethernet-кадры, не применяя для их передачи протокол PPP. Это вызвано тем, что использовать протокол PPP во многих случаях не обязательно, например, при подключении Ethernet-рабочей станции, не поддерживающей протокол TCP/IP, или в том случае, когда использование протокола PPP является избыточным.

В стандарте 802.1x определяется три основных элемента:

- Супликант (Supplicant) — пользователь, который нуждается в сетевой аутентификации.
- Сервер аутентификации (Authentication Server) — обычно RADIUS-сервер, который производит фактическую аутентификацию, например Cisco ACS или Microsoft IAS.
- Аутентификатор (Authenticator) — сетевое устройство, находящееся между суппликантом и сервером аутентификации и предоставляющее доступ в сеть, например, точка доступа или Ethernet-коммутатор.

Ключевым моментом является то, что сетевые устройства — аутентификаторы — могут быть достаточно простыми, поскольку для реализации функций 802.1x в них требуются минимальные аппаратные затраты, в то время как весь интеллект концентрируется в RADIUS-сервере. Такая схема имеет дополнительные выгоды и позволяет организовать тесную интеграцию управления сетевым оборудованием и сетевым ПО, что значительно облегчает управление информационной системой большого предприятия.

*Аутентификаторы могут быть достаточно простыми, поскольку для реализации функций 802.1x в них требуются минимальные аппаратные затраты.*


**Рис. 7.10. Модель контроля доступа к портам стандарта 802.1x. Схема работы**

Протокол передачи EAP-сообщений в стандарте 802.1x называется EAPOL (EAP encapsulation over LAN) и в настоящее время определен для Ethernet ЛВС, а также беспроводных сетей стандартов серии IEEE 802.11 и ЛВС, использующих технологии token ring и FDDI. На рис. 7.10 показана модель контроля доступа к портам стандарта 802.1x.

*Протокол передачи EAP-сообщений в стандарте 802.1x называется EAPOL (EAP encapsulation over LAN) и в настоящее время определен для Ethernet ЛВС, а также беспроводных сетей стандартов серии IEEE 802.11 и ЛВС.*

Схема работы протокола EAPOL достаточно проста. При этом можно выделить следующие основные режимы работы:

1. Аутентификатор посылает запрос на аутентификацию (EAP-Request/Identity) суппликанту, как только он определит, что какой-то из его Ethernet-портов перешел в активное состояние (link active), то есть к нему подключен сетевой адаптер. Таким образом, если отключить клиентскую станцию, которая уже прошла аутентификацию, и снова подключить к сетевому порту, то потребуется пройти аутентификацию еще раз.

2. Суппликант посылает сообщение/ответ (EAPResponse/Identity) аутентификатору, которое затем передается им на сервер аутентификации (RADIUS).

3. Сервер аутентификации в ответ посылает пакет-запрос (challenge) аутентификатору, который затем переупаковывает его из IP-транспорта в EAPOL и передает суппликанту. В различных схемах аутентификации число таких сообщений может изменяться. В EAP поддерживается как аутентификация клиентской стороны, так и взаимная «сильная» аутентификация клиента и сервера. Заметим, что только последний вариант считается приемлемым для использования в беспроводных сетях.

4. Суппликант отвечает на запрос в соответствие с выбранным алгоритмом и передает его аутентификатору, который пересылает его на сервер аутентификации.

5. Если суппликант предоставляет правильный ответ на запрос, сервер посылает сообщение об успешной аутентификации суппликанту. В этой ситуации аутентификатор

открывает клиенту доступ к ЛВС, который может зависеть от дополнительных параметров, передаваемых ему RADIUS-сервером, например, от номера ВЛВС (VLAN) или определенного уровня качества обслуживания (QoS).

Таким образом, использование сетевой аутентификации позволяет предоставлять пользователю определенный номер ВЛВС или уровень качества обслуживания независимо от точки подключения в корпоративную ЛВС. Это обеспечивает как мобильность пользователей, так и постоянное соблюдение профиля безопасности сети — если даже сетевые кабели будут случайно перепутаны, пользователь не сможет войти в ВЛВС (подключиться не к своему сегменту сети), доступ к которой ему запрещен. Стандарт включает три вида протокола EAP:

- EAP-MD5 — с хэшированием имени пользователя и пароля по алгоритму MD5;
- EAP-OTP — с поддержкой доступа по одноразовым паролям;
- EAP-TLS — аутентификация с установлением защищенного канала (SSL).

## 7.7. Протокол EAP-TLS с использованием российской криптографии

### 7.7.1. Общие сведения

Протокол EAP предоставляет стандартный механизм поддержки дополнительных методов аутентификации в протоколе PPP. При использовании EAP можно добавить несколько схем аутентификации, включая смарт-карты, Kerberos, открытые ключи, одноразовые пароли и др. Существующие в настоящее время реализации протокола EAP фокусируются на аутентификации клиента к серверу. Однако зачастую требуется взаимная аутентификация клиента и сервера.

Протоколы PPP (такие, как 3DES, Triple-DES Encryption Protocol) используют сессионные ключи, поэтому необходимо иметь механизм получения таких ключей. Такой механизм реализован в протоколе EAP-TLS. Этот протокол позволяет обеим сторонам, взаимодействующим по протоколу PPP, осуществлять защищенный обмен по заданным алгоритмам, взаимную аутентификацию и управление ключами на основе протокола TLS.

Протокол EAP-TLS используется в стандарте сетевой аутентификации IEEE 802.1x для осуществления первоначальной аутентификации субъекта локальной вычислительной сети (виртуальной частной сети, VPN).

*Протоколы PPP (такие, как DES, Triple-DES Encryption Protocol) используют сессионные ключи, поэтому необходимо иметь механизм получения таких ключей. Такой механизм реализован в протоколе EAP-TLS.*

Обмен по протоколу EAP-TLS начинается со стандартного обмена аутентификатора и аутентифицируемого субъекта в соответствии с протоколом EAP. Аутентификатор отправляет пакет EAP-запроса на аутентификацию субъекту, который в свою очередь отправляет аутентификатору пакет EAP-ответа, содержащий его идентификатор.

С этой точки зрения в процессе обмена по протоколу EAP между аутентификатором и субъектом аутентификатор может действовать как промежуточное устройство, накапливающее приходящие от противоположной стороны пакеты для передачи их RADIUS-сер-

веру или защищенному серверу. В протоколе EAP-TLS под понятием EAP-сервер подразумевается конечная точка в такой цепи, которая общается с субъектом, нуждающимся в аутентификации.

Субъекта, взаимодействующего с EAP-сервером, в дальнейшем будем называть EAP-клиентом. Получив идентификатор EAP-клиента, EAP-сервер отправляет стартовый пакет EAP-TLS. Последний представляет собой пакет EAP-запроса, в котором EAP-Type=EAP-TLS, выставлен стартовый бит (S) и отсутствуют данные.

После этого начинается непосредственный обмен по протоколу EAP-TLS. EAP-клиент отправляет пакет EAP-ответа, в котором EAP-Type=EAP-TLS. Поле данных этого пакета инкапсулируется из сообщения client_hello протокола TLS. Шифрование и компрессия пакета не производятся.

Сообщение client_hello содержит версию протокола TLS, идентификатор сессии, случайные данные клиента, а также набор поддерживаемых клиентом шифр-сюит. Отправляемая клиентом версия должна быть не ниже, чем TLS версии 1.0.

После этого EAP-сервер отвечает пакетом EAP-запроса, в котором EAP-Type=EAP-TLS. Поле данных этого пакета инкапсулируется из одного или более TLS-пакетов. Эти пакеты содержат в своей совокупности ряд сообщений, формируемый в зависимости от того, создается ли новая сессия или восстанавливается предыдущая. Если создается новая сессия, то отправляются сообщения server_hello, certificate, certificate_request и server_hello_done. В случае восстановления предыдущей сессии отправляются сообщения server_hello, change_cipher_spec и finished.

Сообщение server_hello содержит версию протокола TLS, идентификатор сессии, случайные данные сервера и выбранную сервером шифр-сюиту.

Если отправленный клиентом идентификатор сессии был нулевым или он не поддерживается сервером, то сервер выбирает идентификатор для установления новой сессии. В противном случае сервер пытается восстановить ранее установленную сессию с заданным идентификатором.

Идентификатор сессии в рамках протокола TLS служит для повышения эффективности в тех случаях, когда клиент повторно производит аутентификацию для сервера через небольшой промежуток времени. То же самое можно сказать и для PPP-аутентификации.

Сообщение certificate представляет собой сертификат, содержащий открытый ключ сервера, соответствующий выбранной им шифр-сюите в сообщении server_hello.

Сообщение certificate_request (запрос на сертификат клиента) оправляется в том случае, если сервер требует аутентификацию клиента. В случае EAP-сервера создание такого запроса желательно, но не обязательно.

После получения этих сообщений EAP-клиент отправляет пакет EAP-ответа, в котором EAP-Type=EAP-TLS. Поле данных этого пакета инкапсулируется из одного или более TLS-пакетов, содержащих сообщения change_cipher_spec, client_key_exchange, finished, а также возможно сообщения certificate и certificate_verify. Если сервер восстановил предыдущую сессию, то клиент отправляет только сообщения change_cipher_spec и finished.

Что касается сообщения certificate, то возможны следующие варианты ответной реакции клиента:

1. Сервер не запросил сертификат клиента. В зависимости от настройки верхнего уровня связь прервется, или продолжается установление связи в соответствии с пунктом 4).

2. Сервер запросил сертификат клиента. У клиента есть сертификат, соответствующий параметрам Диффи—Хеллмана сертификата сервера. Он передает сообщением certificate этот сертификат серверу.

3. Сервер запросил сертификат клиента, но у клиента нет сертификата, соответствующего параметрам Диффи—Хеллмана сертификата сервера. Если клиент не хочет аутенти-

фикации его сервером, он посылает сообщение certificate пустым. В этом случае система может быть настроена как на прекращение связи, так и на ее продолжение. Если связь продолжается, осуществляется переход к пункту 4).

4. У клиента нет сертификата, соответствующего параметрам Диффи—Хеллмана сертификата сервера, но есть сертификат, соответствующий другим параметрам или соответствующий закрытый ключ предназначен только для подписи. В этом случае клиент передает свой сертификат сообщением certificate и генерирует эфемеральную пару ключей (закрытый/открытый) с параметрами сертификата сервера. Если у клиента нет никакого сертификата, то он генерирует эфемеральную пару ключей (закрытый/открытый) с параметрами сертификата сервера, но сообщение certificate не передает.

Клиент на основе своего закрытого ключа, соответствующего сертификату с параметрами Диффи—Хеллмана сервера, или закрытого ключа выработанной им эфемеральной пары и открытого ключа сертификата сервера формирует ключ Диффи—Хеллмана (ключ обмена), генерирует премастер-ключ (32 случайных байта) и сообщением client_key_exchange передает ее серверу. Если клиент вырабатывал эфемерную пару ключей Диффи—Хеллмана, то в этом же сообщении он отправляет также открытый ключ эфемеральной пары.

После этого у клиента имеются пары ключей Диффи—Хеллмана (закрытый, открытый), шифр-сюита сессии, премастер-ключ и случайные данные клиента и сервера (32 байта каждая).

Если клиент в сообщении client_key_exchange передал открытый ключ из эфемеральной пары и передал свой сертификат, он передает сообщение certificate_verify, содержащее подпись на закрытом ключе из этого сертификата (с параметрами, отличными от параметров сертификата сервера) хэш-значения всего диалога клиент/сервер до данного момента.

Клиент переключается на криптографические алгоритмы, соответствующие выбранной сервером шифр-сюите, и сообщает об этом переходе серверу сообщением change_cipher_spec.

Из имеющихся данных клиент вырабатывает мастер-ключ (48 байт), а из него и рабочие ключи. Помимо этого из мастер-ключа и хэш-значения всего диалога клиент/сервер до данного момента вырабатывается сообщение finished и отправляется серверу.

В случае восстановления сессии сообщение finished вырабатывается из сохраненного в сессии мастер-ключа.

После этого EAP-сервер отвечает пакетом EAP-запроса, в котором EAP-Type=EAP-TLS. Поле данных этого пакета инкапсулируется из одного или более TLS-пакетов. Эти пакеты содержат сообщения change_cipher_spec, finished.

Сервер, если клиент использует эфемеральную пару ключей, вычисляет хэш-значение всего диалога клиент-сервер до данного момента и проверяет подпись хэша из сообщения certificate_verify. Этим он проверяет одновременно открытый ключ эфемеральной пары, переданный клиентом в сообщении client_key_exchange.

Сервер на основе своего закрытого ключа (соответствующего его сертификату) и открытого ключа клиента (из сертификата клиента с параметрами сервера или из выработанной им эфемеральной пары) формирует ключ Диффи—Хеллмана, ключ обмена и дешифрует премастер-ключ.

Сервер переключается на криптографические алгоритмы, соответствующие выбранной сервером шифр-сюите, и сообщает об этом переходе клиенту сообщением change_cipher_spec.

После всех операций у сервера имеются пары ключей Диффи—Хеллмана (закрытый, открытый), шифр-сюита сессии, премастер-ключ и случайные данные клиента и сервера (32 байта каждая).

Из этих данных сервер вырабатывает мастер-ключ (48 байт), а из него и рабочие ключи. Помимо этого, из мастер-ключа и хэш-значения всего диалога клиент/сервер до данного момента вырабатывается сообщение finished, шифруется и отправляется клиенту.

Если аутентификация пройдена успешно, то EAP-клиент отправляет пакет EAP-ответа, в котором EAP-Type = EAP-TLS. Поле данных этого пакета пустое.

EAP-сервер в случае успешной аутентификации отвечает аналогичным пакетом EAP-запроса.

Если аутентификация оказывается неуспешной, то формируется пакет, в котором EAP-Type=EAP-TLS, а в поле запроса содержится TLS-сообщение об ошибке.

### 7.7.2. Получение ключей

Ключи шифрования, используемые для PPP-шифрования, получаются из TLS мастер-ключа. Ключи для обеих сторон производятся следующим образом: из мастер-ключа, полученного в процессе обмена, псевдослучайной функции PRF и случайных данных, получаемых конкатенацией случайных данных клиента и сервера, вычисляется значение функции PRF (master secret, "client EAP encryption", random) — 128 байт, после чего вычисляется значение PRF (“”, "client EAP encryption", random) — 64 байта (где “” —пустая строка).

Из полученных первым преобразованием PRF 128 байт последовательно создаются 4 ключа по 32 байта каждый: ключ шифрования клиента (используется для шифрования данных, отправляемых EAP-серверу), ключ шифрования сервера (используется для шифрования данных, отправляемых клиенту), ключ аутентификации клиента (используется для подсчета MAC при отправке сообщений серверу) и ключ аутентификации сервера (используется для подсчета MAC при отправке сообщений клиенту).

Из полученных вторым преобразованием PRF 64 байт последовательно создаются два вектора инициализации — клиента и сервера — используемые в процессе шифрования сообщений.

Поскольку в основе протокола EAP-TLS лежит механизм установления аутентичного соединения в соответствии с протоколом TLS, то использование российской криптографии целесообразно проводить именно для выполнения основных функций протокола TLS.

В реализации протокола TLS на базе российской криптографии используются следующие алгоритмы:

- формирования/проверки подписи при аутентификации клиента и сервера (алгоритмы ГОСТ Р 34.10—94 и ГОСТ Р 34.10—2001);
- шифрования/дешифрования информации (алгоритм ГОСТ 28147—89);
- контроля целостности передаваемой информации (ключевой хэш на базе алгоритма хэширования ГОСТ Р 34.11—94);
- обмена ключей (с использованием алгоритма Диффи—Хеллмана на базе алгоритмов ГОСТ Р 34.10—94 и ГОСТ Р 34.10—2001).

Подробное описание данных алгоритмов приводится в стандартах RFC 4490 и RFC 4491. Описание специфичных для российских криптографических алгоритмов шифр-сюит, функций выработки и шифрования ключей приводится в информационном документе RFC 4357.

Кроме того, компанией «КРИПТО-ПРО» был разработан и опубликован документ draft-chudov-cryptopro-cptls, в котором приводится подробное описание использования российских криптографических алгоритмов в протоколе TLS.

## 7.8. Стандарт IEEE 802.1x в операционных системах Microsoft

До последнего времени клиент IEEE 802.1x в операционных системах компании Microsoft 802.1x был изначально встроен только в ОС Windows XP, однако относительно недавно компания выпустила свободно распространяемое дополнение для Windows 2000, позволяющее и данной ОС производить сетевую аутентификацию по протоколу 802.1x.

В стандартной комплектации Windows XP поддерживает три метода EAP:

- EAP-MD5. Сервер аутентификации запрашивает идентификационные данные у инициатора взаимодействия. Инициатор взаимодействия объединяет запрос со своим идентификатором и паролем, создает хэш MD5 из всех этих данных и посылает его обратно на сервер аутентификации. Последний дешифрует принятые данные, и в случае совпадения их с исходным запросом аутентификация завершается успешно. Заметим, что Windows не позволяет использование метода EAP-MD5 для беспроводных подключений по протоколу 802.1x.
- EAP-TLS. Сервер аутентификации открывает сеанс TLS с инициатором взаимодействия. Сервер посылает свой цифровой сертификат инициатору взаимодействия, а он проверяет действительность этого сертификата. После этого инициатор взаимодействия посылает свой цифровой сертификат на сервер, а тот проверяет действительность сертификата. Таким образом, клиент и сеть производят взаимную проверку подлинности, и если каждая из сторон доверяет сертификату другой стороны и этот сертификат является действительным, аутентификация завершается успешно.
- Защищенный EAP (PEAP). Обмен данными по протоколу PEAP начинается так же, как и в случае EAP: сервер аутентификации открывает сеанс TLS с инициатором взаимодействия и посылает ему свой цифровой сертификат для проверки. Если инициатор взаимодействия доверяет этому сертификату, он удостоверяет свою подлинность серверу одним из нескольких методов. В настоящее время единственным доступным методом аутентификации со стороны просителя в Windows является MS-CHAPv2, в котором инициатор взаимодействия использует традиционные учетные записи (имена и пароли пользователей и компьютеров) для проверки подлинности. Это называется PEAP-EAP-MS-CHAPv2. Обратите внимание, что можно также выбрать вариант PEAP-EAP-TLS, хотя на самом деле в его использовании нет смысла. Он предусматривает открытие независимого второго сеанса TLS внутри первого; такое удвоение сеансов TLS замедляет работу по сравнению с методом EAP-TLS.

ПК с установленными ОС семейства Windows 2000/2003 Server и службой IAS может исполнять функции RADIUS-сервера, который, в свою очередь, выполняет аутентификацию и авторизацию клиентов, использующих протоколы EAP-TLS, PEAP-MS-CHAP v2 или PEAP-EAP-TLS.

Клиенты Microsoft 802.1x Authentication Client packages для Windows 98 и Windows NT 4.0 Workstation доступны только партнерам компании Microsoft, которые имеют контракты по технической поддержке уровня Premier и Alliance. Однако для версий Windows 98, NT 4.0 или Linux можно использовать утилиты от ряда независимых производителей, например, Funk Software (http://www.funk.com/).

## 7.9. Cisco NAC

Технология Cisco Network Admission Control (NAC) усиливает сетевую инфраструктуру, сокращая вред от воздействия вирусов и «червей». С помощью Cisco NAC организации могут обеспечить сетевой доступ для таких оконечных устройств, как персональные компьютеры, карманные компьютеры и серверы, полностью соблюдая при этом заданную политику безопасности. Cisco NAC позволяет отказать в доступе устройствам, которые не соответствуют политике защиты, и поместить их в карантинную область или предоставить им ограниченный доступ к информационным ресурсам. Cisco NAC — первый шаг многоэтапной концепции защиты Cisco Self-Defending Network по идентификации угроз и их предотвращению.

*С помощью Cisco NAC организации могут обеспечить сетевой доступ для таких оконечных устройств, как персональные компьютеры, карманные компьютеры и серверы.*

Существующие антивирусные решения, опирающиеся на распознавание сигнатур атаки, не содержат и не могут распознавать вирусы с момента их появления (Day-zero) и атаки типа «отказ в обслуживании», которые этими вирусами порождаются. Особенность саморазмножения новейших атак делает их особенно опасными и разрушительными.

Персональные компьютеры и серверы, которые не соответствуют корпоративной политике безопасности, являются повсеместным явлением и их трудно обнаружить, ограничить и очистить. Блокирование и изоляция таких систем требуют много времени и ресурсов и приводят к тому, что заражение, которое, казалось бы, излечено, проявляется снова через некоторый срок. Проблема заключается в сложности нынешней сетевой среды, которая состоит из следующих элементов:

- типы пользователей — сотрудники, поставщики, подрядчики;
- типы конечных точек — персональные компьютеры в офисе, домашние ПК, серверы;
- типы доступа — кабельный, беспроводной, виртуальные частные сети и доступ по коммутируемым линиям связи.

Cisco NAC учитывает увеличивающуюся угрозу сетям, усложнение сетевой среды, и обеспечивает, в первую очередь, технологию защиты хостов, но не фокусируется на доступности и отказоустойчивости всей сети компании.

Cisco NAC предлагает полнофункциональное решение, которое позволяет организациям проводить политику применения «заплаток» на оконечных узлах, а также помещать несоответствующие политике безопасности и потенциально уязвимые системы в карантинные области с ограниченным сетевым доступом или без него. Предлагаемая Cisco NAC возможность комбинировать информацию о статусе защиты конечного устройства со сведениями об осуществлении доступа к сети позволяет организации значительно улучшить защиту своей вычислительной инфраструктуры.

Cisco NAC разрешает сетевой доступ соответствующим политике и доверенным оконечным устройствам (например, персональные компьютеры, серверы, карманные компьютеры) и ограничивает доступ тем устройствам, которые не соответствуют политике безопасности. Решение о доступе принимается на основе информации об антивирусной защищенности и об уровне применения программных обновлений («заплаток») в операционной системе.

Cisco NAC умножает отдачу средств, инвестированных в сетевую инфраструктуру и технологию защиты хостов, связывая их вместе и обеспечивая этим возможность контроля сетевого допуска. Например, организация может быть уверена, что применение антивирусных программ обеспечивается сетью Cisco — маршрутизаторами, коммутаторами, беспроводными устройствами и устройствами защиты. Таким образом, Cisco NAC дополняет — а не заменяет — классические и широко распространенные технологии защиты: межсетевые экраны, системы обнаружения вторжений, аутентификацию пользователей и защиту коммуникаций.

*Cisco NAC разрешает сетевой доступ соответствующим политике и доверенным оконечным устройствам и ограничивает доступ тем устройствам, которые не соответствуют политике.*

Состав компонентов Cisco NAC приведен на рис. 7.11.

