ключ D_A остается секретным. Если Алиса раскроет свой секретный ключ, этот аргумент перестанет быть убедительным, так как послать сообщение мог кто угодно, включая самого Боба.

Проблема может возникнуть, если Боб, например, является биржевым брокером Алисы. Алиса заказывает Бобу купить некоторое количество акций. Сразу после этого цена акций резко падает. Чтобы отречься от своего сообщения, посланного Бобу, Алиса заявляет в полицию, что ее дом был обворован, а компьютер вместе с секретным ключом украден. В зависимости от законов ее страны или штата она может быть признана или не признана ответственной перед законом, особенно если она заявляет, что обнаружила, что ее квартира взломана, только через несколько часов после возвращения с работы.

Другая проблема данной схемы цифровой подписи возникает в случае, если Алиса решит сменить свой ключ. Подобное действие абсолютно законно, более того, рекомендуется периодически менять ключ, чтобы гарантировать его высокую надежность. В этом случае, если дело дойдет до судебного разбирательства, судья попытается применить к подписи $D_A(P)$ *текущий* ключ E_A и обнаружит, что в результате не получается сообщение P. При этом Боб будет выглядеть довольно глупо.

В принципе, для цифровых подписей можно использовать любой алгоритм с открытым ключом. Алгоритм RSA, фактически, стал промышленным стандартом. Он применяется во многих программах, предназначенных для обеспечения безопасности. Однако в 1991 г. Национальный институт стандартов и технологий США NIST (National Institute of Standards and Technology) предложил использовать для нового **стандарта цифровой подписи DSS** (Digital Signature Standard) вариант алгоритма с открытым ключом Эль-Гамаля, основанный не на трудности разложения больших чисел на множители, а на сложности вычисления дискретных алгоритмов.

Как обычно, попытка правительства навязать новые криптографические стандарты вызвала много шума. Стандарт DSS критиковали за то, что он:

- слишком засекречен (протокол, использующий алгоритм Эль-Гамаля, разрабатывался Агентством национальной безопасности США);
- слишком медленный (от 10 до 40 раз медленнее алгоритма RSA для проверки подписей);
- слишком новый (алгоритм Эль-Гамаля еще не был достаточно тщательно проверен);
- слишком ненадежен (фиксированный 512-разрядный ключ).

При последующей переработке четвертый пункт претензий стал спорным, так как было разрешено использовать ключи длиной до 1024 разрядов. Однако первые два пункта актуальны и по сей день.

Профили сообщений

Многие методы цифровых подписей критикуются за то, что в них совмещаются две различные функции: аутентификация и секретность. Довольно часто требуется только аутентификация. К тому же, например, получить лицензию на экспорт обычно проще, если система обеспечивает только аутентификацию, но не секретность. Далее будет описана схема аутентификации, не требующая шифрования всего сообщения.

Эта схема основана на идее необратимой хэш-функции, которая принимает на входе участок открытого текста произвольной длины и по нему вычисляет строку битов фиксированной длины. У этой хэш-функции, часто называемой **профилем сообщения** (message digest, MD), есть четыре следующих важных свойства:

1. По заданному открытому тексту P легко сосчитать значение хэш-функции $MD(P)$.
2. По цифровой подписи $MD(P)$ практически невозможно определить значение открытого текста P.
3. Для данного P практически невозможно подобрать такой P', чтобы выполнялось равенство $MD(P') = MD(P)$.
4. Изменение даже одного бита входной последовательности приводит к очень непохожему результату.

Чтобы удовлетворять требованию 3, результат хэш-функции должен быть длиной, по крайней мере, в 128 бит, желательно даже больше. Чтобы удовлетворять требованию 4, хэш-функция должна искажать входные значения очень сильно. Этим данный метод напоминает алгоритмы с симметричными ключами, которые мы рассматривали ранее.

Профиль сообщения по части открытого текста вычисляется значительно быстрее, чем шифруется все сообщение с помощью алгоритма с открытым ключом. Поэтому профили сообщений могут использоваться для ускорения работы алгоритмов цифровых подписей. Чтобы понять, как все это работает, рассмотрим снова протокол передачи цифровой подписи, показанный на рис. 8.15. Вместо того чтобы посылать открытый текст P вместе с $K_{BB}(A, t, P)$, Большой Брат теперь вычисляет профиль сообщения $MD(P)$, применяя функцию хэширования MD к открытому тексту P. Затем он помещает $K_{BB}(A, t, MD(P))$ как пятый элемент в список, который зашифровывает ключом K_B, и отправляет его Бобу вместо $K_{BB}(A, t, P)$.

В случае возникновения спора Боб может предъявить на суде как открытый текст P, так и $K_{BB}(A, t, MD(P))$. По просьбе судьи Большой Брат расшифровывает $K_{BB}(A, t, MD(P))$, в результате чего суду предъявляются также цифровая подпись $MD(P)$, подлинность которой гарантируется Большим Братом, и сам открытый текст P, подлинность которого суд должен выяснить. Поскольку практически невозможно создать другой открытый текст, соответствующий данной цифровой подписи, суд убеждается в том, что Боб говорит правду. Использование профиля сообщения экономит время шифрования и затраты на транспортировку и хранение.

Профиль сообщения может также применяться для гарантии сохранности сообщения при передаче его по сети в системах шифрования с открытым ключом, как показано на рис. 8.17. Здесь Алиса сначала вычисляет профиль сообщения

для своего открытого текста. Затем она подписывает профиль сообщения и посылает зашифрованный профиль сообщения и открытый текст Бобу. Если злоумышленник попытается подменить по дороге открытый текст *P*, Боб обнаружит это, сосчитав значение профиля сообщения *MD*(*P*).

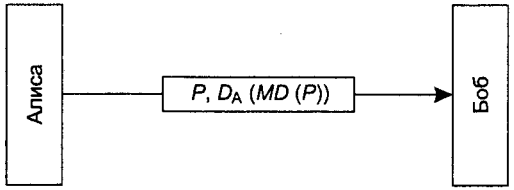


Рис. 8.17. Цифровая подпись с использованием профиля сообщения

MD5

Было предложено несколько вариантов функций, вычисляющих профиль сообщения. Самое широкое распространение получили алгоритмы MD5 (Rivest, 1992) и SHA (NIST, 1993). Алгоритм **MD5** (Message Digest 5 — профиль сообщения 5) представляет собой пятую версию хэш-функций, разработанных Рональдом Ривестом (Ronald Rivest). Он перемешивает входные биты достаточно сложным образом, так что каждый выходной бит зависит от каждого входного бита. Сначала сообщение дополняется до длины 448 бит по модулю 512. Затем к нему добавляется исходная длина сообщения, рассматриваемая как 64-разрядное число, в результате чего получается блок битов, длина которого кратна 512. Последний шаг подготовки к вычислениям инициализирует 128-разрядный буфер, задавая его содержимое равным некоему фиксированному значению.

Затем начинаются вычисления. На каждом этапе берется 512-разрядный блок входного текста и тщательно перемешивается со 128-разрядным буфером. Для пущей навористости в кастрюлю также кидается содержимое таблицы синусов. Именно синусы используются не потому, что их результат более случаен, чем результат других генераторов случайных чисел (в которых часто также применяются тригонометрические функции), а чтобы избежать каких бы то ни было подозрений в создании потайной лазейки, через которую потом разработчик (или заказчик) мог бы войти. Отказ корпорации IBM раскрыть принципы устройства S-блоков, применяемых в стандарте шифрования DES, привел к появлению большого количества слухов и домыслов о потайных ходах. Каждый входной блок обрабатывается за четыре итерации. Процесс продолжается, пока не будут обработаны все входные блоки. Содержимое 128-разрядного буфера и образует профиль сообщения.

MD5 появился около десяти лет назад, и за это время было предпринято множество атак на этот алгоритм. Были обнаружены некоторые слабые места, однако существуют определенные внутренние процедуры, позволяющие защититься от взлома. И все же, если упадут и эти последние барьеры, в один прекрасный день MD5 может оказаться ненадежным. Несмотря на это на момент написания книги этот алгоритм еще держится на плаву.

SHA-1

Второй широко применяемой функцией вычисления профиля сообщения является **SHA** (Secure Hash Algorithm — надежный алгоритм хэширования), разработанный Агентством национальной безопасности США (NSA) и получивший благословение национального института стандартов и технологий NIST (выразившееся в федеральном стандарте FIPS 180-1). Как и MD5, алгоритм SHA обрабатывает входные данные 512-битовыми блоками, но, в отличие от MD5, он формирует 160-разрядный профиль сообщения. Типичный случай отправки Алисой несекретного, но подписанного сообщения Бобу показан на рис. 8.18. Открытый текст обрабатывается алгоритмом SHA-1, на выходе получается 160-битный хэш SHA-1. Он подписывается Алисой (закрытым ключом RSA) и отправляется вместе с открытым текстом Бобу.

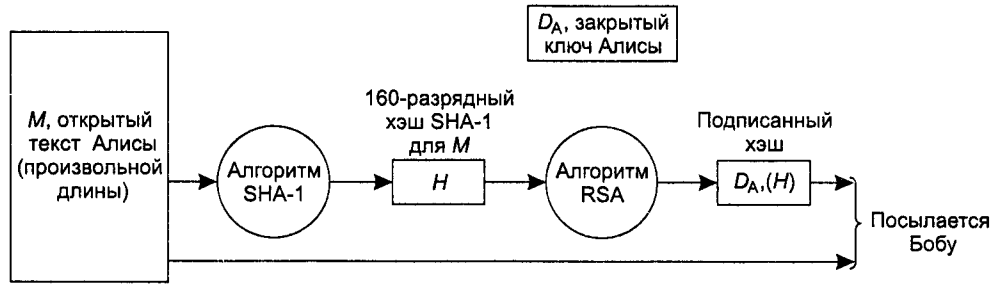


Рис. 8.18. Применение SHA-1 и RSA для создания подписей несекретных сообщений

При получении сообщения Боб сам вычисляет хэш-функцию с помощью алгоритма SHA-1 и применяет открытый ключ Алисы к подписанному хэшу для того, чтобы получить исходный хэш, *H*. Если они совпадают, сообщение считается корректным. Так как Труди не может, перехватив сообщение, изменить его таким образом, чтобы значение *H* совпадало с контрольным, Боб легко узнает обо всех подменах, которые совершила Труди. Для сообщений, чья неприкосновенность существенна, а секретность не имеет значения, часто применяется схема, показанная на рис. 8.18. При относительно небольших затратах на вычисления она гарантирует, что любые изменения, внесенные на пути следования сообщения, будут с высокой вероятностью выявлены.

Давайте теперь вкратце рассмотрим, как работает SHA-1. Для начала алгоритм SHA-1 также дополняет сообщение единичным битом в конце, за которым следует такое количество нулевых бит, чтобы в итоге получилось общее число битов, кратное 512. Затем 64-разрядное число, содержащее длину сообщения (до битового дополнения), логически складывается (операция ИЛИ) с 64 младшими битами. На рис. 8.19, *а* показано сообщение с дополнением, расположенным справа, потому что английский текст и рисунки читаются слева направо (то есть правая граница рисунка воспринимается как его конец, а левая — как начало). Применительно к вычислительной технике такое расположение соответствует обратному порядку хранения байтов (сначала передается самый значимый, старший бит).

Такая реализация присуща, например, SPARC. Однако вне зависимости от используемой техники SHA-1 вставляет битовое дополнение в конец сообщения.

Во время выполнения вычислений SHA-1 работает с пятью 32-битными переменными (H_0 ... H_4), в которых накапливается значение хэш-функции. Они показаны на рис. 8.19, *б*. Их начальные значения — это постоянные величины, определенные стандартом.

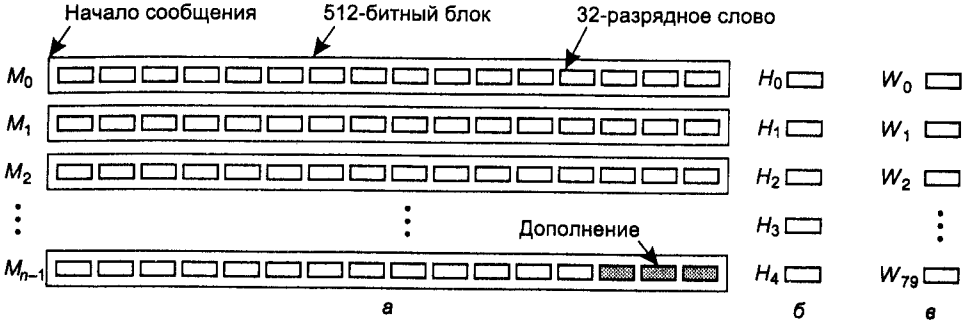


Рис. 8.19. Сообщение, дополненное до размера, кратного 512 битам (*а*); выходные переменные (*б*); массив слов (*в*)

Затем поочередно обрабатываются блоки с M_0 по M_{n-1}. Для текущего блока 16 слов сначала копируются в начало вспомогательного массива W размером 80 слов, как показано на рис. 8.19, *в*. 64 оставшихся слова вычисляются с использованием следующей формулы:

$$W_i = S^1(W_{i-3} \text{ XOR } W_{i-8} \text{ XOR } W_{i-14} \text{ XOR } W_{i-16}) \qquad (16 \le i \le 79),$$

где $S^b(W)$ представляет собой поворот 32-разрядного слова W на b бит. Теперь по значениям H_0 ... H_4 инициализируются переменные от A до E.

Сами вычисления на псевдо-С можно записать таким образом:

```
for(i=0;i<80;i++){
temp = S5(A) + f1(B, C, D) + E + W1 + K1;
E=D; D=C; C=S30(B); B=A; A=temp;
}
```

где постоянные K_i определяются стандартом. Смешивающие функции f_i задаются следующим образом:

$$f_i(B, C, D) = (B \text{ AND } C) \text{ OR } (\text{NOT } B \text{ AND } D) \qquad (0 \le i \le 19),$$

$$f_i(B, C, D) = B \text{ XOR } C \text{ XOR } D \qquad (20 \le i \le 39),$$

$$f_i(B, C, D) = (B \text{ AND } C) \text{ OR } (B \text{ AND } D) \text{ OR } (C \text{ AND } D) \qquad (40 \le i \le 59),$$

$$f_i(B, C, D) = B \text{ XOR } C \text{ XOR } D \qquad (60 \le i \le 79).$$

После 80 итераций цикла значения переменных A ... E добавляются к H_0 ... H_4 соответственно.

После обработки первого 512-битного блока начинается обработка следующего. Массив W инициализируется заново с помощью нового блока, однако H сохраняется неизменным. По окончании этого блока обрабатывается следующий, и так далее, пока все 512-разрядные блоки сообщения не попадут в эту кастрюлю. После обработки последнего блока пять 32-разрядных слов в массиве H выводятся в качестве 160-битного значения криптографической хэш-функции. Полный код SHA-1 приведен в RFC 3174.

В настоящее время идет работа над новыми версиями SHA-1 с 256-, 384- и 512-разрядными значениями хэш-функций.

Задача о днях рождения

В мире шифров многое оказывается совсем не таким, каким кажется на первый взгляд. Можно, например, предполагать, что для ниспровержения профиля сообщения, состоящего из m разрядов, потребуется порядка 2^m операций. На самом деле, часто оказывается достаточно $2^{m/2}$ операций, если использовать метод, основанный на **задаче о днях рождения**, опубликованный в ставшей классической книге (Yuval, 1979).

В основе идеи этого метода лежит задача, часто приводимая в качестве примера профессорами математики на курсах по теории вероятности. Вопрос: сколько студентов должно находиться в классе, чтобы вероятность появления двух человек с совпадающими днями рождения превысила 1/2? Большинство студентов обычно ожидают, что ответ будет значительно больше 100. На самом же деле, теория вероятности утверждает, что это число равно 23. Не вдаваясь в тонкости анализа этой проблемы, дадим интуитивно понятное объяснение: из 23 человек мы можем сформировать $(23 \cdot 22)/2 = 253$ различных пары, у каждой из которой дни рождения могут совпасть с вероятностью 1/365. Теперь этот ответ уже не кажется таким удивительным.

В более общем случае, если имеется некое соответствие между n входами (людьми, сообщениями и т. д.) и k возможными выходами (днями рождения, профилями сообщений и т. д.), мы имеем $n(n-1)/2$ входных пар. Если $n(n-1)/2 > k$, то вероятность того, что будет хотя бы одно совпадение выхода при различных входах, довольно велика. Таким образом, вероятность существования двух сообщений с одинаковыми профилями велика уже при $n > \sqrt{k}$. Это означает, что 64-разрядный профиль сообщения можно с большой вероятностью взломать (то есть найти два различных сообщения с одинаковым профилем), перебрав 2^{32} сообщений.

Рассмотрим практический пример. На кафедре компьютерных наук Государственного университета появились вакансия и два кандидата на эту должность, Том и Дик. Том работает на факультете на два года дольше Дика, поэтому его кандидатура будет рассматриваться первой. Если он получит эту должность, значит, Дику не повезло. Том знает, что заведующая кафедрой Мэрилин высоко ценит его работу, поэтому он просит ее написать для него рекомендательное письмо декану факультета, который будет решать дело Тома. После отправки все письма становятся конфиденциальными.

Мэрилин просит написать это письмо декану свою секретаршу Элен, подчеркивая, что она хотела бы видеть в этом письме. Когда письмо готово, Мэрилин просматривает его, подписывает 64-разрядной подписью и посылает декану. Позднее Элен может послать это письмо электронной почтой.

К несчастью для Тома, у Элен роман с Диком, и она хочет обмануть Тома. Поэтому она пишет следующее письмо с 32 вариантами в квадратных скобках.

Уважаемый господин декан,

Это [*письмо* | *обращение*] отражает мое [*искреннее* | *откровенное*] мнение о проф. Томе Уилсоне, являющемся [*кандидатом* | *претендентом*] на профессорскую должность [*в настоящее время* | *в этом году*]. Я [*знакома* | *работала*] с проф. Уилсоном в течение [*почти* | *около*] шести лет. Он является [*выдающимся* | *блестящим*] исследователем, обладающим [*большим талантом* | *большими возможностями*] и известным [*во всем мире* | *не только в нашей стране*] своим [*серьезным* | *созидательным*] подходом к [*большому числу* | *широкому спектру*] [*сложных* | *перспективных*] вопросов.

Он также является [*высоко* | *весьма*] [*уважаемым* | *ценимым*] [*преподавателем* | *педагогом*]. Его студенты дают его [*занятиям* | *лекциям*] [*самые высокие* | *высочайшие*] оценки. Он самый [*популярный* | *любимый*] [*преподаватель* | *учитель*] [*нашей кафедры* | *нашего университета*].

[*Кроме* | *Помимо*] того, [*гранты* | *контракты*] проф. Уилсона [*существенно* | *значительно*] пополнили [*фонды* | *финансовые запасы*] нашей кафедры. Эти [*денежные* | *финансовые*] средства [*позволили нам* | *дали возможность*] [*выполнить* | *осуществить*] [*много* | *ряд*] [*важных* | *специальных*] программ, [*таких как* | *среди которых*], государственная программа 2000 года. Без этих средств было бы невозможным продолжение этой программы, такой [*важной* | *значительной*] для нас. Я настойчиво рекомендую вам предоставить ему эту должность.

К несчастью для Тома, закончив печатать это письмо, Элен тут же принимается за второе:

Уважаемый господин декан,

Это [*письмо* | *обращение*] отражает мое [*искреннее* | *откровенное*] [*мнение* | *суждение*] о проф. Томе Уилсоне, являющемся [*кандидатом* | *претендентом*] на профессорскую должность в [*настоящее время* | *этом году*]. Я [*знакома* | *работала*] с проф. Уилсоном в течение [*почти* | *около*] шести лет. Он является [*слабым* | *недостаточно талантливым*] [*исследователем* | *ученым*], почти не известным в той области науки, которой он занимается. В его работах практически не заметно понимания [*ключевых* | *главных*] [*проблем* | *вопросов*] современности.

[*Более* | *Кроме*] того, он также не является сколько-нибудь [*уважаемым* | *ценимым*] [*преподавателем* | *педагогом*]. Его студенты дают его [*занятиям* | *лекциям*] [*самые низкие* | *негативные*] оценки. Он самый непопулярный [*преподаватель* | *учитель*] нашей кафедры, [*славящийся* | *печально известный*] своей [*привычкой* | *склонностью*] [*высмеивать* |*ставить в неудобное положение*] студентов, осмелившихся задавать вопросы на его [*лекциях* | *занятиях*].

[*Кроме* | *Помимо*] того, [*гранты* | *контракты*] проф. Уилсона [*почти* | *практически*] не пополняют [*фондов* | *финансовых запасов*] нашей кафедры. Если не удастся быстро найти новый источник финансирования, [*мы будем вынуждены* | *нам придется*] [*закрыть* | *прекратить*] [*много* | *ряд*] [*важных* | *специальных*] программ, [*таких как* | *среди которых*] государственная программа 2000 года. К сожалению, при таких [*условиях* | *обстоятельствах*] я не могу [*предлагать* | *рекомендовать*] его вам на эту должность.

Затем Элен заставляет свой компьютер сосчитать 2^{32} профиля сообщения для каждого варианта обоих писем, что занимает всю ночь. Есть шансы, что один профиль первого письма совпадет с одним из профилей второго письма. Если нет, она может добавить еще несколько вариантов слов и выражений в каждое письмо и попытаться еще раз за выходные. Предположим, что ей удалось найти такое совпадение. Назовем положительный отзыв письмом *A*, а отрицательный — письмом *B*.

Элен отправляет письмо *A* электронной почтой Мэрилин на утверждение. Мэрилин, конечно, утверждает письмо, вычисляет 64-разрядный профиль сообщения, подписывает профиль и посылает по почте подписанный профиль декану. Независимо Элен посылает декану письмо *B* (вместо письма *A*, которое следует отправить на самом деле).

Получив письмо и подписанный профиль, декан запускает алгоритм вычисления профиля сообщений для письма *B*, видит, что его профиль совпадает с тем, что ему прислала Мэрилин, и увольняет Тома. Он даже не может себе представить, что Элен удалось создать два письма с одинаковыми профилями сообщений и отправить ему совсем не тот вариант, который читала и подписала Мэрилин. (Вариант с хэппи-эндом, столь любимым американцами: Элен сообщает Дику о своих проделках. Потрясенный Дик порывает с ней. Элен в ярости бежит сознаваться во всем Мэрилин. Мэрилин звонит декану. В конце концов, Том получает профессуру.) При использовании алгоритма MD5 подобная атака шифра невозможна, так как даже если компьютер сможет вычислять по 1 млрд профилей в секунду, потребуется около 500 лет, чтобы перебрать по 2^{64} варианта для обоих писем, что все равно не даст 100-процентной гарантии совпадения. Конечно, если заставить параллельно работать 5000 компьютеров, вместо 500 лет потребуется 5 недель. В этом смысле SHA-1 подходит лучше (так как работает дольше).

Управление открытыми ключами

Криптография с использованием открытых ключей позволяет передавать секретные данные, не обладая общим ключом, а также создавать электронные подписи сообщений без необходимости привлечения третьей, доверительной стороны. Наконец, подписанные профили сообщений позволяют легко проверять целостность и аутентичность полученных данных.

Однако мы как-то слишком ловко обошли один вопрос: если Алиса и Боб друг друга не знают, как они смогут обменяться открытыми ключами перед началом общения? Вот, казалось бы, очевидное решение: выложить открытый ключ на веб-сайте. Однако так делать нельзя, и вот почему: представим себе, что Алиса хочет найти открытый ключ Боба на его веб-сайте. Каким образом она это

делает? Набирает URL сайта. Браузер ищет DNS-адрес домашней страницы Боба и посылает запрос *GET*, как показано на рис. 8.20. К сожалению, Труди в этот момент перехватывает запрос и посылает Алисе фальшивую страницу. В качестве таковой может выступать, к примеру, копия страницы Боба, на которой вместо его открытого ключа выложен открытый ключ Труди. После того как Алиса зашифрует свое первое сообщение с помощью E_T, Труди расшифрует его, прочтет, зашифрует с помощью открытого ключа Боба и перешлет сообщение Бобу, который даже не подозревает обо всех этих перипетиях. Но гораздо хуже то, что Труди может изменять сообщения перед повторной шифрацией и отправкой Бобу. Очевидно, нужен некий механизм секретного обмена открытыми ключами.

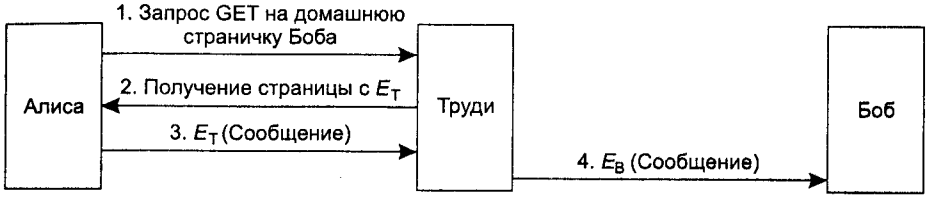


Рис. 8.20. Способ вторжения в систему с открытыми ключами

Сертификаты

Первая попытка организации защищенного обмена ключами может состоять в создании круглосуточного интернет-центра распространения ключей по требованию. Один из множества недостатков такого решения заключается в том, что данную систему не удастся масштабировать, и сам этот центр очень скоро станет узким местом. А если он не выдержит нагрузки, вся интернет-безопасность в один момент сойдет на нет.

По этой причине было придумано новое решение, не требующее, чтобы центр распространения ключей был доступен постоянно. В общем-то, даже не требуется, чтобы он вообще работал в подключенном (онлайновом) режиме. Вместо этого на него возлагается обязанность сертификации открытых ключей, принадлежащих как физическим, так и юридическим лицам. Организация, занимающаяся сертификацией открытых ключей, в настоящее время называется **Управлением сертификации** (CA — Certification Authority).

В качестве примера рассмотрим такую ситуацию: Боб хочет разрешить Алисе и другим лицам устанавливать с ним защищенные соединения. Он приходит в Управление сертификации со своим открытым ключом, а также паспортом или другим удостоверением личности и просит зарегистрировать ключ. Управление выдает ему сертификат, напоминающий тот, что показан на рис. 8.21, и подписывает хэш SHA-1 своим закрытым ключом. Затем Боб оплачивает услуги Управления и получает дискету с сертификатом и подписанным хэшем.

Основная задача сертификата состоит в связывании открытого ключа с именем принципала (физического или юридического лица). Сертификаты сами по себе никак не защищаются и не хранятся в тайне. Например, Боб может выложить его на свой сайт и поставить ссылку: «Здесь можно посмотреть на сертификат моего открытого ключа». Перейдя по этой ссылке, пользователь увидит и сертификат, и блок с подписью (подписанный хэш SHA-1 сертификата).

Настоящим удостоверяю, что открытый ключ
19836A8B03030CF83737E3837837FC3s87092827262643FFA82710382828282A
принадлежит
Роберту Джону Смиту
Университетская улица 12345
Беркли, CA 94702
1958 род. 5 июля 1958Кг.
Электронный адрес: bob@superdupernet.com

Хэш SHA-1 данного сертификата подписан закрытым ключом Управления сертификации

Рис. 8.21. Пример сертификата и подписанного хэша

Давайте теперь снова взглянем на сценарий, показанный на рис. 8.20. Что может сделать Труди, перехватив запрос страницы Боба, посланный Алисой? Она может выложить на странице свой собственный сертификат и блок с подписью на подложной странице, однако Алиса, читая этот сертификат, сразу догадается, что она разговаривает не с Бобом: в нем его имени просто нет. Труди может изменить домашнюю страницу Боба «на лету», заменив его открытый ключ своим собственным. И все же, проверив сертификат алгоритмом SHA-1, она получит значение хэш-функции, не согласующееся с тем, которое будет вычислено по открытому ключу Управления сертификации и блоку подписи. Так как Труди не имеет доступа к закрытому ключу Управления сертификации, она никак не может сгенерировать блок подписи, содержащий хэш модифицированной страницы с выложенным на ней открытым ключом Труди. Таким образом, Алиса может не беспокоиться о подлинности ключа Боба. И вот, как мы и обещали, при такой схеме не требуется, чтобы Управление сертификации постоянно работало в подключенном режиме. Тем самым ликвидируется потенциально узкое место системы.

Сертификат может связывать открытый ключ не только с принципалом, но и с **атрибутом**. Например, сертификат может содержать такую информацию: «Данный открытый ключ принадлежит лицу старше 18 лет». Этим можно подтвердить статус принципала или убедить окружающих в том, что ему разрешен доступ к некоторым специфическим данным, на которые накладываются возрастные ограничения. При этом имя принципала может и не раскрываться. Обычно владелец сертификата отсылает его на веб-сайт, принципалу или тому процессу, который обеспокоен возрастом клиента. В ответ генерируется случайное число, шифруемое с помощью открытого ключа (считываемого с сертификата). Если клиент сможет расшифровать его и отослать обратно, это будет служить подтверждением того, что он действительно обладает указанными в сертификате характеристиками. Еще это случайное число может быть использовано в качестве сеансового ключа для будущего соединения.

Сертификат может содержать атрибут еще в одном случае: если речь идет об объектно-ориентированной распределенной системе. Каждый объект обычно обладает некоторым набором методов. Владелец объекта может предоставлять каждому клиенту сертификат с указанием тех методов, которыми он может пользоваться. Этот список в виде поразрядной карты отображения информации (битовой карты) можно связать с открытым ключом, используя подписанный сертификат. Опять же, если владелец сертификата сможет подтвердить факт обладания соответствующим закрытым ключом, он сможет воспользоваться методами, указанными в списке. Особенностью такого использования сертификатов является отсутствие необходимости указывать имя владельца. Это бывает полезно в ситуациях, когда важна конфиденциальность.

X.509

Если бы все желающие подписать что-нибудь обращались в Управление идентификации, обслуживание клиентов с разного рода документами, требующими подписи, вскоре стало бы проблемой. Во избежание этого был разработан и утвержден организацией ITU специальный стандарт сертификатов. Он называется **X.509** и широко применяется в Интернете. В свет, начиная с 1988 года, вышло уже три версии стандарта, и мы будем рассматривать новейшую из них — третью.

На стандарт X.509 сильное влияние оказал мир OSI, в связи с чем в нем появились некоторые неприятные вещи, как, например, определенный принцип кодирования и именования. Удивительно, что проблемная группа по развитию Интернета, IETF, согласилась с данной концепцией, особенно учитывая то, что практически во всех областях, начиная с машинных адресов и заканчивая транспортными протоколами и форматами электронных писем, IETF всегда игнорировала OSI и пыталась сделать что-нибудь более толковое. IETF-версия X.509 описана в RFC 3280.

По сути, X.509 — это способ описания сертификатов. Основные поля сертификата перечислены в табл 8.3. Из описаний, приведенных в правой колонке, должно быть понятно, что для чего служит поле. За дополнительной информацией обращайтесь к RFC 2459.

Например, если Боб работает в отделе ссуд банка Money Bank, его X.500-адрес будет выглядеть так:

```
/C=US/O=MoneyBank/OU=Loan/CN=Bob/
```

где `C` — страна, `O` — организация, `OU` — отдел организации, `CN` — имя. Управление сертификации и другие сущности именуются похожим образом. Существенная проблема с системой именования X.500 заключается в том, что если Алиса пытается соединиться с *bob@moneybank.com* и имеет данные сертификата с именем в этом формате, для нее вовсе не очевидно, что этот сертификат имеет отношение именно к тому Бобу, который ей нужен. К счастью, начиная с третьей версии, разрешено использование имен DNS вместо X.500, поэтому данная проблема может счастливо разрешиться.

Сертификаты шифруются с использованием **системы записи абстрактного синтаксиса 1** (**ASN** — Abstract Syntax Notation) OSI. Ее можно рассматривать как нечто подобное структуре в языке C, за тем исключением, что эта запись очень странная и многословная. Более подробную информацию можно найти в (Ford и Baum, 2000).

Таблица 8.3. Основные поля сертификата в стандарте X.509

Поле	Значение
Version	Версия X.509
Serial number	Это число вместе с названием Управления сертификации однозначно идентифицирует сертификат
Signature algorithm	Алгоритм генерации подписи сертификата
Issuer	X.500-имя Управления
Validity period	Начало и конец периода годности
Subject name	Сущность, ключ которой сертифицируется
Public key	Открытый ключ сущности и идентификатор использующего его алгоритма
Issuer ID	Необязательный идентификатор, единственным образом определяющий эмитента (создателя) сертификата
Subject ID	Необязательный идентификатор, единственным образом определяющий владельца сертификата
Extensions	Различные возможные расширения
Signature	Подпись сертификата (генерируется с помощью закрытого ключа Управления сертификации)

Инфраструктуры систем с открытыми ключами

Понятно, что одного Управления сертификации на весь мир недостаточно. Оно бы быстро перестало функционировать из-за огромной нагрузки, да еще и стало бы эпицентром всех проблем, связанных с безопасностью сетей. Возможно, следует создать целый набор таких Управлений, использующих один и тот же закрытый ключ для подписания сертификатов, под покровительством одной и той же организации. Однако, хотя это и решит проблему распределения нагрузки, возникнет новый вопрос, связанный с *утечкой ключа*. Если по всему миру будут разбросаны десятки серверов, хранящих закрытый ключ Управления сертификации, велик шанс, что рано или поздно этот ключ будет украден или пропадет каким-то иным образом. Если ключ будет рассекречен, всю мировую инфраструктуру электронной безопасности можно будет похоронить. Вместе с тем, наличие всего одного центрального Управления сертификации — это тоже риск.

Далее, какая организация будет заведовать Управлением? Довольно трудно представить себе какую-либо законную структуру с большим кредитом доверия мирового масштаба. В некоторых странах предпочтительно, чтобы это было какое-нибудь правительственное учреждение, а где-то — наоборот, чтобы это было чем угодно, но не правительством.

По этим причинам был разработан альтернативный способ сертификации открытых ключей. Он известен под общим названием **PKI** (Public Key Infrastructure — инфраструктура систем с открытыми ключами). В этом разделе мы

рассмотрим только общие принципы действия PKI, поскольку было внесено множество предложений по ее модификации и некоторые детали могут со временем измениться.

PKI состоит из множества компонентов, среди которых Управления сертификации, сами сертификаты, а также каталоги. Инфраструктура систем с открытыми ключами предоставляет возможность структурной организации этих компонентов и определяет стандарты, касающиеся различных документов и протоколов. Одним из простейших видов PKI является иерархия Управлений, представленная на рис. 8.22. На рисунке представлены три уровня, однако реально их может быть как больше, так и меньше. Управление сертификации верхнего уровня (root) мы будем называть Центральным управлением (ЦУ). Центральное управление сертифицирует управления второго уровня — назовем их Региональными отделами (РО), — так как они могут обслуживать некоторый географический регион, например, страну или континент. Этот термин не стандартизован. Названия для уровней иерархии вообще не оговариваются стандартом. Региональные отделы, в свою очередь, занимаются легализацией реальных Управлений сертификации (УС), эмитирующих сертификаты стандарта X.509 для физических и юридических лиц. При легализации Центральным управлением нового Регионального отдела последнему выдается сертификат, подтверждающий его признание. Он содержит открытый ключ нового РО и подпись ЦУ. Аналогичным образом РО взаимодействуют с Управлениями сертификации: выдают и подписывают сертификаты, содержащие открытые ключи УС и признающие легальность деятельности.

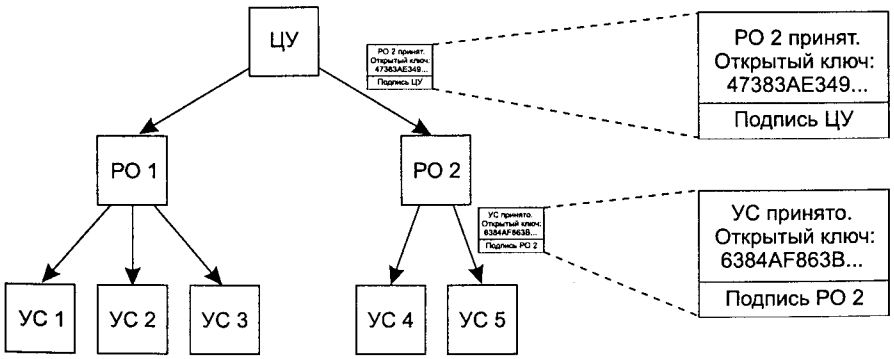


Рис. 8.22. Иерархия PKI (*а*); цепочка сертификатов (*б*)

Итак, наш PKI работает следующим образом. Допустим, Алисе нужен открытый ключ Боба, чтобы она могла с ним пообщаться. Она ищет и находит содержащий его сертификат, подписанный УС 5. Однако Алиса никогда ничего не слышала про УС 5. Этим «Управлением» может оказаться, на самом деле, десятилетняя дочка Боба. Алиса может отправиться в УС 5 и попросить подтвердить легитимность. Управление в ответ может показать сертификат, полученный от РО 2 и содержащий открытый ключ УС 5. Теперь, вооружившись открытым ключом УС 5, Алиса может удостовериться в том, что сертификат Боба действительно подписан УС 5, а значит, является легальным.

Если только РО 2 не является двенадцатилетним сыном Боба. Если Алисе вдруг придет в голову такая мысль, она может запросить подтверждение легитимности РО 2. Ответом будет служить сертификат, подписанный Центральным управлением и содержащий открытый ключ РО 2. Вот теперь Алиса может не сомневаться, что она получила открытый ключ Боба, а не кого-то другого.

А если Алиса хочет узнать открытый ключ ЦУ? Как это сделать? Загадка. Предполагается, что открытый ключ ЦУ знают все. Например, он может быть «зашит» внутрь ее браузера.

Но наш Боб — добряк, он не хочет доставлять Алисе лишние хлопоты. Он понимает, что она захочет проверить легитимность УС 5 и РО 2, поэтому он сам собирает соответствующие сертификаты и отправляет их ей вместе со своим. Теперь, зная открытый ключ ЦУ, Алиса может проверить по цепочке все интересующие ее организации. Ей не придется никого беспокоить для подтверждения. Поскольку все сертификаты подписаны, она может запросто уличить любые попытки подлога. Цепочка сертификатов, восходящая к ЦУ, иногда называется **доверительной цепочкой** или **путем сертификата**. Описанный метод широко применяется на практике.

Конечно, остается проблема определения владельца ЦУ. Следует иметь не одно Центральное управление, а несколько, причем связать с каждым из них свою иерархию региональных отделов и управлений сертификации. На самом деле, в современные браузеры действительно «зашиваются» открытые ключи более 100 центральных управлений, иногда называемые **доверительными якорями**. Как видите, можно избежать проблемы одного всемирного учреждения, занимающегося сертификацией.

Однако встает вопрос, какие доверительные якоря производители браузеров могут считать надежными, а какие — нет. Все, на самом деле, сводится к тому, насколько конечный пользователь доверяет разработчику браузера, насколько он уверен в том, что решения генерируются грамотно и доверительные якоря не принимаются от всех желающих за умеренную плату. Большинство браузеров обеспечивают возможность проверки ключей ЦУ (обычно это делается с помощью сертификатов, подписанных им) и удаления подозрительных ключей.

Каталоги

Инфраструктура систем с открытыми ключами должна решать еще один вопрос. Он касается места хранения сертификатов (и цепочек, ведущих к какому-нибудь доверительному якорю). Можно заставить всех пользователей хранить свои сертификаты у себя. Это безопасно (так как невозможно подделать подписанные сертификаты незаметно), но не слишком удобно. В качестве каталога для сертификатов было предложено использовать DNS. Прежде чем соединиться с Бобом, Алисе, видимо, все равно придется узнать с помощью службы имен доменов (DNS) его IP-адрес. Так почему бы не заставить DNS возвращать вместе с IP-адресом всю цепочку сертификатов?

Кому-то это кажется выходом из положения, однако некоторые считают, что лучше иметь специализированные серверы с каталогами для хранения сертификатов X.509. Такие каталоги могли бы с помощью имен X.500 обеспечивать возможность поиска. Например, теоретически можно представить себе услугу сервера каталогов, позволяющую получать ответы на запросы типа «Дайте мне полный список всех людей по имени Алиса, работающих в отделе продаж в любом месте США или Канады». Хранить такую информацию можно, например, при помощи LDAP.

Аннулирование

Реальный мир полон разного рода сертификатов, среди которых, например, паспорта, водительские удостоверения. Иногда эти сертификаты необходимо аннулировать (например, водительское удостоверение надо аннулировать за езду в нетрезвом состоянии). Та же проблема возникает и в мире цифровых технологий: лицо, предоставившее сертификат, может отозвать его за нарушение противоположной стороной каких-либо условий. Это необходимо делать и тогда, когда закрытый ключ, в сущности, перестал быть защищенным или, что еще хуже, ключ УС потерял кредит доверия. Таким образом, инфраструктура систем с открытыми ключами должна как-то обеспечивать процедуру аннулирования.

Первым шагом в этом направлении является принуждение всех УС к периодическому выпуску **списка аннулированных сертификатов** (CRL — Certificate Revocation List). В нем перечисляются порядковые номера всех аннулированных сертификатов. Поскольку в сертификатах содержится дата окончания срока годности, в CRL следует включать номера только тех из них, срок годности которых еще *не* истек. По истечении срока годности сертификаты перестают быть действительными автоматически, поэтому нужно различать случаи аннулирования «по старости» и по другим причинам. В любом случае их использование необходимо запрещать.

К сожалению, возникновение списков аннулированных сертификатов означает, что лицо, собирающееся использовать сертификат, должно вначале убедиться в том, что его нет в этом списке. В противном случае от использования надо отказаться. Тем не менее, сертификат мог быть аннулирован тотчас же после выпуска самого свежего варианта черного списка. Получается, что единственный надежный способ — это узнать о состоянии сертификата непосредственно у УС. Причем эти запросы придется посылать при каждом использовании сертификата, так как нет никакой гарантии, что его аннулирование не произошло несколько секунд назад.

Еще больше усложняет ситуацию то, что аннулированный сертификат иногда требуется восстанавливать. Например, если причиной отзыва была неуплата каких-нибудь взносов, после внесения необходимой суммы не остается никаких причин, которые не позволяли бы восстановить сертификат. Обработка ситуаций аннулирования и восстановления сводят на нет такое ценное свойство сертификатов, как возможность их использования без помощи УС.

Где хранить списки аннулированных сертификатов? Было бы здорово хранить их там же, где и сами сертификаты. Одна из стратегий подразумевает, что

УС периодически выпускает «черные» списки и заставляет вносить обновления в каталоги (удаляя отозванные сертификаты). Если для хранения сертификатов каталоги не используются, можно кэшировать их в разных удобных местах в сети. Поскольку «черный» список сам по себе является подписанным документом, любые попытки подлога тотчас будут замечены.

Если сертификаты имеют большие сроки годности, списки аннулированных сертификатов также будут довольно длинными. Например, количество отозванных кредитных карточек со сроком годности 5 лет будет гораздо больше списка отозванных трехмесячных карточек. Стандартным способом борьбы с длинными списками является довольно редкий выпуск самих списков и частый — обновлений к ним. Кроме всего прочего, это помогает снизить необходимую для распространения списков пропускную способность.

Защита соединений

Мы закончили изучение прикладных инструментов. Были описаны большинство применяемых методов и протоколов. Оставшаяся часть главы будет посвящена применению этих методов на практике для обеспечения безопасности сетей. Кроме того, будут высказаны некоторые мысли относительно социального аспекта этого вопроса.

Следующие четыре раздела посвящены безопасности соединений, то есть тому, как секретно и без риска подмены данных передавать биты от пункта отправления до пункта назначения, а также тому, как не пускать на линию посторонние биты. Это ни в коем случае не полный список проблем сетевой безопасности, однако перечисленные вопросы являются одними из самых важных.

IPsec

Проблемная группа IETF в течение многих лет мирилась с отсутствием безопасности в Интернете. Обеспечить ее было действительно непросто, и прежде всего потому, что разгорелись жаркие споры вокруг того, какую часть Интернета следует, собственно, защищать. Большинство экспертов по вопросам безопасности уверены в том, что по-настоящему надежная система должна выполнять сквозную шифрацию и сквозное обеспечение целостности данных (то есть все это должно быть сделано на прикладном уровне). Это означает, что процесс-источник шифрует и/или ставит защиту целостности данных и отправляет их процессу-приемнику, который, соответственно, дешифрует данные и проверяет их целостность. Тогда можно будет заметить любые попытки взлома (даже на уровне операционной системы на любой из сторон). Беда такого подхода в том, что для обеспечения безопасности требуется вносить изменения во все приложения. Это означает, что необходимо «спустить» шифрацию на транспортный уровень или организовать новый специализированный подуровень между прикладным и транспортным уровнями. Он должен быть сквозным, но в то же время не требующим внесения изменений в приложения.

Противоположная точка зрения состоит в том, что пользователи все равно не осознают необходимости применения мер безопасности и просто не способны корректно использовать все предоставленные возможности. При этом никто не захочет каким-либо образом изменять существующие программы, поэтому сетевой уровень должен выполнять проверку подлинности и/или шифровать сообщения незаметно для пользователя. Долгие годы сражений привели к победе этой точки зрения: был разработан стандарт безопасности, ориентированный на сетевой уровень. Одним из аргументов было то, что шифрование на сетевом уровне, с одной стороны, не помешает тем пользователям, которые серьезно относятся к безопасности, и а с другой — в некоторой степени убережет беспечных пользователей.