- Доверенный агент Cisco (Cisco trust agent) — программное обеспечение, размещаемое на оконечных узлах (рабочие станции, серверы и т. п.). Доверенный агент собирает информацию о защищенности от множества программных клиентов, таких, как антивирусные клиенты, а затем передает эти сведения устройствам сетевого доступа Cisco, которые осуществляют контроль допуска. Компания Cisco лицензировала технологию доверенного агента для своих партнеров в области антивирусов, и теперь этот агент может интегрироваться с их клиентским ПО. Доверенный агент будет также интегрироваться и с Cisco Security Agent, чтобы контролировать доступ, например, в зависимости от уровня применения программных обновлений в операционной системе оконечного устройства. Cisco Security Agent програм-

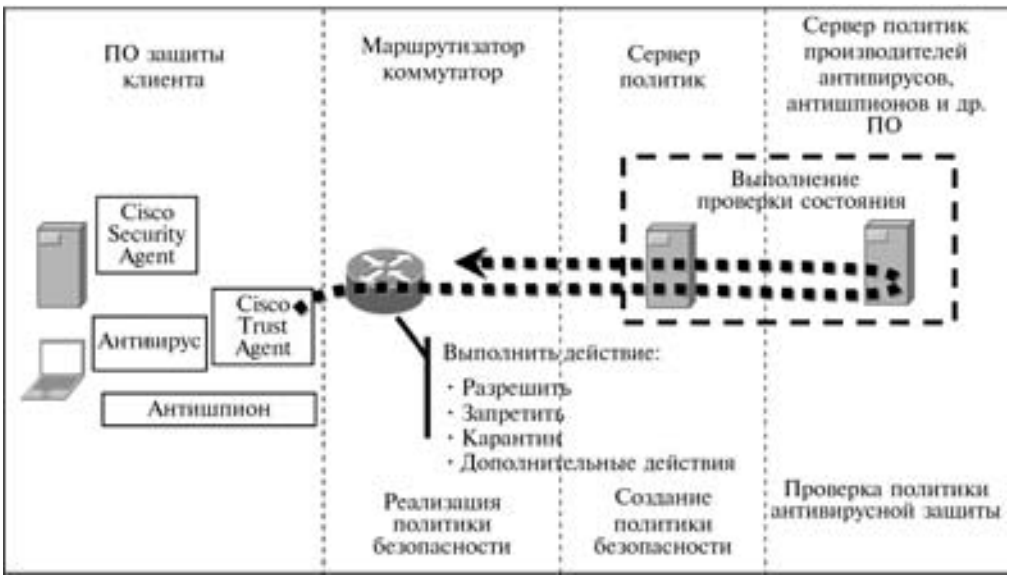



**Рис. 7.11. Компоненты Cisco NAC**

мное решение защиты хоста от несанкционированного доступа, будет получать доступ к информации о версии операционной системы, о программных «заплатках» и текущем состоянии системы и передавать эту информацию доверенному агенту Cisco (Cisco Trust Agent). Хостам, не имеющим необходимых программных обновлений, может быть предложен ограниченный доступ или вообще отказано в доступе к сети.

- Устройства сетевого доступа — к сетевым устройствам, осуществляющим политику контроля допуска, относятся маршрутизаторы, коммутаторы, точки беспроводного доступа и устройства защиты. Эти устройства требуют наличия электронных удостоверений от оконечного узла, и передают эту информацию на серверы контроля политики, где и принимается решения о доступе в сеть. На основе политики заказчика сеть будет проводить соответствующее решение о контроле сетевого доступа — разрешить, запретить, поместить под наблюдение, ограничить.
- Сервер политики — оценивает защищенность конечной точки по информации, полученной от устройств сетевого доступа, и определяет для них соответствующую политику доступа. Основой системы серверов политики являются сервер аутентификации, авторизации и отчетности (AAA) RADIUS — Cisco Secure Access Control Server (ACS). Система работает совместно с серверами приложений партнеров Cisco NAC, обеспечивающих более широкие возможности проверки электронных удостоверений, такими как серверы антивирусной политики.
- Система управления — CiscoWorks VPN/Security Management Solution (VMS) управляет элементами Cisco NAC, а решение CiscoWorks Security Information Management Solution (SIMS) предлагает инструменты мониторинга и создания отчетов. Партнеры Cisco NAC предлагают решения для управления своим программным обеспечением на конечных узлах.

На рис. 7.12 показаны сценарии развертывания Cisco NAC в различных сегментах сети.

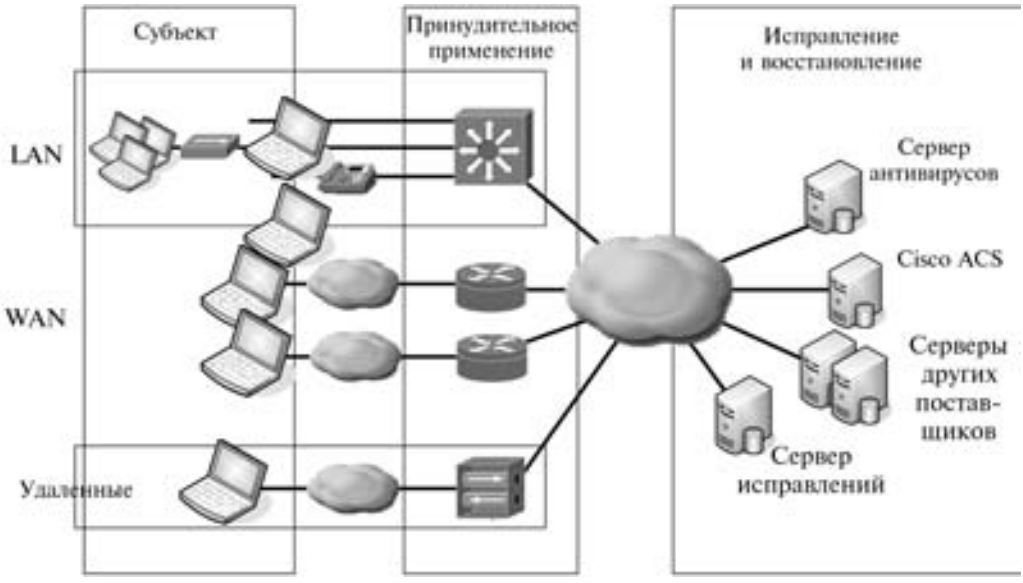



**Рис. 7.12. Сценарии развертывания Cisco NAC**

Cisco NAC помогает гарантировать соответствие хостов в удаленных или домашних офисах, которые подключаются к ресурсам центральной компании либо по территориальным сетям передачи данных, либо по защищенному каналу в Интернете. Сюда входит проверка соответствия на главном маршрутизаторе или на маршрутизаторе филиала.

- *Защита удаленного доступа.* Еще до того, как мобильный или удаленный работник получит доступ в сеть по коммутируемой телефонной линии, по протоколу IPSec или иному соединению, Cisco NAC гарантирует, что на компьютере работника установлены последние версии антивирусов и программных обновлений для операционной системы.
- *Защита беспроводной сети.* Cisco NAC проверяет подключенные по беспроводным каналам хосты на предмет правильности их программных обновлений. Для выполнения этой проверки используется протокол 802.1x и комбинация аутентификации пользователя и устройства.
- *Защита сетевого доступа и центра обработки данных.* Cisco NAC контролирует серверы и персональные компьютеры в офисе еще до того, как им будет предоставлен доступ к сети, позволяет гарантировать их соответствие антивирусной политике и политике использования программных обновлений, принятых в компании. Расширяет контроль допуска на коммутаторах Уровня 2, что позволяет уменьшить риск заражения вирусами или «червями».
- *Соответствие внешних (партнерских) сетей.* С помощью Cisco NAC можно проверять не только системы, управляемые IT-подразделением предприятия, но и любую систему, которая пытается получить доступ в сеть. На соответствие политикам антивирусов и операционных систем могут быть проверены управляемые и неуправляемые хосты, принадлежащие в том числе партнерам и подрядчикам. Если на опрашиваемом хосте отсутствует доверенный агент Cisco (Cisco Trust Agent), то используется политика доступа по умолчанию.

## Контрольные вопросы

1. Какие протоколы включены в механизм аутентификации Point-to-Point Protocol (PPP)?
2. Перечислите основные элементы стандарта 802.1x.
3. Какие методы EAP стандарта 802.1x включены в стандартную комплектацию Windows XP?
4. Опишите взаимодействие между пользователем, клиентом и сервером RADIUS.
5. Опишите метод получения ключей шифрования, используемых для PPP.

# Глава 8
# АУТЕНТИФИКАЦИЯ В ЗАЩИЩЕННЫХ СОЕДИНЕНИЯХ

Протоколы безопасности на транспортном уровне SSL и Secure Shell Protocol (SSH) обеспечивают безопасную передачу данных между клиентом и сервером. Оба протокола разработаны рабочей группой IETF по безопасности транспортного уровня (Transport Layer Security — TLS). Безопасный протокол передачи гипертекста S-HTTP предоставляет надежный механизм Web-транзакций, однако в настоящее время наиболее распространен протокол SSL. Рамочная структура SOCKS позволяет приложениям клиент/сервер в доменах TCP и UDP удобно и безопасно пользоваться услугами межсетевого экрана. Протокол безопасности IP (IPSec) представляет собой набор стандартов поддержки целостности и конфиденциальности данных на сетевом уровне (в сетях IP). X.509 — это стандарт безопасности, определяющий структуру данных цифрового сертификата и описывающий вопросы обращения общих ключей. X.509 является важнейшим компонентом инфраструктуры открытых ключей (PKI).

## 8.1. Протоколы SSL, TLS

SSL — открытый протокол, разработанный компанией Netscape, который определяет механизм поддержки безопасности данных на уровне между протоколами приложений (такими, как Hypertext Transfer Protocol (http), Telnet, Network News Transfer Protocol (NNTP) или File Transfer Protocol (FTP)) и протоколом TCP/IP. Он поддерживает шифрование данных, аутентификацию серверов, целостность сообщений и (в качестве опции) аутентификацию клиентов в канале TCP/IP. SSL был представлен рабочей группе по безопасности консорциума W3 (W3C) для утверждения в качестве стандартного средства безопасности Web-браузеров и серверов в сети Интернет.

*SSL поддерживает шифрование данных, аутентификацию серверов, целостность сообщений и (в качестве опции) аутентификацию клиентов в канале TCP/IP.*

Основная цель протокола SSL состоит в том, чтобы обеспечить защиту и надежность связи между двумя подключенными друг к другу приложениями. Этот протокол состоит из двух уровней. Нижний уровень, который располагается поверх надежного транспортного протокола (например, TCP), называется SSL Record Protocol и используется для встраивания различных протоколов высокого уровня. Один из таких встроенных протоколов — SSL Handshake Protocol позволяет серверу и клиенту аутентифицировать друг друга и согласовывать алгоритм шифрования и криптографические ключи, прежде чем протокол приложения произведет обмен первыми битами данных. Одно из преимуществ SSL состоит в том, что он независим от протоколов приложений. Протокол высокого уровня может располагаться поверх протокола SSL. Протокол SSL поддерживает безопасность связи, обеспечивая ей следующие возможности:

- Защищенность. После первоначального квитирования связи применяются средства шифрования и определяется секретный ключ. Для шифрования данных используются средства симметричной криптографии (например, DES, RC4 и т. д.).
- Аутентификация участника сеанса связи с помощью общих ключей, т. е. средствами асимметричной криптографии (например, RSA, DSS и т. д.).
- Надежность. Транспортные средства проводят проверку целостности сообщений с помощью шифрованного кода целостности (MAC). Для вычисления кодов MAC используются безопасные хэш-функции (например, безопасный хэш-алгоритм (SHA), MD5 и т. д.).

*Основная цель протокола SSL состоит в том, чтобы обеспечить защиту и надежность связи между двумя подключенными друг к другу приложениями.*

Протокол SSL состоит из нескольких уровней. На каждом уровне сообщения имеют ряд полей для указания длины, описания и содержания. SSL воспринимает данные, предназначенные для передачи, делит их на управляемые блоки, проводит компрессию данных (если это необходимо), использует код MAC, производит шифрование и передает результат. Принятые данные дешифруются, проверяются, декомпрессируются и реассемблируются, а затем передаются клиентам более высокого уровня.

Протокол SSL принят только в рамках HTTP. Другие протоколы могут работать с SSL, но используют его не часто.

Протокол обеспечения безопасности на уровне передачи данных TLS (Transport Layer Security Protocol) создан на основе SSL 3.0 компании Netscape. Идея заключалась в том, чтобы опубликовать этот протокол как формальный RFC и придать ему до некоторой степени некоммерческий статус. Ниже приведено общее описание TLS. Более подробное описание приведено в RFC 2276 и спецификации SSL.

Протокол TLS состоит из двух уровней: TLS Record Protocol (протокол записей) и TLS Handshake Protocol (протокол установления связи). TLS Record Protocol действует поверх TCP и UDP и выполняет следующие функции:

- симметричное шифрование для шифрования. Ключи для такого шифрования генерируются отдельно для каждого соединения и при их создании используется секретная информация, полученная с помощью другого протокола (такого, как TLS Handshake Protocol). TLS Record Protocol может быть использован и без шифрования;
- передача сообщений: сюда входит проверка целостности переданных сообщений с помощью кода идентификации по ключу. Для этого применяется хэширование (например, SHA, MD5 и др.);
- выступает в качестве оболочки для протоколов более высокого уровня. Примером такого протокола может служить TLS Handshake Protocol, который позволяет серверу и клиенту идентифицировать друг друга, выбрать алгоритм шифрования и используемые ключи до того, как начнут передаваться данные.

*Протокол TLS состоит из двух уровней: TLS Record Protocol (протокол записей) и TLS Handshake Protocol (протокол установления связи).*

TLS Handshake Protocol защищает соединение, обеспечивая следующие три функции:

- идентификация другой стороны с помощью шифрования с открытым ключом (например, RSA, DSS и др.). Такая проверка не является обязательной, но обычно выполняется, по крайней мере, для одной из сторон;
- разделяемый секретный ключ становится недоступным для перехвата и не может быть получен даже в случае, если атакующая сторона сможет поместить себя между соединяемыми сторонами;
- обнаружение злоумышленника при попытке изменить данные.

## 8.2. Протокол SSH

Протокол Secure Shell (SSH) предназначен для защиты удаленного доступа и других сетевых услуг в незащищенной сети. Он поддерживает безопасный удаленный вход в сеть, безопасную передачу файлов и безопасную эстафетную передачу сообщений по протоколам TCP/IP и X11. SSH, может автоматически шифровать, аутентифицировать и сжимать передаваемые данные. В настоящее время SSH достаточно хорошо защищен от криптоанализа и протокольных атак. Он хорошо работает при отсутствии глобальной системы управления ключами и инфраструктуры сертификатов и при необходимости может поддерживать инфраструктуры сертификатов, которые существуют в настоящий момент (например, DNSSEC, простую инфраструктуру открытых ключей (SPKI), X.509).

*Протокол Secure Shell (SSH) поддерживает безопасный удаленный вход в сеть, безопасную передачу файлов и безопасную эстафетную передачу сообщений по протоколам TCP/IP и X11.*

Протокол SSH состоит из трех основных компонентов:

- Протокол транспортного уровня. Обеспечивает аутентификацию сервера, конфиденциальность и целостность данных с отличной защитой эстафетной передачи. В качестве опции может поддерживаться компрессия данных.
- Протокол аутентификации пользователя. Позволяет серверу аутентифицировать клиента.
- Протокол соединения. Мультиплексирует шифрованный туннель, создавая в нем несколько логических каналов.

Все сообщения шифруются с помощью IDEA или одного из нескольких других шифровальных средств (тройного DES с тремя ключами, DES, RC4-128, Blowfish). Обмен ключами шифрования происходит с помощью алгоритма RSA, а данные, использованные при этом обмене, уничтожаются каждый час (ключи нигде не сохраняются). Для защиты от «подслушивания» (спуфинга) сети IP используется шифрование; для защиты от DNS и спуфинга маршрутизации — аутентификация с помощью общих ключей. Кроме того ключи RSA используются для аутентификации центральных компьютеров.

Недостатком протоколов безопасности, действующих на уровне сессий, является их зависимость от инструкций протокола транспортного уровня. В случае SSL это означает, что атака на TCP может быстро прервать сессию SSL и потребовать формирования новой сессии, в то время как TCP будет считать, что все идет нормально.

Более подробные технические детали о протоколе SSH можно получить в рабочей группе IEFT Secure Shell (secsh).

К преимуществам средств безопасности транспортного уровня (например, SSL или SSH) относятся:

- возможность действий на сквозной основе (end-to-end) с существующими стеками TCP/IP, существующими интерфейсами прикладного программирования (API) (WinSock, Berkeley Standard Distribution (BSD) и т. д.);
- повышенная эффективность по сравнению с медленными каналами, поддержка технологии Van Jacobson для компрессии заголовков, поддержка различных средств контроля за переполнением сети, просматривающих заголовки TCP/IP;
- отсутствие каких-либо проблем с фрагментацией, определением максимального объема блоков, передаваемых по данному маршруту (MTU) и т. д.;
- сочетание компрессии с шифрованием. На этом уровне такое сочетание оказывается гораздо более эффективным, чем на уровне пакетов.

## 8.3. Протокол S-HTTP

S-HTTP представляет собой безопасный протокол связи, ориентированный на сообщения и разработанный для использования в сочетании с HTTP. Он предназначен для совместной работы с моделью сообщений HTTP и легкой интеграции с приложениями HTTP. Этот протокол предоставляет клиенту и серверу одинаковые возможности (он одинаково относится к их запросам и ответам, а также к предпочтениям обеих сторон). При этом сохраняется модель транзакций и эксплуатационные характеристики HTTP.

Клиенты и серверы S-HTTP допускают использование нескольких стандартных форматов криптографических сообщений. Клиенты, поддерживающие S-HTTP, могут устанавливать связь с серверами S-HTTP, и наоборот, эти серверы могут связываться с клиентами S-HTTP, хотя в процессе подобных транзакций функции безопасности S-HTTP скорее всего не будут использованы. S-HTTP не требует от клиента сертификатов открытых ключей (или самих открытых ключей), потому что этот протокол поддерживает только операции с симметричными ключами шифрования. Хотя S-HTTP может пользоваться преимуществами глобальных сертификационных инфраструктур, для его работы такие структуры не обязательны.

Протокол S-HTTP поддерживает безопасные сквозные (end-to-end) транзакции, что выгодно отличает его от базовых механизмов аутентификации HTTP. Последние требуют, чтобы клиент попытался получить доступ и получил отказ и лишь затем включают механизм безопасности. Клиенты могут быть настроены таким образом, чтобы любая их транзакция автоматически защищалась (обычно с помощью специальной метки в заголовке сообщения). Такая настройка, например, часто используется для передачи заполненных бланков. Если вы используете протокол S-HTTP, вам никогда не придется отправлять важные данные по сети в незащищенном виде.

*Протокол S-HTTP предназначен для совместной работы с моделью сообщений HTTP и легкой интеграции с приложениями HTTP, поддерживает безопасные сквозные (end-to-end) транзакции, что выгодно отличает его от базовых механизмов аутентификации HTTP.*

S-HTTP поддерживает высокий уровень гибкости криптографических алгоритмов, режимов и параметров. Для того чтобы клиенты и серверы смогли выбрать единый режим транзакции (так, например, им нужно решить, будет ли запрос только шифроваться или только подписываться или и шифроваться и подписываться одновременно. Такое же решение нужно принять и для ответов), используется механизм согласования опций, криптографических алгоритмов (RSA или DSA для подписи, DES или RC2 для шифрования и т. д.), и выбора сертификатов (например, «Подписывайтесь своим сертификатом Verisign»). S-HTTP поддерживает криптографию общих ключей, функцию цифровой подписи и обеспечивает конфиденциальность данных.

Отметим, что протокол S-HTTP не получил широкого распространения.

## 8.4. Протокол SOCKS

SOCKS разработан для того, чтобы дать возможность приложениям клиент/сервер в доменах TCP и UDP удобно и безопасно пользоваться услугами межсетевого экрана. Он дает пользователям возможность преодолевать межсетевой экран организации и получать доступ к ресурсам, расположенным в Интернете. SOCKS служит «посредником уровня приложений»: он взаимодействует с общими сетевыми средствами (например, Telnet и браузер Netscape) и с помощью центрального сервера (прокси-сервера) от имени компьютера пользователя устанавливает связь с другими центральными компьютерами.

*Протокол SOCKS дает пользователям возможность преодолевать межсетевой экран организации и получать доступ к ресурсам, расположенным в Интернете.*

SOCKS был разработан много лет назад Дейвом Кобласом (Dave Koblas) из компании SGI, и сегодня этот код можно бесплатно получить через Интернет. С момента первого выпуска этот код пережил несколько крупных модификаций, но каждая из них распространялась бесплатно. SOCKS версии 4 решает вопрос незащищенного пересечения межсетевых экранов приложениями клиент/сервер, основанными на протоколе TCP, включая Telnet, FTP и распространенные информационные протоколы, например HTTP, Wide Area Information Server (WAIS) и GOPHER. SOCKS версия 5, RFC 1928, является дальнейшим расширением четвертой версии SOCKS. Он включает в себя UDP, расширяет общую рамочную структуру, придавая ей возможность использования мощных обобщенных схем аутентификации, и расширяет систему адресации, включая в нее имя домена и адреса IP v.6.

В настоящее время предлагается создать механизм управления входящими и исходящими многоадресными сообщениями IP, которые проходят через межсетевой экран. Это достигается за счет определения расширений для существующего протокола SOCKS V.5, что создает основу для аутентифицированного перехода межсетевого экрана одноадресным пользовательским трафиком TCP и UDP. Однако из-за того, что поддержка UDP в текущей версии SOCKS 5 имеет проблемы с масштабируемостью и другие недостатки (и их обязательно нужно разрешить, прежде чем переходить к многоадресной передаче), расширения определяются двояко: как базовые и как многоадресные расширения UDP.

Протокол SOCKS заменяет стандартные сетевые системные вызовы в приложении их специальными версиями. Эти новые системные вызовы устанавливают связь с прокси-сер-

вером SOCKS (который конфигурируется самим пользователем в приложении или системным файлом конфигурации), подключаясь к хорошо известному порту (обычно это порт 1080/TCP). После установления связи с сервером SOCKS приложение отправляет серверу имя машины и номер порта, к которому хочет подключиться пользователь. Сервер SOCKS реально устанавливает связь с удаленным центральным компьютером, а затем прозрачно передает данные между приложением и удаленной машиной. При этом пользователь даже не подозревает, что в канале связи присутствует сервер SOCKS.

Трудность использования SOCKS состоит в том, что кто-то должен проводить работу по замене сетевых системных вызовов версиями SOCKS (этот процесс обычно называется «SOCKS-ификацией» приложения). К счастью, большинство обычных сетевых приложений (Telnet, FTP, finger, whois) уже SOCKS-ифицированы, и многие производители включают поддержку SOCKS в свои коммерческие приложения. Кроме того, SOCKS 5 включает эти процедуры в свою общую библиотеку: на некоторых системах (например, на машинах Solaris) можно автоматически SOCKS-ифицировать приложение, поставив общую библиотеку SOCKS перед «shared libc» в строке поиска библиотек (переменная среды LD_LIBRARY_PATH в системах Solaris).

Более подробные технические детали можно получить в рабочей группе IETF, работающей над проблемой аутентифицированного пересечения межсетевых экранов.

## 8.5. Семейство протоколов IPSec

IP Security (IPSec) — это семейство протоколов, которые обеспечивают шифрование, аутентификацию и защиту при транспортировке IP-пакетов; в его состав сейчас входят почти 20 предложений по стандартам и 18 RFC.

IPSec добавляет возможности шифрования и аутентификации в стек протокола TCP/IP на более низком уровне, чем протоколы прикладного уровня, такие как Secure Socket Layer (SSL) и Transport Layer Security (TLS). Защита IPSec прозрачна для приложений, поскольку она осуществляется на нижнем уровне стека TCP/IP. Для защиты приложений средствами IPSec необходимо, чтобы передача информации осуществлялась через определенный порт. Если для всех соединений приложения применяют произвольные порты, определить фильтр IPSec, идентифицирующий потоки сетевых данных этих программ, практически невозможно.

*IPSec добавляет возможности шифрования и аутентификации в стек протокола TCP/IP на более низком уровне, чем протоколы прикладного уровня.*

Приложения не обязательно должны быть IPSec-совместимы, так как данные от клиента к серверу передаются открытым текстом. Протокол IPSec шифрует полезные данные после их отправки клиентом и дешифрует до того, как они достигнут приложения на сервере.

Например, протокол Telnet передает реквизиты пользователя и данные программ открытым текстом. Ниже показан сценарий, когда и клиент, и сервер настроены на согласование ассоциации безопасности IPSec при обмене данными по Telnet (рис. 8.1).

1. Клиент отправляет пакет серверу. Пакет отправляется с произвольного порта клиента, но всегда на TCP-порт 23 Telnet-сервера.

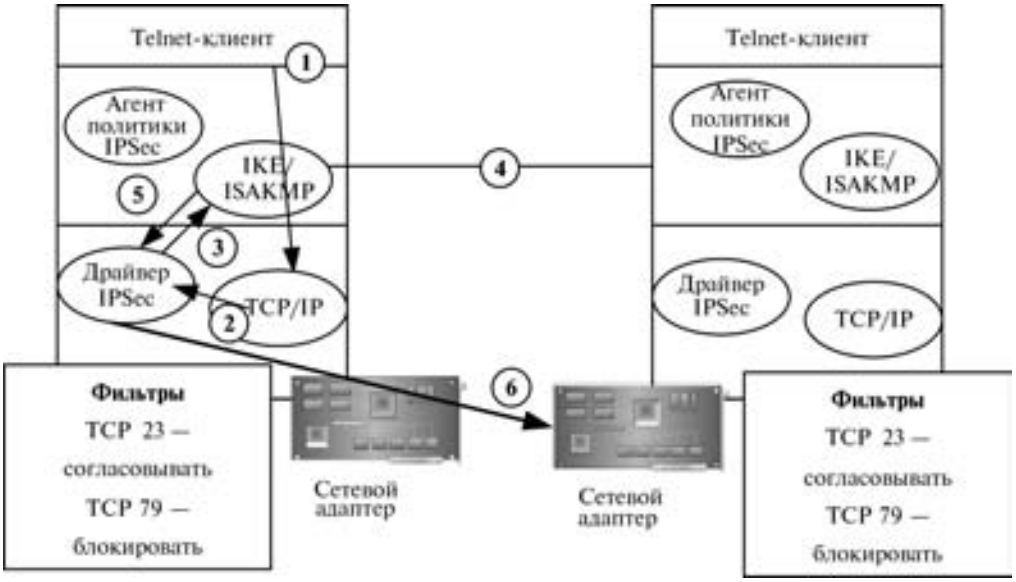



**Рис. 8.1. Обмен данными по протоколу Telnet через IPSec**

2. Драйвер IPSec на компьютере клиента перехватывает пакет, когда он достигает уровня IP, и сравнивает его со списком фильтров IPSec, настроенным на клиентском компьютере. В нашем случае фильтр соответствует TCP 23. В результате начинается согласование ассоциации (сопоставления) безопасности SA (security association) между клиентом и сервером. SA определяет параметры IPSec-сеанса между клиентом и сервером, а также протоколы IPSec и аутентификации, алгоритмы шифрования и проверки целостности.

3. Драйвер IPSec передает пакет протоколу Internet Security Association and Key Management Protocol (ISAKMP) для согласования SA между клиентом и сервером. ISAKMP используется для определения типа SA, установленного между клиентом и сервером.

4. Клиент и сервер продолжают процесс ISAKMP, используя протокол Internet Key Exchange (IKE) и осуществляя соединение по протоколу User Datagram Protocol (UDP) 500. Процесс ISAKMP устанавливает необходимые SA между клиентом и сервером. *SA включает протокол и алгоритмы шифрования, применяемые для защиты обмена данными между клиентом и сервером.*

5. Результаты SA возвращаются драйверу IPSec, поэтому он может выполнять все необходимые задачи и производить любые изменения для обеспечения безопасности данных до их передачи от клиента к серверу.

6. Драйвер IPSec применяет к данным шифрование или алгоритм целостности или и то, и другое одновременно и отправляет данные на сетевой адаптер для передачи клиенту.

### *8.5.1. Набор стандартов, используемых в IPSec*

Безопасный протокол IP (IPSec) представляет собой набор стандартов, используемых для защиты данных и для аутентификации на уровне IP. Текущие стандарты IPSec включают независимые от алгоритмов базовые спецификации, которые являются стандартными RFC.

Эти RFC, перечисленные ниже, сейчас пересматриваются с целью разрешить различные проблемы безопасности, которые имеются в текущих спецификациях:

- RFC 2401 (Security Architecture for the Internet Protocol) — Архитектура защиты для протокола IP.
- RFC 2402 (IP Authentication header) — аутентификационный заголовок IP.
- RFC 2403 (The Use of HMAC-MD5-96 within ESP and AH) — Использование алгоритма хэширования MD-5 для создания аутентификационного заголовка.
- RFC 2404 (The Use of HMAC-SHA-1-96 within ESP and AH) — Использование алгоритма хэширования SHA-1 для создания аутентификационного заголовка.
- RFC 2405 (The ESP DES-CBC Cipher Algorithm With Explicit IV) — Использование алгоритма шифрования DES.
- RFC 2406 (IP Encapsulating Security Payload (ESP)) — Шифрование данных.
- RFC 2407 (The Internet IP Security Domain of Interpretation for ISAKMP) — Область применения протокола управления ключами.
- RFC 2408 (Internet Security Association and Key Management Protocol (ISAKMP)) — Управление ключами и аутентификаторами защищенных соединений.
- RFC 2409 (The Internet Key Exchange (IKE)) — Обмен ключами.
- RFC 2410 (The NULL Encryption Algorithm and Its Use With IPsec) — нулевой алгоритм шифрования и его использование.
- RFC 2411 (IP Security Document Roadmap) — Дальнейшее развитие стандарта.
- RFC 2412 (The OAKLEY Key Determination Protocol) — Проверка аутентичности ключа.

Для защиты данных IPSec предлагает два протокола: Authentication Headers (AH) и Encapsulating Security Payloads (ESP). В своей простейшей форме AH предоставляет службы аутентификации и проверки целостности для передаваемых данных, а ESP — службы шифрования. AH и ESP — независимые протоколы. Их можно использовать отдельно или в комбинации для обеспечения целостности и защиты данных от просмотра.

*Для защиты данных IPSec предлагает два протокола: Authentication Headers (AH) и Encapsulating Security Payloads (ESP).*

### 8.5.2. Протокол Authentication Headers (AH)

AH обеспечивает аутентификацию, целостность и защиту от повтора данных, передаваемых в сети. Он не помешает просмотреть данные, но исключает их изменение при передаче.

AH-пакеты используются для аутентификации компьютеров, участвующих в обмене данными, и для обеспечения целостности передаваемых пакетов, чтобы злоумышленник не мог изменить или воспроизвести пересылаемые данные. Протокол AH рекомендуется использовать, когда соединения в рамках рабочей группы или проекта должны быть ограничены определенными компьютерами. AH гарантирует взаимную аутентификацию соединенных компьютеров, поэтому в обмене данными могут участвовать только аутентифицированные компьютеры.

Преимущество AH в том, что он обеспечивает возможность взаимной аутентификации для тех протоколов, которые ее не поддерживают. Если аутентификация перемещается на более низкий уровень в стеке сетевого протокола, все приложения смогут поддерживать IPSec.

### 8.5.3. Протокол Encapsulating Security Payloads (ESP)

ESP-пакеты позволяют шифровать данные. Кроме того, ESP предоставляет механизмы аутентификации, целостности и защиты от повтора. Протокол ESP шифрует заголовки TCP или UDP и данные приложений в IP-пакете. Если туннельный режим IPSec не используется, то исходный заголовок IP не шифруется.

При выработке решения IPSec протоколы AH и ESP можно совместно использовать в одном SA для IPSec. Оба протокола обеспечивают целостность данных, при этом AH защищает весь пакет от изменения, а ESP — только заголовок TCP/UDP и полезные данные от проверки. Протокол ESP позволяет шифровать данные. Он необходим, когда приложение не распознает протокол безопасности на прикладном уровне, например SSL.

Поскольку процесс шифрования и дешифрования IPSec происходит на уровне IP/IPSec, приложение не обязано поддерживать IPSec. Фактически приложение ничего «не знает» о защите данных средствами IPSec (рис. 8.2).

Шифрование ESP могут применять только операционные системы и сетевые устройства, поддерживающие IPSec. Если операционная система или сетевое устройство не поддерживают IPSec, то либо SA IPSec должно разрешить обмен открытым текстом, либо нужно применить альтернативный процесс шифрования.

Кроме поддержки шифрования, протокол ESP обеспечивает цифровую подпись данных. Протоколы AH и ESP различаются лишь способами защиты от изменений: AH защищает пакет целиком, а подпись ESP не защищает IP-заголовок, применяемый для маршрутизации пакета в сети. При необходимости шифровать данные и обеспечить защиту всех полей в пакетах нужно настроить SA на внедрение обоих (AH и ESP) протоколов в IPSec.

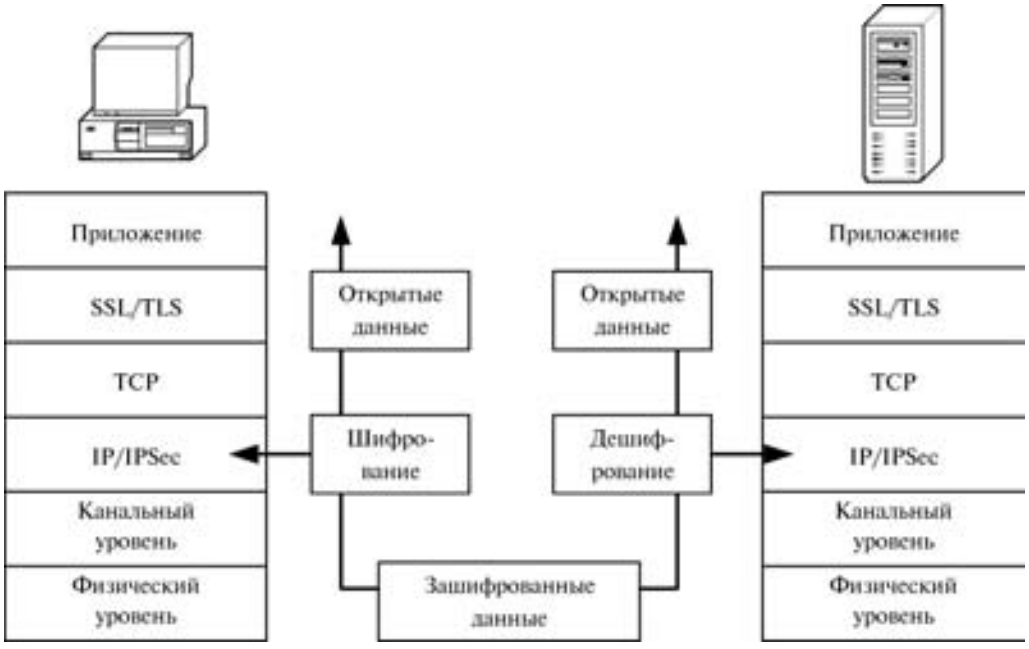



**Рис. 8.2. «Прозрачная» защита данных средствами IPSec**

### 8.5.4. Работа протоколов IPSec

Для прохождения трафика IPSec через межсетевой экран нужно разрешить прохождение через него пакетов, использующих UDP 500 и идентификатор протокола 51 для AH или 50 для ESP. Кроме того, межсетевой экран не должен транслировать сетевой адрес (Network Address Translation, NAT). Пакеты IPSec не могут пройти через NAT, поскольку поля, измененные в результате процесса NAT, уже защищены IPSec и при их изменении пакет станет недействительным. При прохождении пакета через службу NAT исходный IP-адрес источника преобразуется в общий IP-адрес, а порт источника — в реальный порт, который задан на сервере, осуществляющем преобразование.

Протокол IPSec также включает криптографические методы для управления ключами на сетевом уровне безопасности. Протокол управления ключами Ассоциации безопасности в Интернете (Internet Security Association Key Management Protocol — ISAKMP) создает рамочную структуру для управления ключами в сети Интернет и предоставляет конкретную протокольную поддержку для согласования атрибутов безопасности. Это не создает ключей сессии, однако процедуру можно использовать с разными протоколами, создающими такие ключи (например, с Oakley).

*Протокол IPSec также включает криптографические методы для управления ключами на сетевом уровне безопасности.*

Протокол определения ключей Oakley Key Determination Protocol пользуется гибридным методом Диффи—Хеллмана, чтобы создать ключи Интернет-сессии для центральных компьютеров и маршрутизаторов. Протокол Oakley решает важную задачу обеспечения полной безопасности эстафетной передачи данных. Он основан на криптографических методах, прошедших серьезное испытание практикой. Полная защита эстафетной передачи означает, что если хотя бы один ключ раскрыт, раскрыты будут только те данные, которые шифрованы этим ключом. Что же касается данных, шифрованных последующими ключами, они останутся в полной безопасности.

Протоколы ISAKMP и Oakley были совмещены в рамках гибридного протокола IKE — Internet Key Exchange. Протокол IKE, включающий ISAKMP и Oakley, использует рамочную структуру ISAKMP для поддержки подмножества режимов обмена ключами Oakley. Новый протокол обмена ключами обеспечивает (в виде опции) полную защиту эстафетной передачи данных, ассоциаций и согласования атрибутов, а также поддерживает методы аутентификации, допускающие отказ от авторства и не допускающие такого отказа. Этот протокол можно, например, использовать для создания виртуальных частных сетей (VPN) и предоставления удаленным клиентам (использующим динамически распределяемые адреса IP) доступ к защищенной сети.

*Протоколы ISAKMP и Oakley были совмещены в рамках гибридного протокола IKE — Internet Key Exchange.*

Стандарт IPSec позволяет поддерживать на уровне IP потоки безопасных и аутентичных данных между взаимодействующими устройствами, включая центральные компьютеры, межсетевые экраны (сетевые фильтры) различных типов и маршрутизаторы. Ниже приводится пример использования IPSec для обеспечения обмена аутентифицирован-


**Рис. 8.3. Безопасность обмена данными между удаленным маршрутизатором и межсетевым экраном**

ными конфиденциальными данными между удаленным маршрутизатором и межсетевым экраном (рис. 8.3).

Прежде чем пройти через межсетевой экран предприятия, весь трафик, идущий от удаленного маршрутизатора, должен быть аутентифицирован. Маршрутизатор и межсетевой экран должны согласовать ассоциацию безопасности (SA), т. е. прийти к согласию относительно политики в области безопасности. SA включает:

- алгоритм шифрования;
- алгоритм аутентификации;
- общий ключ сессии;
- срок действия ключа.

Ассоциация безопасности SA является однонаправленной, поэтому для двусторонней связи нужно устанавливать две SA, по одной для каждого направления. Как правило, в обоих случаях политика остается той же самой, но существует возможность и для асимметричной политики в разных направлениях. Согласование SA проводится через ISAKMP. Кроме того, SA могут определяться вручную. На рис. 8.4 показан процесс согласования через ISAKMP, который происходит, когда на маршрутизатор поступает пакет, предназначенный для межсетевого экрана предприятия.

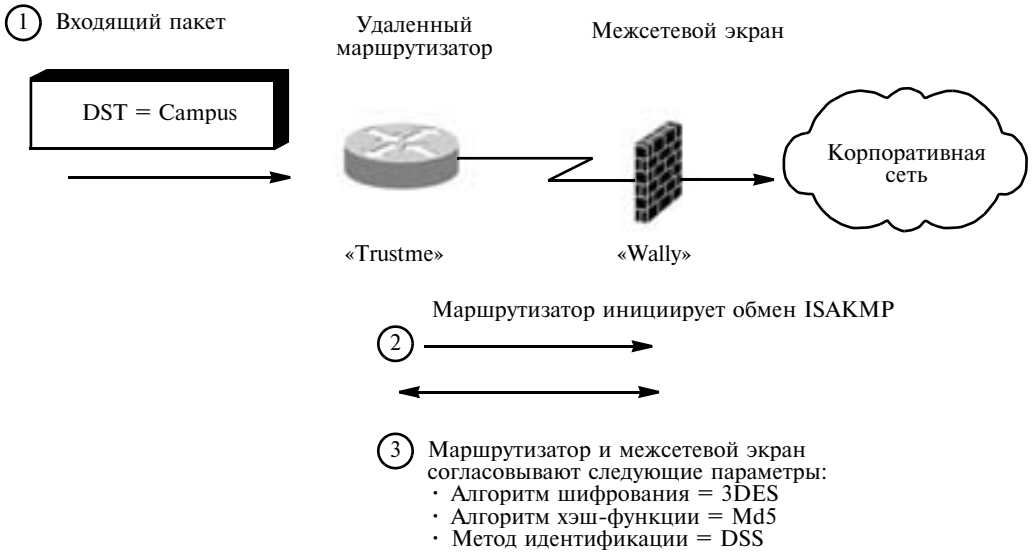



**Рис. 8.4. Согласование SA через ISAKMP**

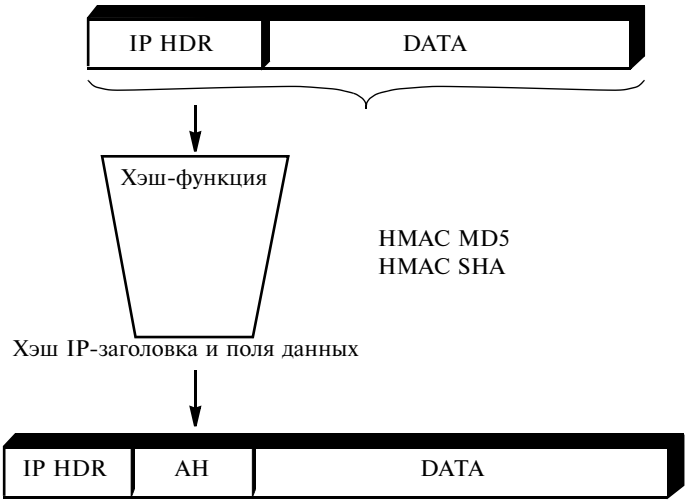



**Рис. 8.5. Создание нового аутентификационного заголовка IP**

После согласования SA принимается решение о том, следует ли использовать средства аутентификации, конфиденциальности и целостности данных или ограничиться только аутентификацией. Если использоваться будут только средства аутентификации, текущий стандарт предполагает применение хэш-функции, а точнее алгоритма не ниже MD5 с 128-разрядными ключами. Заголовок пакета и данные пропускаются через хэш-функцию, и результаты этого вычисления вводятся в специальное поле заголовка AH, как показано на рис. 8.5.

Новый пакет с аутентификационным заголовком, расположенным между заголовком IP и данными, отправляется через маршрутизатор в пункт назначения. Когда этот пакет попадает на межсетевой экран, последний проверяет его аутентичность, вычисляя хэш с помощью хэш-функции, указанной в SA. Обе стороны должны использовать одни и те же хэш-функции. Как показано на рис. 8.6, межсетевой экран сравнивает вычисленный им хэш с параметрами, указанными в соответствующем поле AH. Если эти величины совпадают, аутентичность и целостность данных считаются доказанными (если пакет передан из удаленной точки и при передаче не был искажен ни один бит).

Отметим, что вставка заголовка AH расширяет пакет и поэтому для данного пакета может потребоваться фрагментация, которая производится после заголовка AH для исходящих пакетов и перед ним для входящих пакетов.

Если помимо всего сказанного стороны пожелают использовать средства поддержки конфиденциальности, SA указывает, что весь трафик, поступающий из удаленного маршрутизатора на межсетевой экран предприятия, должен аутентифицироваться и шифроваться. В противном случае межсетевой экран его не пропустит. ESP поддерживает аутентификацию, целостность и конфиденциальность данных и работает в двух режимах: туннельном и транспортном, как показано на рис. 8.7 и 8.8.

В туннельном режиме вся датаграмма IP, заголовок IP и данные встраиваются в заголовок ESP. В транспортном режиме шифруются только данные, а заголовок IP передается в нешифрованном виде. Современные стандарты требуют использования DES в режиме цепочки шифрованных блоков (CBC).

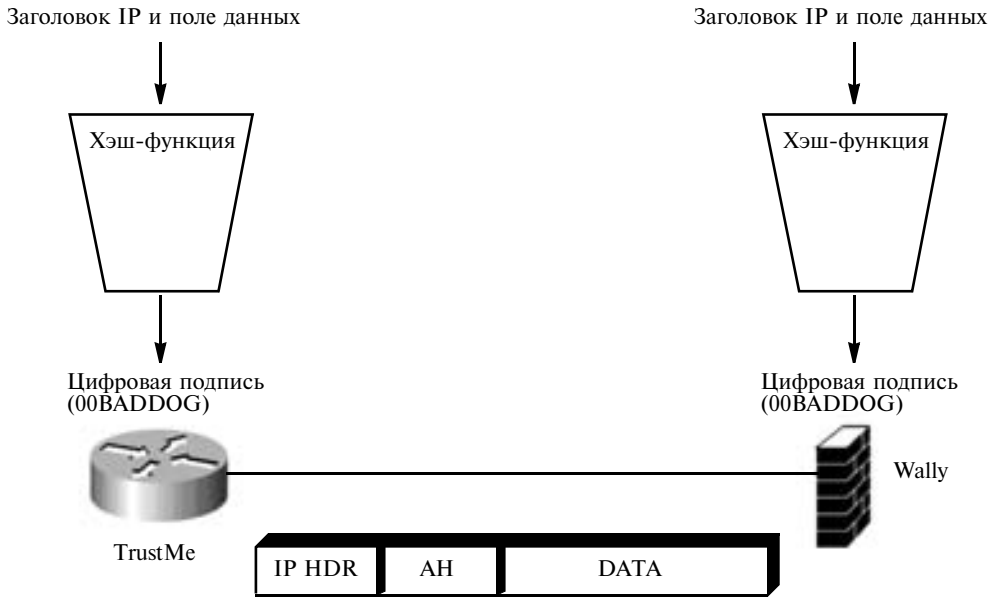



**Рис. 8.6. Проверка аутентичности и целостности данных**

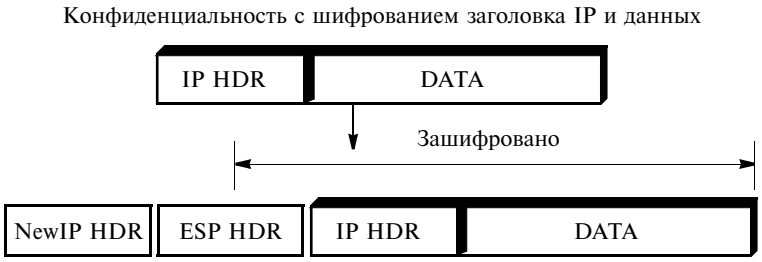



**Рис. 8.7. Туннельный режим ESP**

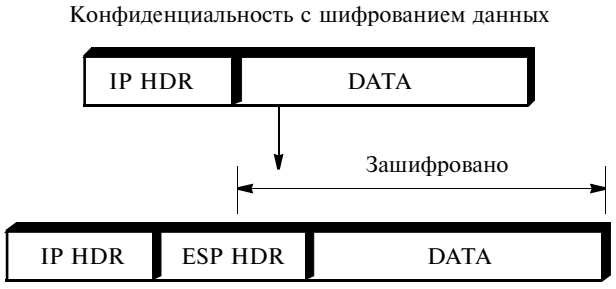



**Рис. 8.8. Транспортный режим ESP**

Отметим, что вставка заголовка AH расширяет пакет и поэтому для данного пакета может потребоваться фрагментация, которая производится после ESP для исходящих пакетов и перед ESP для входящих пакетов.