Результатом всех этих дискуссий было создание стандарта **IPsec** (IP security — IP-безопасность), описанного в RFC 2401, 2402, 2406 и др. Не всем пользователям требуется шифрация соединений (выполнение соответствующих процедур может занимать существенную часть вычислительных ресурсов). Однако вместо того чтобы делать шифрацию необязательной, пользователю предлагается в случае необходимости выбирать пустой алгоритм. В RFC 2410 расписываются такие достоинства пустого алгоритма, как простота, легкость реализации и высокая скорость.

IPsec служит основой для множества услуг, алгоритмов и модулей. Причиной наличия множества услуг является то, что далеко не все хотят постоянно платить за все возможные услуги, поэтому нужные сервисы предоставляются порционно. Основные услуги таковы: секретность, целостность данных, защита от взлома методом повторения сообщений (когда жулик повторяет подслушанный разговор). Все это основано на криптографии с симметричными ключами, поскольку здесь критична высокая производительность.

Для чего нужен целый набор алгоритмов? Дело в том, что считающийся сегодня надежным алгоритм завтра может быть сломан. Если сделать IPsec независимым от конкретного алгоритма, стандарт выживет даже в случае взлома одного из алгоритмов.

Для чего нужны разные модули? Для того чтобы можно было защищать и одно TCP-соединение, и весь трафик между парой хостов, и весь трафик между парой защищенных хостов, и т. д.

Несколько удивительно, что IPsec ориентирован на соединение, несмотря на его присутствие на уровне IP. На самом деле, это не так странно, как кажется. Ведь безопасность можно обеспечить только созданием ключа и использованием его в течение какого-то времени. А это, по сути дела, разновидность соединения. К тому же все соединения погашают расходы на их установление за счет передачи большого количества пакетов. «Соединение» в контексте IPsec называется **защищающей связью** (security connection). Защищающая связь — это симплексное соединение между двумя конечными точками, с которым связан специальный идентификатор защиты. Если требуется передача защищенных данных в обоих направлениях, понадобятся две защищающие связи. Идентификаторы защиты передаются в пакетах, следующих по этим надежным соединениям, и используются по прибытии защищенных пакетов для поиска ключей и другой важной информации.

Технически IPsec состоит из двух основных частей. Первая описывает два новых заголовка, которые можно добавлять к пакету для передачи идентификатора защиты, данных контроля целостности и другой информации. Вторая часть, **ISAKMP** (Internet Security and Key Management Protocol — интернет-безопасность и протокол управления ключами), предназначена для создания ключей. Мы не станем вдаваться в подробности устройства ISAKMP, потому что, во-первых, это очень сложная тема и, во-вторых, основной протокол **IKE** (Internet Key Exchange — обмен ключами в Интернете) работает очень некорректно и требует замены (Perlman и Kaufman, 2000).

IPsec может работать в двух режимах. В **транспортном режиме** заголовок IPsec вставляется сразу за заголовком IP. Поле *Protocol* заголовка IP изменяется таким образом, чтобы было понятно, что далее следует заголовок IPsec (перед заголовком TCP). В заголовке IPsec содержится информация, касающаяся безопасности, — в частности, идентификатор защищающей связи, новый порядковый номер и, возможно, проверка целостности поля полезной нагрузки.

В **режиме туннелирования** весь IP-пакет вместе с заголовком вставляется внутрь нового IP-пакета с совершенно новым заголовком. Этот режим хорош тогда, когда туннель заканчивается где-нибудь вне конечного пункта. В некоторых случаях концом туннеля является шлюз, обеспечивающий безопасность, например, корпоративный брандмауэр. В этом режиме брандмауэр вставляет и извлекает пакеты, проходящие через него в разные стороны. При такой организации машины ЛВС компании гарантированно будут обслужены по стандарту IPsec. Об этом совершенно не приходится беспокоиться: все заботы берет на себя брандмауэр.

Еще режим туннелирования полезен, когда несколько TCP-соединений объединяются вместе и обрабатываются в виде единого шифрованного потока, поскольку в данном случае взломщик не может узнать, кто и кому передает пакеты, а также в каком количестве. А ведь иногда даже объем трафика, передаваемого одним лицом другому, является ценной информацией. Например, если во время военного кризиса трафик между Пентагоном и Белым домом резко снижается и при этом так же резко растет трафик между Пентагоном и какой-нибудь военной базой в Колорадо, перехватчик может сделать из этого далеко идущие выводы.

Изучение структуры потока по проходящим пакетам называется **анализом трафика**. Если используется туннелирование, такой анализ становится задачей весьма сложной. Недостаток режима туннелирования заключается в том, что приходится расширять заголовок IP-пакетов, за счет чего заметно возрастает суммарный размер пакетов. В транспортном режиме размер пакетов изменяется незначительно.

Первый из новых заголовков называется **заголовком идентификации** (AH — Authentication Header). С его помощью проверяется целостность данных и выполняется защита от взлома путем повторной передачи. Однако он не имеет никакого отношения к секретности (то есть шифрации данных). Применение AH в транспортном режиме показано на рис. 8.23. В стандарте IPv4 он располагается

между заголовком IP (вместе со всеми необязательными полями) и заголовком TCP. В IPv6 это просто еще один дополнительный заголовок. Так он и воспринимается. Формат AH действительно очень близок к формату дополнительного заголовка IPv6. К полезной нагрузке иногда добавляют заполнение, чтобы достичь определенной длины, необходимой алгоритму идентификации. Это показано на рисунке.

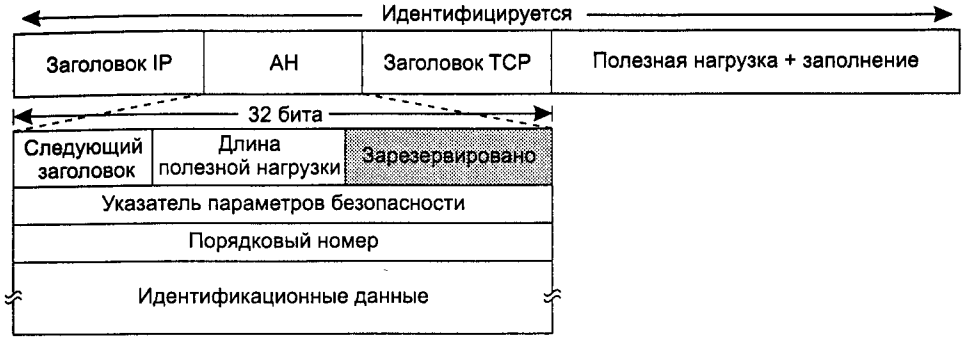


Рис. 8.23. Заголовок идентификации IPsec в транспортном режиме для IPv4

Рассмотрим заголовок AH. Поле *Следующий заголовок* хранит предыдущее значение, которое в поле *Протокол* заголовка IP ранее было заменено на 51, чтобы показать, что далее следует заголовок AH. Обычно здесь встречается код для TCP (6). Поле *Длина полезной нагрузки* хранит количество 32-разрядных слов заголовка AH минус 2.

Поле *Указатель параметров безопасности* — это идентификатор соединения. Он вставляется отправителем и ссылается на конкретную запись в базе данных у получателя. В этой записи содержится общий ключ и другая информация данного соединения. Если бы этот протокол был придуман ITU, а не IETF, это поле, скорее всего, называлось бы *Номером виртуального канала.*

Поле *Порядковый номер* применяется для нумерации всех пакетов, посылаемых по защищенной связи. Все пакеты получают уникальные номера, даже если они посылаются повторно. Имеется в виду, что повторно передаваемый пакет имеет номер, отличный от номера оригинального пакета (даже если порядковый номер TCP тот же самый). Это поле служит для предотвращения взлома путем повторной передачи. Порядковые номера никогда не повторяются. Если же окажутся использованными все 2^{32} номера, для продолжения общения устанавливается новая защищающая связь.

Наконец, поле переменной длины *Данные идентификации* содержит цифровую подпись, вычисляемую относительно полезной нагрузки. При установке защищающей связи стороны договариваются об используемом алгоритме генерации подписей. Чаще всего здесь не применяется шифрование с открытыми ключами, так как все известные алгоритмы этого типа работают слишком медленно, а пакеты необходимо обрабатывать с очень большой скоростью. Протокол IPsec основан на шифровании с симметричными ключами, поэтому перед установкой защищающей связи отправитель и получатель должны договориться о значении общего ключа, применяемого при вычислении подписи. Один из простейших способов заключается в вычислении хэш-функции для пакета и общего ключа. Отдельно общий ключ, конечно, не передается. Подобная схема называется **HMAC** (Hashed Message Authentication Code — код идентификации хэшированного сообщения). Вычисление этого кода выполняется гораздо быстрее, чем последовательный запуск SHA-1 и RSA.

Заголовок AH не позволяет шифровать данные. Его основная польза выявляется, когда важна проверка целостности, но не нужна секретность. Стоит отметить, что при проверке целостности при помощи AH охватываются некоторые поля заголовка IP, в частности, те из них, которые не изменяются при прохождении пакета от маршрутизатора к маршрутизатору. Поле *Время жизни*, например, меняется при каждой пересылке через маршрутизатор, поэтому его нельзя охватить при проверке целостности. Однако IP-адрес источника охватывается, тем самым предотвращается возможность его подмены взломщиком.

Альтернативой заголовку IPsec служит заголовок **ESP** (Encapsulating Security Payload — инкапсулированная защищенная полезная нагрузка). Как показано на рис. 8.24, этот заголовок может применяться как в транспортном режиме, так и в режиме туннелирования.

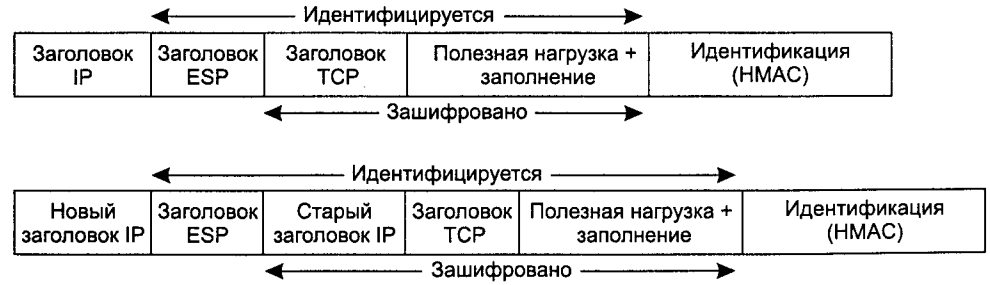


Рис. 8.24. ESP в транспортном режиме (*а*); ESP в режиме туннелирования (*б*)

Заголовок ESP состоит из двух 32-разрядных слов: *Указателя параметров безопасности* и *Порядкового номера.* Мы их уже встречали в заголовке AH. Третье слово, которое обычно следует за ними, однако технически не является частью заголовка, — это *Вектор инициализации* (если только не применяется пустой алгоритм шифрования, тогда это поле опускается).

ESP, как и AH, обеспечивает проверку целостности при помощи HMAC, однако вместо того, чтобы включать хэш в заголовок, его вставляют после поля полезной нагрузки. Это видно на рис. 8.24. Такое расположение полей дает преимущество при аппаратной реализации метода. Оно заключается в том, что HMAC может подсчитываться во время передачи битов полезной нагрузки по сети и добавляться к ним в конец. Именно поэтому в Ethernet и других стандартах локальных сетей циклический контроль избыточности вставляется в концевик, а не в заголовок. При применении заголовка AH пакет приходится буферизировать и вычислять подпись, только после его можно отправлять. Это потенциаль-

но приводит к уменьшению числа пакетов, которые можно передать за единицу времени.

Казалось бы, если ESP умеет делать все то же самое, что и AH, и даже больше, причем он еще и гораздо эффективнее, тогда зачем мучиться с AH? Причины этого в основном исторические. Изначально заголовок AH обеспечивал только проверку целостности, а ESP — только секретность. Позднее ESP научили использовать для проверки целостности, но разработчики AH не хотели, чтобы он канул в Лету после всей той работы, которую они проделали. Единственный аргумент в пользу AH заключается в том, что с его помощью можно частично проверять заголовок IP, чего не умеет ESP. И все же это аргумент довольно слабый. Еще один сомнительный аргумент состоит в том, что система, поддерживающая AH, но не поддерживающая ESP, возможно, будет иметь меньше проблем при получении лицензии на экспорт, поскольку этот заголовок не имеет отношения к секретности и шифрованию. Похоже, что AH со временем все-таки исчезнет с горизонта.

Брандмауэры

Возможность соединять любые компьютеры друг с другом в некоторых случаях является достоинством, а в других, наоборот, недостатком. Возможность бродить по Интернету доставляет много радости домашним пользователям. Менеджерам отдела безопасности корпораций эта возможность кажется кошмаром. Большинство компаний располагает огромными объемами конфиденциальной информации, размещенной на компьютерах, подключенных к сети, — коммерческие тайны, планы развития производства, рыночные стратегии, аналитические отчеты финансового состояния и т. д. Раскрытие этих сведений перед конкурентами может иметь ужасные последствия.

Помимо опасности утечки информации наружу, имеется опасность проникновения вредной информации, такой как вирусы, черви и прочей цифровой заразы, способной взламывать секреты, уничтожать ценные данные, на борьбу с которой уходит масса времени сетевых администраторов. Часто эту инфекцию заносят беззаботные сотрудники, желающие поиграть в новую модную компьютерную игру.

Таким образом, требуются специальные средства, удерживающие «доброкачественную» информацию внутри, а «вредную» — снаружи. Один из способов состоит в применении IPsec. Этот метод защищает данные при их пересылке. Однако шифрование не спасает от вирусов и хакеров, способных проникнуть в локальную сеть. Помочь защитить сети от нежелательного проникновения снаружи может установка брандмауэров, к рассмотрению которых мы сейчас обратимся.

Брандмауэры представляют собой современную реализацию средневекового принципа обеспечения безопасности. Они напоминают ров, вырытый вокруг замка. Суть конструкции заключается в том, что все входящие и выходящие из замка должны проходить по одному подъемному мосту, где полиция ввода-вывода сможет проверить их личность. Тот же принцип может быть применен и в сетях:

у компании может быть несколько локальных сетей, соединенных произвольным образом, но весь внешний трафик должен проходить через электронный подъемный мост (брандмауэр), как показано на рис. 8.25.

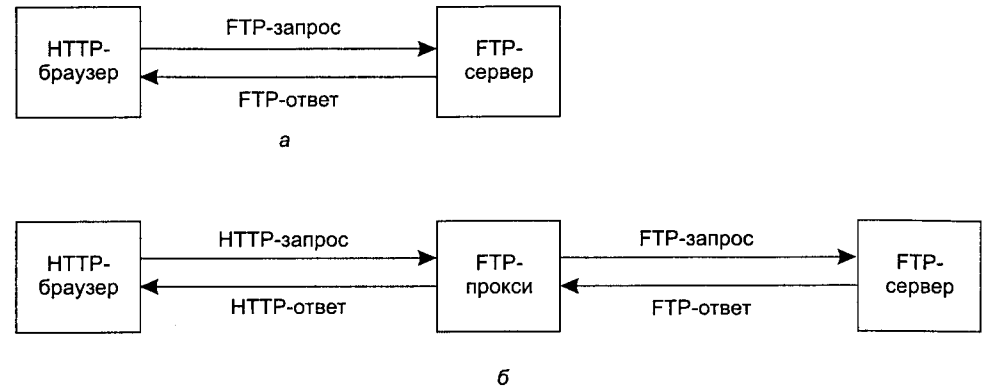


Рис. 8.25. Брандмауэр, состоящий из двух пакетных фильтров и шлюза прикладного уровня

Брандмауэр в данной конфигурации состоит из двух компонентов: двух маршрутизаторов, фильтрующих пакеты, и шлюза прикладного уровня. Существуют также и более простые конструкции, но преимущество такой разработки состоит в том, что каждый пакет, желающий войти или выйти, должен пройти через два фильтра и один шлюз прикладного уровня. Других путей нет. Читатели, полагающие, что достаточно одного контрольно-пропускного пункта, видимо, давно не летали международными авиалиниями.

Каждый **пакетный фильтр** представляет собой стандартный маршрутизатор с расширенными функциями, позволяющими анализировать входящие и выходящие пакеты. Пакеты, удовлетворяющие определенным критериям, пропускаются сквозь фильтр. Не сумевшие пройти проверку пакеты удаляются.

Показанный на рис. 8.25 пакетный фильтр внутренней локальной сети проверяет выходящие пакеты, а пакетный фильтр внешней локальной сети проверяет входящие пакеты. Пакеты, преодолевшие первый барьер, проходят к шлюзу прикладного уровня для дальнейшего исследования. Размещение двух фильтров в разных локальных сетях позволяет гарантировать, что ни один пакет не попадет из одной сети в другую, не пройдя через шлюз прикладного уровня. Обходного пути вокруг него нет.

Пакетные фильтры обычно управляются таблицами, настраиваемыми системным администратором. В этих таблицах перечислены допустимые и блокируемые отправители и получатели, а также правила, описывающие действия над исходящими и входящими пакетами.

В общем случае настроек TCP/IP информация о получателе или отправителе состоит из IP-адреса и номера порта. Номера портов определяют требуемую службу. Например, порт 23 используется для программы Telnet, порт 79 — для Finger, а порт 119 — для новостей сети USENET. Компания может заблокировать все входящие пакеты для комбинаций всех IP-адресов с одним из этих но-

меров портов. Таким образом, никто посторонний не сможет войти в сеть через Telnet или просмотреть список текущих пользователей сети с помощью программы Finger. Кроме того, компания может таким образом не допустить, чтобы ее сотрудники весь день читали новости USENET.

Блокирование исходящих пакетов сложнее. Несмотря на то что названия большинства сайтов чаще всего соответствуют стандартным соглашениям об именах, никто не обязывает их придерживаться. Кроме того, для некоторых важных служб, таких как FTP (File Transfer Protocol — протокол передачи файлов), номера портов назначаются динамически. Более того, хотя блокирование TCP-соединения является непростым делом, блокировать UDP-пакеты еще тяжелее, так как почти ничего нельзя сказать заранее о том, что они собираются делать. Многие пакетные фильтры по этой причине просто запрещают UDP-трафик совсем.

Вторая составляющая механизма брандмауэра представляет собой **шлюз прикладного уровня**. Вместо того чтобы просто разглядывать пакеты, этот шлюз работает на прикладном уровне. Например, может быть установлен почтовый шлюз, просматривающий каждое входящее и выходящее сообщение. Каждое сообщение пропускается или отвергается в зависимости от содержимого полей заголовков, размера сообщения и даже содержимого (например, шлюз, работающий на военном объекте, может реагировать особым образом на ключевые слова вроде «атомная» или «бомба»).

Может быть одновременно установлено несколько шлюзов для специфических приложений, тем не менее, осторожные организации нередко разрешают обмен электронной почтой и даже WWW, однако запрещают все остальное как слишком рискованное. В сочетании с шифрованием и фильтрацией пакетов подобные меры обеспечивают некоторый уровень безопасности ценой некоторого неудобства.

Даже в случае идеально настроенного брандмауэра остается множество проблем, связанных с безопасностью. Например, если входящие пакеты пропускаются только со стороны конкретных сетей (например, со стороны ЛВС дочерней фирмы компании), взломщик, находящийся вне зоны действия брандмауэра, может просто фальсифицировать адрес отправителя и тем самым преодолеть барьер. Если же нечестный сотрудник компании решит переслать секретную документацию, он может зашифровать ее или вообще сфотографировать, и тогда эти данные смогут проникнуть через любые лингвистические анализаторы. Мы даже не обсуждаем тот факт, что в 70 % случаев мошенники находятся в зоне действия брандмауэра. Очень часто ими являются недовольные сотрудники (Schneier, 2000).

К тому же, существует целый класс атак, с которыми не способны справиться никакие брандмауэры. Идея, лежащая в основе брандмауэров, заключается в том, чтобы не давать взломщикам проникнуть в систему, а секретным данным — уходить наружу. К сожалению, в мире есть много людей, которые не могут найти себе лучшего занятия, нежели препятствовать работоспособности сайтов. Они отправляют вполне легитимные сообщения до тех пор, пока сайт не перестанет функционировать из-за чрезмерной нагрузки. Например, такое хулиганство может заключаться в рассылке пакетов *SYN* для установки соединений. Сайт выделит часть таблицы под это соединение и пошлет в ответ пакеты *SYN + ACK*. Если взломщик не ответит, табличная запись будет продолжать оставаться зарезервированной в течение нескольких секунд до наступления тайм-аута. Если одновременно посылаются тысячи запросов на соединение, никакие запросы от честных граждан просто не пробьются к серверу, так все ячейки таблицы окажутся заняты. Атаки, целью которых является нарушение деятельности объекта, а не получение секретных данных, называются атаками типа **DoS** (Denial of Service — отказ в обслуживании (запроса) — сравните с сокращением QoS — качество обслуживания). Обычно адрес отправителя в пакетах с запросами фальсифицирован, поэтому найти вандала не так просто.

Существует и более жестокий вариант такой атаки. Если сетевому хулигану уже удалось взломать несколько сотен компьютеров, расположенных по всему миру, он может приказать им всем забивать запросами один и тот же сервер. Тем самым не только повышается «убойная сила», но и уменьшаются шансы на обнаружение негодяя, так как пакеты приходят с самых разных компьютеров, ничем плохим себя ранее не зарекомендовавших. Этот тип атаки носит название **DDoS** (Distributed Denial of Service — распределенный отказ в обслуживании). С этой напастью бороться трудно. Даже если атакуемая машина сможет быстро распознать поддельный запрос, на его обработку и отвержение потребуется некоторое время, в течение которого придут другие запросы, и в итоге центральный процессор будет постоянно занят их обработкой.

Виртуальные частные сети

Многие компании владеют множеством подразделений, расположенных в разных городах, иногда даже в разных странах. До появления общедоступных сетей передачи данных обычным делом было арендовать выделенную телефонную линию для организации связи между некоторыми или всеми парами подразделений. В некоторых компаниях такой подход применяется до сих пор. Сеть, состоящая из компьютеров, принадлежащих компании, и выделенных телефонных линий, называется **частной сетью**. Пример частной сети, соединяющей три подразделения, показан на рис. 8.26, *а*.