### 8.5.5. Выбор протокола защиты данных

Решение об использовании протоколов AH, ESP или их комбинации зависит от требований к защите IPSec. Включение в проект IPSec протокола AH обеспечит выполнение следующих задач:

1. *Защита от изменения всего пакета целиком.* Для защиты пакета от изменений при передаче протокол AH осуществляет цифровую подпись всего пакета, включая исходный заголовок IP. Протокол AH используется для подписи пакета, когда требуется защитить весь пакет от попыток изменить заголовок IP и маршрут пакета в сети.

2. *Взаимная аутентификация клиента и сервера.* В случае AH IPSec требует взаимной аутентификации компьютеров, участвующих в обмене данными. Аутентификация осуществляется между компьютерами, а не между работающими на них пользователями.

3. *Ограничение соединений в рамках проекта только полномочными компьютерами.* AH требует взаимной аутентификации компьютеров, обменивающихся данными. Если компьютеры не могут согласовать SA, соединения не будут установлены. AH обеспечивает установление соединения только между авторизованными компьютерами.

Использование в проекте IPSec протокола ESP обеспечивает выполнение следующих задач:

1. *Защита полезных данных приложения от просмотра во время передачи.* Пакеты ESP шифруют исходные полезные данные, не позволяя просмотреть содержимое пакета при передаче по сети.

2. *Защита заголовка TCP/UDP и данных приложения от изменения во время передачи.* Протокол ESP применяет к пакету данных цифровую подпись, но не обеспечивает защиту всего пакета от изменения. От изменения защищены только заголовок ESP, заголовок TCP/UDP, данные приложения и трейлер ESP.

Если же надо шифровать данные и защитить весь пакет от изменения, следует применять AH и ESP одновременно. Для обеспечения общей защиты передаваемых данных можно согласовать SA. Для этого требуются оба протокола.

> *Для обеспечения общей защиты передаваемых данных можно согласовать SA, требующее наличия протоколов AH и ESP.*

### 8.5.6. Планирование режимов IPSec

IPSec можно использовать в одном из двух режимов: транспортном и туннельном. Иногда требуется обеспечить защиту IPSec на всем протяжении пути от клиента до сервера назначения. Этот режим называется транспортным (рис. 8.9).

Защита пересылаемых данных осуществляется с помощью протоколов AH, ESP или обоих одновременно. Данные защищаются на всем пути между двумя компьютерами.

При использовании туннельного режима IPSec (рис. 8.10) защита данных осуществляется только между двумя определенными точками туннеля или шлюзами. Туннельный режим IPSec обеспечивает защиту данных, пересылаемых между шлюзами.


**Рис. 8.9. Транспортный режим IPSec**

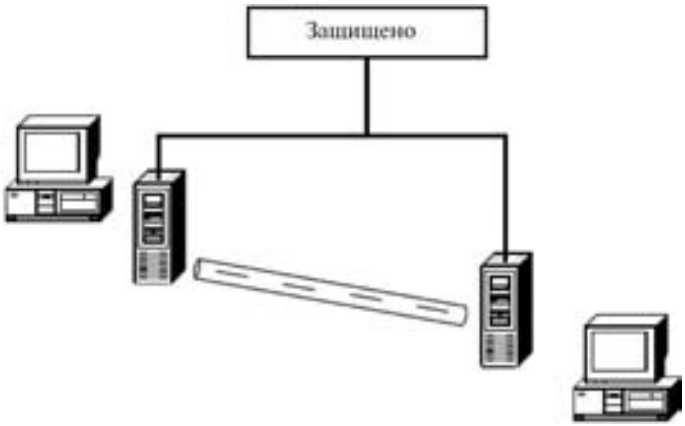



**Рис. 8.10. Туннельный режим IPSec**

От клиента к серверу данные до достижения начального шлюза пересылаются в незащищенном виде. Затем при передаче пакетов в сеть назначения к ним применяются определенные в SA протоколы AH или ESP. На принимающем шлюзе происходит процесс дешифрования и верификации данных. В конце процесса данные в виде открытого текста передаются на целевой компьютер.

Обычно туннельный режим применяется, когда пакеты должны пройти через открытую или незащищенную сеть. Туннельный режим подходит для локальных сетей, в которых нет необходимости защищать данные.

Пакеты туннельного режима IPSec отличаются от пакетов транспортного режима тем, что при передаче между шлюзами к пакету добавляется новый заголовок IP. В него вставляется AH-заголовок между новым и исходным заголовками IP. Поля, включенные в заголовок аутентификации, не различаются.

Аналогично пакет туннельного режима ESP отличается от пакета транспортного режима ESP расположением заголовка ESP. Как и в пакете туннельного режима AH, заголовок ESP помещается между новым и старым заголовками IP.

Поля, включенные в ESP-заголовок в туннельном и транспортном режимах, не различаются. Единственное отличие — местоположение ESP-заголовка в пакете туннельного режима и защите информации исходного IP-заголовка.

*Пакеты туннельного режима IPSec отличаются от пакетов транспортного режима тем, что при передаче между шлюзами к пакету добавляется новый заголовок IP.*

### 8.5.7. Аутентификация при использовании протокола IPSec

Перед согласованием SA участники IPSec-соединения должны взаимно аутентифицироваться одним из трех способов:

1. **Kerberos.** Стандартный механизм проверки подлинности в Windows 2000/2003 обеспечивает стойкую аутентификацию и легко настраивается, так как все компьютеры с Windows 2000/2003 осуществляют аутентификацию с аналогичными компьютерами в лесу, не требуя дополнительной настройки. Но Kerberos не годится для аутентификации между лесами.

2. **Сертификаты.** Для аутентификации участников сети на основе сертификатов в сеансе IPSec. Для этого сертификаты должны быть выданы центром сертификации (CA), которому доверяют оба компьютера. Сертификаты обеспечивают надежную аутентификацию для компьютеров в разных сетях. Компьютеры должны получить сертификаты для аутентификации до начала согласования SA IPSec.

3. **Общие секретные ключи.** Общие ключи представляют собой текстовые строки, введенные на обоих компьютерах для подтверждения их подлинности. Использование общих ключей возможно, когда нельзя применять Kerberos или сертификаты, или при тестировании фильтров IPSec перед внедрением параметров IPSec в сети.

### 8.5.8. Алгоритмы шифрования и проверки целостности IPSec

Свойства фильтров IPSec настраиваются для определения алгоритмов IPSec, применяемых при согласовании безопасности. Для потоков данных, защищенных протоколами AH и ESP, можно использовать разные алгоритмы.

Если требуется защита AH, в качестве алгоритма проверки целостности можно использовать алгоритмы Message Digest v5 (MD5) или Secure Hash Algorithm v1 (SHA1). Если требуется шифрование ESP, используется алгоритм цифровой подписи MD5 или SHAI и алгоритм шифрования Data Encryption Standard (DES) или Triple DES (3DES). Алгоритм проверки целостности SHAI считается сильнее MD5, а алгоритм шифрования 3DES — сильнее DES.

### 8.5.9. Фильтры протокола IPSec

Для идентификации протоколов, которые необходимо защитить средствами AH или ESP, надо определить IPSec-фильтры. Для идентификации протокола используются следующие характеристики:

- *IP-адрес источника.* Может быть конкретным IP-адресом, IP-адресом определенной подсети или любым адресом.

- *IP-адрес назначения.* Может быть конкретным IP-адресом, IP-адресом определенной подсети или любым адресом.
- *Тип протокола.* Это идентификатор протокола или используемый транспортный протокол. Например, PPTP использует пакеты GRE, которые определяются по их идентификатору протокола (47). С другой стороны, Telnet использует в качестве транспортного протокола TCP, поэтому в IPSec-фильтре для Telnet нужно указывать тип протокола как TCP.
- *Порт источника.* Если используются протоколы TCP или UDP, для защищенного соединения можно определить порт источника. В зависимости от протокола порт источника устанавливается для конкретного или произвольного порта. Большинство протоколов в качестве порта источника используют случайный порт.
- *Порт назначения.* Если применяется TCP или UDP, то для получения данных служит конкретный порт на сервере. Например, Telnet настраивает сервер для прослушивания соединений на TCP-порте 23.

После определения защищаемых протоколов следует задать действия, предпринимаемые в случае соответствия принимаемых или получаемых пакетов фильтру IPSec. Действия фильтра IPSec могут быть следующие:

- **Разрешить (Permit)** передачу пакетов без защиты IPSec. Например, протокол SNMP включает поддержку устройств, не совместимых с IPSec. Включение IPSec для протокола SNMP может помешать этим устройствам управлять сетью. Для разрешения передачи пакетов SNMP без защиты IPSec в сетях с высокой безопасностью можно создать специальный фильтр IPSec для SNMP.
- **Блокировать (Block)** — запрещает существование в сети протокола, соответствующего IPSec-фильтру. Все пакеты, подпадающие под условия этого фильтра, отбрасываются.
- **Согласовать безопасность (Negotiate Security)** — позволяет администратору определить уровень шифрования и алгоритм хэширования для защиты трафика, соответствующего фильтру IPSec.

Если для установления виртуальной частной сети применяется протокол Layer Two Tunneling Protocol (L2TP), определять фильтры IPSec для включения защиты IPSec в сети Windows 2000/2003 не нужно. Windows 2000/2003 автоматически включает защиту IPSec для туннеля L2TP. В этом случае нет необходимости определять фильтры IPSec, так как Windows 2000 средствами ESP автоматически защищает данные, пересылаемые по туннелю L2TP.

IPSec не может защитить:

- *Широковещательные IP-адреса.* Фильтры IPSec можно определять только для отдельных получателей пакетов. Для пакетов, предназначенных нескольким получателям, определить IPSec нельзя, так как SA должны быть установлены между парами компьютеров.
- *Групповые адреса.* Как и в случае с широковещательными сообщениями, нельзя обеспечить защиту пакетов, отправляемых нескольким получателям. Групповые адреса включают все IP-адреса класса D (224.0.0.0-237.255.255.255).
- *Протокол* Resource ReSerVation Protocol (RSVP). Для запроса части пропускной способности сети компьютер использует протокол RSVP (идентификатор протокола 46). Служба IPSec может защитить протокол, для которого RSVP запрашивает качество обслуживания, но не сами RSVP-пакеты этого запроса.
- *Протокол* Kerberos. Протокол Kerberos используется для аутентификации двух компьютеров, участвующих в обмене данными IPSec. Защита аутентификации Kerberos осуществляется протоколом Kerberos и не требует защиты IPSec.

- *Протокол* Internet Key Exchange (IKE). Протокол IKE применяется для согласования SA между двумя компьютерами, участвующими в передаче данных IPSec. Процесс согласования IPSec шифровать нельзя. Согласование осуществляется с помощью пакетов с открытым текстом. При этом определяется порядок защиты следующих пакетов.

Чтобы обеспечить соответствие фильтров потребностям предприятия, при их разработке нужно учитывать следующее:

- Для одного компьютера можно назначать только одну политику IPSec. Если фильтрованию нужно подвергнуть несколько разных протоколов, необходимо создать перечень фильтров, включающий все протоколы, и вставить его в список фильтров.
- В среде Windows политики IPSec определяются не для пользователей, а для компьютеров. Фильтры IPSec определяются только для компьютеров в сети. При этом не имеет значения, кто из пользователей работает на компьютере.
- Для выбора правильного фильтра нужно задать требования к протоколу. Необходимо определить следующие атрибуты для каждого фильтра:
  1) IP-адрес источника;
  2) порт источника;
  3) IP-адрес назначения;
  4) порт назначения;
  5) тип протокола.
- Идентифицировать шифрованный трафик IPSec при прохождении через межсетевой экран нельзя. Если к пакету был применен протокол ESP, определить, какой протокол зашифрован в пакете, невозможно. Это может привести к тому, что через межсетевой экран сможет пройти нежелательный трафик, если межсетевой экран настроен на IKE-пакеты (UDP-порт 500) и ESP-пакеты (идентификатор протокола 50). Так как данные в пакете зашифрованы, межсетевой экран не может определить, какой исходный протокол был защищен IPSec.
- Если определено несколько фильтров, первым вычисляется наиболее конкретный из них, а наименее конкретный — последним. Порядок вывода на экран при этом не имеет значения.
- В случае транспортного режима IPSec всегда следует применять пакетные фильтры с отражением. Отражение обеспечивает шифрование ответных пакетов при их передаче обратно источнику. В ответных пакетах информация об источнике и назначении будет обращена. Отражение правил обеспечивает шифрование ответов.
- При определении соединений туннельного режима IPSec для каждого направления следует создавать фильтр IPSec. Для туннельного режима IPSec нельзя использовать отраженные пакетные фильтры, так как для заголовка трафика на каждом направлении должны быть заданы разные конечные точки туннеля.

### 8.5.10. Преимущества протокола IPSec

К преимуществам поддержки безопасности на сетевом уровне с помощью IPSec следует отнести:

- поддержку совершенно немодифицированных конечных систем, хотя в этом случае шифрование нельзя назвать в полном смысле слова сквозным (end-to-end);
- частичную поддержку виртуальных частных сетей (VPN) в незащищенных сетях;
- поддержку других транспортных протоколов, а не только TCP (например, UDP);

- защиту заголовков транспортного уровня от перехвата и, следовательно, более надежную защиту от анализа трафика;
- при использовании AH и средств обнаружения повторяющихся операций обеспечивается защита от атак типа «отказ от обслуживания», основанных на «затоплении» систем ненужной информацией (например, от атак TCP SYN).

## 8.6. Протоколы защищенного взаимодействия и аутентификации для корпоративных беспроводных локальных сетей

На сегодняшний день лидеры рынка предлагают ряд технологий, которые существенно повышают безопасность корпоративных сетей WLAN, предоставляющую следующие возможности для клиентских устройств WLAN:

- Поддержка стандарта IEEE 802.11i.
- Поддержка сертификатов безопасности Wi-Fi Alliance — Wi-Fi Protected Access (WPA) и Wi-Fi Protected Access 2 (WPA2).
- Двусторонняя аутентификация и управление динамическими ключами шифрования благодаря поддержке IEEE 802.1X.
- Шифрование данных с помощью алгоритмов Advanced Encryption Standard (AES) или Temporal Key Integrity Protocol (TKIP).
- Поддержка типов аутентификации 802.1X, клиентских устройств и клиентских операционных систем.
- Подавление активных и пассивных сетевых атак.
- Интеграция с решениями Network Admission Control (NAC) компании Cisco Systems.
- Предотвращение сетевых вторжений (Intrusion Prevention System, IPS) и слежения за перемещением абонента — прозрачное представление сети в реальном времени.
- Конвергенция безопасности внутренней и внешней сетей Wi-Fi.

Главным постулатом безопасности любой сети, не только беспроводной, является управление доступом и конфиденциальностью. Одним из надежных способов управления доступом к WLAN является аутентификация, позволяющая предотвратить доступ несанкционированных пользователей к передаче данных через точки доступа. Действенные меры управления доступом к WLAN помогают определить круг разрешенных клиентских станций и связать их только с доверенными точками доступа, исключая несанкционированные или опасные точки доступа. В настоящее время компании, использующие сети WLAN, внедряют четыре отдельных решения для безопасности WLAN и управления доступом и конфиденциальностью:

- открытый доступ;
- базовая безопасность;
- повышенная безопасность;
- безопасность удаленного доступа.

### *8.6.1. Открытый доступ*

Все продукты для беспроводных локальных сетей, соответствующие спецификациям Wi-Fi, например продукты серии Cisco Aironet, поставляются для работы в режиме открытого доступа с выключенными функциями безопасности. Открытый доступ или отсут-

ствие безопасности могут устраивать и удовлетворять требования общественных хот-спотов, таких как кофейни, университетские городки, аэропорты или другие общественные места, однако для предприятий этот вариант не подходит. Функции безопасности должны быть включены на беспроводных устройствах в процессе их установки. Некоторые компании, однако, не включают функции безопасности сетей WLAN, и тем самым повышают уровень риска для своих сетей.

### 8.6.2. Базовая безопасность

Базовая безопасность заключается в использовании идентификаторов сети SSID (Service Set Identifier), открытой аутентификации, аутентификации с помощью общих и статических WEP-ключей и аутентификации по MAC-адресу. С помощью этой комбинации можно настроить элементарные средства управления доступом и конфиденциальностью, однако каждый отдельный элемент такой защиты может быть взломан.

Идентификатор SSID — общее имя сети для устройств в подсистеме WLAN — служит для логического обособления данной подсистемы. Он предотвращает доступ любого клиентского устройства, не имеющего SSID. Однако по умолчанию точка доступа передает в эфир среди своих сигналов и свой SSID. Даже если отключить передачу в эфир SSID, взломщик или хакер может обнаружить нужный SSID с помощью скрытого мониторинга сети. Стандарт 802.11, группа спецификаций для сетей WLAN, выработанная IEEE, поддерживает два средства аутентификации клиента: открытую аутентификацию и аутентификацию с помощью открытых ключей.

*Идентификатор SSID — общее имя сети для устройств в подсистеме WLAN, служит для логического обособления данной подсистемы.*

Открытая аутентификация лишь ненамного отличается от предоставления правильного идентификатора SSID. При аутентификации с помощью открытых ключей точка доступа посылает на клиентское устройство тестовый текстовый пакет, который клиент должен зашифровать правильным WEP-ключом и вернуть на точку доступа. Без правильного ключа аутентификация будет прервана и клиент не будет допущен в группу пользователей точки доступа.

Аутентификация с помощью общих ключей считается ненадежной, поскольку взломщик, получивший в свое распоряжение начальное тестовое текстовое сообщение и это же сообщение, зашифрованное WEP-ключом, может дешифровать сам WEP-ключ. При открытой аутентификации, даже если клиент проходит аутентификацию и получает доступ в группу пользователей точки доступа, WEP-защита не позволяет клиенту передавать данные с этой точки доступа без правильного WEP-ключа.

WEP-ключи могут состоять из 40 или 128 бит и обычно статически определяются сетевым администратором на точке доступа и каждом клиенте, передающем данные через эту точку доступа. При использовании статических WEP-ключей сетевой администратор должен потратить много времени на ввод одинаковых ключей в каждое устройство сети WLAN.

Если устройство, использующее статические WEP-ключи, потеряно или украдено, обладатель пропавшего устройства может получить доступ к сети WLAN. Администратор не сможет определить, что в сеть проник несанкционированный пользователь до тех пор, пока не будет известно о пропаже. После этого администратор должен сменить WEP-ключ

на каждом устройстве, использующем тот же статический WEP-ключ, что и пропавшее устройство. В сети крупного предприятия, включающей сотни или даже тысячи пользователей, это может оказаться невыполнимым. Если же статический WEP-ключ был дешифрован с помощью такого инструмента, как AirSnort, администратор никак не узнает о том, что ключ был взломан несанкционированным пользователем.

Некоторые поставщики решений WLAN поддерживают аутентификацию на базе физического или MAC-адреса, клиентской сетевой карты (NIC). Точка доступа позволит клиенту ассоциироваться с точкой доступа только в случае, если MAC-адрес клиента соответствует одному из адресов в таблице аутентификации, используемой точкой доступа. Однако аутентификация по MAC-адресу не является адекватной мерой безопасности, поскольку MAC-адрес можно подделать, а сетевую карту — потерять или украсть.

Другая форма доступной на сегодняшний день базовой безопасности — это WPA или WPA2 с помощью общих ключей (Pre-Shared Key, PSK). Общий ключ проверяет пользователей с помощью пароля или кода идентификации (также называемого «фраза—пароль») как на клиентской станции, так и на точке доступа. Клиент может получить доступ к сети только в том случае, если пароль клиента соответствует паролю точки доступа. Общий ключ также предоставляет данные для генерации ключа шифрования, который используется алгоритмами TKIP (Tenporal Integrity Protocol) или AES для каждого пакета передаваемых данных. Являясь более защищенным, чем статический WEP-ключ, общий ключ также хранится на клиентской станции и может быть взломан, если клиентская станция потеряна или украдена. Рекомендуется использовать общую фразу-пароль, включающую разнообразные буквы, цифры и не алфавитно-цифровые символы.

**Выводы.** Базовая безопасность сетей WLAN, основанная на комбинации SSID, открытой аутентификации, статических WEP-ключей, MAC-аутентификации и общих ключей WPA/WPA2, является достаточной только для очень небольших компаний или тех, которые не доверяют жизненно важные данные своим сетям WLAN. Всем прочим организациям рекомендуется вкладывать средства в надежные решения безопасности сетей WLAN класса предприятия.

*Некоторые поставщики решений WLAN поддерживают аутентификацию на базе физического адреса или MAC-адреса, клиентской сетевой карты (NIC).*

### 8.6.3. Повышенная безопасность

Повышенный уровень безопасности рекомендуется для тех компаний, которым требуется безопасность и защищенность класса предприятия. При этом для обеспечения безопасности повышенного уровня, полностью поддерживаемого WPA и WPA2 с двусторонней аутентификацией 802.1x и шифрования алгоритмами TKIP и AES, целесообразно реализовать следующие возможности:

- 802.1X для двусторонней аутентификации и динамических ключей шифрования для каждого пользователя и каждой сессии;
- TKIP для расширения шифрования на базе RC4, например, для хэширования ключей (для каждого пакета), проверки целостности сообщения (MIC), изменений вектора инициализации (IV) и ротации широковещательных ключей;
- AES для шифрования данных максимальной защищенности;
- интеграция с Cisco Self-Defending Network и NAC;

- возможности системы предотвращения сетевых вторжений (Intrusion Prevention System, IPS) и слежения за перемещением абонента — прозрачное представление сети в реальном времени.