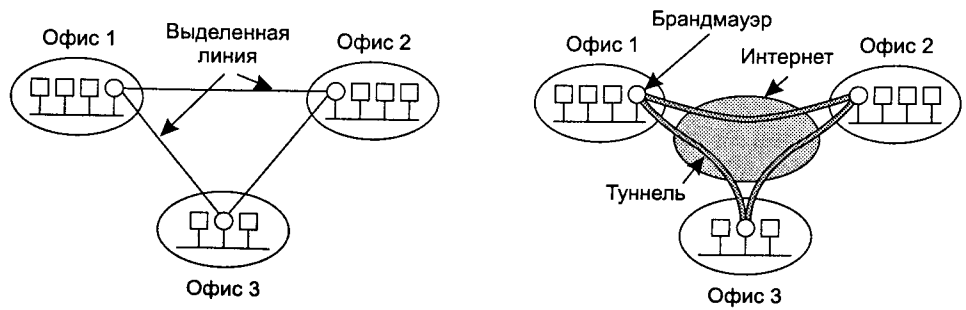


Рис. 8.26. Частная сеть на основе выделенной линии (*а*); виртуальная частная сеть (*б*)

Частные сети работают хорошо и обладают высокой защищенностью. Если бы были доступны только выделенные линии, то отсутствовала бы проблема утечки трафика, и взломщикам пришлось бы физически подключаться к линиям, чтобы перехватить данные, а это не так просто. Беда в том, что стоимость аренды одного выделенного канала T1 составляет тысячи долларов (в месяц!), а аренда линии T3 во много раз дороже. Когда появились общедоступные сети передачи данных, у компаний возникло естественное желание воспользоваться ими для передачи данных (а может, и голоса). При этом, правда, не хотелось терять свойства защищенности, присущие частной сети.

Это соображение вскоре привело к изобретению **виртуальных частных сетей** (VPN — Virtual Private Networks), которые являются оверлейными сетями, работающими поверх обычных общедоступных сетей, но обладающими свойствами частных сетей. Они называются «виртуальными», потому что такие сети — это почти иллюзия; аналогичным образом виртуальные каналы — это не реальные каналы, а виртуальная память — это не реальная память.

Хотя виртуальные частные сети могут строиться на основе ATM (или сетей с коммутацией кадров — frame relay), все более популярным становится организация VPN прямо в Интернете. При этом обычно каждый офис оборудуется брандмауэром и создаются интернет-туннели между всеми парами офисов, как показано на рис. 8.26, *б*. Если IPsec работает в режиме туннелирования, можно собрать весь трафик между любыми двумя парами офисов в один надежный поток и установить защищающую связь, обеспечив тем самым контроль целостности, секретности и даже определенный иммунитет против анализа трафика.

При запуске системы каждая пара брандмауэров должна договориться о параметрах защищающей связи, таких как набор услуг, режимов, алгоритмов и ключей. Во многие брандмауэры встроен специальный инструментарий для работы с виртуальными частными сетями, но можно построить систему и на обычных маршрутизаторах. Тем не менее, поскольку брандмауэры — это почти неотъемлемая часть систем сетевой безопасности, вполне естественно начинать и заканчивать туннели именно на брандмауэрах, проводя четкую границу между компанией и Интернетом. Таким образом, наиболее распространенная комбинация подразумевает наличие брандмауэров, виртуальных частных сетей и IPsec с ESP в режиме туннелирования.

После установки защищающей связи начинается передача данных. С точки зрения маршрутизатора, работающего в Интернете, пакет, проходящий по туннелю VPN, — это самый обычный пакет. Единственное, что его отличает от остальных, это наличие заголовка IPsec после заголовка IP. Но поскольку дополнительные заголовки на процесс пересылки никак не влияют, маршрутизаторы не сильно беспокоит заголовок IPsec.

Основное преимущество такой организации виртуальной частной сети состоит в том, что она совершенно прозрачна для всего пользовательского ПО. Установкой и управлением защищающих связей занимаются брандмауэры. Единственный человек, которому есть дело до настройки сети, — это системный администратор, который обязан сконфигурировать и поддерживать брандмауэры. Для всех остальных виртуальная частная сеть мало чем отличается от частной сети на основе выделенной линии. Более подробно про VPN написано в (Brown, 1999; Izzo, 2000).

Безопасность в беспроводных сетях

Оказывается, удивительно просто создать систему, которая логически полностью надежна, то есть состоит из VPN и брандмауэров, и при этом на практике протекает, как решето. Такая ситуация может возникнуть, если в сети есть беспроводные машины, передающие данные с помощью радиосигнала, проходящего прямо над брандмауэром в обе стороны. Радиус действия сетей типа 802.11 может составлять несколько сотен метров, поэтому шпион, желающий перехватить информацию, может просто приехать на автостоянку перед зданием фирмы, оставить в машине ноутбук с приемопередатчиком 802.11, записывающим все, что слышно в эфире, и пойти гулять по городу. К вечеру на жестком диске он обнаружит массу ценной информации. Теоретически такого происходить не должно. Правда, теоретически же ограбления банков тоже не должны происходить.

За многие проблемы безопасности стоит сказать спасибо производителям беспроводных базовых станций (точек доступа), пытающихся сделать свою продукцию дружественной по отношению к пользователю. Обычно, если пользователь вынимает свое устройство из сумки и вставляет в розетку, оно сразу начинает работать, и практически всегда все окружающие в зоне действия радиопередатчика смогут услышать любые секреты, о которых он проболтается. Если же затем это устройство подключить к Ethernet, весь трафик, проходящий по локальной сети, может быть перехвачен ноутбуком, стоящим в припаркованной неподалеку машине. Беспроводная связь — это мечта шпиона, ставшая реальностью: информация сама идет в руки, только успевай ее ловить. Очевидно, что вопрос безопасности в беспроводных сетях стоит куда острее, чем в проводных. В этом разделе мы рассмотрим некоторые методы, позволяющие в какой-то мере обезопасить системы такого рода. Дополнительную информацию можно найти в (Nichols и Lekkas, 2002).

Безопасность в сетях 802.11

Стандарт 802.11 описывает протокол безопасности уровня передачи данных под названием **WEP** (Wired Equivalent Privacy — секретность, эквивалентная проводным сетям), предназначенный для того, чтобы обезопасить беспроводные ЛВС так же надежно, как и проводные. По умолчанию в проводных сетях вопрос безопасности отсутствует как таковой, поэтому добиться этой цели несложно, и WEP, как мы увидим далее, справляется со своей задачей.

При наличии системы безопасности в сети 802.11 каждая станция имеет общий закрытый ключ с базовой станцией. Метод распространения ключей стандартом не оговаривается. Скажем, они могут быть прошиты в устройствах или программах производителем. Ими можно обменяться заранее по проводной сети. Наконец, либо базовая станция, либо пользовательская машина может случайным образом выбирать ключ и отсылать его противоположной стороне, предварительно зашифровав при помощи открытого ключа этой стороны. После уста-

новки ключи могут оставаться неизменными в течение нескольких месяцев или даже лет.

Шифрация при помощи WEP использует потоковый шифр, основанный на алгоритме RC4. Создателем RC4 был Роналд Ривест (Ronald Rivest). Этот алгоритм хранился в тайне до тех пор, пока в 1994 году он не просочился в Интернет. Как мы уже упоминали, практически нереально сохранить какой-либо алгоритм в тайне, даже с такой скромной целью, как соблюдение закона об интеллектуальной собственности (случай RC4), не говоря уже о том, чтобы сохранить его в тайне от взломщиков (а такой задачи создатели RC4 перед собой даже не ставили). В WEP RC4 генерирует потоковый шифр, который суммируется по модулю 2 с открытым текстом, в результате чего получается шифрованный текст.

Полезная нагрузка каждого пакета шифруется с использованием метода, показанного на рис. 8.27. Вначале проверяется контрольная сумма (по многочлену CRC-32), которая добавляется к полезной нагрузке. Так формируется открытый текст, передаваемый алгоритму шифрования. Этот открытый текст складывается по модулю 2 с отрезком ключевого потока, равного ему по размеру. Результатом этих преобразований является зашифрованный текст. Вектор инициализации, необходимый для запуска RC4, передается вместе с шифром. После получения пакета приемник извлекает из него зашифрованные данные (полезную нагрузку), создает ключевой поток из общего закрытого ключа и только что принятого вектора инициализации, затем ключевой поток суммируется по модулю 2 с полезной нагрузкой, что позволяет восстановить открытый текст. Наконец, можно проверить контрольную сумму, чтобы убедиться в подлинности принятой информации.

На первый взгляд, такой подход кажется довольно убедительным, однако метод его взлома уже опубликован (Borisov и др., 2001). Далее будут подведены итоги этого. Во-первых, как ни странно, очень многие используют одинаковые общие ключи для всех пользователей, из-за этого все пользователи могут запросто читать весь трафик друг друга. Это, конечно, подход вполне эквивалентный подходу, принятому вEthernet, однако не слишком безопасный.

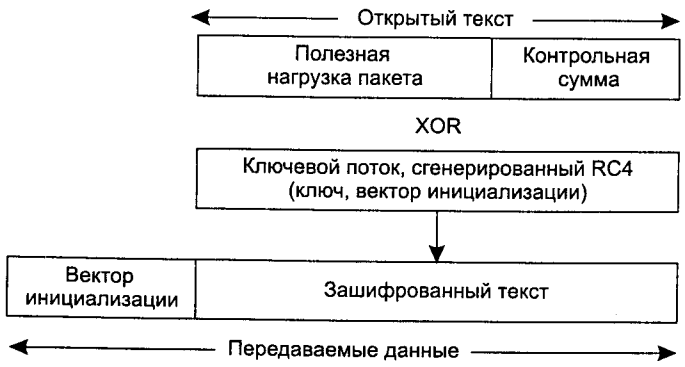


Рис. 8.27. Шифрация пакета с использованием WEP

Но даже если всем пользователям раздать разные ключи, WEP все равно может быть взломан. Так как ключи не изменяются в течение больших периодов времени, стандарт WEP рекомендует (но не обязывает) изменять вектор инициализации при передаче каждого пакета во избежание атак посредством повторного использования (мы обсуждали это в разделе «Режимы шифрования»). К сожалению, многие сетевые карты стандарта 802.11 для ноутбуков сбрасывают вектор инициализации в 0, когда ее вставляют в разъем, и увеличивают на единичку с каждым пересылаемым пакетом. Так как сетевые карты вставляются и вынимаются весьма часто, малые числа, выступающие в качестве векторов инициализации, — обычное дело. Если Труди удастся собрать несколько пакетов, посланных одним и тем же пользователем, с одинаковыми значениями вектора инициализации (который сам по себе посылается открытым текстом вместе с пакетом), она сможет вычислить сумму по модулю 2 двух блоков открытого текста, и, возможно, у нее получится взломать шифр.

Даже если сетевая карта 802.11 будет подбирать значения вектора инициализации для каждого пакета случайным образом, все равно благодаря ограниченной длине вектора (24 разряда) после передачи 2^{24} пакетов векторы начнут повторяться. Хуже того, при случайном методе подбора значений ожидаемое число пакетов, которые можно послать, не опасаясь повторного выпадения того же значения вектора инициализации, равно примерно 5000. Это связано с «задачей о днях рождения», которую мы обсуждали в разделе «Задача о днях рождения». Итак, прослушивая линию в течение нескольких минут, Труди почти наверняка захватит два пакета с одинаковыми векторами инициализации и ключами. Складывая по модулю эти два пакета, она получит сумму (по модулю 2, разумеется) открытых текстов. А уж эту битовую последовательность можно атаковать самыми разными способами с целью восстановления исходных данных. Приложив некоторые усилия, можно подобрать ключевой поток для данного вектора инициализации. Продолжив свои исследования, Труди сможет составить целый словарь ключевых потоков для разных векторов. Взломав вектор инициализации, можно расшифровывать все проходящие по сети пакеты.

Далее при случайном выборе значений векторов Труди достаточно определить рабочую пару (вектор, ключевой поток), чтобы начать генерировать собственные пакеты произвольного содержания. Это может сильно помешать нормальному обмену данными. Теоретически, принимающая сторона может заметить, что слишком много пакетов имеют одинаковые значения векторов инициализации, но ведь WEP этого не запрещает, и к тому же, все равно никто это не проверяет, на самом-то деле.

Наконец, проверка при помощи CRC — это тоже довольно наивный метод, так как Труди может изменить полезную нагрузку так, чтобы она соответствовала циклическому коду по избыточности, для этого даже не придется расшифровывать само сообщение. Короче говоря, сломать защиту 802.11 очень несложно, а ведь мы перечислили далеко не все методы, обнаруженные Борисовым и др.

В августе 2001, спустя месяц после опубликования работы Борисова и др., был обнародован еще один документ, еще больше низвергающий WEP (Fluhrer и др., 2001). В нем отмечается слабость самого алгоритма RC4. Флурер (Fluhrer)

и его соавторы обнаружили, что очень многие ключи обладают одним неприятным свойством: некоторые разряды ключа можно извлечь, анализируя ключевой поток. Если несколько раз повторить попытки атаки шифра, в конце концов удастся извлечь ключ целиком. Для этого даже не понадобится совершать большие усилия. Впрочем, свои теоретические выводы Флурер не пытался применить, взламывая какие-нибудь сети 802.11.

Вместе с тем, когда один студент-практикант и двое ученых из компании AT&T Labs узнали о работе Флурера, они решили испробовать описанный метод на практике (Stubblefield и др., 2002). За неделю был взломан 128-разрядный ключ, использовавшийся в сети 802.11. Причем, большая часть недели ушла на поиски самой дешевой сетевой платы 802.11, получение разрешения на ее приобретение, а также на ее установку и тестирование. Программирование заняло два часа.

После объявления ими результатов своей деятельности телекомпания CNN затеяла историю под названием «Крутой хакер взламывает систему безопасности беспроводных сетей», в которой некоторые гуру этой индустрии пытались высмеять результаты эксперимента. Мол, результаты работы Флурера делают эксперимент слишком тривиальным. С технической точки зрения, это действительно так, но суть не в этом, а в том, что объединенными усилиями двух команд стандарты WEP и 802.11 были просто низвергнуты.

7 сентября 2001 года институт IEEE представил свой ответ на падение стандарта WEP в виде небольшого бюллетеня. В нем названы шесть позиций, которые резюмированы далее:

1. Мы предупреждали, что уровень безопасности, обеспечиваемый WEP, не выше, чем в Ethernet.
2. Гораздо опаснее просто забыть обеспечить безопасность.
3. Надо попробовать разработать какую-нибудь другую систему безопасности (например, на транспортном уровне).
4. Следующая версия, 802.11i, будет, несомненно, обладать более надежной защитой.
5. В будущем сертификация будет подразумевать обязательное использование версии 802.11i.
6. Мы постараемся решить, что делать до того, как появится 802.11i.

Мы более или менее детально рассмотрели эту историю, чтобы читатель мог осознать, что обеспечение безопасности — непростая задача даже для профессионалов.

Безопасность в системах Bluetooth

Радиус действия систем Bluetooth значительно короче, чем сетей 802.11, поэтому взломщику не удастся произвести атаку, оставив ноутбук в припаркованной рядом со зданием машине, однако вопрос безопасности важен и тут. Например, предположим, что компьютер Алисы оборудован беспроводной клавиатурой стандарта Bluetooth. Если не установить систему защиты, то Труди, находясь за стенкой, в соседнем офисе, сможет без труда прочесть все, что набирает Алиса, включая исходящую почту. Можно захватить все, что передается на беспроводной принтер, если расположиться неподалеку от него (включая входящую почту и конфиденциальные бумаги). К счастью, в Bluetooth есть рабочая схема защиты, нарушающая планы всевозможных личностей типа Труди. Далее мы опишем основные черты этой схемы.

Система защиты Bluetooth может работать в трех режимах, начиная от полного бездействия и заканчивая тотальной шифрацией данных и контролем целостности. Как и в случае с 802.11, если система защиты отключена (по умолчанию это именно так), о какой-либо безопасности говорить не приходится. Большинство пользователей не включают защиту до тех пор, пока не грянет гром. Можно привести сельскохозяйственный пример такого подхода: ворота конюшни закрывают только после исчезновения лошади.

Bluetooth обеспечивает безопасность на нескольких уровнях. На физическом уровне для этого применяются скачкообразные изменения частот, но поскольку любое устройство, появляющееся в микросети, должно узнать последовательность скачков частоты, эта последовательность, очевидно, не является секретной. Настоящая защита информации начинает проявляться тогда, когда вновь прибывшее подчиненное устройство пытается запросить канал для связи с управляющим устройством. Предполагается, что оба устройства совместно используют предварительно установленный закрытый ключ. В некоторых случаях он прошивается в обоих устройствах (например, в гарнитуре и мобильном телефоне, продающихся вместе). В других случаях в одном из устройств (например, в гарнитуре) ключ прошит, а в сопряженное устройство (например, мобильный телефон) пользователь должен ввести ключ вручную в виде десятичного числа. Общие ключи такого типа называются **отмычками**.

Перед установкой канала подчиненное и управляющее устройства должны выяснить, владеют ли они отмычками. В случае положительного ответа им необходимо договориться о том, каким будет канал: шифрованным, с контролем целостности или и таким, и таким. Затем выбирается 128-разрядный ключ сеанса, некоторые биты которого могут быть сделаны общедоступными. Такое послабление сделано в целях соответствия системы ограничениям, введенным правительствами разных стран и запрещающим экспорт или использование ключей, длина которых больше той, что способно взломать правительство.

Шифрация выполняется с применением потокового шифра $\mathbf{E}_0$, контроль целостности — с применением **SAFER+**. И тот, и другой представляют собой традиционные блочные шифры с симметричными ключами. SAFER+ пытались использовать в AES, однако очень быстро отказались от этой мысли, так как он работал гораздо медленнее других. Работа над Bluetooth завершилась еще до того, как был выбран шифр AES; в противном случае, вероятно, использовался бы алгоритм Rijndael.

Процесс шифрации с использованием ключевого потока показан на рис. 8.12. На нем видно, что открытый текст суммируется по модулю 2 с ключевым потоком. В результате получается шифрованный текст. К сожалению, алгоритм E_0, как и RC4, чрезвычайно слаб (Jacobsson и Wetzel, 2001). Несмотря на то, что на

момент написания книги он еще не взломан, его сходство с шифром A5/1, чей провал угрожает безопасности всего GSM-трафика, наводит на грустные мысли (Biryukov и др., 2000). Многим (в том числе и автору) кажется удивительным тот факт, что в игре «кошки-мышки» между шифровальщиками и криптоаналитиками так часто побеждают последние.

Еще одна проблема безопасности, связанная с Bluetooth, состоит в том, что система идентифицирует только устройства, а не пользователей. Это приводит к тому, что вор, укравший устройство Bluetooth, получит доступ к финансовым и другим счетам жертвы. Тем не менее, система безопасности в Bluetooth реализована и на верхних уровнях, поэтому даже в случае взлома защиты на уровне передачи данных некоторые шансы еще остаются, особенно если приложение для выполнения транзакции требует ввода PIN-кода вручную с помощью какой-нибудь разновидности клавиатуры.

Безопасность в WAP 2.0

Надо признать, что форум разработчиков WAP извлек уроки из нестандартного стека протоколов, придуманного для WAP 1.0. В отличие от первой версии, WAP 2.0 характеризуется стандартными протоколами на всех уровнях. Это касается и вопросов безопасности. Базируясь на IP, он полностью поддерживает все возможности IPsec на сетевом уровне. На транспортном уровне TCP-соединения можно защитить TLS — стандартом IETF, который мы изучим далее в этой главе. На более высоких уровнях применяется идентификация клиентов в соответствии с RFC 2617. Криптографическая библиотека прикладного уровня обеспечивает контроль целостности и обнаружение ложной информации. В конечном итоге, так как WAP 2.0 базируется на известных стандартах, есть шанс, что услуги защиты, в частности, секретность, идентификация и обнаружение ложной информации, будут реализованы значительно лучше, чем в 802.11 и Bluetooth.

Протоколы аутентификации

Аутентификация (или идентификация) — это метод, с помощью которого процесс удостоверяется в том, что его собеседник является именно тем, за кого он себя выдает. Проверка подлинности удаленного процесса при активных злонамеренных попытках проникновения представляет собой удивительно сложную задачу и требует сложных протоколов, основанных на криптографии. В данном разделе мы познакомимся с несколькими протоколами аутентификации, применяемыми в незащищенных компьютерных сетях.

Следует отметить, что понятия аутентификации и авторизации иногда путают. Аутентификация связана с вопросом подлинности вашего собеседника. Авторизация имеет дело с разрешениями. Например, клиентский процесс обращается к файловому серверу и говорит: «Я процесс Скотта, и я хочу удалить файл cookbook.old». Файл-сервер должен решить следующие два вопроса:

1. Действительно ли это процесс Скотта (аутентификация)?
2. Имеет ли Скотт право удалять файл cookbook.old (авторизация)?

Только после того, как на оба вопроса будет получен недвусмысленный утвердительный ответ, может быть выполнено запрашиваемое действие. Ключевым является первый вопрос. После того как сервер узнает, с кем он разговаривает, для проверки прав доступа потребуется лишь просмотреть содержимое локальных таблиц или баз данных. По этой причине в данном разделе мы уделим особое внимание вопросу аутентификации.

Общая схема, используемая всеми протоколами аутентификации, состоит из следующих действий. Алиса желает установить защищенное соединение с Бобом или считающимся надежным **Центром распространения ключей**. Затем в разных направлениях посылаются еще несколько сообщений. По мере их передачи хулиганка по имени Труди может перехватить, изменить и снова воспроизвести эти сообщения, чтобы обмануть Алису и Боба или просто сорвать сделку.

Тем не менее, когда протокол завершит свою работу, Алиса должна быть уверена, что разговаривает с Бобом, а Боб — что разговаривает с Алисой. Кроме того, в большинстве протоколов собеседники также установят секретный **ключ сеанса**, которым будут пользоваться для последующего обмена информацией. На практике весь обмен данными шифруется с помощью одного из алгоритмов с секретным ключом (AES или тройной DES), так как их производительность намного выше производительности алгоритмов с открытым ключом. Тем не менее, алгоритмы с открытым ключом широко применяются в протоколах аутентификации и для определения ключа сеанса.

Цель использования нового, случайно выбираемого ключа сеанса для каждого нового соединения состоит в минимизации трафика, посылаемого с использованием закрытых и открытых ключей пользователя, уменьшении количества шифрованного текста, который может достаться злоумышленнику, а также минимизации вреда, причиняемого в случае, если процесс даст сбой и дамп ядра попадет в чужие руки. Поэтому после установки соединения в процессе должен храниться только один временный ключ сеанса. Все постоянные ключи должны быть тщательно стерты.

Аутентификация, основанная на общем секретном ключе

В нашем первом протоколе аутентификации мы уже предполагали, что у Алисы и Боба есть общий секретный ключ K_{AB}. Об этом секретном ключе можно договориться при личной встрече или по телефону, но, в любом случае, не по сети.