Решение по обеспечению повышенного уровня безопасности, как правило, должно предоставлять:

- Безопасное подключение к сетям WLAN — динамические ключи шифрования должны автоматически изменяться в соответствии с настройками для обеспечения конфиденциальности передаваемых данных.

  — Шифрование WPA-TKIP должно расширяться такими функциями, как проверка целостности сообщения (MIC), хэширование ключей (для каждого пакета), изменение вектора инициализации (IV) и ротация широковещательных ключей.

  — WPA2-AES, «золотой эталон» шифрования данных.
- Доверительные отношения и идентификацию в сетях WLAN. Надежное управление доступом к WLAN должно обеспечивать подключение уполномоченных клиентов только к доверенным точкам доступа и исключать неавторизованные точки доступа. Для этого должна применяться двусторонняя аутентификация каждого пользователя и каждой сессии с применением IEEE 802.1X, разнообразных типов расширяемого протокола аутентификации (Extensible Authentication Protocol, EAP) и сервера аутентификации RADIUS (Remote Authentication Dial-In User Service) или сервера аутентификации, авторизации и учета AAA (Authentication, Authorization and Accounting).

  — Поддержка типов аутентификации 802.1X, клиентских устройств и клиентских операционных систем.

  — Поддержка записей биллинга протокола RADIUS для всех попыток аутентификации.
- Защиту от атак на сети WLAN. Обнаружение несанкционированного доступа, сетевых атак и несанкционированных точек доступа с помощью надежных средств предотвращения вторжений IPS, WLAN NAC и расширенных сервисов обнаружения местоположения. Необходимо использовать средства предотвращения сетевых вторжений (IPS) класса предприятия, позволяющие непрерывно сканировать радиодиапазон, обнаруживать несанкционированные точки доступа и прочие несанкционированные события.

Протокол NAC Network Admission Control был специально разработан для адекватной защиты всех проводных и беспроводных оконечных устройств (таких как персональные компьютеры, ноутбуки, серверы и КПК), обращающихся к сетевым ресурсам, от угроз безопасности. Использование протокола NAC позволяет организациям анализировать и контролировать все устройства, подключающиеся к сети.

### 8.6.4. Поддержка WPA и WPA2

WPA был представлен Wi-Fi Alliance в 2003 г., а WPA2 был представлен Wi-Fi Alliance в 2004 г. Все продукты, сертифицированные Wi-Fi на соответствие требованиям WPA2, обязательно взаимодействуют с продуктами, сертифицированными Wi-Fi на соответствие требованиям WPA. Cisco Unified Wireless Network включает поддержку сертифицированных Альянсом Wi-Fi механизмов WPA и WPA2.

WPA и WPA2 предоставляют конечным пользователям и сетевым администраторам высокий уровень уверенности в том, что их данные останутся конфиденциальными, а доступ к их сетям будет предоставляться только санкционированным пользователям. Оба стан-

*Таблица 8.1*

**Сравнение типов режимов WPA и WPA2**

| | *WPA* | *WPA2* |
|---|---|---|
| Корпоративный режим (коммерческие, правительственные, образовательные структуры) | • Аутентификация: IEEE 802.1X/EAP<br>• Шифрование: TKIP/MIC | • Аутентификация: IEEE 802.1X/EAP<br>• Шифрование: AES-CCMP |
| Персональный режим (небольшие компании, домашние и персональные системы) | • Аутентификация: PSK<br>• Шифрование: TKIP/MIC | • Аутентификация: PSK<br>• Шифрование: AES-CCMP |

дарта обладают персональным и корпоративным режимами работы, удовлетворяющими отдельным требованиям этих двух сегментов рынка. Корпоративный режим использует для аутентификации IEEE 802.1X и EAP, а персональный режим — общие ключи (PSK). Не рекомендуется применять персональный режим для коммерческих или государственных решений из-за использования общих PSK-ключей при аутентификации пользователей. PSK-ключи не считаются достаточно надежной мерой для внедрения на предприятии.

WPA позволяет закрыть все известные уязвимости WEP исходного стандарта безопасности IEEE 802.11. Представляет собой быстрое решение для обеспечения безопасности сетей WLAN как для предприятий, так и для небольших компаний или домашних систем. WPA использует алгоритм шифрования TKIP.

WPA2 — это следующее поколение безопасности Wi-Fi. Он представляет собой предложенный Wi-Fi Alliance вариант ратифицированного стандарта IEEE 802.11i. В его состав входит рекомендованный Национальным институтом стандартов и технологий (NIST) алгоритм шифрования AES, использующий протокол CCMP (Counter Mode with Cipher Block Chaining Message Authentication Code Protocol). WPA2 обеспечивает соответствие требованиям правительственного стандарта FIPS 140-2 (табл. 8.1).

*WPA позволяет закрыть все известные уязвимости WEP исходного стандарта безопасности IEEE 802.11, WPA2 представляет собой предложенный Wi-Fi Alliance вариант ратифицированного стандарта IEEE 802.11i.*

### *8.6.5. IEEE 802.1X-аутентификация и протокол EAP (Extensible Authentication Protocol)*

IEEE приняла 802.1X в качестве стандарта аутентификации для проводных и беспроводных сетей. 802.1X поддерживается корпоративными режимами WPA и WPA2. 802.1X предоставляет сетям WLAN средства мощной двусторонней аутентификации между клиентом и сервером аутентификации. В дополнение к этому, 802.1X предоставляет динамические ключи шифрования для каждого пользователя и каждой сессии, избавляя таким образом администраторов от необходимости обслуживания ненадежных статических ключей шифрования.

Средствами 802.1X конфиденциальная информация, используемая для аутентификации, такая, как пароли для входа, никогда не передается в открытом виде без шифрования по беспроводным сетям. 802.1X обеспечивает надежную аутентификацию для беспроводных локальных сетей, однако для шифрования дополнительно к 802.1X необходимы алго-

ритмы TKIP и AES, поскольку стандартное WEP-шифрование стандарта 802.11 уязвимо перед сетевыми атаками.

Существует несколько типов 802.1X-аутентификации, предоставляющих различные подходы к аутентификации, однако, опирающихся на единую структуру, и протокол EAP с целью обеспечения передачи данных между клиентом и точкой доступа. К ним относятся следующие: Cisco LEAP, EAP-Flexible Authentication via Secure Tunneling (EAP-FAST), EAP-Transport Layer Security (EAP-TLS), Protected Extensible Authentication Protocol (PEAP), EAP-Tunneled TLS (EAP-TTLS) и EAP-Subscriber Identity Module (EAP-SIM).

Для выбора наиболее подходящего типа EAP-аутентификации для развертывания 802.1X. рекомендуется провести анализ собственных сетей и систем безопасности. При выборе типа EAP следует обращать внимание на механизм обеспечения безопасности, используемый для передачи данных для аутентификации, базу данных аутентификации пользователей, используемые клиентами операционные системы, доступные клиентам способы обращения, тип необходимого входа пользователей в сеть, а также наличие серверов RADIUS или AAA.

Разница между уровнями безопасности зависит от типа управляемости EAP, поддерживаемых операционных систем, клиентских устройств, преимуществ клиентского программного обеспечения и передачи аутентификационных данных, требований сертификации, простоты использования и поддержки устройств инфраструктурой WLAN. Кроме того, возможно применение нескольких типов EAP в рамках сети — для поддержки специфического типа аутентификации, клиентского устройства или потребностей конечных пользователей.

Для аутентификации по 802.1X может быть использован широкий ассортимент серверов RADIUS, таких, например, как Cisco Secure Access Control Server (ACS) и AAA RADIUS-серверов различных производителей, например, Interlink Networks (AAA RADIUS).

Применение одного из типов 802.1X-аутентификации, позволяющего аутентифицировать клиентскую станцию с помощью вводимых пользователем данных, а не физических атрибутов клиентского устройства, дает возможность понизить риск, связанный с потерей устройства или его сетевой карты WLAN. 802.1X предоставляет и другие преимущества, включая снижение опасности появления угрозы «человек посередине» («man-in-the-middle») при аутентификации, централизованное управление шифрованием ключей с ротацией ключей на базе установленной политики, а также защиту от атак, осуществляемых методом перебора ключей (brute-force) (рис. 8.11).

*Применение одного из типов 802.1X-аутентификации, позволяющего аутентифицировать клиентскую станцию с помощью вводимых пользователем данных, дает возможность понизить риск, связанный с потерей устройства или его сетевой карты WLAN.*

Другим преимуществом 802.1X-аутентификации является *централизованное управление группами пользователей сетей WLAN*, включающее ротацию ключей на базе политик, динамическое распределение ключей, динамическое назначение VLAN и запрет SSID. Эти функции осуществляют ротацию ключей шифрования. Они также позволяют назначить пользователям определенные VLAN-сети для гарантии того, что пользователи имеют доступ только к определенным ресурсам.

После удачного проведения двусторонней аутентификации и клиент, и RADIUS-сервер генерируют одинаковые ключи шифрования, используемые для шифрования всех пере-

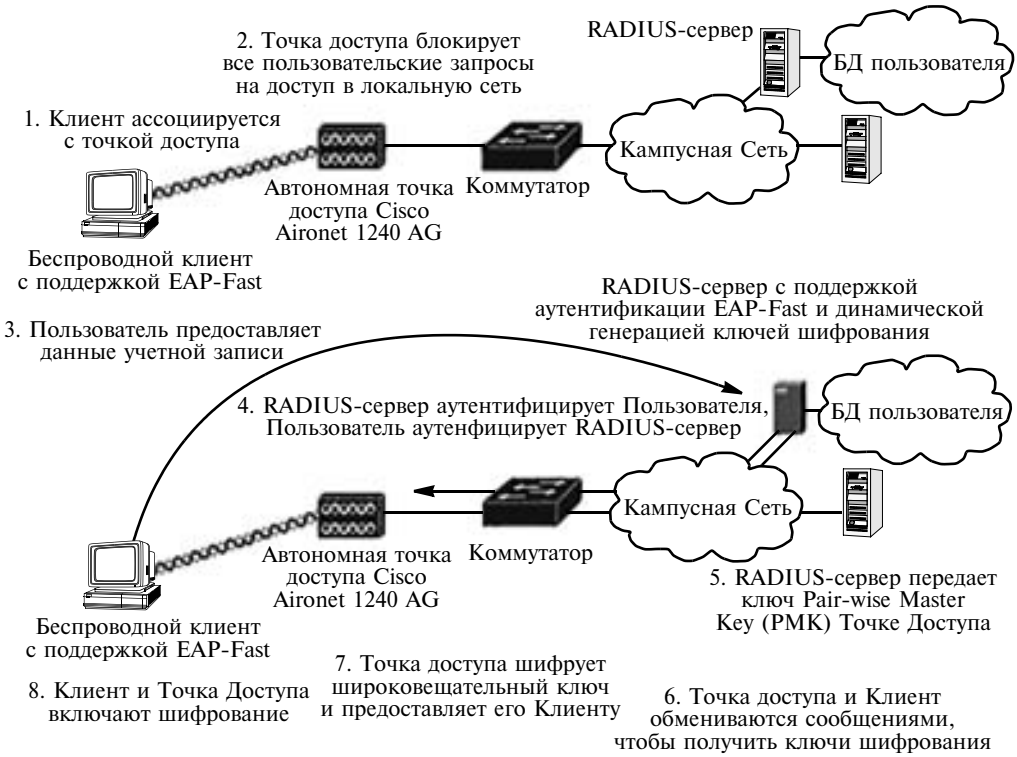



**Рис. 8.11. Применение EAP-FAST одного из типов 802.1X для обеспечения безопасности класса предприятия**

даваемых данных. По защищенному каналу проводной локальной сети RADIUS-сервер посылает этот ключ автономной точке доступа или беспроводному контроллеру локальной сети, которые сохраняют его для клиента. В результате мы имеем ключи шифрования для каждого пользователя и каждой сессии, а продолжительность сессии определяется политикой на RADIUS-сервере. После окончания сессии или при переходе клиента с одной точки доступа на другую происходит процесс повторной аутентификации, в результате которой генерируется новый ключ сессии. Повторная аутентификация проходит незаметно для пользователя.

Параметры имени/идентификатора VLAN и параметры SSID передаются на автономную точку доступа или контроллер беспроводной локальной сети вместе с ключами шифрования и таймером повторной аутентификации. После получения автономной точкой доступа или контроллером беспроводной локальной сети имени/идентификатора VLAN, назначаемого для указанного пользователя, он указывает данного пользователя в метке указанного имени/идентификатора VLAN. В случае, если точке доступа или контроллеру также передается список разрешенных SSID, точка доступа или контроллер пытаются обеспечить данного пользователя действительным SSID для доступа к сети WLAN. Если пользователь предоставляет SSID, не указанный в списке разрешенных SSID, точка доступа или контроллер беспроводной локальной сети удаляет ассоциацию с пользователем из сети WLAN.

*Снижение риска атак, осуществляемых методом перебора ключей (brute-force).* Традиционные схемы WLAN на основе статических ключей шифрования легко поддаются взлому сетевыми атаками, осуществляемыми методом перебора ключей. Атака представляет собой попытку взломщика получить ключ шифрования путем перебора. Для взлома стандартной 128-битовой WEP-защиты потребуется перебрать максимум 2104 разных ключей. Использование динамических ключей шифрования стандарта 802.1X, получаемых каждым пользователем для каждой сессии, делает атаку методом перебора ключей хоть и теоретически возможной, но крайне сложной для проведения и практически бесполезной.

### 8.6.6. WPA-шифрование, протокол целостности временных ключей Temporal Key Integrity Protocol

TKIP представляет собой следующее поколение стандарта обеспечения безопасности WEP. Как и WEP, TKIP использует метод шифрования, разработанный инженером Роном Райвестом и известный как алгоритм шифрования Ron's Code 4 (RC4). Однако TKIP улучшает WEP за счет ликвидации известных уязвимостей WEP и добавления таких функций, как хэширование ключа каждого пакета, MIC и ротация широковещательных ключей.

TKIP реализует RC4-кодирование потока 128-битовыми ключами для шифрования и 64-битовыми ключами для аутентификации. За счет шифрования данных ключом, который может быть использован только предписанным пользователем этих данных, TKIP позволяет гарантировать получение передаваемых данных в открытом виде только теми, для кого они предназначены. Шифрование TKIP приводит к 280 триллионам возможных комбинаций ключей для каждого отдельного пакета данных.

*TKIP реализует RC4-кодирование потока 128-битовыми ключами для шифрования и 64-битовыми ключами для аутентификации.*

Например, в рамках решения Cisco Unified Wireless Network реализованы алгоритмы Cisco TKIP и WPA TKIP для автономных точек доступа Cisco Aironet, устройств Cisco Aironet и совместимых с Cisco клиентских устройств для работы с беспроводной локальной сетью. Несмотря на то, что Cisco TKIP и WPA TKIP не могут взаимодействовать друг с другом, автономные точки доступа серии Cisco Aironet могут работать одновременно в этих режимах при использовании нескольких VLAN. Системным администраторам требуется выбрать один набор TKIP-алгоритмов для активации на клиентских устройствах предприятия, поскольку клиенты не могут поддерживать оба набора TKIP-алгоритмов одновременно. Cisco рекомендует по возможности использовать для клиентских устройств и точек доступа алгоритм WPA TKIP. Контроллеры беспроводной локальной сети Cisco и простые точки доступа Cisco Aironet поддерживают только WPA TKIP.

*Хэширование ключей для каждого пакета с целью снижения риска атак типа «Слабый вектор инициализации» (Weak IV).* При использовании WEP-ключа для шифрования (дешифрования) передаваемых данных каждый пакет включает вектор инициализации (IV), представляющий собой 24-битовое поле, меняющееся с каждым пакетом. Алгоритм обновления ключей TKIP RC4 генерирует вектор на базе основного WEP-ключа. Уязвимость в реализации WEP-алгоритма RC4 позволяет создавать «слабые» векторы, дающие возможность взлома основного ключа. С помощью таких инструментов, как AirSnort, взломщик может воспользоваться данной уязвимостью путем сбора пакетов, шифрованных одним ключом и подстановки слабых векторов инициализации для нахождения основного ключа.


**Рис. 8.12. Снижение риска атак для схемы беспроводной локальной сети с использованием WPA-802.1X EAP/TKIP**

TKIP содержит средства хэширования ключей или создания ключей для каждого пакета для снижения риска атак с использованием слабых векторов инициализации. При внедрении поддержки хэширования ключей как на точке доступа, так и на всех ассоциированных клиентских устройствах, отправитель данных хэширует базовый ключ с помощью вектора инициализации для создания нового ключа для каждого пакета. Обеспечивая шифрование каждого пакета своим ключом, хэширование ключа снимает вероятность определения WEP-ключа с помощью уязвимости векторов инициализации (рис. 8.12).

*Message Integrity Check, проверка целостности сообщения для защиты от активных сетевых атак*. Использование MIC позволяет избежать активных сетевых атак, нацеленных на поиск ключа шифрования, применяемого для шифрования перехваченных пакетов. При внедрении MIC как на точке доступа, так и на всех ассоциированных клиентских устройствах отправитель пакета данных добавляет несколько байтов (для проверки целостности сообщения) к пакету перед его шифрованием и отправкой. При получении пакета получатель дешифрует его и проверяет байты MIC. Если байты MIC пакета соответствуют расчетным данным (рассчитываемым из функции MIC), получатель принимает пакет; в противном случае получатель уничтожает пакет. С помощью MIC становится возможным отбрасывать пакеты, измененные злоумышленниками.

*Ротация широковещательных ключей*. TKIP позволяет сетевым администраторам осуществлять ротацию как присвоенных конкретному устройству, так и широковещательных ключей, используемых для шифрования широковещательных и мультивещательных сообщений. Сетевые администраторы могут конфигурировать политики ротации широковещательных ключей для точек доступа. Поскольку статический широковещательный ключ подвержен тем же атакам, что и присвоенные конкретному устройству или статические WEP-ключи, поддерживается ротация значений ключа для широковещательных ключей, позволяющая закрыть эту уязвимость.

## Контрольные вопросы

1. Какие возможности обеспечивает протокол SSL для безопасности связи?
2. Что включает в себя ассоциация безопасности?
3. Перечислите способы аутентификации при использовании протокола IPSec.
4. Какие протоколы IPSec защитить не может?
5. Преимущества протокола IPSec?

# Глава 9
# ПРИМЕНЕНИЕ АППАРАТНЫХ СРЕДСТВ АУТЕНТИФИКАЦИИ И ХРАНЕНИЯ КЛЮЧЕВОЙ ИНФОРМАЦИИ

## 9.1. Аппаратные средства защиты в современных PKI-решениях

По оценкам компании IDC примерно 74% финансовых потерь связано с проблемами так называемого «человеческого фактора». Для сравнения, потери от вирусных и хакерских атак составляют соответственно 4 и 2%.

Поэтому важнейшими приоритетными задачами обеспечения информационной безопасности (ИБ) является снижение риска, связанного с человеческим фактором (ненадежность паролей, сложность их выбора и смены, ошибки администрирования и т. п.).

Таким образом, нельзя построить защищенную систему, не обеспечив надежное решение проблемы «трех А»:

- аутентификации (ответа на вопрос «Ты кто и как ты можешь доказать, что ты — это ты?»);
- авторизации («Какие у тебя есть полномочия на доступ к информации и права на работу в системе?»);
- администрирования («Как безопасно управлять и централизованно администрировать всю информационную систему?»).

Практически все ведущие продавцы, разработчики операционных систем, систем ИБ, ERP-систем и бизнес-приложений (Microsoft, Novell, Linux, IBM, Oracle, Cisco, SAP, Check Point и др.) поддержали PKI и включили в состав своих продуктов поддержку современных средств двухфакторной аутентификации (смарт-карты, USB-ключи) для безопасного хранения закрытых ключей и удобной работы с цифровыми сертификатами, обеспечив тем самым надежное решение проблемы «трех А».

В данной главе приводится сравнение существующих аппаратных и программных средств защиты закрытых ключей пользователя, применяемых в современных PKI-решениях. Показывается, что использование программных контейнеров для хранения закрытых ключей пользователя в памяти компьютера, который сам по себе является потенциально *небезопасным,* или на отчуждаемом носителе является недостаточным для того, чтобы обеспечить должный уровень безопасности закрытого ключа пользователя. В связи с чем PKI-решения, предполагающие применение таких средств, заведомо уязвимы.

Единственным надежным средством, обеспечивающим сохранность в тайне закрытых ключей на всех этапах их жизненного цикла (генерация ключевой пары (закрытый ключ — открытый ключ), хранение, использование и уничтожение закрытого ключа), являются специализированные устройства, построенные на основе микросхемы смарт-карты. В них выполняются все криптографические операции с закрытым ключом, а сам закрытый ключ никогда не покидает устройство и не может быть из него извлечен.

В настоящее время только аппаратные решения — смарт-карты и USB-ключи на основе микросхемы смарт-карты, аппаратно реализующие криптографические алгоритмы в соответствии с национальными стандартами — в состоянии надежно защитить закрытый ключ пользователя даже при работе в небезопасных средах. Об этом следует обязательно помнить, проектируя архитектуру PKI-решений или планируя их внедрение.

## 9.2. Необходимость применения аппаратных средств аутентификации и хранения ключевой информации

В асимметричной криптографии, как известно, применяется два типа отличных друг от друга ключей. Один из них — открытый ключ — используется для того, чтобы выполнить «публичные операции» (например, шифрование, проверку и подтверждение подлинности цифровой подписи (ЭЦП)). Другой — закрытый ключ — используется для «закрытых операций» (например, дешифрование, генерация ЭЦП). Таким образом, все, что зашифровано с помощью публичного ключа, может быть расшифровано с помощью закрытого или секретного ключа, а подлинность ЭЦП, выработанной с помощью закрытого ключа, может быть проверена с помощью открытого ключа. Такая система позволяет не только избежать необходимости делиться секретной (ключевой) информацией с другими пользователями, но и обеспечить такое важное свойство этой информации, как неотказуемость пользователя от авторства, так как только владелец закрытого ключа в состоянии реализовать соответствующие процедуры.

Благодаря инфраструктуре открытых ключей (PKI), которая базируется на использовании сертификатов, определяющих владельцев закрытых ключей и их полномочия, мы имеем возможность соотнести публичные ключи с их владельцами. Это, безусловно, важно, так как при криптографическом преобразовании сообщения решающим фактором является использование публичного ключа, который принадлежит легальному получателю, а не какому-либо «подставному» лицу. Таким образом, технология PKI по существу является способом безопасного распределения публичных ключей, которая дает гарантию того, что сообщение зашифровано с применением публичного ключа, принадлежащего нужному нам адресату. По аналогичной схеме после получения подписанного документа мы в состоянии проверить идентичность подписывающего его лица.

Другая не менее важная составляющая асимметричной криптографии — управление закрытыми ключами пользователя. В этой главе мы обсудим, где эти ключи должны храниться и почему. При этом будем считать решенным вопрос, как распределены публичные ключи, и каким образом мы узнаем о соответствии ключа пользователю.

### 9.2.1. Безопасность закрытых ключей

Основной постулат криптографии заключается в том, что закрытые ключи доступны *только* их владельцам. Если злоумышленник может получить закрытый ключ какой-либо из сторон, участвующих в информационном обмене, это значит, что он легко может расшифровать все сообщения, посланные этой стороне. Кроме того, он может подписать любое сообщение от имени легального пользователя и успешно исполнить его роль в информационном обмене. Таким образом, ни о какой безопасности здесь не может быть и речи.

Более тонкий вопрос, который возникает в этом случае, заключается в том, что неотказуемость, как свойство ЭЦП, можно также считать утраченным, как только пользователь получит возможность *утверждать*, что его закрытый ключ мог быть нелегально использован (неважно, действительно ли фактическое воровство имело место или нет).

Признание того факта, что закрытый ключ пользователя мог быть нелегально использован, является основанием для поддержки судом заявления о том, что подписанный заказ, финансовый документ или приказ является фальсификацией. Безусловно, наличие такой возможности для недобросовестных пользователей носило бы разрушительный характер для системы электронных цифровых подписей в юридическом контексте транзакций.

Безопасность закрытого ключа пользователя должна быть обеспечена на каждом этапе его жизненного цикла:

- *Генерация ключевой пары* (открытый и закрытый ключи).
- *Хранение закрытого ключа*.
- *Использование закрытого ключа* (выполнение криптографических операций, требующих использования закрытого ключа пользователя, например, формирование электронной цифровой подписи).
- *Уничтожение закрытого ключа*.

*Безопасность закрытого ключа должна быть обеспечена на каждом этапе его жизненного цикла: генерация, хранение, использование закрытого ключа, уничтожение.*

*Генерация ключевой пары* должна выполняться в среде, исключающей как возможность влияния злоумышленника на сам процесс генерации, так и возможность получения какой-либо информации о закрытом ключе, которая может впоследствии быть использована при попытке его восстановления.

При *хранении* закрытого ключа должны быть обеспечены его *конфиденциальность* и *целостность* — ключ должен быть надежно защищен от несанкционированного доступа, а также модификации.

При *использовании* закрытого ключа важно исключить возможности его перехвата, а также несанкционированного использования (помимо воли и желания владельца ключа).

И, наконец, на этапе *уничтожения* закрытого ключа необходимо обеспечить гарантированное уничтожение информации и полностью исключить возможность его повторного использования (например, путем восстановления ранее удаленного хранилища).

### 9.2.2. Подходы к обеспечению безопасности закрытых ключей

Сейчас наиболее распространены и часто используются следующие подходы к обеспечению безопасности закрытых ключей:

1. **Программные хранилища** (software token) предназначены для хранения закрытых ключей на диске компьютера, чаще всего в зашифрованном виде. Примерами реализации данного подхода являются криптопровайдер Microsoft Enhanced CSP, входящий в состав операционной системы Microsoft Windows или браузер Mozilla/Netscape. Программные хранилища не обеспечивают безопасность закрытых ключей пользователя. Они создают только иллюзию защищенности.

*Программные хранилища не обеспечивают безопасность закрытых ключей пользователя. Они создают только иллюзию защищенности.*

2. **Аппаратные устройства** (hardware token) предназначены для хранения закрытых ключей и выполнения криптографических операций, требующих использования закрытого ключа. Наиболее распространенными представителями устройств данного класса являются смарт-карты и USB-ключи eToken PRO компании Aladdin, построенные с использованием микросхем смарт-карты.

3. **Репозитории** (credentials repository) представляют собой выделенные серверы (часто специализированное аппаратное обеспечение) для централизованного хранения закрытых ключей нескольких пользователей. Для того, чтобы выполнить операцию с помощью своего закрытого ключа (например, подписать электронное письмо), пользователь должен вначале аутентифицироваться на сервере, передать данные для обработки на сервер, затем получить результат.

В настоящее время в России наиболее распространенными средствами хранения закрытых ключей являются программные хранилища и аппаратные устройства в виде смарт-карт и USB-ключей. Репозитории стоят очень дорого и не позволяют обеспечить *мобильность* пользователя. В основном они используются для хранения ключей, используемых при аутентификации устройств, например, серверов, при организации защищенного информационного обмена, а также в финансовой сфере.

Ниже мы сравним два наиболее распространенных в сфере информационной безопасности подхода к хранению закрытых ключей пользователей. Один из вариантов предполагает хранение закрытых ключей с использованием программных средств, а второй — в специальном внешнем аппаратном устройстве (смарт-карте или USB-ключе). Последний вариант обладает значительно большей эффективностью для безопасности системы в целом.

### 9.2.3. Программные хранилища и их уязвимость

Защита закрытых ключей программными средствами называется программным хранилищем. Такое хранилище является программной эмуляцией аппаратного электронного ключа и выполняет практически аналогичные функции.

Для того чтобы воспользоваться закрытым ключом (например, для формирования ЭЦП документа), закрытый ключ пользователя должен быть предварительно извлечен из программного хранилища и загружен в память компьютера, после чего он может быть использован для выполнения криптографических операций.

Отметим, что за удобство и простоту использования программных хранилищ приходится дорого расплачиваться. В частности, для обеспечения мобильности пользователя программное хранилище должно быть продублировано на всех компьютерах пользователя (рабочий компьютер, домашний компьютер, ноутбук, карманный компьютер и др.). Это значительно увеличивает уязвимость закрытого ключа и может привести к его компрометации, как будет показано ниже.

### 9.2.4. Основные угрозы

Важно понимать и отдавать себе отчет в том, что *любые персональные компьютеры* представляют собой потенциально *небезопасную среду* из-за двух основных угроз:

- **Возможность физического доступа.** В основе угрозы лежит тот факт, что компьютер в большинстве случаев является физически доступным и незащищенным. И дело даже не в краже того же ноутбука, хотя и это не исключено. Во многих компаниях практикуется посменная работа сотрудников. При этом все они используют один и тот же компьютер — каждый в свою смену. Соответственно никто из них не может знать, что происходит с компьютером и кто за ним сидит в другую смену.

Нередки случаи, когда, например, секретарь может покинуть свое рабочее место, оставив пришедшего в офис посетителя предоставленным самому себе. Кто знает с какой целью он пришел?

Во всех этих случаях получить физический доступ к «бесхозному» компьютеру не составляет никакой сложности. А это значит, что потенциально злоумышленник может получить доступ к программному хранилищу.

Фактически любой офис является небезопасной средой и представляет возможности для проникновения злоумышленника, особенно обладающего навыками социальной инженерии. Программное хранилище закрытых ключей легального пользователя — сотрудника компании — может быть захвачено, скопировано или украдено вместе с ноутбуком или карманным компьютером. Причем в качестве злоумышленника можно легко представить не только «случайного гостя», но и коллегу по работе или даже уборщицу!

Важно отметить, что простой блокировки компьютера недостаточно, ведь злоумышленник может просто вручную извлечь жесткий диск и прочесть хранящиеся на нем файлы. (Конечно, можно принять контрмеры против кражи программного хранилища таким способом, например, зашифровав содержимое жесткого диска компьютера, однако в большинстве случаев даже такой «грубый» вариант все еще остается возможным).

- **Злонамеренное программное обеспечение (malicious software).** Вторую серьезную угрозу представляет быстро растущее и по качеству, и по количеству злонамеренное ПО, к которому относятся вирусы, сетевые черви, трояны и др. Стремительное распространение вредоносных программ ежегодно наносит компаниям колоссальный ущерб.

Заражение с помощью любого типа злонамеренного ПО может иметь разрушительные последствия для хранящегося на компьютере пользователя закрытого ключа, защищенного программными средствами. Написанная специально для этих целей программа-шпион (spyware) может просто считать файл программного хранилища без ведома хозяина компьютера и послать его своему автору. Такие программы незаметно «подсаживаются» на компьютер в ходе просмотра обычных с виду web-страниц.

Специализированные вирусы и троянские программы — реальность! Пример — вирус Caligula, выпущенный в конце 90-х гг. Он был предназначен именно для похищения закрытых ключей пользователя системы PGP, которые хранились в программном хранилище.

Более того, описанная выше атака может носить массовый характер, и, таким образом, множество ключей, хранящихся на компьютерах, могут быть считаны и украдены одним единственным «шпионом».

*Специализированные вирусы и троянские программы — реальность! Пример — вирус Caligula, выпущенный в конце 90-х гг. Он был предназначен именно для похищения закрытых ключей пользователя системы PGP, которые хранились в программном хранилище.*

## Основные контрмеры и обеспечиваемая ими безопасность

Учитывая вышеупомянутые угрозы, ясно, что программные хранилища не являются простыми, типичными файлами, сохраненными в личном каталоге пользователя. Это сделало бы доступ к содержимому программного токена чрезмерно простой задачей для злоумышленника. Программные хранилища защищены, и для этого используются следующие методики.

*Шифрование данных:* все содержимое программного хранилища находится в зашифрованном виде; используется надежный алгоритм шифрования; ключ шифрования формируется из пароля пользователя (например, с использованием хэш-функции).

*Путаница:* данный способ состоит в том, чтобы максимально усложнить поиск программного хранилища тому, кто несанкционированно попытается получить доступ к закрытому ключу. По существу, программное хранилище скрывается благодаря маскировке на жестком диске пользователя. Чтобы усложнить задачу хакера, используется множество способов, имеющих эффект «путаницы». Например, один из возможных методов состоит в том, чтобы сначала зашифровать программное хранилище с закрытым ключом и затем «прятать» ключ в различных местах на диске пользователя.

Рассмотрим безопасность каждой из описанных выше контрмер. Мы вполне можем согласиться с тем, что эффект «путаницы» действительно может помочь защититься от неопытных сетевых злоумышленников. Однако опытный хакер, приложив некоторые усилия, может обойти этот способ защиты. Таким образом, всецело доверять обеспечение безопасности «путанице» не стоит, так как она не обеспечивает необходимого для PKI-решений уровня безопасности.

Что же касается шифрования с помощью пароля, его безопасность в значительной степени зависит от качества используемого пароля. Важно понимать следующее:

- как только злоумышленник стал обладателем зашифрованного программного хранилища, он может попытаться подобрать пароль и по этому паролю расшифровать нужную информацию;
- число попыток подбора пароля для программного хранилища неограниченно.

Пробуя вводить подобранные пароли, атакующий может угадать необходимую для доступа комбинацию. Такое нападение называют «атакой по словарю» или «словарной атакой» и оно в достаточной мере эффективно. Единственный способ предотвратить эту угрозу состоит в применении сложного пароля, содержащего символьные, цифровые, буквенные значения разных регистров, и к тому же состоящего из случайного набора символов. Качественный пароль может выглядеть примерно так: g1UY$^M#&6430Ff`\`@Nk. К сожалению, запомнить качественные пароли для большинства пользователей не представляется возможным, особенно если принять во внимание тот факт, что таких паролей может быть несколько. Таким образом, в большинстве случаев «словарные атаки» очень успешны.

### 9.2.5. Выводы о безопасности программных хранилищ

Основная проблема программных хранилищ состоит в том, что они зависят от безопасности среды, в которой они находятся, т. е. компьютера. Однако, как мы уже неоднократно упоминали, персональные компьютеры небезопасны: как только получен физический доступ или машина заражена вирусом, закрытый ключ пользователя может быть скомпрометирован.

Стоит добавить, что сам факт атаки может остаться незамеченным для пользователя, и злоумышленник в течение долгого времени сможет расшифровывать корреспонденцию легального пользователя и отправлять от его имени и за его подписью фальсифицированную информацию. Таким образом, важнейшее свойство ЭЦП — неотказуемость — не может быть достигнуто. Именно поэтому программное хранилище обеспечивает относительно низкий уровень безопасности, и его использование в организациях, оперирующих конфиденциальной информацией, должно быть исключено.

Для усиления безопасности закрытых ключей пользователя и обеспечения его мобильности широко используются внешние отчуждаемые носители. Следует отметить, что ис-

пользование внешних носителей для сохранения программного хранилища (в качестве носителя может выступать, например, дискета, CD-диск) обеспечивает мобильность, однако несущественно повышает безопасность. Содержимое такого носителя загружается на локальный компьютер всякий раз, когда пользователю необходима ключевая информация (например, для формирования ЭЦП документа). Удобство и простота — важные преимущества использования портативного внешнего устройства по сравнению со стандартным программным хранилищем. Однако эти преимущества не повышают защищенность системы. Почему?

На первый взгляд, кажется, что раз закрытый ключ пользователя не сохранен на ноутбуке или стационарном компьютере, он надежно защищен. Ведь теперь злоумышленник нуждается в физическом доступе к внешнему устройству пользователя, а такое устройство мобильно и его значительно проще обезопасить, чем тот же ноутбук или персональный компьютер. Его можно хотя бы просто всегда носить с собой.

Однако можно вполне обоснованно утверждать, что достигаемое повышение защищенности (по сравнению с программным хранилищем) крайне незначительно.

- Во-первых, если внешний носитель, например, дискета, даже на короткое время окажется в руках злоумышленника, он сможет считать с нее информацию. Причем легальный пользователь — хозяин дискеты — может никогда не узнать о факте копирования его данных и, следовательно, их компрометации.
- Во-вторых, загруженный с дискеты на компьютер закрытый ключ уязвим для любого вредоносного программного обеспечения, имеющегося на компьютере пользователя.
- В-третьих, опасность для утечки данных представляют так называемые файлы подкачки (swap-файлы). В большинстве операционных систем они используются для временного хранения данных, которые выгружаются для ускорения работы системы. Облегчая работу компьютеру, технология подкачки всегда несет опасность записи на жесткий диск в открытом виде данных, которые должны оставаться зашифрованными.

Исходя из вышеперечисленного, можно сделать вывод о том, что уровень безопасности закрытых ключей пользователя, обеспечиваемый внешними носителями, ненамного выше уровня безопасности, обеспечиваемого программными хранилищами. Усилить защиту ключевой информации, записанной на взятую нами в качестве примера дискету, можно с помощью пароля. Однако, как мы уже говорили выше, простой пароль уязвим, и, следовательно, уровня обеспечиваемой им защиты недостаточно. Также не стоит забывать о вирусах, которые может содержать компьютер пользователя. Парольная защита здесь бессильна.

Использование программных хранилищ — это слишком большой риск для организации, так как обеспечивает слишком низкий уровень безопасности. При этом последствия вирусной атаки могут быть необратимыми. Это необходимо учитывать при построении системы информационной безопасности.

### 9.2.6. Аппаратные устройства с криптографическими возможностями

Итак, основная проблема рассмотренной выше системы защиты состоит в том, что закрытый ключ импортируется в небезопасную среду локального компьютера. Решить эту проблему можно, лишь используя отчуждаемое устройство, способное аппаратно выполнять криптографические операции. Таким образом, внешний носитель должен быть оснащен микропроцессором, способным зашифровать и отправить обратно сообщение, посланное на это устройство локальным компьютером пользователя. Благодаря возмож-

ности выполнения криптографических операций аппаратные устройства обеспечивают более высокий уровень защиты ключевой информации, так как закрытые ключи никогда не экспортируются из устройства.

*Благодаря возможности выполнения криптографических операций аппаратные устройства обеспечивают более высокий уровень защиты ключевой информации, так как закрытые ключи никогда не экспортируются из устройства.*

Тот факт, что закрытый ключ никогда не экспортируется из памяти устройства, является фундаментальным шагом вперед к максимально безопасному хранению закрытых ключей. Рассмотрим снова обозначенную нами парадигму угроз безопасности, но теперь через призму аппаратного устройства с криптографическими возможностями:

1. *Злонамеренное программное обеспечение (malicious software)*. Предположим, что пользователь подключает аппаратное устройство с криптографическими возможностями на инфицированную вирусом машину и вводит пароль для авторизации в появившемся на экране монитора окне. Существует вероятность того, что вирус, находящийся в компьютере, может подменить собой пользователя и, действуя от его имени, *использовать* аппаратное устройство (токен) для подписи сообщения. Однако реализация такой атаки ограничена во времени — она осуществима только на время физического подключения токена к компьютеру.

2. *Физический доступ (phisycal access)*. Вспомним еще раз о том, что в обсуждаемом нами типе токенов закрытый ключ хранится в защищенной памяти устройства и никогда не покидает ее. Поэтому воспользоваться им для проведения криптографических преобразований можно только в случае получения злоумышленником физического доступа к устройству (кража, похищение и др.). Если злоумышленник сумеет получить токен легального пользователя, возникнет угроза компрометации хранящейся в его памяти информации. Безопасность в этом случае обеспечивается лишь степенью физической защиты, обеспечиваемой самим устройством. У простейших токенов она минимальна, поэтому сломать ее и извлечь ключевую информацию не составляет большого труда. Справедливости ради, отметим, что для взлома защиты часто необходимо разрушить сам токен или просто украсть его, а, значит, нападение будет обнаружено легальным пользователем. В некоторых случаях, эта цена слишком высока.

3. В дополнение к перечисленным выше рискам и угрозам, существует потенциально более разрушительный тип нападения на аппаратные токены — *атака на побочные каналы (side channel attack)*. Получив физический доступ к токену, атакующий может получить информацию о закрытых ключах пользователя, измерив такие показатели, как время и мощность, затраченные в ходе выполнения токеном криптографических преобразований. Да, возможно, такой тип атаки покажется неправдоподобным, однако на деле он представляет собой высокоэффективный способ считывания закрытого ключа, к тому же не повреждающий само устройство.

### 9.2.7. Смарт-карты и USB-ключи на основе микросхем смарт-карт

Смарт-карта представляет собой специализированную микросхему, содержащую микропроцессор и операционную систему, управляющую работой микропроцессора.

Микросхема смарт-карты, которую использует этот класс устройств, обеспечивает безопасное хранение и использование ключей шифрования и ЭЦП, а также надежное

хранение цифровых сертификатов. Устройства, использующие технологии смарт-карт, разработаны специально для надежного противостояния различным типам атак и обеспечивают максимально высокий уровень безопасности для хранения и использования закрытых ключей.

*Устройства, использующие технологии смарт-карт, разработаны специально для надежного противостояния различным типам атак и обеспечивают максимально высокий уровень безопасности для хранения и использования закрытых ключей.*

Устройства на основе микросхем смарт-карт могут выпускаться как в виде смарт-карты, так и в виде USB-ключа, что существенно расширяет область их применения. Более того, устройства, базирующиеся на технологиях смарт-карт, можно дополнить RFID-меткой для радиочастотной идентификации, благодаря чему они могут использоваться не только для входа в сеть, но и для контроля доступа в помещения.

Мобильность смарт-карт и USB-ключей позволяет пользователю безопасно работать в «запрещенной среде», так как ключи шифрования и ЭЦП генерируются аппаратно микросхемой смарт-карты, никогда не покидают ее и не могут быть извлечены или перехвачены.

Большие возможности смарт-карт и USB-ключей на основе микросхемы смарт-карты позволяют им работать со всеми приложениями, использующими технологии смарт-карт, что делает их незаменимым средством для проведения защищенных финансовых транзакций, применения в приложениях, предназначенных для электронной коммерции, а также для безопасного доступа в корпоративную сеть, к защищенным информационным ресурсам, порталам и др.

Реализуя принцип двухфакторной аутентификации, данный тип устройств может быть использован злоумышленником только в том случае, если он будет иметь физический доступ к устройству и знать его PIN-код, защищенный от подбора. Во всех остальных случаях кража смарт-карты или USB-ключа на основе микросхемы смарт-карты бесполезна.

Также важно отметить, что существуют международные стандарты безопасности, которые разработаны специально для смарт-карт и USB-ключей, среди которых наиболее широко применяются стандарт CWA 14169 (стандарт для изделий, реализующих электронную подпись — безопасные устройства создания подписи secure signature-creation devices, SSCD) и профиль защиты для смарт-карт «Smart Card Protection Profile (SCSUG-SCPP)». Многие производители смарт-карт и USB-ключей на основе микросхем смарт-карт для приложений информационной безопасности (среди них в первую очередь компания Aladdin — SafeNet) сертифицируют свою продукцию на соответствие этим стандартам.

К основным особенностям смарт-карт можно отнести следующие:

- Смарт-карты изначально проектируются с учетом требований обеспечения безопасности хранящихся и обрабатываемых на них данных. Существует ряд международных открытых стандартов (например, семейство стандартов ITSEC) в области обеспечения безопасности для смарт-карт, которыми руководствуются разработчики.
- Физически (на уровне «железа») и логически (средствами встроенной операционной системы) обеспечивается защищенное хранение данных и защищенная обработка данных внутри микросхемы смарт-карты (а не на внешних модулях памяти).
- Аппаратная реализация криптографических алгоритмов.

Другие типы персональных идентификаторов (Dallas Touch Memory, магнитные карты, радио-метки, средства генерации одноразовых паролей и пр.) сильно проигрывают смарт-картам и USB-ключам.

**Преимуществами использования технологии смарт-карт** в качестве средств аутентификации и хранения ключевой информации пользователей являются:

- Архитектурное решение, специально спроектированное для использования в системах обеспечения информационной безопасности.
- Защищенность микросхем смарт-карт от различных видов атак (по анализу потребляемой мощности, от послойного сканирования, от изучения разрушающими методами и пр.)
- Для микросхем смарт-карт существуют общепринятые международные стандарты по безопасности, которыми руководствуются разработчики микросхем смарт-карт и операционных систем для смарт-карт, такие как стандарт CWA 14169 — стандарт для изделий, реализующих электронную подпись — безопасные устройства создания подписи secure signature-creation devices, SSCD; профиль защиты для смарт-карт «Smart Card Protection Profile (SCSUG-SCPP).
- Сертификация изделий производится на соответствие международным стандартам в независимых испытательных лабораториях, а не самим производителем на соответствие собственным декларациям.

- Аппаратная реализация криптографических функций, в том числе функций генерации ключевых пар (открытый/закрытый ключи), формирования ЭЦП с использованием закрытого ключа.
- Обеспечение безопасности закрытых ключей пользователя на всех этапах их жизненного цикла (генерация, хранение, использование, уничтожение).
- Поддержка смарт-карт включена в наиболее популярные версии операционных систем для компьютеров — Microsoft Windows (PC/SC), Linux и его клоны, MAC OS (Open Card Framework). Это снижает объем ПО от разработчика средств аутентенификации и хранения ключевой информации, требуемый к установке на рабочей станции пользователя. Упрощается процесс развертывания устройств в корпоративной среде, их поддержки и сопровождения.
- Простая конструкция устройства, за счет чего повышаются надежность и стабильность работы устройства в целом (так как используется меньшее число компонентов).
- Становится возможным сертифицировать устройство аутентификации и хранения ключевой информации *в целом*, а не только отдельных его компонентов. Для устройств аутентификации и хранения ключевой информации, построенных на основе микроконтроллера, чаще всего разработчику удается сертифицировать лишь один из компонентов устройства, но не устройство в целом.
- Возможность создания устройств аутентификации и хранения ключевой информации пользователей в различных форм-факторах: смарт-карты и USB-ключа. Формат смарт-карты наиболее востребован корпоративными заказчиками, которые применяют смарт-карты как единое устройство для контроля логического доступа (доступ к информационным ресурсам) и физического доступа (контроль доступа в помещения, визуальный контроль по фотографии, напечатанной на поверхности смарт-карты). Смарт-карты также необходимы на защищенных рабочих станциях, где по требованиям безопасности недопустимо использование портов USB.