В основе этого протокола лежит принцип, применяемый во многих протоколах аутентификации: одна сторона посылает другой случайное число, другая сторона преобразует его особым образом и возвращает результат. Такие протоколы называются протоколами типа **оклик—отзыв**. В этом и последующих протоколах аутентификации будут использоваться следующие условные обозначения:

A и B — Алиса и Боб;
R_i — оклик, где индекс означает его отправителя;
K_i — ключи, где индекс означает владельца ключа;
K_S — ключ сеанса.

Последовательность сообщений нашего первого протокола аутентификации с общим ключом показана на рис. 8.28. В первом сообщении Алиса посылает свое удостоверение личности, A, Бобу тем способом, который ему понятен. Боб, конечно, не знает, пришло ли это сообщение от Алисы или от злоумышленника, поэтому он выбирает большое случайное число R_B и посылает его в качестве оклика «Алисе» открытым текстом (сообщение 2). Затем Алиса шифрует это сообщение секретным ключом, общим для нее и Боба, и отправляет шифрованный текст $K_{AB}(R_B)$ в сообщении 3. Когда Боб видит это сообщение, он понимает, что оно пришло от Алисы, так как злоумышленник не должен знать ключа K_{AB} и поэтому не смог бы сформировать такое сообщение. Более того, поскольку оклик R_B выбирался случайно в большом пространстве чисел (например, 128-разрядных случайных чисел), очень маловероятно, чтобы злоумышленник мог уже видеть этот оклик и ответ на него в предыдущих сеансах.

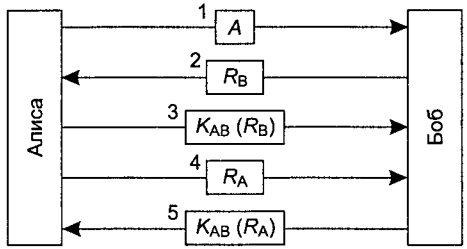


Рис. 8.28. Двусторонняя аутентификация при помощи протокола оклик—отзыв

К этому моменту Боб уверен, что говорит с Алисой, однако Алиса еще пока не уверена ни в чем. Злоумышленник мог перехватить сообщение 1 и послать обратно оклик R_B. Возможно, Боба уже нет в живых. Далее протокол работает симметрично: Алиса посылает оклик, а Боб отвечает на него. Теперь уже обе стороны уверены, что говорят именно с тем, с кем собирались. После этого они могут установить временный ключ сеанса K_S, который можно переслать друг другу, закодировав его все тем же общим ключом K_{AB}.

Количество сообщений в этом протоколе можно сократить, объединив в каждом сообщении ответ на предыдущее сообщение с новым окликом, как показано на рис. 8.29. Здесь Алиса сама в первом же сообщении посылает Бобу оклик. Отвечая на него, Боб помещает в то же сообщение свой оклик. Таким образом, вместо пяти сообщений понадобилось всего три.

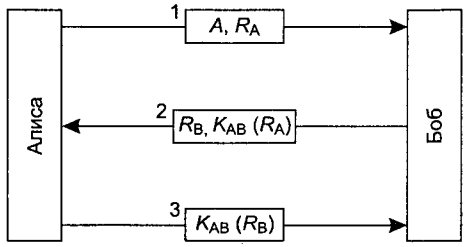


Рис. 8.29. Укороченный двусторонний протокол аутентификации

Лучше ли этот протокол, чем предыдущий? С одной стороны, да: он короче. Но, к сожалению, пользоваться таким протоколом не рекомендуется. При некоторых обстоятельствах злоумышленник может атаковать этот протокол способом, известным под названием **зеркальная атака**. В частности, Труди может взломать его, если ей будет позволено одновременно открыть несколько сеансов связи с Бобом. Такое вполне возможно, если, скажем, Боб — это банк, позволяющий устанавливать несколько одновременных соединений с банкоматами.

Схема зеркальной атаки показана на рис. 8.30. Она начинается с того, что Труди, объявляя себя Алисой, посылает оклик R_T. Боб, как обычно, отвечает своим собственным окликом R_B. Теперь, казалось бы, Труди в тупике. Что ей делать? Она ведь не знает $K_{AB}(R_B)$.

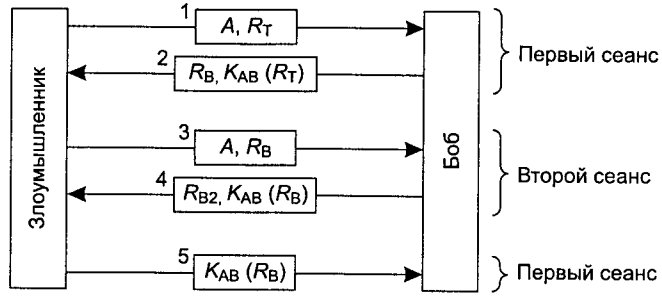


Рис. 8.30. Зеркальная атака

Злоумышленник может открыть второй сеанс сообщением 3 и подать в качестве оклика Бобу оклик самого Боба, взятый из второго сообщения. Боб спокойно шифрует его и посылает обратно $K_{AB}(R_B)$ в сообщении 4. Теперь у Труди есть необходимая информация, поэтому она завершает первый сеанс и прерывает второй. Боб теперь уверен, что злоумышленник — это Алиса, поэтому предоставляет Труди доступ к банковским счетам Алисы и позволяет перевести деньги с ее текущего счета на секретный счет в Швейцарском банке без каких-либо колебаний.

Мораль этой истории такова:

Разработать корректный протокол аутентификации сложнее, чем это может показаться.

Приведем четыре общих правила, которые часто оказываются полезными:

1. Инициатор сеанса должен подтверждать свою личность прежде, чем это сделает отвечающая сторона. В этом случае злоумышленник не сможет получить ценной для него информации, прежде чем подтвердит свою личность.
2. Следует использовать два раздельных общих секретных ключа: один для инициатора сеанса, а другой для отвечающего, K_{AB} и K'_{AB}.
3. Инициатор и отвечающий должны выбирать оклики из различных непересекающихся наборов. Например, инициатор должен пользоваться четными номерами, а отвечающий — нечетными.
4. Протокол должен уметь противостоять атакам, при которых запускается второй параллельный сеанс, информация для которого извлекается при помощи первого сеанса.

Если нарушается хотя бы одно из этих правил, протокол оказывается уязвимым. В приведенном примере были нарушены все четыре правила, что привело к разрушительным последствиям.

Вернемся к ситуации, показанной на рис. 8.28. Можно ли с уверенностью сказать, что этот протокол не подвержен зеркальным атакам? Это зависит от различных факторов. Ситуация с этим очень шаткая. Труди удалось справиться с нашим протоколом, используя зеркальную атаку, потому что он позволял запустить параллельный сеанс с Бобом и ввести его в заблуждение, передав ему его собственный оклик. А что произойдет, если вместо живой Алисы, сидящей за компьютером, стоит обычный компьютер общего назначения, принимающий параллельные сеансы связи? Посмотрим, что Труди сможет сделать.

Чтобы понять, каким образом Труди взламывает протокол, обратимся к рис. 8.31. Алиса объявляет свои идентификационные данные в сообщении 1. Труди это сообщение перехватывает и запускает собственный сеанс, посылая сообщение 2 и прикидываясь Бобом. Здесь мы, как и раньше, изобразили серыми квадратиками сообщения второго сеанса. Алиса отвечает на сообщение 2 так: «Ты представляешься Бобом? Это необходимо подтвердить в сообщении 3». Здесь Труди заходит в тупик: она не может подтвердить, что она — это Боб.

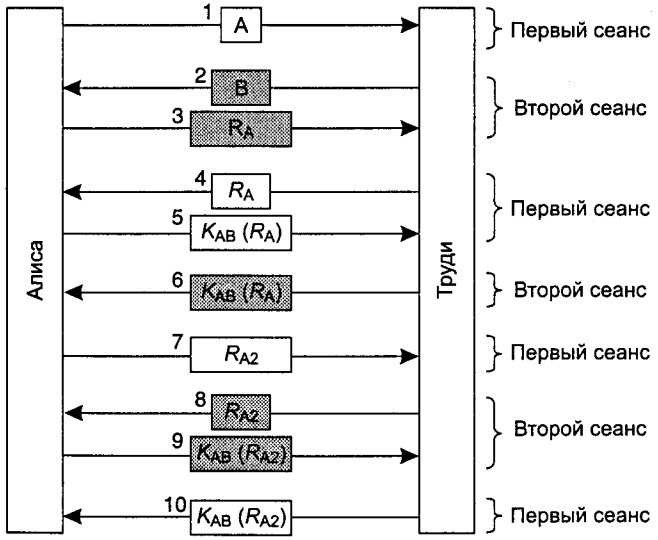


Рис. 8.31. Зеркальная атака протокола, показанного на рис. 8.28

Что же теперь Труди может сделать? Она возвращается к первому сеансу, где как раз наступает ее очередь отправки оклика. При этом отправляется R_A, полученный в сообщении 3. Алиса любезно отвечает на это в сообщении 5, предоставляя тем самым Труди информацию, необходимую ей для создания сообщения 6 в сеансе 2. Труди может теперь выбирать сеанс, так как она корректно ответила на оклик Алисы во втором сеансе. Сеанс 1 можно закрыть, переправить в сеансе 2 какое-нибудь старое число и получить в итоге заверенный сеанс связи с Алисой.

Однако Труди просто невыносима, и она доказывает это своим дальнейшим поведением. Вместо того чтобы отправить какое-нибудь старое число для завершения регистрации сеанса 2, она ждет, пока Алиса пошлет сообщение 7 (ее оклик для сеанса 1). Конечно, Труди понятия не имеет, как ответить на это, поэтому она вновь проводит зеркальную атаку, отправляя R_{A2} в качестве сообщения 8. Алиса очень кстати шифрует R_{A2} в сообщении 9. Труди переключается на сеанс 1 и отправляет Алисе то число, какое ей хочется, в сообщении 10. Откуда она берет это число? Очевидно, из сообщения 9, пришедшего от Алисы. С этого момента Труди может гордиться тем, что у нее есть два полностью заверенных сеанса связи с Алисой.

Эта атака приводит к несколько иному результату, нежели протокол с тремя сообщениями, показанный на рис. 8.30. На этот раз Труди удается установить сразу два заверенных соединения с Алисой. В предыдущем примере одно заверенное соединение было установлено с Бобом. Опять же, если бы протокол удовлетворял всем четырем перечисленным требованиям, атака успеха бы не имела. Детальное обсуждение различных типов атак и методов противодействия им приведено в (Bird и др., 1993). Там также описана методика построения протоколов, корректность которых можно строго доказать. Однако даже простейший из таких протоколов достаточно сложен, поэтому сейчас мы обратимся к другому классу (вполне корректных) протоколов.

Итак, новый протокол аутентификации показан на рис. 8.32 (Bird и др., 1993). Тут мы видим тот же самый HMAC, который мы уже обсуждали при изучении IPsec. Для начала Алиса посылает Бобу отметку времени R_A в виде сообщения 1. Боб при ответе выбирает собственную отметку времени, R_B, и высылает ее вместе с HMAC. HMAC формирует структуру данных, состоящую из временных отметок Алисы и Боба, их идентификаторов, а также общего закрытого ключа K_{AB}. Затем вся эта структура с помощью хэш-функции (например, SHA-1) помещается в HMAC. После приема сообщения 2 Алиса становится счастливым обладателем R_A (это значение выбрано ею же), R_B, полученного в виде открытого текста, двух идентификаторов и закрытого ключа K_{AB}, известного и так. Имея все эти данные, она может вычислить HMAC самостоятельно. Если он согласуется с HMAC, содержащимся в сообщении, она убеждается, что говорит с Бобом, поскольку Труди не знает K_{AB} и, следовательно, не может угадать HMAC, который следует отослать. В ответе Алисы Бобу содержится HMAC, состоящий из двух временных отметок.

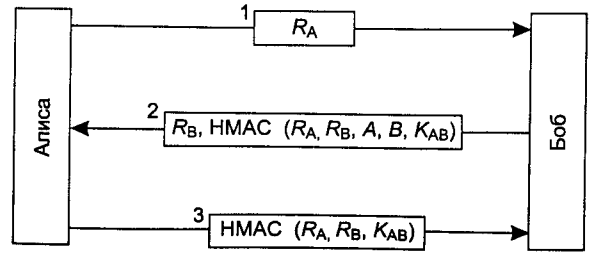


Рис. 8.32. Аутентификация с применением HMAC

Вопрос: может ли Труди как-нибудь взломать такой протокол? Нет, потому что она не может заставить ни одну из сторон шифровать выбранное ею самой значение или применять к нему хэш-функцию, как это было в ситуации на рис. 8.30. Оба HMAC включают в себя значения, выбранные отправителем. Труди не способна их контролировать каким-либо образом.

Использование HMAC — это далеко не единственное, что можно сделать. Альтернативная схема, которая применяется довольно часто, заключается в шифровании элементов данных последовательно с помощью сцепления блоков шифра.

Установка общего ключа: протокол обмена ключами Диффи—Хеллмана

Итак, мы предположили, что у Алисы и Боба есть общий секретный ключ. Предположим теперь, что у них его нет (поскольку до сих пор не разработана универсальная инфраструктура PKI создания подписей и распространения сертификатов). Как им получить такой ключ? Алиса может позвонить Бобу и передать ему ключ по телефону, но он, возможно, спросит: «Как вы докажете, что вы — Алиса, а не злоумышленник?» Они могут попытаться организовать встречу, на которую каждый придет с паспортом, водительскими правами и тремя кредитными картами, но, будучи занятыми людьми, они, возможно, не смогут найти устраивающую обоих дату встречи в течение нескольких месяцев. К счастью, существует способ для совершенно незнакомых людей установить общий секретный ключ средь бела дня, даже если злоумышленник старательно записывает каждое сообщение.

Протокол, позволяющий не встречавшимся ранее людям устанавливать общий секретный ключ, называется **протоколом обмена ключами Диффи—Хеллмана** (Diffie и Hellman, 1976) и работает следующим образом. Алиса и Боб договариваются о двух больших простых числах, n и g, где $(n-1)/2$ также является простым числом, кроме того, на число g накладываются некоторые дополнительные условия. Эти числа могут быть открытыми, поэтому каждый из них может просто выбрать n и g и открыто сообщить о них другому. Затем Алиса выбирает большое (например, 512-разрядное) число x и держит его в секрете. Аналогично, Боб выбирает большое секретное число y.

Алиса начинает протокол обмена ключами с того, что посылает Бобу сообщение, содержащее $(n, g, g^x \bmod n)$, как показано на рис. 8.33. Боб отвечает Алисе сообщением, содержащим $g^y \bmod n$. Теперь Алиса берет число, присланное ей Бобом, и возводит его в степень x, получая $(g^y \bmod n)^x$. Боб выполняет подобные вычисления и получает $(g^x \bmod n)^y$. В соответствии с законами арифметики оба вычисления должны быть равны $g^{xy} \bmod n$. Таким образом, у Алисы и Боба есть общий секретный ключ $g^{xy} \bmod n$.

Конечно, злоумышленник видел оба сообщения. Ему известны значения n и g из первого сообщения. Если бы ему удалось вычислить значения x и y, ему бы удалось получить секретный ключ. Беда в том, что, зная $g^x \bmod n$ и n, найти значение x очень трудно. На сегодняшний день неизвестен алгоритм вычисления дискретного логарифма модуля очень большого простого числа.

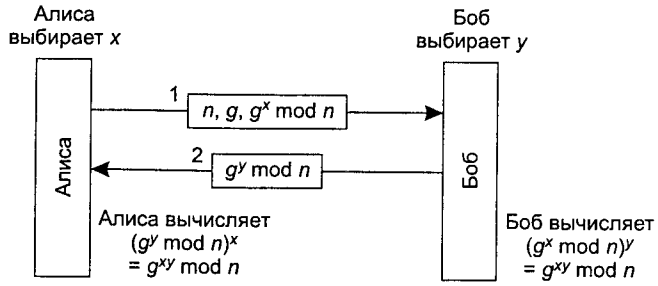


Рис. 8.33. Обмен ключами Диффи—Хеллмана

Для примера возьмем (совершенно нереальные) значения $n = 47$ и $g = 3$. Алиса выбирает значение $x = 8$, а Боб выбирает $y = 10$. Оба эти числа хранятся в секрете. Сообщение Алисы Бобу содержит числа (47, 3, 28), так как $3^8 \bmod 47 = 28$. Боб отвечает Алисе числом 17. Алиса вычисляет $17^8 \bmod 47$ и получает 4. Боб вычисляет $28^{10} \bmod 47$ и получает также 4. Таким образом, независимо друг от друга Алиса и Боб определили, что значение секретного ключа равно 4. Злоумышленнику придется решить уравнение $3^x \bmod 47 = 28$, что можно сделать путем полного перебора для таких небольших чисел, но только не для чисел длиной в несколько сотен бит.

Несмотря на всю элегантность алгоритма Диффи—Хеллмана, имеется одна проблема: когда Боб получит три числа (47, 3, 28), как он сможет удостовериться в том, что они посланы Алисой, а не злоумышленником? Способа узнать это не существует. К сожалению, злоумышленник может воспользоваться этим, чтобы обмануть Алису и Боба, как показано на рис. 8.34. Здесь, пока Алиса с Бобом выбирают значения x и y, злоумышленник выбирает свое случайное число z. Алиса посылает Бобу сообщение 1. Злоумышленник перехватывает его и отправляет вместо него Бобу сообщение 2, используя правильные значения n и g (которые посылались открытым текстом), но со своим значением z вместо x. Он также посылает обратно Алисе сообщение 3. Позднее Боб отправляет Алисе сообщение 4, которое злоумышленник снова перехватывает и хранит у себя.

Теперь все занимаются вычислением остатков от деления. Алиса вычисляет значение секретного ключа: $g^{xz} \bmod n$. Те же самые вычисления производит злоумышленник (для общения с Алисой). Боб вычисляет $g^{yz} \bmod n$, что также делает и злоумышленник (для общения с Бобом). Каждое сообщение, посылаемое Алисой в шифрованном сеансе, перехватывается злоумышленником, сохраняется, изменяется, если это нужно, и отправляется (по желанию злоумышленника) Бобу. То же самое происходит и в обратном направлении. Злоумышленник видит все сообщения и может изменять их по своему усмотрению, в то время как Алиса и Боб полагают, что у них имеется защищенный канал для связи друг с другом. Подобные действия злоумышленника называются атакой типа **«пожарная цепочка»**, поскольку слегка напоминают старинных пожарных, передававших друг другу по цепочке ведра с водой. Еще одно название этой атаки — **«человек посередине»**.

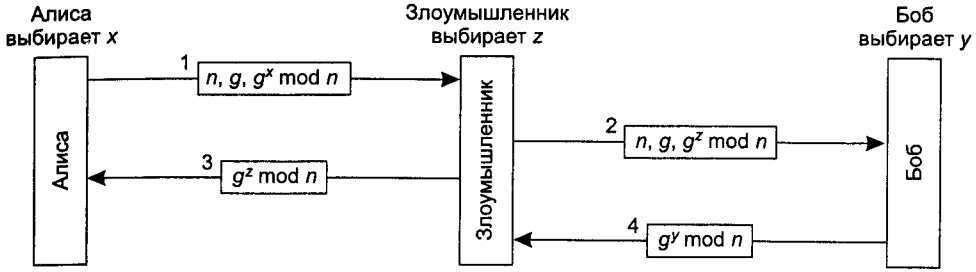


Рис. 8.34. Атака типа «пожарная цепочка»

Аутентификация с помощью центра распространения ключей

Итак, установка общего секретного ключа с незнакомцем почти удалась. С другой стороны, вероятно, не следовало вообще этим заниматься. Чтобы общаться с m людьми, вам понадобится хранить n ключей. Для людей, чей круг общения широк, хранение ключей может превратиться в серьезную проблему, особенно если все эти ключи придется хранить на отдельных пластиковых картах.

Другой подход состоит в организации доверительного центра распространения ключей (KDC, key distribution center). При такой схеме у каждого пользователя всего один ключ, общий с KDC-центром. Операции с ключами аутентификации и сеансовыми ключами проходят через KDC-центр. Простейший протокол аутентификации с помощью центра распространения ключей, включающий две стороны и доверенный KDC-центр, изображен на рис. 8.35.

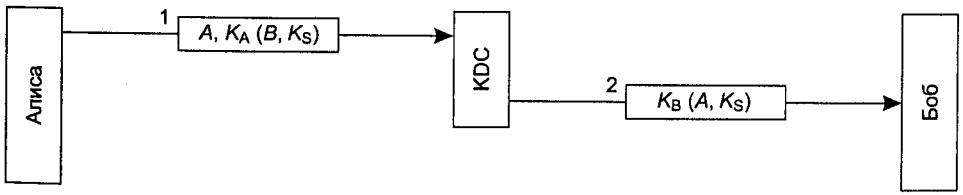


Рис. 8.35. Первая попытка протокола аутентификации с помощью KDC-центра

Идея, лежащая в основе протокола, проста: Алиса выбирает ключ сеанса, K_S, и заявляет KDC-центру, что она желает поговорить с Бобом при помощи ключа K_S. Это сообщение шифруется секретным ключом K_A, которым совместно владеют только Алиса и центр распространения ключей. Центр распространения ключей расшифровывает это сообщение и извлекает из него идентификатор личности Боба и ключ сеанса. Затем он формирует новое сообщение, содержащее идентификатор личности Алисы и ключ сеанса, и посылает его Бобу. Это сообщение зашифровывается ключом K_B — секретным ключом, общим для Боба и центра распространения ключей. Расшифровав это сообщение, Боб узнает, что Алиса желает с ним поговорить и какой ключ она хочет использовать.

Аутентификация в данном случае происходит сама собой. KDC знает, что сообщение 1 пришло от Алисы, так как больше никто не может зашифровать его секретным ключом Алисы. Аналогично, Боб знает, что сообщение 2 пришло от KDC, так как кроме него их общий секретный ключ никому не известен.

К сожалению, этот протокол содержит серьезную ошибку. Труди нужны деньги, поэтому она придумывает некую легальную услугу, которую она могла бы выполнить для Алисы. Затем Труди делает Алисе заманчивое предложение и получает эту работу. Выполнив ее, Труди вежливо предлагает Алисе оплатить услуги, переведя деньги на ее банковский счет. Чтобы оплатить работу, Алиса устанавливает ключ сеанса со своим банкиром Бобом. Затем она посылает Бобу сообщение с просьбой перевести деньги на счет Труди.

Тем временем Труди возвращается к своим темным делам. Она копирует сообщение 2 (см. рис. 8.35) и запрос на перевод денег, следующий за ним. Затем она воспроизводит оба сообщения для Боба. Боб получает их и думает: «Должно быть, Алиса снова наняла Труди. Похоже, она хорошо справляется с работой». Боб перечисляет еще столько же денег со счета Алисы на счет Труди. Получив пятидесятый запрос на перевод денег, Боб выбегает из офиса, чтобы найти Труди и предложить ей большую ссуду, чтобы она могла расширить свой чрезвычайно успешный бизнес. Подобная проблема получила название **атаки повторным воспроизведением**.