Разработчик средств аутентификации и хранения ключевой информации может расширить функции устройства путем написания так называемых пакетов для операционной системы смарт-карты или Java-аплетов (если речь идет о Java-карте). Функции могут быть расширены разработчиком устройства, например для добавления поддержки национальных алгоритмов шифрования или расширения функций устройства. Хорошим примером является USB-ключ eToken PRO (Java) производства компании Aladdin, в котором поддержка российских криптографических алгоритмов (в соответствии с ГОСТ Р 34.10—2001 и ГОСТ Р 34.11—94) реализована в виде Java-аплета, исполняемого на смарт-карте.

## 9.3. Типовые требования к средствам аутентификации и хранения ключевой информации

Современные средства аутентификации и хранения ключевой информации пользователей должны не только обеспечивать защищенное хранение данных в памяти устройства, но и аппаратно поддерживать выполнение криптографических операций в доверенной среде в соответствии с требованиями национальных стандартов.

Поскольку национальные требования в области информационной безопасности различаются в разных странах, задача создания национального средства аутентификации и хранения ключевой информации пользователя *не может* быть решена только путем импорта аналогичных средств, используемых в других странах.

Чтобы полностью отвечать требованиям потребителей на национальных рынках, поставщик средств аутентификации и хранения ключевой информации пользователей должен обеспечить соответствие устройства как минимум следующим требованиям:

- поддержка национальных и международных криптографических стандартов;
- сертификация;
- возможность использования устройства без установки пользователем дополнительного ПО на клиентской рабочей станции;
- возможность размещения и исполнения нескольких приложений на устройстве;
- наличие инфраструктурного решения для управления жизненным циклом устройств;
- наличие двух форм исполнения: USB-ключ и смарт-карта;
- встроенная поддержка в наиболее распространенные клиентские операционные системы и приложения;
- обратная совместимость с наиболее широко распространенными средствами аутентификации и хранения ключевой информации пользователей.

### 9.3.1. Поддержка национальных криптографических стандартов

Данная поддержка должна быть реализована:

- на аппаратном уровне (реализация национальных криптографических алгоритмов самим устройством);
- на программном уровне (ПО промежуточного слоя — библиотека PKCS#11 и/или криптопровайдер стандарта Microsoft CSP, — позволяющее разработчикам прикладного ПО легко встроить поддержку устройств в свои продукты).

При аппаратной реализации национальных криптографических алгоритмов разработчик устройства в первую очередь должен обеспечить поддержку национального стандарта ЭЦП (генерация устройством пар открытый/закрытый ключ, их безопасное хранение, использование и гарантированное уничтожение).

Также следует предусмотреть совместимость с наиболее распространенными типами национальных средств криптографической защиты информации (СКЗИ) (в России — КриптоПро CSP, Домен-К, Signal-COM, Верба-OW и др.) и возможность импорта ключевой информации пользователей этих СКЗИ с незащищенных носителей (дискета, реестр Windows) в устройство.

### 9.3.2. Поддержка международных криптографических стандартов

Для участия в международном информационном обмене следует предусмотреть аппаратную и программную поддержку устройством международных и межгосударственных криптографических стандартов. Примером действующего международного стандарта является алгоритм RSA. В качестве межгосударственного стандарта стран СНГ выступает алгоритм формирования и проверки ЭЦП в соответствии с ГОСТ Р 34.310—2004.

### 9.3.3. Сертификация средств аутентификации и хранения ключевой информации

В соответствии с требованиями законодательства аутентификационные данные пользователей и ключевая информация должны сохраняться в тайне, а в ряде случаев они могут быть отнесены к категории конфиденциальной информации. Средства аутентифи-

кации и хранения ключевой информации реализуют защищенное хранение данных и должны быть сертифицированы в системе сертификации средств защиты информации. Средства аутентификации и хранения ключевой информации также реализуют аппаратно (и программно) национальные криптографические алгоритмы. Они относятся и к категории СКЗИ, что потребует от разработчика наличия лицензии на деятельность в данной области, а также выполнения предусмотренных процедур их проектирования и разработки (например, положение ПКЗ-2005).

Таким образом, разработчик должен иметь лицензии на соответствующие виды деятельности и предусмотреть сертификацию создаваемых средств:

- в системе сертификации средств защиты информации (ФСТЭК России);
- в Федеральной Службе Безопасности России (как СКЗИ);
- в отраслевых системах добровольной сертификации (например, ГАЗПРОМСЕРТ) — по требованию потребителя.

### 9.3.4. Возможность использования без установки дополнительного ПО на клиентской рабочей станции

Данное требование вытекает из сценариев использования устройства пользователем в системах Интернет-банкинга, системах дистанционного банковского обслуживания (ДБО) и других сферах, где очень важны как доступность услуги (сервиса) самому широкому кругу пользователей, так и обеспечение юридической значимости транзакций (действий), выполняемых пользователем в процессе работы.

Выполнение требования доступности сервиса (услуги) приводит к тому, что необходимо предусмотреть возможность полноценной работы пользователя с рабочих станций (терминалов), где у него нет прав локального администратора и поэтому нет возможности проводить установку ПО. Все компоненты, необходимые для функционирования устройства, должны либо входить в состав наиболее популярных операционных систем, либо быть доступны для автоматического скачивания с сайтов обновлений к ОС (например, для ОС семейства Microsoft Windows, это сайт Windows Update).

### 9.3.5. Размещение и исполнение нескольких приложений на устройстве

Устройство аутентификации и хранения ключевой информации пользователя не должно быть одноаппликационным, а должно иметь возможность размещения и исполнения нескольких приложений на устройстве (каждое приложение — со своим набором данных).

Данное требование предполагает наличие на устройстве достаточного количества памяти для размещения приложений и их данных. Оптимальный объем памяти — 72 Кбайт, который в большинстве случаев достаточен для размещения необходимых приложений и их данных.

### 9.3.6. Инфраструктурное решение для управления жизненным циклом устройств

Массовый характер использования средств аутентификации и хранения ключевой информации предусматривает их эмиссию большому числу пользователей. При этом необходимо иметь единую централизованную систему управления этими устройствами, которая позволяет:

- вести реестр выпущенных устройств;
- управлять их жизненным циклом (при любых масштабах эмиссии);
- обеспечивать пользователей web-сервисами самообслуживания для самостоятельного (и удаленного) решения задач, возникающих в процессе эксплуатации устройств;
- вести аудит использования устройств (во внутрикорпоративной сети);
- создавать отчеты (управленческие, для служб ИТ и ИБ).

### *9.3.7. Два форм-фактора исполнения: USB-ключ и смарт-карта*

Форм-фактор USB-ключа является наиболее популярным для конечных (в том числе и индивидуальных) пользователей, так как для работы с устройством аутентификации и хранения ключевой информации не требуется устройство чтения смарт-карт (достаточно наличие на рабочей станции порта USB). Следует отметить, что предпочтительной является версия 2.0 интерфейса USB, поскольку она обеспечивает значительно более высокую скорость передачи данных по сравнению с устаревшей версией 1.1.

Смарт-карты наиболее востребованы корпоративными заказчиками, которые применяют смарт-карты как единое устройство для контроля логического доступа (доступ к информационным ресурсам) и физического доступа (контроль доступа в помещения, визуальный контроль по фотографии, напечатанной на поверхности смарт-карты). Смарт-карты также необходимы на защищенных рабочих станциях, где по требованиям безопасности нельзя использовать порты USB.

### 9.3.8. Встроенная поддержка устройств в наиболее распространенных клиентских операционных системах и приложениях

Выполнение разработчиком устройства данного требования обеспечит беспроблемное использование устройства на большинстве имеющихся рабочих станций.

### 9.3.9. Обратная совместимость с наиболее широко распространенными средствами аутентификации и хранения ключевой информации пользователей

В настоящее время наиболее распространенным средством аутентификации и хранения ключевой информации пользователей являются электронные ключи eToken компании Aladdin. По состоянию на начало 2008 г., компания Aladdin контролировала 70—75% российского рынка этих средств. Оставшийся сегмент рынка был распределен между изделиями отечественных производителей — электронные ключи ruToken (компания Актив), Шипка (ОКБ САПР) и зарубежных — электронные ключи iKey (Rainbow Technologies, ныне — SafeNet), RSA SecurID (RSA, ныне — подразделение EMC).

В процессе перехода на новые средства аутентификации и хранения ключевой информации должна быть обеспечена полная обратная совместимость с уже имеющимися у заказчика аналогичными средствами. Так как электронные ключи eToken PRO занимают доминирующее положение на российском рынке средств аутентификации и хранения ключевой информации, а электронные ключи eToken PRO (Java) полностью обратно совместимы с eToken PRO, то выбор именно eToken PRO (Java) в качестве платформы для средств следующего поколения представляется наиболее целесообразным.

## 9.4. Особенности корпоративного использования персональных средств аутентификации и хранения ключевой информации

### 9.4.1. Многоуровневая ролевая модель доступа

При корпоративном внедрении крайне важна возможность реализации многоуровневой ролевой модели доступа. Как правило, эта модель включает следующие уровни разграничения полномочий доступа к персональным средствам аутентификации и хранения ключевой информации:

- **Уровень пользователя**. Конечный пользователь использует пользовательский PIN-код для выполнения ежедневных задач аутентификации, доступа к данным и т. д.
- **Уровень администратора** информационной безопасности (офицера ИБ). Администратор ИБ устанавливает ряд параметров для применяемых в организации средств аутентификации и хранения ключевой информации (например, требования к качеству PIN-кода), соответствующих действующей политике ИБ. Администратор ИБ использует PIN-код администратора при необходимости разблокирования персонального идентификатора и/или смены забытого PIN-кода пользователя.

- **Уровень производителя.** PIN-код этого уровня используется для:
  - ✓ полного переформатирования персонального средства аутентификации и хранения ключевой информации с полным удалением всех хранящихся в памяти устройства данных;
  - ✓ установки параметров, соответствующих требованиям действующей политики ИБ на предприятии (например, максимально допустимое число последовательных неудачных попыток ввода PIN-кода до блокирования устройства).

В данной модели каждый из субъектов действует на своем уровне в рамках делегированных ему полномочий и установленных ограничений. Применяемые средства аутентификации и хранения ключевой информации должны поддерживать все три уровня разграничения полномочий.

### *9.4.2. Интеграция с системами контроля и управления доступом в помещения (СКУД)*

Смарт-карты и USB-ключи могут выступать как единое средство доступа к различным информационным ресурсам, а смарт-карты с нанесенной на них информацией о владельце и его фотографией — еще и как средство визуальной идентификации.

*Смарт-карты и USB-ключи могут применяться как единое средство для контроля физического доступа в помещения и контроля логического доступа к информационным ресурсам.*

Смарт-карта и токен также могут быть дополнены бесконтактной радио-меткой (RFID-чипом, Proximity), используемым в бесконтактных «электронных проходных». Это позволит повысить уровень безопасности и удобства — смарт-карта или токен не может быть оставлен подключенным к компьютеру или передан другому лицу, так как без него нельзя покинуть помещение. Также становится возможным отслеживание перемещения сотрудников, а при интеграции с системами телефонии — перевод звонков в те помещения, где в данный момент находится сотрудник.

### 9.4.3. Централизованная система управления

При корпоративном внедрении проектов, использующих технологии PKI, необходимо централизованно управлять распределением и вести учет всех средств аутентификации и хранения ключевой информации, используемых в организации. Это могут быть смарт-карты, USB-ключи, генераторы одноразовых паролей, комбинированные устройства (например, USB-ключ с микросхемой смарт-карты и генератором одноразовых паролей).

Система централизованного учета и управления необходима для поддержки исполнения политики ИБ организации и является эффективным средством интеграции различных средств защиты информации (СЗИ).

Централизованная система управления должна:

- производить централизованное автоматическое тиражирование/обновление/удаление пользовательских данных в памяти устройств, а также проводить их разблокировку и форматирование;
- обеспечивать блокировку утерянных персональных средств аутентификации и хранения ключевой информации, выполнять отзыв (блокировку) содержавшихся на них аутентификационных данных и ключевой информации;
- позволять использовать групповые политики для делегирования и отзыва прав и полномочий пользователям, а также реализовать единую политику назначения PIN-кодов персональных идентификаторов;
- определять список приложений ИБ, доступ к которым имеет владелец данного средства аутентификации;
- собирать статистику и вести аудит использования средств аутентификации и хранения ключевой информации;
- интегрироваться со службой каталога (например, Microsoft Active Directory или OpenLDAP);
- осуществлять дистанционное обновление программного обеспечения на клиентских рабочих местах.

## 9.5. Централизованная система управления средствами аутентификации и хранения ключевой информации пользователей

Все большее число организаций используют надежные аппаратные средства аутентификации и хранения ключевой информации (USB-ключи и смарт-карты, далее по тексту — токены). Но без системы управления, используемой для централизованного учета и обслуживания токенов, их практическое использование может оказаться сложным или даже невозможным. Системы управления токенами, предоставляющие возможности по поддержке и управлению их жизненным циклом, делают внедрение аппаратных средств аутентификации и хранения ключевой информации реальностью.

Система управления токенами — это основа, объединяющая и управляющая всей инфраструктурой аутентификации путем организации точки централизованного администрирования всех типов аппаратных средств аутентификации, инструментов самообслуживания пользователя, встраивания в уже существующие системы управления учетными записями пользователей, политики и приложения безопасности в организации. Система упрощает процессы распространения токенов и управления ими, поддерживает управление возможностями повышения безопасности пользователя в течение всей его деятельности в организации и повышает эффективность работы пользователя. Открытая, мощная и гибкая система управления позволяет постоянно увеличивать число поддерживаемых решений безопасности, улучшая и расширяя возможности решений, включающих токены.

### *9.5.1. Проблема управления жизненным циклом*

Применение аппаратных средств аутентификации и хранения ключевой информации пользователей ставит задачи контроля их распространения и управления их жизненным циклом в рамках организации. Без системы управления обеспечение таких связей может оказаться чрезвычайно сложным и длительным, что приводит к большим затратам на внедрение и росту вероятности возникновения ошибок.

Для примера возьмем сотрудника, который только поступил на работу в организацию.

***Назначение и регистрация средства аутентификации.*** Чтобы начать работу, сотруднику необходимо получить токен. Без системы управления администратор должен перед этим вручную настроить и «привязать» токен к учетной записи пользователя, используя различные системы для формирования и сохранения учетных записей нового сотрудника в памяти токена. Такой процесс регистрации токенов вручную несет в себе вероятность ошибки. Предоставление новому сотруднику неправильных учетных записей и/или прав доступа, может привести к нарушению режима ИБ.

***Восстановление PIN-кода токена.*** Если сотрудник забыл PIN-код своего токена, то он не сможет пройти аутентификацию, войти в сеть и выполнить свою работу. Без инструмента самообслуживания для восстановления PIN-кода токена, вместо того, чтобы быстро восстановить PIN-код самостоятельно, сотрудник должен связаться со службой поддержки пользователей. При этом тратятся время и средства компании.

***Замена утерянного токена.*** Сотрудник может потерять токен. При этом администратор должен вручную:

- отозвать (заблокировать) все учетные записи пользователя для доступа ко всем информационным системам, с которыми работал пользователь;
- сформировать новые учетные записи;
- выдать пользователю новый токен и занести в его память новые учетные записи.

При этом опять возникает вероятность ошибки, так как администратор может забыть отозвать некоторые учетные записи, с помощью которых пользователь позже сможет получить несанкционированный доступ.

*При наличии в организации 100 и более токенов система управления их жизненным циклом обязательна.*

Для организации с более чем 100 сотрудниками администрирование парка устройств аутентификации и хранения ключевой информации без системы управления может оказаться очень трудной задачей. Грамотное решение предполагает использование автоматических инструментов и процедур, которые не только значительно разгружают технический отдел, но и уменьшают вероятность ошибок и отвечают требованиям администрации по организации, управлению и отслеживанию всего парка устройств для аутентификации и соответствующих приложений, имеющих отношение к безопасности в организации. Система управления токенами отвечает всем этим требованиям.

Система управления предоставляет всесторонние возможности управления токенами и связанными с ними решениями по безопасности на протяжении всего жизненного цикла токена, от момента его выпуска для пользователя и до отзыва. Для упрощения этих действий система управления предоставляет набор инструментов управления для администратора и пользователей.

### 9.5.2. Жизненный цикл токенов

***Регистрация.*** Токен «связывается» с конкретным пользователем и регистрируется в системе, в соответствии с ролью пользователя в организации и корпоративными политиками безопасности. В память токена записываются аутентификационные данные и ключевая информация.

***Использование и поддержка.*** Аутентификационные данные и ключевая информация пользователя, а также настройки самого токена периодически изменяются. Например, это может произойти из-за перевода сотрудника на другую должность, когда изменяется круг приложений, с которыми работает пользователь, а также при регламентной смене аутентификационных данных. Пользователь самостоятельно осуществляет поддержку токена (например, смену PIN-кода). При утере или повреждении токена хранящиеся в его памяти аутентификационные данные и ключевая информация либо аннулируются, либо восстанавливаются, если это возможно.

***Отзыв.*** Когда сотрудник покидает организацию, назначение токена ему отменяется. Аутентификационные данные и ключевая информация пользователя, хранящиеся в памяти токена, аннулируются.

***Возврат в эксплуатацию.*** Токен может быть выдан новому сотруднику. Перед регистрацией токена он должен быть приведен в исходное состояние и персонализирован для нового пользователя.

Ниже приведен обзор главных стадий и процессов жизненного цикла токенов.

### 9.5.3. Выпуск и распространение

Токены могут быть вначале зарегистрированы в системе и только потом назначены пользователям, либо регистрироваться в системе непосредственно в момент их выпуска.

Когда токены выдаются пользователям, система управления позволяет создавать и обновлять записи о физическом списке токенов во время выпуска. В идеале выпуск токенов может производиться как централизованно (например, на рабочем месте администратора системы), так и самими пользователями.

### 9.5.4. Регистрация токенов

Каждый токен должен быть назначен конкретному пользователю и подготовлен для него. В процессе регистрации в память токена загружаются пользовательские данные, цифровые сертификаты и пароли. При этом токен защищается индивидуальным PIN-кодом, известным лишь пользователю, для которого токен предназначен. Система управления может облегчить регистрацию токена путем автоматического создания и сохранения необходимых реквизитов пользователя, основанных на правах доступа конкретного пользователя в соответствии с политикой организации. Например, в организации, использующей инфраструктуру открытых ключей (PKI), авторизованные пользователи могут автоматически создавать запросы на получение сертификата и генерировать ключевые пары аппаратно на самих токенах в процессе регистрации токена.

Системы управления, включающие возможность самостоятельной регистрации пользователей, обеспечивают большую эффективность, уменьшая необходимость вмешательства администраторов.

*Система управления облегчает процесс регистрации токенов путем автоматического создания и последующего сохранения аутентификационных данных и ключевой информации каждого пользователя. Система должна быть основана на действующих правилах доступа и политике ИБ организации.*

### 9.5.5. Обработка типовых событий: «Потеря токена» и «Повреждение токена»

Система управления обеспечивает быстрый процесс замены утерянного или поврежденного токена. Это позволяет значительно снизить потерю производительности пользователя. Утерянный или поврежденный токен может быть быстро заменен на новый, в памяти которого содержатся все необходимые пользователю аутентификационные данные и ключевая информация.

### 9.5.6. Отзыв токена

Данная операция необходима для предотвращения несанкционированного доступа к информационным ресурсам с помощью токена (например, если им завладел злоумышленник или принято решение об увольнении/экстренном отстранении сотрудника от исполнения им служебных обязанностей). Система управления токенами позволяет администратору из единой точки управления аннулировать (заблокировать) все аутентификационные данные и ключевую информацию, хранящиеся в памяти токена, чтобы запретить доступ к корпоративной сети и приложениям пользователям, больше не имеющим на это права.

### 9.5.7. Инструменты управления токенами

Системы управления токенами предоставляют различные инструменты для администраторов и пользователей, облегчающие управление жизненным циклом токенов.

### 9.5.8. Инструменты для администраторов

Система управления токенами — это платформа, с которой можно отслеживать и управлять токенами в сочетании с корпоративным хранилищем учетных записей, правилами, политикой и приложениями безопасности. Она может включать как программы, установленные на серверах организации, так и веб-приложения.

### 9.5.9. Инструменты самообслуживания пользователя

Для операций, которые пользователь может совершать без обращения к администратору, таких как регистрация токена или смена PIN-кода, существуют инструменты самообслуживания пользователя, которые предоставляют значительную экономию времени и средств. Пользователи могут быстро совершать все действия по управлению токенами, уменьшая потерю производительности и загруженность отдела информационных технологий. Инструменты самообслуживания пользователей могут применяться как в виде установленных на компьютеры пользователей программ, так и в виде веб-приложений для удаленного администрирования.

*Инструменты самообслуживания обеспечивают значительную экономию средств и времени для организации.*

## 9.6. Типовые требования к системе управления токенами

При выборе системы управления токенами важно помнить, что она имеет огромное значение для организации. Потенциальная сложность решений на основе токенов делает выбор системы управления важным стратегическим решением. В приведенном ниже списке рассматриваются важные аспекты, которые должны быть приняты во внимание в организации при выборе системы управления токенами.

### 9.6.1. Функциональность

Система должна предоставлять администратору полный набор инструментов для управления жизненными циклами различных устройств и приложений. Она должна давать организации возможность централизованно управлять всеми аспектами использования токенов.

### 9.6.2. Простое и интуитивное использование

Система должна быть простой, доступной и управляться интуитивно. Кроме того, для уменьшения участия службы поддержки пользователей и максимальной экономии важна простота использования инструментов самообслуживания пользователя.

### 9.6.3. Открытая архитектура

Открытая и основанная на стандартах архитектура делает возможным совмещение системы управления токенами с внешними ресурсами, использующими известные и принятые стандарты. Открытая архитектура не только позволяет всесторонне встроить систему управления токенами в уже существующие в организации инфраструктуру информационных технологий и системы управления идентификацией, но и оказывать хорошо налаженную поддержку целому спектру приложений безопасности. Системы управления токенами, для которых поставляются комплекты разработчика (SDK), позволяющие встраивать их в другие приложения, предоставляют увеличенные возможности для расширения спектра поддерживаемых решений.

### 9.6.4. Гибкость

Система управления токенами должна быть достаточно гибкой, чтобы отвечать различным и развивающимся нуждам разных организаций. Она должна позволять организациям легко расширять набор поддерживаемых средств управления при появлении новых решений на основе токенов в будущем.