Существует несколько решений этой проблемы. Первое решение состоит в помещении в каждое сообщение временного штампа. Все устаревшие сообщения просто игнорируются. Беда здесь в том, что системные часы в сети синхронизировать с большой степенью точности невозможно, поэтому должен существовать какой-то срок годности временного штампа. Труди может обмануть протокол, послав повторное сообщение во время этого интервала.

Второе решение заключается в помещении в сообщение уникального порядкового номера, обычно называемого **нонсом** (nonce — данный случай, данное время). Каждая сторона должна запоминать все предыдущие нонсы и отвергать любое сообщение, содержащее использованный ранее нонс. Однако нонсы должны храниться вечно, иначе Труди попытается воспроизвести сообщение пятилетней давности. Кроме того, если машина потеряет список нонсов в результате сбоя, она снова станет уязвимой к атакам повторным воспроизведением. Можно комбинировать временные штампы и нонсы, чтобы ограничить срок хранения нонсов, но так или иначе, протокол должен быть значительно усложнен.

Более сложный метод аутентификации состоит в использовании многостороннего протокола оклик—отзыв. Хорошо известным примером такого протокола является **протокол аутентификации Нидхэма—Шрёдера** (Needham—Schroeder, 1978), один из вариантов которого показан на рис. 8.36.

Работа протокола начинается с того, что Алиса сообщает KDC-центру, что она желает поговорить с Бобом. Это сообщение содержит в качестве нонса большое случайное число R_A. Центр распространения ключей посылает обратно сообщение 2, содержащее случайное число Алисы, ключ сеанса и так называемый билет, который она может послать Бобу. Цель посылки случайного числа R_A состоит в том, чтобы убедить Алису в том, что сообщение 2 является свежим, а не

повторно воспроизведенным. Идентификатор Боба также помещается в сообщение 2 на случай, если злоумышленник (Труди) вздумает заменить его идентификатор на свой в сообщении 1, так чтобы KDC-центр зашифровал билет в конце сообщения 2 ключом K_T (ключ Труди) вместо K_B. Билет, зашифрованный ключом K_B, помещается внутри зашифрованного сообщения, чтобы злоумышленник не смог заменить его чем-либо другим, пока сообщение 2 добирается до Алисы.

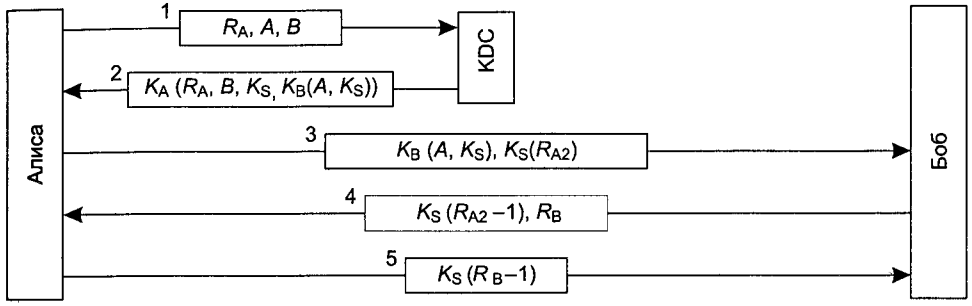


Рис. 8.36. Протокол аутентификации Нидхэма—Шрёдера

Затем Алиса посылает билет Бобу вместе с новым случайным числом R_{A2}, зашифрованным ключом сеанса K_S. В сообщении 4 Боб посылает обратно $K_S(R_{A2}-1)$, чтобы доказать Алисе, что она разговаривает с настоящим Бобом. Отсылать обратно просто $K_S(R_{A2})$ бессмысленно, так как это число могло быть украдено злоумышленником из сообщения 3.

Получив сообщение 4, Алиса убеждается, что разговаривает с Бобом и что до сих пор не было использовано повторных сообщений. Между отправкой случайного числа R_{A2} и получением ответа на него в виде $K_S(R_{A2}-1)$ проходит довольно короткий промежуток времени. Цель сообщения 5 — убедить Боба, что он действительно разговаривает с Алисой и что в этом сеансе связи также отсутствуют повторно воспроизведенные данные. Возможность атаки с помощью повторного воспроизведения ранее записанной информации исключается этим протоколом благодаря тому, что каждая сторона формирует оклик другой стороны и получает на него отзыв.

Несмотря на всю кажущуюся солидность протокола, в нем, тем не менее, имеется небольшое слабое место. Если злоумышленнику удастся каким-либо способом раздобыть старый ключ сеанса K_S, он сможет инициировать новый сеанс с Бобом, повторно воспроизведя сообщение 3 с использованием скомпрометированного ключа, и выдать себя за Алису (Denning и Sacco, 1981). На этот раз злоумышленник может украсть деньги со счета Алисы, даже не выполнив никаких услуг.

Позднее Нидхэм и Шрёдер опубликовали протокол, решающий эту проблему (Needham и Shroeder, 1987). В том же выпуске того же журнала Отуэй (Otway) и Рис (Rees) также опубликовали протокол, решающий эту проблему более коротким путем. На рис. 8.37 показан слегка видоизмененный протокол Отуэя—Риса.

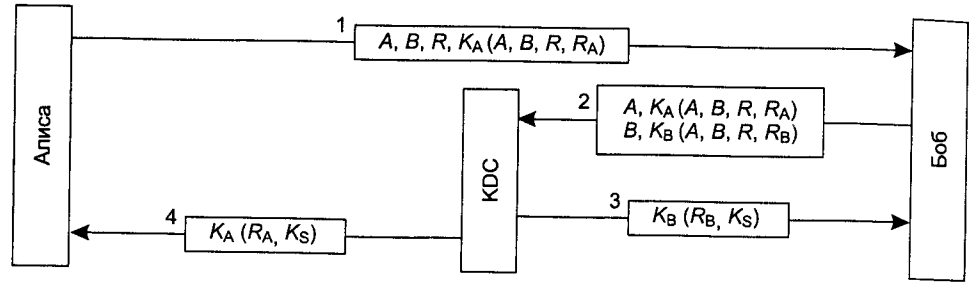


Рис. 8.37. Протокол аутентификации Отуэя—Риса (слегка упрощенный)

В протоколе Отуэя—Риса Алиса начинает с формирования пары случайных номеров: R, который будет использоваться в качестве общего идентификатора, и R_A, который Алиса будет использовать в качестве оклика Боба. Получив это сообщение, Боб формирует новое сообщение из зашифрованной части сообщения Алисы и аналогичной собственной части. Обе части сообщения, зашифрованные ключами K_A и K_B, идентифицируют Алису и Боба, содержат общий идентификатор и оклики.

Центр распространения ключей проверяет, совпадают ли общие идентификаторы R в обеих частях сообщения. Они могут не совпадать, если злоумышленник подменил R в сообщении 1 или заменил часть сообщения 2. Если оба общих идентификатора R совпадают, KDC-центр считает сообщение, полученное от Боба, достоверным. Затем он формирует ключ сеанса K_S и отправляет его Алисе и Бобу, зашифровав ключ сеанса ключами Алисы и Боба. Каждое сообщение также содержит случайное число получателя в доказательство того, что эти сообщения посланы KDC-центром, а не злоумышленником. К этому моменту Алиса и Боб обладают одним и тем же ключом сеанса и могут начать обмен информацией. После первого же обмена данными они увидят, что обладают одинаковыми копиями ключа сеанса K_S, на чем процесс аутентификации можно будет считать завершенным.

Аутентификация при помощи протокола Kerberos

Во многих реально работающих системах применяется протокол аутентификации **Kerberos**, основанный на одном из вариантов протокола Нидхэма—Шрёдера. Он назван по имени трехглавого пса греческих мифов Кербера (чаще называемого Цербером благодаря латинскому написанию. — *Примеч. перев.*), охранявшего выход из Аида. Кербер пропускал в Аид всякого, но не выпускал оттуда никого. Протокол Kerberos был разработан в Массачусетском технологическом институте для обеспечения пользователям рабочих станций надежного доступа к сетевым ресурсам. Его основное отличие от протокола Нидхэма—Шрёдера состоит в предположении о довольно хорошей синхронизации всех часов в сети. Было разработано несколько последовательных версий протокола. Версия V4 наиболее широко применяется в промышленности, поэтому она будет здесь описана. Затем

будет сказано несколько слов о следующей версии, V5. Дополнительную информацию см. в (Steiner и др., 1988).

В работе протокола Kerberos, помимо рабочей (клиентской) станции Алисы, принимают участие еще три сервера:

- сервер аутентификации (AS, Authentication Server): проверяет личность пользователей при входе в сеть;
- сервер выдачи билетов (TGS, Ticket Granting Server): выдает «билеты, подтверждающие подлинность»;
- Боб, то есть сервер, предоставляющий услуги Алисе.

Сервер аутентификации AS аналогичен центру распространения ключей KDC в том, что у него есть общий секретный пароль для каждого пользователя. Работа сервера выдачи билетов TGS состоит в выдаче свидетельств, убеждающих другие серверы в том, что владелец билета действительно является тем, за кого он себя выдает.

Чтобы начать сеанс, Алиса усаживается за клавиатуру произвольной общедоступной рабочей станции и вводит свое имя. Рабочая станция посылает введенное имя открытым текстом на сервер аутентификации, как показано на рис. 8.38. Сервер аутентификации AS возвращает рабочей станции Алисы ключ сеанса и билет $K_{TGS}(A, K_S)$ для сервера выдачи билетов TGS. Эти данные упаковываются вместе и зашифровываются секретным ключом Алисы так, чтобы только Алиса могла их расшифровать. Только после получения сообщения 2 рабочая станция запрашивает пароль Алисы. С помощью этого пароля формируется ключ K_A, которым расшифровывается сообщение 2, и из него извлекаются ключ сеанса и билет для получения доступа к серверу выдачи билетов TGS. После расшифровки рабочая станция сразу же уничтожает хранящийся в ее памяти пароль. Если вместо Алисы на рабочей станции попытается зарегистрироваться Труди, введенный ею пароль окажется неверным, что будет обнаружено рабочей станцией, так как стандартная часть сообщения 2 окажется неверной.

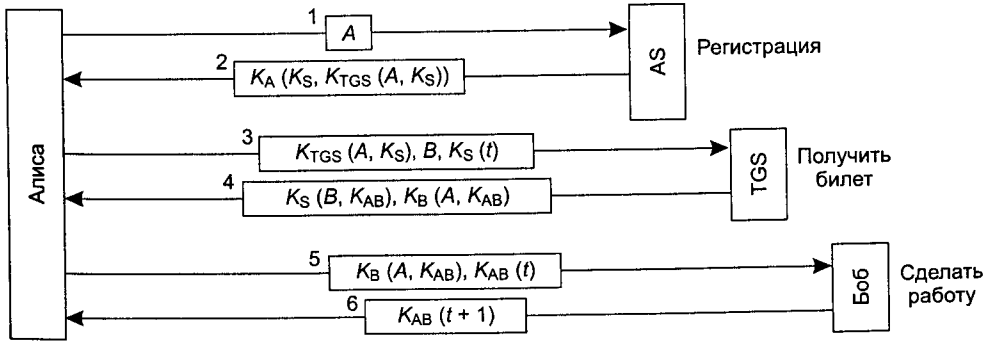


Рис. 8.38. Работа протокола Kerberos V4

После регистрации в сети Алиса может сообщить рабочей станции, что она хочет вступить в контакт с файловым сервером, то есть Бобом. При этом рабочая

станция посылает серверу выдачи билетов сообщение 3 с просьбой выдать билет для общения с Бобом. Ключевым элементом этого запроса является билет $K_{TGS}(A, K_S)$, который зашифрован секретным ключом TGS-сервера и используется для подтверждения личности отправителя. Сервер выдачи билетов отвечает созданием ключа сеанса K_{AB}, которым будут пользоваться Алиса и Боб. Он отправляет Алисе две версии этого ключа. Один ключ зашифрован ключом сеанса K_S, поэтому Алиса может его прочитать. Второй ключ шифруется ключом Боба K_B, что позволяет Бобу его прочитать.

Злоумышленник может скопировать сообщение 3 и попытаться использовать его снова, но ему помешает временной штамп t, отправляемый вместе с этим сообщением. Злоумышленник не может заменить этот временной штамп на более новый, так как не знает ключа сеанса K_S, которым пользуется Алиса для общения с сервером выдачи билетов. Даже если злоумышленник очень быстро повторит сообщение 3, все равно, единственное, что он получит в ответ, это сообщение 4, которое он не смог расшифровать в первый раз и не сможет расшифровать и во второй раз.

После этого Алиса может послать Бобу ключ K_{AB} для установки сеанса с Бобом. Эти сообщения также содержат временные штампы. Сообщение 6, получаемое в ответ, подтверждает, что Алиса говорит именно с Бобом, а не со злоумышленником.

Наконец, после этой серии обмена сообщениями Алиса сможет обмениваться с Бобом данными, используя ключ сеанса K_{AB}. Если после этого Алиса решит, что ей необходим другой сервер, например Кэрол (Carol, C), она может просто послать серверу выдачи ключей сообщение, аналогичное третьему, заменив в нем B на C (то есть идентификатор Боба на идентификатор Кэрол). TGC-сервер мгновенно ответит сообщением, содержащим билет, зашифрованный ключом K_C. Этот билет Алиса пошлет Кэрол, для которой он будет служить гарантией подлинности Алисы.

Достоинство этого протокола состоит в том, что теперь Алиса может получать защищенный доступ к любому серверу сети, и в то же время ее пароль ни разу не передавался по сети. В действительности он только на несколько миллисекунд появлялся в ее рабочей станции. Однако обратите внимание на то, что каждый сервер выполняет свою собственную процедуру авторизации. Когда Алиса предъявляет свой билет Бобу, это всего лишь подтверждает Бобу подлинность предъявителя билета. К чему же Алиса может получить доступ на сервере, решает Боб.

Поскольку разработчики системы Kerberos не рассчитывали, что весь мир станет доверять одному единственному серверу аутентификации, они обеспечили существование нескольких **областей**, каждая из которых имеет свой собственный сервер аутентификации и сервер выдачи билетов. Чтобы получить билет для сервера, расположенного в удаленной области, Алиса должна запросить у своего TGS-сервера билет, который будет принят TGS-сервером удаленной области. Если удаленный TGS-сервер зарегистрировался на локальном TGS-сервере (так же, как это делают локальные серверы), локальный TGS-сервер выдаст Алисе билет, действительный на удаленном TGS-сервере. После этого она может получить у удаленного TGS-сервера билеты к серверам данной удаленной области.

Обратите внимание на то, что для того чтобы две стороны, расположенные в различных областях, могли установить друг с другом защищенный сеанс связи, каждая из сторон должна доверять TGS-серверу другой стороны.

Протокол Kerberos V5 сложнее четвертой версии и подразумевает большее количество накладных расходов. Кроме того, он использует язык OSI ASN.1 для описания типов данных. Претерпели небольшие изменения и протоколы. Помимо этого в системе Kerberos V5 время жизни билетов более длительное, билеты могут обновляться и даже датироваться задним числом. Также, по крайней мере в теории, пятая версия системы Kerberos не является зависимой от стандарта DES, как V4, и поддерживает различные области.

Аутентификация с помощью шифрования с открытым ключом

Взаимная аутентификация также может выполняться с помощью шифрования с открытым ключом. Для начала Алисе нужно получить открытый ключ Боба. Если инфраструктура PKI реализована на основе сервера каталогов, выдающего сертификаты на открытые ключи, Алиса может потребовать сертификат Боба, что показано в виде сообщения 1 на рис. 8.39. Ответ, содержащийся в сообщении 2, — это сертификат X.509 с открытым ключом Боба. Проверив корректность подписи, Алиса может отправить Бобу сообщение со своим идентификатором и нонсом.

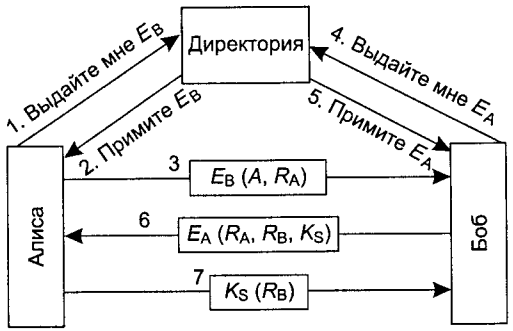


Рис. 8.39. Взаимная идентификация с помощью открытого ключа

Когда Боб получает это сообщение, он не знает, пришло ли оно от Алисы или от злоумышленника, но он делает вид, что все в порядке, и просит сервер каталогов выдать ему открытый ключ Алисы (сообщение 4). Вскоре он его получает (в сообщении 5). Затем он отсылает Алисе сообщение, содержащее случайное число Алисы R_A, свой собственный нонс R_B и предлагаемый ключ сеанса K_S. Все это сообщение зашифровывается открытым ключом Алисы.

Алиса расшифровывает полученное сообщение 6 своим закрытым ключом. Она видит в нем свое случайное число, R_A, и очень этому рада: это подтверждает, что сообщение пришло от Боба, так как у злоумышленника не должно быть способа определить значение этого числа. Кроме того, случайное число R_A свидетельствует о свежести этого сообщения. Алиса соглашается на установку сеанса, отправляя сообщение 7. Когда Боб видит свое случайное число R_B, зашифрованное ключом сеанса, который он сам же сформировал, он понимает, что Алиса получила сообщение 6 и проверила значение R_A.

Может ли злоумышленник каким-либо образом обмануть этот протокол? Он может сфабриковать сообщение 3 и спровоцировать Боба на проверку Алисы, но Алиса увидит число R_A, которого она не посылала, и не станет продолжать. Злоумышленник не сможет убедительно подделать сообщение 7, так как ему не известны значения оклика R_B или ключа K_S, и он не может определить их, не имея закрытого ключа Алисы. Так что ему не везет.

Конфиденциальность электронной переписки

При пересылке между двумя удаленными пользователями сообщение обычно преодолевает по пути десяток других машин. Любая из них может читать и записывать проходящую через нее почту. Конфиденциальности не существует, что бы ни думали об этом многие пользователи. Тем не менее, многие пользователи желали бы иметь возможность посылать электронную почту так, чтобы ее мог прочитать только тот, для кого она предназначается, и никто другой: ни шеф, ни хакеры, ни даже правительство. Эта потребность стимулировала применение некоторыми группами и отдельными разработчиками криптографических принципов к электронной почте. В следующих разделах мы познакомимся с широко распространенной системой защиты электронной почты PGP, а также дадим общее представление о двух других: PEM и S/MIME. Дополнительную информацию см. в (Kaufman и др., 2002; Schneier, 1995).

PGP — довольно неплохая конфиденциальность

Наш первый пример, система **PGP** (Pretty Good Privacy — довольно хорошая конфиденциальность), создана всего одним человеком, Филом Циммерманом (Phil Zimmermann, 1995a, 1995b). Циммерман является сторонником безопасности в сетях, и его девиз таков: «Если конфиденциальность объявлена вне закона, значит, пользоваться ею будут только нарушители закона». Выпущенная в 1991 году система PGP представляет собой полный пакет для электронной почты, обеспечивающий конфиденциальность, аутентификацию, цифровые подписи и сжатие. Все это делается в легкой и удобной форме. Более того, полный пакет, включающий все исходные тексты программ, свободно распространяется через Интернет. Благодаря своему качеству, цене (нулевой) и простоте установки на различных платформах, включая UNIX, Linux, Windows и Mac OS, в настоящее время система PGP получила широкое распространение.

PGP кодирует данные с помощью блочного шифра **IDEA** (International Data Encryption Algorithm — международный алгоритм шифрования данных), ис-

пользующего 128-разрядные ключи. Он был изобретен в Швейцарии в те времена, когда DES уже считался устаревшим, а AES еще не был придуман. Концептуально IDEA похож на DES/AES: производится смешивание разрядов в серии, однако детали реализации функций отличают его от DES и AES. Управление ключами происходит с помощью RSA, а для задач обеспечения целостности данных применяется MD5. Все эти методы мы обсуждали ранее.

История создания системы PGP весьма запутана с первого дня ее существования (Levy, 1993). Поскольку она свободно распространялась через Интернет, правительство США объявило, что Циммерман нарушил закон, запрещающий экспорт военного имущества. Следствие по этому делу длилось пять лет, однако в один прекрасный день прекратилось по двум основным причинам. Во-первых, Циммерман собственноручно не выкладывал PGP в Интернете, и адвокат аргументировал позицию защиты тем, что обвиняемый *сам* никогда не занимался экспортом чего бы то ни было (кроме того, еще надо доказать, что создание сайта равносильно экспорту). Во-вторых, правительство вдруг осознало, что выигрыш дела означал бы, что любой веб-сайт, содержащий загружаемые программы, связанные с секретностью, подпадает под действие закона о торговле такими предметами, как танки, подводные лодки, военные самолеты и ядерное оружие. Этим можно было бы добиться лишь одного: бурного протеста общественности. Это не решение проблемы.

Честно говоря, законы, касающиеся экспорта, кажутся несколько диковатыми. Правительство решило, что размещение программы на веб-странице можно приравнять к нелегальному экспорту, и надоедало Циммерману целых 5 лет. С другой стороны, если кто-то опубликует в книге полный исходный код PGP на языке C (крупным шрифтом, да еще и с контрольной суммой в конце каждой страницы, что облегчит сканирование) и затем займется экспортом этой книги, правительство и глазом не моргнет: книги по закону не являются военным имуществом. *Оружие сильней пера в законе дяди Сэма.*

Еще одна проблема, с которой внезапно столкнулась PGP, была связана с посягательством на патентные права. Владелец патента на RSA, корпорация RSA Security, сослалась на то, что использование метода RSA в PGP является посягательством на патент. Эта проблема разрешилась в версиях начиная с 2.6. Забавно, что вместо RSA в PGP стали применять IDEA, что поначалу тоже вызывало некоторые вопросы.

Так как PGP — это система с открытым исходным кодом, появилось множество модификаций, созданных различными группами и отдельными заинтересованными лицами. Некоторые из них пытались каким-то образом обойти законы об экспорте оружия, другие старались избежать применения запатентованных алгоритмов, а третьи работали над превращением PGP в коммерческий продукт с закрытым исходным кодом. Несмотря на то, что законы об экспорте оружия несколько смягчились (тем не менее, продукцию, использующую AES, до сих пор нельзя экспортировать за пределы США), а срок действия патента RSA закончился в сентябре 2000 года, следствием всех этих проблем стало появление и распространение нескольких несовместимых версий PGP, имеющих разные названия. Далее обсуждается классический вариант PGP, он же является самым старым

и простым. Еще одна популярная версия, Open PGP, описана в RFC 2440. Можно отметить еще GNU Privacy Guard.

В системе PGP намеренно используются уже существующие криптографические алгоритмы, а не изобретаются новые. Все они прошли тщательную проверку ведущими криптоаналитиками мира, и история создания этих алгоритмов не запятнана участием каких-либо государственных организаций, пытающихся их ослабить. Последнее качество является особенно большим преимуществом для всех, кто склонен не доверять правительству.

Система PGP поддерживает сжатие текста, секретность и цифровые подписи, а также предоставляет исчерпывающие средства управления ключами. Как ни странно, не поддерживаются средства электронной почты. Она больше всего похожа на препроцессор, берущий на входе открытый текст и создающий на выходе шифр base64. Разумеется, эти выходные данные можно отправить по электронной почте. Некоторые реализации на последнем шаге обращаются к пользовательскому агенту, чтобы упростить задачу реальной отправки сообщения.

Чтобы понять, как работает система PGP, рассмотрим пример на рис. 8.40. Алиса хочет надежным способом послать Бобу открытым текстом подписанное сообщение P. У Алисы и у Боба есть закрытый (D_X) и открытый (D_Y) RSA-ключи. Предположим, что каждому из них известен открытый ключ другого. Способы передачи ключей мы рассмотрим позднее.

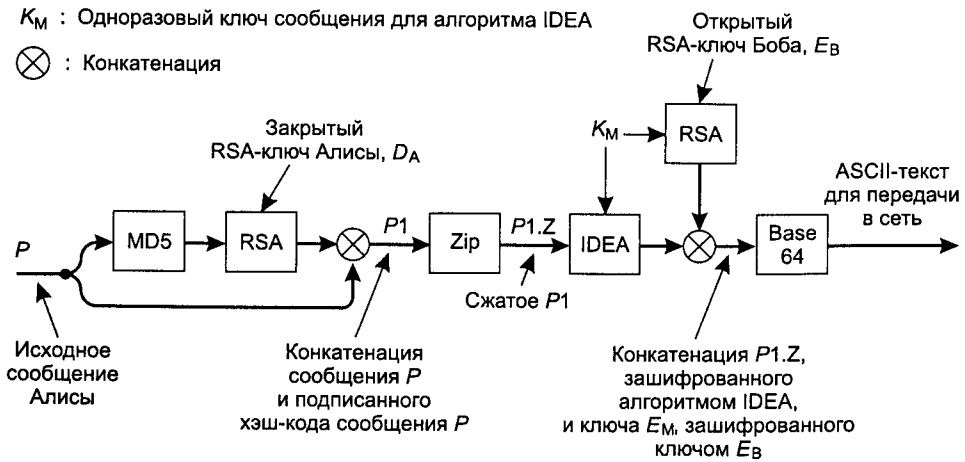


Рис. 8.40. Использование системы PGP для передачи сообщения

Алиса начинает с того, что запускает на своем компьютере программу PGP. Программа PGP сначала хэширует ее сообщение P с помощью алгоритма MD5, а затем шифрует полученный хэш-код при помощи ее закрытого RSA-ключа D_A. Получив это сообщение, Боб может расшифровать хэш-код открытым ключом Алисы и убедиться в его правильности. Даже если какой-либо злоумышленник мог получить хэш на этой стадии и расшифровать его известным открытым ключом Алисы, сила алгоритма MD5 гарантирует невозможность создания другого сообщения с тем же хэш-кодом (из-за трудоемкости вычислений).

Затем зашифрованный хэш-код и оригинальное сообщение объединяются в единое сообщение $P1$, которое сжимается с помощью программы ZIP, использующей алгоритм Зива—Лемпеля (Ziv—Lempel, 1977). Будем называть результат этого этапа $P1.Z$.

Затем программа PGP предлагает Алисе ввести случайную текстовую строку. При формировании 128-разрядного ключа сообщения K_M для алгоритма IDEA учитываются как содержимое, так и скорость ввода. (В PGP-литературе этот ключ назван сеансовым, что является неправильным употреблением термина, так как никакого сеанса нет.) Затем $P1.Z$ шифруется алгоритмом IDEA с помощью ключа K_M в режиме шифрованной обратной связи. Кроме того, ключ K_M шифруется открытым ключом Боба, E_B. Эти два компонента объединяются и преобразуются в кодировку base64, о которой уже рассказывалось в главе 7, когда мы говорили о стандартах MIME. Получающееся в результате сообщение содержит только буквы, цифры и символы +, * и =, что означает, что это сообщение может быть помещено в тело письма стандарта RFC 822, и можно надеяться, что оно прибудет к получателю без изменений.

Получив сообщение, Боб выполняет обратное преобразование Base64 и расшифровывает IDEA-ключ своим закрытым RSA-ключом. С помощью IDEA-ключа он расшифровывает сообщение и получает $P1.Z$. Распаковав zip-файл, Боб отделяет зашифрованный хэш-код от открытого текста и расшифровывает его открытым ключом Алисы. Если в результате обработки открытого текста алгоритмом MD5 получается тот же самый хэш-код, это означает, что сообщение P действительно пришло от Алисы.

Следует отметить, что алгоритм RSA используется здесь только в двух местах: для зашифровки 128-разрядного MD5-хэша и 128-разрядного IDEA-ключа. Алгоритм RSA медленный, но ему нужно зашифровать всего лишь 256 бит, что совсем немного. Более того, все эти 256 бит в высшей степени случайны, поэтому злоумышленнику придется очень сильно попотеть, чтобы угадать ключ. Основное шифрование выполняется алгоритмом IDEA, который на порядок быстрее, чем RSA. Итак, система PGP обеспечивает секретность, сжатие и цифровую подпись, и делает это намного эффективнее, чем схема, показанная на рис. 8.16.

Система PGP поддерживает четыре длины ключа RSA. Пользователь может самостоятельно выбирать нужную длину. Предлагаются следующие варианты длины:

1. Несерьезная (384 бит): шифр может быть взломан сегодня же организациями с большим бюджетом.
2. Коммерческая (512 бит): возможно, шифр смогут взломать организации из трех букв.
3. Военная (1024 бит): никто на Земле не сможет взломать этот шифр.
4. Межпланетная (2048 бит): никто во всей вселенной не сможет взломать шифр.

Поскольку алгоритм RSA используется только для двух небольших вычислений, всем следует всегда применять ключи межпланетного варианта, длиной 2048 бит.

Формат PGP-сообщения показан на рис. 8.41. Сообщение состоит из трех частей: области ключа, области подписи и области сообщения. Область ключа, помимо самого IDEA-ключа, содержит также идентификатор ключа, так как пользователям разрешено иметь несколько открытых ключей.

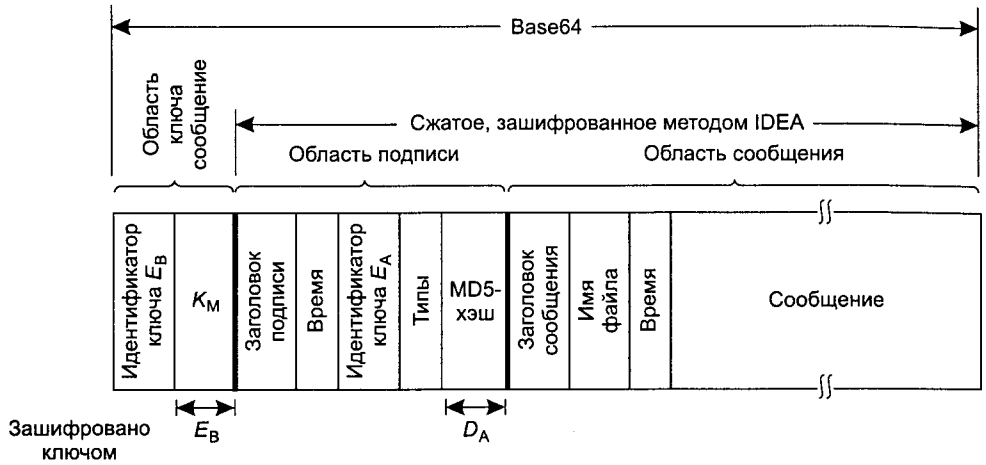


Рис. 8.41. PGP-сообщение

Область подписи содержит заголовок, который нас сейчас не интересует. За заголовком следует временной штамп, идентификатор открытого ключа отправителя, с помощью которого получатель сможет расшифровать хэш-код, используемый в качестве подписи. Следом идет идентификатор использованных алгоритмов шифрования и хэширования (чтобы можно было пользоваться, например, MD6 или RSA2, когда они будут разработаны). Последним в области подписи располагается сам зашифрованный хэш-код.

Часть сообщения также содержит заголовок, имя файла по умолчанию, на случай, если получатель будет сохранять принятое сообщение на диске, временной штамп создания сообщения и, наконец, само сообщение.

Работе с ключами в системе PGP было уделено особенное внимание, так как это ахиллесова пята всех систем защиты. У каждого пользователя локально находится две структуры данных: набор закрытых ключей и набор открытых ключей (эти наборы иногда называют связками). **Связка закрытых ключей** содержит несколько личных пар ключей, состоящих из закрытого и открытого ключей. Несколько пар ключей поддерживаются, чтобы позволить пользователям периодически их менять, когда возникают опасения, что тот или иной ключ скомпрометирован. При этом для смены ключа не требуется принимать каких-либо экстренных мер по передаче нового ключа. У каждой пары ключей есть связанный с ней идентификатор, так что отправителю нужно всего лишь сообщить получателю, которым открытым ключом был зашифрован ключ сообщения. Идентификатор сообщения состоит из младших 64 разрядов открытого ключа. За отсутствие конфликтов между идентификаторами ключей отвечают сами пользователи.

Закрытые ключи на диске зашифрованы специальным паролем (произвольной длины), защищающем их от кражи.

Связка открытых ключей содержит открытые ключи корреспондентов пользователей. Они нужны для зашифровки ключей сообщений, связанных с каждым сообщением. Каждая запись набора открытых ключей содержит не только ключ, но также его 64-разрядный идентификатор и отметку, указывающую степень доверия пользователя этому ключу.

Степень доверия ключу зависит, например, от способа его получения. Предположим, что открытые ключи расположены на электронных досках объявлений (BBS). Злоумышленник может атаковать доску объявлений и подменить размещенный там открытый ключ Боба своим ключом. Когда Алиса попытается воспользоваться подмененным ключом, злоумышленник сможет применить к Бобу атаку типа «человек посередине».

Чтобы предотвратить подобные атаки или хотя бы минимизировать их ущерб, Алисе необходимо знать, насколько она может доверять открытому ключу Боба, хранящемуся в ее наборе открытых ключей. Если Боб лично дал ей дискету с ключом, она может поставить такому ключу максимальную степень доверия. В этом и заключается децентрализованный, контролируемый пользователем подход к управлению открытыми ключами, отличающий PGP от централизованной схемы PKI.

Однако на практике открытые ключи часто получают, опрашивая доверенный сервер ключей. По этой причине после стандартизации X.509 система PGP стала поддерживать сертификаты наряду с традиционным для PGP механизмом связки открытых ключей. Все современные версии PGP имеют поддержку X.509.

PEM — почта повышенной секретности

В противоположность системе PGP, целиком созданной одним человеком, система **PEM** (Privacy Enhanced Mail — почта повышенной секретности) является официальным стандартом Интернета и описана в четырех RFC: с RFC 1421 по RFC 1424. Система PEM обладает примерно тем же набором функций, что и система PGP: секретность и аутентификация для систем электронной почты стандарта RFC 822. Тем не менее, она имеет несколько отличий от системы PGP в методах и технологии.

Сообщения, посылаемые с помощью системы PEM, сначала преобразуются в каноническую форму, удовлетворяющую особому набору правил, касающихся использования пробелов, табуляторов, символов возврата каретки и перевода строки. Затем с помощью алгоритма MD2 или MD5 вычисляется хэш-код сообщения. Потом конкатенация хэш-кода и сообщения шифруются при помощи алгоритма DES. В свете обсуждавшихся недостатков 56-разрядного ключа выбор именно этой системы шифрования, несомненно, вызывает большие подозрения. Затем зашифрованное сообщение преобразуется в кодировку base64 и передается получателю.

Как и в PGP, каждое сообщение шифруется одноразовым ключом, передаваемым вместе с сообщением. Ключ может быть защищен либо алгоритмом RSA, либо тройным применением системы DES в режиме EDE.

Управление ключами в системе PEM более структурированное, чем в PGP. Ключи сертифицируются по стандарту X.509 управлениями сертификации, организованными в виде жесткой иерархии с единым центром управления на ее вершине. Преимущество этой схемы в том, что сертификаты могут отзываться, для чего центр управления периодически публикует «черные списки».

С системой PEM связана только одна небольшая проблема: никто ею никогда не пользовался. Это связано, прежде всего, с политикой: как решить, кто и при каких условиях должен стать центром управления? Недостатка в кандидатах никогда не ощущалось, однако многие боятся доверять безопасность всей системы какой-либо одной компании. Наиболее серьезным кандидатом представлялась корпорация RSA Security, но она собиралась взимать плату за каждый выданный сертификат. Такая идея многим не понравилась. В частности, правительству США всегда было разрешено безвозмездно пользоваться американскими патентами, да и компании за пределами США привыкли бесплатно пользоваться алгоритмом RSA (разработчики забыли запатентовать его за пределами США). Ни те, ни другие не воодушевились внезапной необходимостью оплаты услуг RSA Security, которые они всегда получали бесплатно. В итоге центр управления так и не был выбран, а система PEM потерпела неудачу.

S/MIME

Следующим изобретением IETF в области обеспечения конфиденциальности электронной почты стала система под названием **S/MIME** (Secure/MIME — защищенный MIME). Она описывается в RFC с 2632 по 2643. Подобно PEM, она обеспечивает аутентификацию, целостность данных, секретность и проверку подлинности информации. Обладает неплохой гибкостью, поддерживает разнообразные криптографические алгоритмы. По названию можно догадаться, что S/MIME тесно связана с MIME в том смысле, что позволяет защищать любые типы сообщений. Определено множество новых заголовков MIME, например, для цифровых подписей.

Группа IETF определенно извлекла какие-то уроки из опыта PEM. В S/MIME нет жесткой иерархии сертификатов, отсутствует единый центр управления. Вместо этого пользователи могут работать с набором доверительных якорей. До тех пор, пока сертификат может быть проверен по доверительному якорю, он считается корректным. Система S/MIME использует стандартные алгоритмы и протоколы, которые мы уже рассматривали, поэтому на этом мы закончим ее обсуждение. Более подробную информацию вы найдете в RFC.

Защита информации во Всемирной паутине

Мы только что закончили изучение двух важных областей, в которых требуется защита информации, — соединения и электронная почта. Можно сказать, что это были аперитив и суп. Теперь же мы приступаем к главному блюду: защите ин-

формации во Всемирной паутине. Именно в WWW работает большинство злоумышленников, делая свое грязное дело. В следующих разделах будут рассмотрены некоторые проблемы, относящиеся к безопасности в Паутине.

Эту тему можно разделить на три части. Первая связана с безопасным именованием объектов и ресурсов. Вторая — с установлением аутентифицированных соединений. Третья — с тем, что случается, когда веб-сайт отправляет клиенту исполняемый код. После перечисления возможных опасностей мы рассмотрим все эти вопросы.

Возможные опасности

Практически каждую неделю газеты публикуют статьи о проблемах безопасности во Всемирной паутине. Ситуация складывается действительно довольно мрачная. Посмотрим на некоторые примеры того, что уже имело место. Во-первых, мы помним, как домашние страницы многочисленных организаций самых разных масштабов подвергались атакам хакеров и заменялись подложными страницами. (Термин «хакер» (hacker) приобрел значение «взломщик» благодаря журналистам, которые мало что понимали в компьютерном мире, но попытались воспользоваться профессиональным жаргоном программистов. На самом же деле изначально хакерами называли великих программистов. Взломщиков же мы и называем взломщиками (cracker).) В списке сайтов, которые удалось взломать, находятся такие, как Yahoo!, сайт Вооруженных сил США, ЦРУ, НАСА, а также New York Times. В большинстве случаев взломщики просто заменяли оригиналы на свои странички с каким-нибудь смешным (обычно издевательским) текстом, и уже через несколько часов сайты удавалось восстановить.

Однако были и гораздо более серьезные атаки. Многие сайты были сломаны за счет искусственно созданной чрезмерной нагрузки (атака типа «отказ в обслуживании», DoS), с которой заведомо не может справиться сервер. Зачастую такие нападения совершались сразу с нескольких машин, которые взломщику уже удалось сломать и заставить против воли участвовать в преступлении («распределенный DoS», DDoS). Такие атаки настолько распространены, что уже перестали быть новостью. Тем не менее, ущерб от них исчисляется тысячами долларов.

В 1999 году шведский взломщик проник на сайт Hotmail (корпорации Microsoft) и создал зеркало, на котором все желающие могли ввести имя любого пользователя этого сайта и прочесть всю его текущую почту и почтовые архивы.

А один русский 19-летний взломщик по имени Максим смог украсть с сайта, посвященного электронной коммерции, номера 300 000 кредитных карт. Затем он обратился к их владельцам и сообщил, что если они не заплатят ему 100 000 долларов, он опубликует номера кредиток в Интернете. Они не поддались на провокацию, и тогда он действительно опубликовал номера кредитных карт, что нанесло серьезный ущерб невинным жертвам.

23-летний студент из Калифорнии послал по электронной почте в агентство новостей фальшивый пресс-релиз, в котором сообщалось об огромных убытках корпорации Emulex и об уходе в отставку ее генерального директора. Спустя несколько часов биржевые цены на акции Emulex снизились на 60 %, в результате чего их держатели лишились более 2 миллиардов долларов. Злоумышленник заработал около четверти миллиона долларов, продав акции незадолго до своего ложного заявления. Хотя в данном случае взлом не произошел непосредственно во Всемирной паутине, понятно, что объявление подобного рода, размещенное на сайте компании, привело бы к такому же эффекту.

К сожалению, одно перечисление таких примеров могло бы занять несколько страниц. И теперь нам пора обратиться к технической стороне дела. Более подробную информацию, касающуюся проблем безопасности всех видов, см. в (Anderson, 2001; Garfinkel и Spafford, 2002; Schneier, 2000). Поиск в Интернете также даст неплохие результаты.

Безопасное именование ресурсов

Начнем с чего-нибудь очень простого. Допустим, Алиса хочет посетить веб-сайт Боба. Она набирает в браузере URL, и через несколько секунд появляется страничка. Но в самом ли деле эта страничка создана Бобом? Может, да, а может, нет. Не исключено, что Труди снова принялась за свои шуточки. Например, она могла перехватить исходящие от Алисы пакеты и изучить их. Найдя запрос *GET* на получение страницы Боба, Труди могла сама зайти на эту страницу, изменить ее и отослать Алисе. Алиса не заметила бы ровным счетом ничего. Хуже того, Труди могла изменить цены на акции на более низкие, сделав тем самым предложение Боба очень привлекательным. Вероятность того, что теперь Алиса вышлет номер своей кредитной карты «Бобу» (с целью приобрести акции по выгодной цене), резко повысилась.

Одним из недостатков схемы «человек посередине» является то, что Труди должна быть в состоянии перехватывать исходящий трафик Алисы и подделывать свой исходящий трафик. На практике она должна прослушивать телефонную линию либо Боба, либо Алисы (поскольку прослушивание оптоволоконного кабеля — задача непростая). Это, конечно, возможно, но Труди не только умна и хитра, но и ленива. Она знает более простые способы обмануть Алису.

Обман DNS

Допустим, Труди может взломать систему DNS (например, ту ее часть, которая хранится в кэше DNS у провайдера Алисы) и заменить IP-адрес Боба (например, 36.1.2.3) своим IP-адресом (например, 42.9.9.9). Тогда можно провести атаку. То, как все должно работать в нормальной ситуации, показано на рис. 8.42, *а*: 1) Алиса запрашивает у службы DNS IP-адрес Боба; 2) получает его; 3) она запрашивает домашнюю страничку Боба; 4) получает ее. После того как Труди заменяет IP-адрес Боба на свой собственный, мы получаем ситуацию, показанную на рис. 8.42, *б*. Алиса ищет IP-адрес Боба, а получает вместо него IP-адрес злоумышленницы Труди, поэтому весь трафик Алисы, предназначенный для Боба, приходит, на самом деле, Труди. Та может организовать атаку типа «человек посередине», не мучаясь с установкой «крокодилов» на телефонной линии Алисы. Вместо этого она может заменить всего одну запись на сервере имен DNS. Это, согласитесь, более просто.

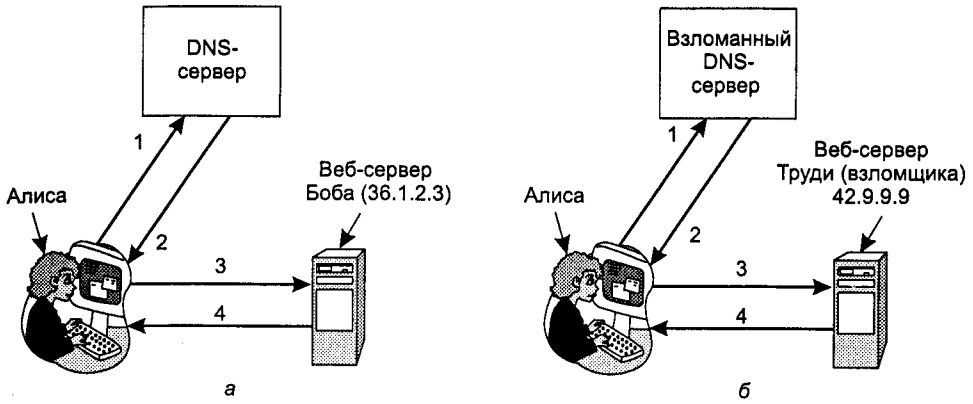


Рис. 8.42. Нормальная ситуация (*а*); атака со взломом DNS и изменением записи, относящейся к Бобу (*б*)

Как Труди удалось обмануть DNS? А это оказалось не таким уж сложным делом. Если не вдаваться в подробности, можно описать процесс так: Труди обманным путем заставляет DNS-сервер провайдера Алисы послать запрос для поиска адреса Боба. К несчастью, так как DNS использует UDP, сервер не может узнать, кто является реальным отправителем ответа. Труди использует это свойство, фальсифицируя ожидаемый ответ и тем самым занося неверные сведения об IP-адресе Боба в кэш DNS-сервера. Для простоты мы будем предполагать, что провайдер Алисы изначально не имеет сведений о веб-сайте Боба, bob.com. Если же такие сведения есть, злоумышленник может выждать, пока срок действия записи истечет, и попробовать еще раз (либо применить другие хитрости).