## 9.7. Token Management System (TMS) компании Aladdin

TMS — надежная система управления, позволяющая осуществлять в организации распространение токенов, подготовку их к работе и последующую поддержку. TMS компании Aladdin поддерживает ряд устройств, включая USB-токены, смарт-карты и устройства одноразовых паролей, а также разнообразные приложения безопасности, такие как программы для входа в сеть, виртуальные частные сети, защищенный доступ к веб-сайтам, приложения для аутентификации с помощью одноразового пароля, защиты электронной почты, шифрования информации и т. д.

TMS — это единственное решение, которое объединяет пользователей, устройства, организационные правила и приложения безопасности в единую, автоматизированную и полностью настраиваемую систему. Это дает возможность легко управлять использованием решений с применением токенов, в частности, решений в инфраструктуре открытых ключей.

*TMS компании Aladdin — это надежная платформа, которая помогает внедрять токены и связанные с ними приложения безопасности и управлять их жизненным циклом в масштабах всей организации.*

TMS полностью встраивается в Active Directory (AD). При внедрении TMS расширяется схема экземпляра AD. TMS хорошо сочетается с уже существующей в организации инфраструктурой информационных технологий. Система предоставляет интерфейс управ-

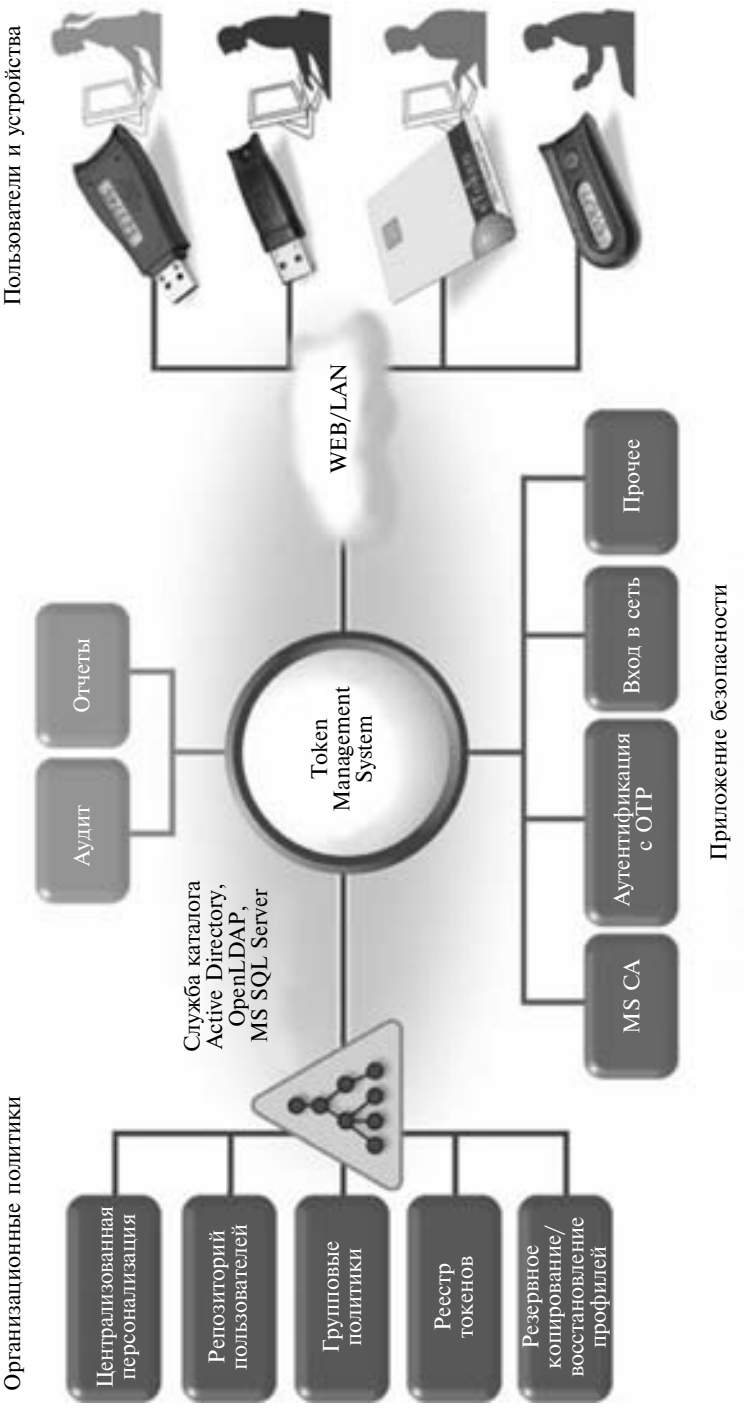



**Рис. 9.1. Система управления TMS**

ления в AD, позволяющий администраторам выполнять свои функции по отношению к токенам средствами, сходными с созданием и управлением групповыми политиками.

В TMS применяется веб-интерфейс для взаимодействия пользователя с системой. Пользователь имеет возможность самостоятельно управлять токеном через настраиваемый веб-сайт предприятия. Это дает пользователям возможность регистрировать токены и сразу начинать их использовать, переустановить забытый PIN-код (разблокировать токен) и выполнять иные действия (рис. 9.1).

Открытая модульная архитектура TMS позволяет управлять использованием токенов с применением решений по безопасности сторонних разработчиков. Эта возможность достигается путем применения «коннекторов» TMS — настраиваемых подключаемых модулей, используемых на сервере. Компания Aladdin предоставляет набор для разработчика коннекторов TMS, который позволяет поставщикам решений безопасности создавать собственные коннекторы и тем самым дополнять свои решения, использующие eToken, средствами администрирования. В настоящее время разработаны коннекторы для УЦ «КриптоПро УЦ» и УЦ RSA Keon.

### 9.7.1. Назначение

- Поэкземплярный учет и регистрация всех аппаратных и программных средств аутентификации и хранения ключевой информации, используемых сотрудниками.
- Ускорение ввода в эксплуатацию электронных ключей и смарт-карт, автоматизация процессов выдачи eToken сотруднику, персонализация eToken, запись ключевой информации и аутентификационных данных в память eToken.
- Управление жизненным циклом средств аутентификации и хранения ключевой информации: обновление аутентификационных данных и ключевой информации, предоставление/отзыв прав доступа к приложениям при изменении служебных обязанностей/увольнении сотрудника, замена устройства при его утере/повреждении, вывод устройства из эксплуатации.
- Аудит использования сотрудником выданного ему средства аутентификации и хранения ключевой информации (фиксируются все факты использования устройства сотрудником на компьютере предприятия, изменения хранящихся в памяти устройства данных).
- Подготовка отчетов для руководителей служб ИТ и ИБ об использовании сотрудниками средств аутентификации и хранения ключевой информации (на основе данных аудита средствами встроенного генератора отчетов, также имеется возможность экспорта данных во внешние средства построения отчетов).
- Техническая поддержка и сопровождение пользователей средств аутентификации через веб-сайт технической поддержки: переустановка забытого пользователем PIN-кода устройства, синхронизация генератора одноразовых паролей, обработка типовых ситуаций «пользователь забыл eToken», «пользователь потерял eToken», «пользователь повредил/сломал eToken».

### 9.7.2. Возможности

- Поддержка всех типов и моделей электронных ключей eToken (смарт-карты, USB-ключи, комбинированные USB-ключи, программные или виртуальные токены).
- Интеграция со службой каталога Microsoft Active Directory.

- Веб-сайты для самостоятельного решения пользователем проблем, возникающих в ходе эксплуатации eToken, оказания технической поддержки пользователям eToken специалистом сервисной службы (help desk web site), веб-сайт для администрирования системы.
- Открытая архитектура, позволяющая добавлять в систему поддержку новых приложений и аппаратных устройств (через механизм коннекторов).
- Масштабируемое, распределенное администрирование:
  - ✓ администрирование систем eToken TMS, установленных в разных доменах, с одного рабочего места администратора;
  - ✓ ролевое администрирование, возможность делегирования полномочий.
- Аудит использования средств аутентификации и хранения ключевой информации сотрудниками, гибкая система построения отчетов на основе данных аудита.
- Отказоустойчивость и масштабируемость системы, поддержка кластерных технологий Microsoft Windows 2003/2008.
- Централизованная установка клиентского ПО на рабочие станции пользователей.
- Широкий спектр поддерживаемых клиентских ОС: Windows, Linux, Mac OS; любая ОС при использовании одноразовых паролей.
- Полная поддержка русского языка.
- Виртуальный токен — уникальное программное решение, позволяющее пользователю, находящемуся вне офиса, даже в случае утери/повреждения eToken продолжить работу с компьютером или получить безопасный доступ к ресурсам.

## 9.8. Практика: комплексная система на базе единого персонального средства аутентификации и хранения ключевой информации

На примере простой гипотетической корпоративной системы попробуем выяснить, с какими трудностями чаще всего приходится сталкиваться при выборе единого персонального средства аутентификации и хранения ключевой информации, на основе каких принципов можно сформулировать критерии такого выбора.

Современные системы обеспечения информационной безопасности состоят, как правило, из нескольких объединенных посредством централизованного управления подсистем, включающих в себя разнородные сервисы безопасности. Следовательно, и задачи управления доступом пользователей с помощью единого персонального средства аутентификации к различным подсистемам, подчас не связанным напрямую между собой, отнюдь не кажутся тривиальными. Тем более что еще несколько лет назад производители систем защиты информации (СЗИ) настраивали их работу только на определенные типы аутентификаторов.

Поэтому до сих пор для доступа к одному защищенному приложению пользователю приходится пользоваться дискетой, к другому — смарт-картой, а к третьему — токеном или труднозапоминаемым паролем. С точки зрения централизованного управления доступом и прозрачного администрирования при наличии десятков информационных систем более удобным является введение единого персонального идентификатора, в котором надежно хранятся параметры доступа пользователя во все разрешенные области информационного пространства предприятия.

### 9.8.1. Бизнес-задачи по защите корпоративной информации

Наша гипотетическая организация в той или иной степени решает такие задачи, как:

- разграничение и контроль доступа зарегистрированных во внутренней сети пользователей к рабочим станциям и корпоративным информационным ресурсам;
- организация защиты корпоративных ресурсов, в том числе баз данных, содержащих информацию, утечка которой критична для существования бизнеса компании;
- предоставление безопасного доступа к порталу компании, в том числе к защищенным веб-страницам;
- защищенный обмен информацией территориально удаленных подразделений;
- внедрение и поддержка сервиса электронной цифровой подписи (ЭЦП);
- защищенный вход в бизнес-приложения;
- защита корпоративной почты и документооборота;
- организация полноценной работы удаленных пользователей;
- физическая защита рабочих помещений (системы контроля и управления доступом);
- оперативное и эффективное управление информационной инфраструктурой.

Данный перечень условимся считать оптимальными бизнес-требованиями комплексной защиты. При этом под задачей организации защищенного доступа к корпоративной сети, как уже упоминалось, будем понимать применение технологий, основанных на использовании закрытых ключей и цифровых сертификатов стандарта X.509.

Например, для сетей под управлением сервера Windows 2000/2003/2008 эта технология основана на возможности запрета доступа к сети по паролю и разрешения доступа по предъявлении пользователем сертификата из защищенной памяти персонального идентификатора (смарт-карты или USB-ключа) и проверке его валидности (протокол Kerberos + PKINIT).

Вопрос защиты корпоративных баз данных рассмотрим на примере СУБД Oracle, поскольку начиная с версии 8i данная СУБД оснащена встроенной поддержкой PKI, что позволяет организовывать доступ к корпоративной базе по предъявлении цифрового сертификата, сформированного штатными средствами Oracle.

Проблемы организации доступа к корпоративному порталу, защищенным приложениям и обеспечение работы удаленных пользователей проанализируем на примере технологий, основанных на использовании цифровых сертификатов.

Для применения ЭЦП в качестве гарантии конфиденциальности (защита от НСД), целостности (защита от внесения изменений), доступности для легальных пользователей, аутентичности (подтверждение авторства) и неотказуемости (несмотря на внесенные изменения в документы и почтовые сообщения) требуется применение средств криптографической защиты информации (СКЗИ). При этом персональное средство аутентификации и хранения ключевой информации должно обеспечить надежное хранение ключевых контейнеров применяемых СКЗИ.

Физическая защита рабочих помещений пользователей обеспечивается, как правило, с помощью систем контроля и управления доступом, основанных на применении RFID-технологий (Radio Frequency Identification Device — радиочастотная идентификация).

Можно выделить основные технологии информационной защиты, позволяющие решать типичные проблемы, составляющие вышеперечисленный оптимальный набор бизнес-задач:

- аутентификация пользователей;
- криптографическая защита электронных документов (шифрование, ЭЦП);
- межсетевое экранирование, защита каналов связи и VPN;
- RFID-технологии.

Иные технологии, не вошедшие в список, будем считать комбинацией перечисленных выше методов (например, защиту удаленного доступа или беспроводных соединений можно рассматривать как комбинацию технологии аутентификации и защиты сетевого трафика).

Каждая из перечисленных технологий в той или иной степени связана с активным использованием персональных устройств аутентификации и хранения ключевой информации:

- в рамках систем аутентификации данные устройства используются в качестве носителей атрибутов доступа пользователей к информационным и (или) вычислительным ресурсам;
- в рамках СКЗИ — для хранения ключевых контейнеров и выполнения криптопреобразований;
- при защите каналов связи и организации VPN — для аутентификации удаленных пользователей и сетевых устройств, выработки сеансовых ключей шифрования трафика;
- в рамках систем контроля и управления доступом (СКУД), основанных на использовании RFID-технологии, персональные устройства играют центральную роль маркеров доступа, в зависимости от состояния которых принимается решение о допуске пользователей в те или иные помещения.

Таким образом, проблема выбора единого персонального средства аутентификации и хранения ключевой информации может быть решена путем приобретения универсального устройства, отвечающего перечисленному оптимальному набору бизнес-задач по защите информации, плюс возможность интеграции с установленными средствами защиты и унаследованными (как правило, не поддерживающими сертификат X.509) приложениями. Рассмотрим типичные трудности, с которыми можно столкнуться при этом.

### 9.8.2. Проблемы выбора персонального устройства

Если вы хотите объединить все установленные системы защиты информации и применить для доступа единое средство аутентификации и хранения ключевой информации, то неизбежно столкнетесь со следующими трудностями:

- системы защиты информации работают с конкретными типами идентификаторов;
- СКЗИ настроены на использование определенных идентификаторов;
- RFID-метки размещаются только в смарт-картах;
- не все типы идентификаторов могут быть универсальными (применимыми к разным системам).

В частности, для того, чтобы полностью отвечать бизнес-задачам защиты информации, персональный идентификатор должен как минимум содержать процессор для выполнения криптографических операций на закрытом ключе пользователя и иметь достаточно большой (порядка 20—40 килобайт) запас свободной энергонезависимой памяти для записи цифровых сертификатов (каждый сертификат занимает в среднем от 1,5 до 2 Кбайт) и других параметров доступа к системам, которые не могут поддерживать использование сертификатов стандарта X.509 для аутентификации. Прежде чем рассмотреть существующие идентификаторы с точки зрения выполнения перечисленных выше задач, попробуем сформулировать критерии выбора.

### 9.8.3. Критерии выбора

Для выбора персонального средства аутентификации и хранения ключевой информации предлагается применять следующие минимальные бизнес-критерии.

1. Обеспечение строгой (двухфакторной) аутентификации при доступе к корпоративной сети, информационным ресурсам, защищенным приложениям.

2. Возможность защищенного хранения в персональном идентификаторе ключевого контейнера, сформированного СКЗИ, закрытого ключа пользователя (в терминах PKI).

3. Обеспечение защиты критичной информации от несанкционированного доступа.

4. Наличие необходимого объема свободной памяти в персональном идентификаторе для записи ключевой информации и других параметров доступа пользователя к защищенным ресурсам.

5. Надежность идентификатора как хранилища ключевой информации (физическая защита чипа, гарантированный срок применения, соответствие отечественным и международным стандартам).

6. Приемлемая средняя стоимость.

Насколько хорошо отвечают широко известные типы идентификаторов (парольная защита, дискета, устройство iButton, USB-ключи/смарт-карты) сформулированным критериям?

Детальный анализ показывает, что парольная защита, дискета и устройства iButton не отвечают указанным критериям. Наиболее приемлемыми (исходя из минимальных критериев) являются устройства класса USB-ключа eToken. Заметим, что функционально eToken PRO и смарт-карта, содержащая чип, идентичны, поскольку eToken — это смарт-карта и считыватель в одном устройстве.

Парольная защита не может использоваться для решения рассматриваемых задач защиты корпоративной информации.

Несложно объяснить, почему до сих пор многие заказчики предпочитают пользоваться персональными идентификаторами в виде дискеты. На этот выбор влияют три фактора: низкая стоимость, большой объем памяти и то, что дискета в качестве носителя поддерживается практически всеми отечественными производителями систем криптографической защиты информации. При этом не учитывается тот факт, что в случае попадания дискеты в руки злоумышленника (этого достаточно — никакой защиты нет) от вашего имени (а во многих случаях и с вашей подписью) могут быть совершены действия, последствия которых приведут к краху бизнеса предприятия. К тому же дискета недолговечна, скажем, в интенсивно используемых приложениях типа систем «клиент-банк» в течение года приходится менять десятки дискет в расчете на одного пользователя. При этом на каждую замену тратятся время и деньги не только пользователя, но и производителя системы.

Не представляется возможным персонифицировать пользователя «таблеток» iButton. Характерно на этот счет высказывание одного из специалистов по защите информации: «При использовании "таблетки" iButton в качестве персонального идентификатора вы получаете защиту корпоративной информации на уровне домофона».

Таким образом, в качестве универсальных персональных идентификаторов можно порекомендовать только устройства класса процессорной смарт-карты и USB-ключа (например, смарт-карты или USB-ключи eToken компании Aladdin).

## Контрольные вопросы

1. На каких этапах должна быть обеспечена безопасность закрытого ключа пользователя?
2. Перечислите подходы к обеспечению безопасности закрытых ключей.
3. Опишите жизненный цикл токенов.
4. Перечислите функции централизованной системы управления.
5. Перечислите основные критерии выбора персонального средства аутентификации и хранения ключевой информации.

## Список использованной литературы


1. A Guide to Understanding Identification and Authentication in Trusted Systems, U.S. National Computer Security Center.
2. Курило А. П. и др. Обеспечение информационной безопасности бизнеса. — М.: БДЦ-Пресс, 2005.
3. Документ RFC 4226, http://www.ietf.org/rfc/rfc4226.txt.
4. eToken NG-OTP. Краткая справочная информация. — Компания Aladdin, эксплуатационная документация, 2006.
5. Федеральный закон Российской Федерации от 10 января 2002 г. N 1-ФЗ «Об электронной цифровой подписи».
6. Ричард Э. Смит. Аутентификация: от паролей до открытых ключей. — М.: Вильямс, 2002.
7. Рекомендации к средствам криптографической защиты информации на взаимодействие удостоверяющих центров, реестров сертификатов, сертификаты ключей формата X.509 и электронные документы формата CMS. — ООО «КРИПТО-ПРО», 2001.
8. Стандарты ГОСТ 28147—89, ГОСТ Р 34.10—94 «Информационная технология. Криптографическая защита информации. Процедуры выработки и проверки электронной цифровой подписи на базе асимметричного криптографического алгоритма», ГОСТ Р 34.10—2001 «Информационная технология. Криптографическая защита информации. Процессы формирования и проверки электронной цифровой подписи», ГОСТ Р 34.11—94 «Информационная технология. Криптографическая защита информации. Функция хэширования».
9. Документы RFC 4357, 4490, RFC 4491.
10. Горбатов В. С., Полянская О. Ю. Основы технологии PKI. — М.: Горячая линия — Телеком, 2004.
11. Рассел Ч., Кроуфорд Ш., Джеренд Дж. Microsoft Windows Server 2003. Справочник администратора. — М.: Издательство «Эком», 2006.
12. Мак-Федрис П. Microsoft Windows XP. Полное руководство. — М.: Вильямс, 2006.
13. [CMS] Housley, R., «Cryptographic Message Syntax (CMS)», RFC 3852, July 2004.
14. [CPALGS] Popov, V., Kurepkin, I., and S. Leontiev, «Additional Cryptographic Algorithms for Use with GOST 28147—89, GOST R 34.10—94, GOST R 34.10—2001, and GOST R 34.11—94 Algorithms», RFC 4357, January 2006.
15. [CPCMS] S. Leontiev, Ed., G. Chudov, Ed., «Using the GOST 28147—89, GOST R 34.11—94, GOST R 34.10—94, and GOST R 34.10—2001 Algorithms with Cryptographic Message Syntax (CMS)», RFC 4490, May 2006.
16. [CPPK] S. Leontiev, Ed. and D. Shefanovskij, Ed., «Using the GOST R 34.10—94, GOST R 34.10—2001, and GOST R 34.11—94 Algorithms with the Internet X.509 Public Key Infrastructure Certificate and CRL Profile», RFC 4491, May 2006.

17. [PKIX] Housley, R., Polk, W., Ford, W., and D. Solo, «Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile», RFC 3280, April 2002.
18. [SMIME] B. Ramsdell, «Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.1 Message Specification», RFC 3851, July 2004.
19. Веб-сайт компании Майкрософт для технических специалистов (на русском языке) http://technet.microsoft.com/ru-ru/default.aspx.
20. Майкл Уэнстром. Организация защиты сетей Cisco (Managing Cisco Network Security). Издательство «Вильямс», 2003, 768 стр. ISBN 5-8459-0387-4, 1-5787-0103-1.
21. Шиндер Д. Основы компьютерных сетей. Серия: Cisco Press, 2002 г., 656 стр., ISBN: 5845902851, 1587130386.
22. Дэвид В. Чепмен мл., Энди Фокс. Брандмауэры Cisco Secure PIX Издательство: Вильямс, 2003 г., 384, с ил. ISBN: 5-8459-0463-3, 1-5870-5035-8.
23. Кэтрин Пакет. Создание сетей удаленного доступа Cisco. Издательство: Вильямс, 2003 г., 672, с ил., ISBN: 5-8459-0443-9, 1-5787-0091-4.
24. Педжман Рошан, Джонатан Лиэри. Основы построения беспроводных локальных сетей стандарта 802.11. Руководство Cisco. Издательство: Вильямс, 2004 г., 304, с ил., ISBN: 5-8459-0701-2, 1-5870-5077-3.
25. Альваро Ретана, Дон Слайс, Расс Уайт. Принципы проектирования корпоративных IP-сетей. Издательство: Вильямс, 2002 г., 368, с ил., ISBN: 5-8459-0248-7, 1-57870-097-3.
26. Джим Гейер. Беспроводные сети. Первый шаг (Cisco). Издательство: Вильямс, 2005 г., 192 стр., с ил.; ISBN 5-8459-0852-3, 1-58-720111-9.
27. Vijay Bollapragada, Mohamed Khalid, Scott Wainner. IPSec VPN Design CiscoPress, 2005, ISBN: 1-58-705111-7.
28. Troubleshooting Virtual Private Networks (VPN). Mark Lewis. Cisco Press, 2007, ISBN: 1-58-705104-4.