Труди начинает свою атаку с того, что посылает провайдеру Алисы запрос на поиск IP-адреса bob.com. Так как соответствующая запись отсутствует, сервер, в свою очередь, опрашивает сервер домена верхнего уровня (.com). Но Труди опережает этот сервер и посылает ложный ответ, в котором сообщается, что IP-адрес bob.com якобы 42.9.9.9. Как мы знаем, в реальности это адрес Труди. Так как этот ответ приходит первым, данные из него заносятся в кэш сервера провайдера, а настоящий ответ, если он приходит позже, отвергается. Установка ложного IP-адреса называется **обманом DNS**. А кэш, в котором хранится заведомо ложный IP-адрес, называется **отравленным кэшем**.

Надо сказать, что на практике все не так просто. Во-первых, провайдер Алисы все-таки проверяет наличие в ответе правильного адреса сервера верхнего уровня. Но Труди может написать в соответствующем поле что угодно и преодолеть эту преграду. Учитывая то, что адреса серверов верхнего уровня общедоступны, сделать это несложно.

Во-вторых, для того чтобы DNS-сервер мог понять, какому запросу соответствует ответ, во все запросы добавляются порядковые номера. Чтобы обмануть

провайдера Алисы, Труди должна знать текущий порядковый номер. Самый простой способ узнать его — это зарегистрировать собственный домен, например, trudy-the-intruder.com.

Предположим, что IP-адрес этого домена также 42.9.9.9. Труди создает DNS-сервер для этого домена: dns.trudy-the-intruder.com. Его IP-адрес тот же самый (42.9.9.9), поскольку оба домена расположены на одном и том же компьютере. Теперь надо заставить провайдера Алисы поинтересоваться DNS-сервером Труди. Сделать это несложно. Требуется лишь запросить, например, foobar.trudy-the-intruder.com, и серверу провайдера Алисы придется опросить сервер верхнего уровня, .com, и узнать у него, кто обслуживает новый домен Труди.

И вот теперь, когда запись dns.trudy-the-intruder.com занесена в кэш провайдера, можно спокойно начинать атаку. Труди запрашивает у провайдера Алисы www.trudy-the-intruder.com, а тот в ответ посылает на DNS-сервер Труди соответствующий запрос. Вот в этом-то запросе и содержится нужный злоумышленнице порядковый номер. Теперь Труди должна действовать без промедления: она ищет с помощью провайдера Алисы Боба и тут же отвечает на собственный вопрос, посылая фальшивку: «Адрес bob.com: 42.9.9.9». Этот подделанный ответ несет в себе порядковый номер на единицу больше только что полученного. За время атаки она может послать еще одну фальшивку, с номером, на два больше полученного, а также еще около дюжины таких «ответов» с увеличивающимися номерами. Задача одного из них нам уже ясна. Остальные никому не нужны, их просто выкинут. После прибытия фальшивого ответа на запрос Алисы он будет помещен в кэш; к тому времени, когда доберется настоящий ответ, он будет отвергнут, так как сервер уже ничего не ожидает.

И вот Алиса ищет IP-адрес bob.com и узнает, что он равен 42.9.9.9. Как мы знаем, это адрес Труди, которая провела успешную атаку типа «человек посередине», не выходя из своей комнаты. Последовательность предпринятых ею шагов показана на рис. 8.43. К сожалению, это еще и не единственный способ обмануть DNS. Этих способов действительно много.

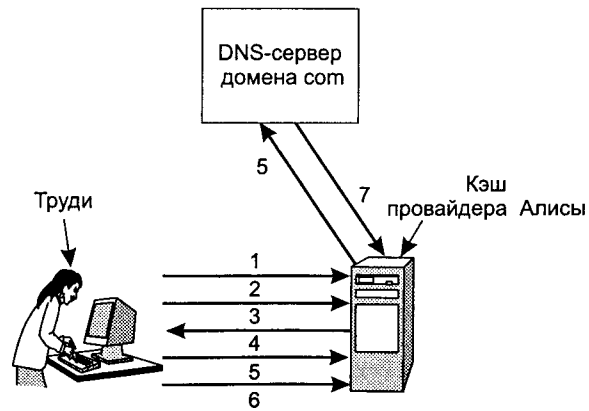


Рис. 8.43. Обман провайдера Алисы

Защита DNS

Избежать атак описанного типа можно, заставив DNS-серверы использовать случайные идентификаторы в своих запросах вместо того, чтобы просто инкрементировать их. К сожалению, заткнув одну «дыру», мы обнаруживаем другую. Настоящая проблема в том, что служба DNS разрабатывалась в те времена, когда Интернет был чисто исследовательской сетью, работавшей в нескольких сотнях университетов, и ни Алиса, ни Боб, ни Труди на этот праздник жизни приглашены не были. Вопрос защиты информации тогда еще не стоял; задачей-максимум было заставить Интернет функционировать. Но с годами среда, в которой приходилось выживать Интернету, сильно изменилась, поэтому в 1994 году IETF основала рабочую группу, задачей которой было защитить DNS. Этот проект известен под названием **DNSsec (Защита DNS)**; результаты работы группы опубликованы в RFC 2535. К сожалению, DNSsec до сих пор не удается развернуть в больших масштабах, поэтому многие DNS-серверы продолжают подвергаться нападениям злоумышленников.

Концептуально система DNSsec очень проста. Она основана на шифровании с открытыми ключами. Каждая зона DNS (в терминах рис. 7.2) обладает парой ключей, а именно, открытым и закрытым. Вся информация, отправляемая DNS-сервером, подписывается с помощью закрытого ключа зоны отправителя, поэтому аутентичность может быть запросто проверена принимающей стороной.

DNSsec предоставляет три основные услуги:

1. Подтверждение места отправления данных.
2. Распространение открытых ключей.
3. Аутентификацию транзакций и запросов.

Самой главной является первая услуга, с ее помощью проверяется то, что пришедшие данные были подтверждены их отправителем. Вторая услуга полезна для безопасного хранения и извлечения открытых ключей. Третья позволяет защититься от атак повторного воспроизведения и обмана сервера. Обратите внимание: секретность здесь не обеспечивается, этой задачи нет, поскольку вся информация DNS считается открытой. Так как процесс введения DNSsec в строй, скорее всего, будет продолжаться в течение нескольких лет, важно предоставить возможность общения между собой серверам, снабженным системой защиты и не снабженным ею. Это неявно подразумевает, что протокол изменять нельзя. Рассмотрим теперь эту систему чуть более детально.

Записи DNS группируются в наборы, называемые **RRSet** (Resource Record Set — набор записей ресурсов). В набор входят все записи с одинаковыми именами, классами и типами. Скажем, в наборе может быть несколько записей *A*, если имя DNS соответствует первичному и вторичному IP-адресам. Наборы расширяются за счет некоторых новых типов записей (обсуждаются далее). Каждый RRSet хэшируется (например, с использованием MD-5 или SHA-1). Хэш подписывается при помощи закрытого ключа зоны (например, по алгоритму RSA). Единицей передаваемой клиентам информации является подписанный RRSet. Получив его, клиент может проверить, действительно ли для генерации подписи

был взят закрытый ключ зоны отправителя. Если подпись корректна, данные принимаются. Так как каждый RRSet содержит собственную подпись, наборы можно кэшировать где угодно, даже на не слишком надежных серверах, не опасаясь за их судьбу.

Система DNSec вводит несколько новых типов записей. Первая из них — это запись *KEY*. В ней хранятся открытый ключ зоны, пользователя, хоста или другого принципала, криптографический алгоритм генерации подписи, наименование протокола передачи и еще несколько бит. Открытый ключ хранится в незащищенном виде. Сертификаты X.509 не используются из-за их громоздкости. В поле алгоритма рекомендуемое значение, соответствующее MD5/RSA, равно 1, для других комбинаций используются другие значения. Поле протокола может указывать на использование IPsec или другого протокола защиты соединений (если таковой вообще применяется).

Второй новый тип записей — *SIG*. В такой записи содержится подписанный хэш, сформированный в соответствии с алгоритмом, указанным в *KEY*. Подпись охватывает все записи RRSet, включая все записи *KEY*, однако не включая саму себя. Здесь также содержатся время начала и конца действия подписи, имя владельца подписи и некоторая дополнительная информация.

Система DNSsec устроена так, что закрытый ключ зоны может храниться в автономном режиме. Один или два раза в день содержимое базы данных зоны можно вручную переносить (например, записав компакт-диск) на машину, работающую в автономном режиме и хранящую закрытый ключ. Там можно сгенерировать подписи для всех наборов, и полученные таким образом записи *SIG* можно снова записать на компакт-диск и перенести на главный сервер. Таким образом, закрытый ключ можно хранить на компакт-диске, запертом в сейфе и вынимаемом только для того, чтобы подписать на автономной машине ежедневное обновление наборов типа RRSet. По окончании генерации подписей все копии ключа удаляются из памяти, а диск и компакт-диск возвращаются в сейф. Эта процедура превращает электронную защиту информации в физическую, что гораздо понятнее пользователям.

Метод предварительного подписания наборов значительно ускоряет процесс обработки запросов, так как отпадает необходимость в шифровании «на лету». Платой за это является большой объем дискового пространства, необходимого для хранения всех ключей и подписей в базах данных DNS. Из-за этого некоторые записи увеличиваются в размере десятикратно.

Получив подписанный RRSet, клиент должен применить открытый ключ зоны для расшифровки хэша, затем вычислить хэш самостоятельно и сравнить два значения. В случае их соответствия данные считаются корректными. Тем не менее, эта процедура не решает вопрос получения клиентом открытого ключа зоны. Одним из способов является запрос ключа у надежного сервера и передача его по защищенному соединению (например, при помощи IPsec).

Однако на практике предполагается, что у клиентов уже есть открытые ключи всех доменов верхнего уровня. Если Алиса пожелает посетить сайт Боба, она запросит у службы DNS набор RRSet для bob.com, в котором будет содержаться IP-адрес и запись *KEY* с открытым ключом Боба. RRSet будет подписан доменом

Защита DNS

Избежать атак описанного типа можно, заставив DNS-серверы использовать случайные идентификаторы в своих запросах вместо того, чтобы просто инкрементировать их. К сожалению, заткнув одну «дыру», мы обнаруживаем другую. Настоящая проблема в том, что служба DNS разрабатывалась в те времена, когда Интернет был чисто исследовательской сетью, работавшей в нескольких сотнях университетов, и ни Алиса, ни Боб, ни Труди на этот праздник жизни приглашены не были. Вопрос защиты информации тогда еще не стоял; задачей-максимум было заставить Интернет функционировать. Но с годами среда, в которой приходилось выживать Интернету, сильно изменилась, поэтому в 1994 году IETF основала рабочую группу, задачей которой было защитить DNS. Этот проект известен под названием **DNSsec (Защита DNS)**; результаты работы группы опубликованы в RFC 2535. К сожалению, DNSsec до сих пор не удается развернуть в больших масштабах, поэтому многие DNS-серверы продолжают подвергаться нападениям злоумышленников.

Концептуально система DNSsec очень проста. Она основана на шифровании с открытыми ключами. Каждая зона DNS (в терминах рис. 7.2) обладает парой ключей, а именно, открытым и закрытым. Вся информация, отправляемая DNS-сервером, подписывается с помощью закрытого ключа зоны отправителя, поэтому аутентичность может быть запросто проверена принимающей стороной.

DNSsec предоставляет три основные услуги:

1. Подтверждение места отправления данных.
2. Распространение открытых ключей.
3. Аутентификацию транзакций и запросов.

Самой главной является первая услуга, с ее помощью проверяется то, что пришедшие данные были подтверждены их отправителем. Вторая услуга полезна для безопасного хранения и извлечения открытых ключей. Третья позволяет защититься от атак повторного воспроизведения и обмана сервера. Обратите внимание: секретность здесь не обеспечивается, этой задачи нет, поскольку вся информация DNS считается открытой. Так как процесс введения DNSsec в строй, скорее всего, будет продолжаться в течение нескольких лет, важно предоставить возможность общения между собой серверам, снабженным системой защиты и не снабженным ею. Это неявно подразумевает, что протокол изменять нельзя. Рассмотрим теперь эту систему чуть более детально.

Записи DNS группируются в наборы, называемые **RRSet** (Resource Record Set — набор записей ресурсов). В набор входят все записи с одинаковыми именами, классами и типами. Скажем, в наборе может быть несколько записей *A*, если имя DNS соответствует первичному и вторичному IP-адресам. Наборы расширяются за счет некоторых новых типов записей (обсуждаются далее). Каждый RRSet хэшируется (например, с использованием MD-5 или SHA-1). Хэш подписывается при помощи закрытого ключа зоны (например, по алгоритму RSA). Единицей передаваемой клиентам информации является подписанный RRSet. Получив его, клиент может проверить, действительно ли для генерации подписи

был взят закрытый ключ зоны отправителя. Если подпись корректна, данные принимаются. Так как каждый RRSet содержит собственную подпись, наборы можно кэшировать где угодно, даже на не слишком надежных серверах, не опасаясь за их судьбу.

Система DNSec вводит несколько новых типов записей. Первая из них — это запись *KEY*. В ней хранятся открытый ключ зоны, пользователя, хоста или другого принципала, криптографический алгоритм генерации подписи, наименование протокола передачи и еще несколько бит. Открытый ключ хранится в незащищенном виде. Сертификаты X.509 не используются из-за их громоздкости. В поле алгоритма рекомендуемое значение, соответствующее MD5/RSA, равно 1, для других комбинаций используются другие значения. Поле протокола может указывать на использование IPsec или другого протокола защиты соединений (если таковой вообще применяется).

Второй новый тип записей — *SIG*. В такой записи содержится подписанный хэш, сформированный в соответствии с алгоритмом, указанным в *KEY*. Подпись охватывает все записи RRSet, включая все записи *KEY*, однако не включая саму себя. Здесь также содержатся время начала и конца действия подписи, имя владельца подписи и некоторая дополнительная информация.

Система DNSsec устроена так, что закрытый ключ зоны может храниться в автономном режиме. Один или два раза в день содержимое базы данных зоны можно вручную переносить (например, записав компакт-диск) на машину, работающую в автономном режиме и хранящую закрытый ключ. Там можно сгенерировать подписи для всех наборов, и полученные таким образом записи *SIG* можно снова записать на компакт-диск и перенести на главный сервер. Таким образом, закрытый ключ можно хранить на компакт-диске, запертом в сейфе и вынимаемом только для того, чтобы подписать на автономной машине ежедневное обновление наборов типа RRSet. По окончании генерации подписей все копии ключа удаляются из памяти, а диск и компакт-диск возвращаются в сейф. Эта процедура превращает электронную защиту информации в физическую, что гораздо понятнее пользователям.

Метод предварительного подписания наборов значительно ускоряет процесс обработки запросов, так как отпадает необходимость в шифровании «на лету». Платой за это является большой объем дискового пространства, необходимого для хранения всех ключей и подписей в базах данных DNS. Из-за этого некоторые записи увеличиваются в размере десятикратно.

Получив подписанный RRSet, клиент должен применить открытый ключ зоны для расшифровки хэша, затем вычислить хэш самостоятельно и сравнить два значения. В случае их соответствия данные считаются корректными. Тем не менее, эта процедура не решает вопрос получения клиентом открытого ключа зоны. Одним из способов является запрос ключа у надежного сервера и передача его по защищенному соединению (например, при помощи IPsec).

Однако на практике предполагается, что у клиентов уже есть открытые ключи всех доменов верхнего уровня. Если Алиса пожелает посетить сайт Боба, она запросит у службы DNS набор RRSet для bob.com, в котором будет содержаться IP-адрес и запись *KEY* с открытым ключом Боба. RRSet будет подписан доменом

верхнего уровня (com), поэтому Алиса запросто сможет проверить подлинность набора. Пример содержимого набора RRSet приведен в табл. 8.4.

Таблица 8.4. Пример набора RRSet для bob.com. Запись KEY содержит открытый ключ Боба. Запись SIG — это хэш A и KEY, подписанный сервером домена верхнего уровня (com) для проверки их аутентичности

Имя домена	Время жизни	Класс	Тип	Значение
bob.com	86400	IN	A	36.1.2.3
bob.com	86400	IN	KEY	3682793A7B73F731029CE2737D...
bob.com	86400	IN	SIG	86947503A8B848F5272E53930C...

Теперь, вооружившись заверенной копией открытого ключа Боба, Алиса может узнать у DNS-сервера Боба IP-адрес www.bob.com. Этот RRSet будет подписан закрытым ключом Боба, поэтому Алиса сможет проверить подлинность подписи возвращенного Бобом набора. Если злоумышленнику каким-то образом удастся внедрить фальшивый RRSet в один из кэшей, Алиса заметит это, так как запись *SIG* будет неправильной.

Тем не менее, DNSsec также предоставляет криптографический механизм для связывания ответов с соответствующими запросами, предотвращающий атаки типа той, что показана на рис. 8.43. Эта (необязательная) мера заключается в добавлении к ответу хэша запроса, подписанного закрытым ключом опрашиваемого. Поскольку Труди неизвестен закрытый ключ сервера верхнего уровня (домена com), она не сможет подделать ответ этого сервера на запрос провайдера Алисы. Она, конечно, может опередить настоящий ответ, но фальшивка будет замечена по неправильной подписи хэшированного запроса.

DNSsec поддерживает и некоторые другие типы записей. Так, для хранения сертификатов (например, стандарта X.509) можно использовать запись *CERT*. Зачем нужна такая запись? Дело в том, что есть желающие превратить DNS в инфраструктуру PKI. Случится это на самом деле или нет, пока еще неизвестно. На этом мы заканчиваем обсуждение DNSsec. Более подробную информацию вы найдете в RFC 2535.

Самозаверяющиеся имена

Защита DNS — это не единственный способ защиты имен. Совершенно другой подход применяется в **защищенной файловой системе** (Mazi res и др., 1999). Авторами этого проекта была создана надежная, масштабируемая файловая система мирового масштаба, не требующая внесения изменений в структуру DNS, не использующая сертификаты и не подразумевающая существование инфраструктуры PKI. В этом разделе мы покажем, как эти идеи можно применить во Всемирной паутине. Соответственно, мы будем пользоваться веб-терминологией, а не понятиями файловых систем (именно последними оперирует документация, описывающая защищенную файловую систему). Однако во избежание недоразумений мы считаем своим долгом предупредить, что эта схема *может быть* применена в веб-технологиях для обеспечения безопасности, однако в данное время

она еще не используется, а для ее внедрения потребуется внести серьезные изменения в программное обеспечение.

Мы начнем с предположения о том, что каждый веб-сервер имеет два ключа: открытый и закрытый. Суть идеи состоит в том, чтобы каждый URL содержал хэш имени сервера и открытого ключа. Например, на рис. 8.44 мы видим URL фотографии Боба. Он начинается с традиционного http://, за которым следует имя сервера (www.bob.com). Далее ставится двоеточие, а за ним — 32-символьный хэш. В конце мы видим обычное имя файла. Если исключить хэш, получится вполне обычный URL. Вместе с хэшем он образует **самозаверяющийся URL**.

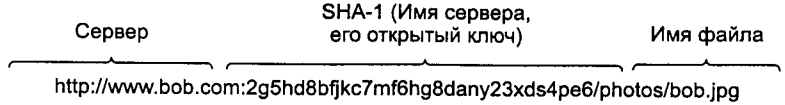


Рис. 8.44. Самозаверяющийся URL, содержащий хэш имени сервера и его открытого ключа

Сразу же возникает вопрос: для чего служит хэш? Он вычисляется конкатенацией (объединением) DNS-имени сервера с его открытым ключом и прогоном получающейся строки через функцию SHA-1, в результате чего получается 160-разрядный хэш. В данной схеме значение хэш-функции представлено последовательностью из 32 цифр и букв нижнего регистра. Из указанного алфавита во избежание путаницы исключены буквы «l» и «o» и цифры «1» и «0». В распоряжении остаются 32 символа, которые можно кодировать 5-разрядными битовыми последовательностями. В результате мы как раз и получаем $32 \cdot 5 = 160$ бит хэша SHA-1. На самом деле, вовсе не обязательно использовать хэш-функцию. Вместо этого можно вставлять и сам открытый ключ. Преимущество применения хэша заключается в том, что по сравнению с незашифрованным ключом уменьшается длина имени ресурса.

В простейшем (но наименее удобном) варианте Алисе, чтобы посмотреть фотографию Боба, придется набрать всю строку, показанную на рис. 8.44. Браузер отправляет на веб-сайт Боба просьбу прислать открытый ключ. По прибытии этого ключа браузер объединяет имя сервера с ключом и вычисляет хэш-функцию. Если результат согласуется с 32-символьным хэшем из защищенного URL, то у браузера не остается сомнений в том, что у него есть подлинный ключ Боба. В конце концов, даже если Труди перехватит запрос и подделает ответ, ей не удастся подобрать открытый ключ для получения корректного хэша благодаря свойствам алгоритма SHA-1. Любые попытки подлога будут сразу же выявлены. Что касается открытого ключа Боба, его можно поместить в кэш для будущего использования.

Теперь Алисе необходимо проверить соответствующий закрытый ключ Боба. Она формирует сообщение, содержащее AES-ключ сеанса, нонс и временной штамп. Затем она шифрует это сообщение с помощью открытого ключа Боба и отправляет его ему. Так как подходящий закрытый ключ есть только у Боба, только он сможет расшифровать сообщение и отослать назад нонс, зашифрованный ключом AES. Принятый Алисой корректный нонс убеждает ее в том, что она действительно разговаривает с Бобом. Теперь и у Алисы, и у Боба есть AES-

ключ сеанса, которым они будут пользоваться при дальнейшем обмене запросами и ответами.

Алисе достаточно один раз загрузить фотографию Боба (или любую веб-страницу). После этого она может поставить на данный ресурс закладку, и ей больше не придется вручную набирать всю эту огромную строку URL. Более того, указатели URL, расположенные на веб-страницах, также могут быть самозаверяющимися, и ими можно пользоваться обычным образом (просто щелкая на них мышкой). Однако такие указатели гарантируют, что вы получите настоящую, а не поддельную информацию. Еще один способ избежать ручного набора длинных строк URL — использовать для их получения защищенное соединение с надежным сервером или вписывать их в сертификаты X.509, подписанные управлением сертификации.

Другой путь получения самозаверяющихся URL заключается в установлении соединения с надежной поисковой машиной. Вначале придется набрать длинную строку, но затем можно будет воспользоваться описанным ранее протоколом, приводящим к защищенному, аутентифицированному соединению с надежной поисковой машиной. Результатом поиска будет страница (с электронной подписью), содержащая список нужных самозаверяющихся URL, на которых можно щелкать вместо того, чтобы вводить вручную в строке адреса в браузере.

Давайте теперь посмотрим, как этот подход противостоит обманным действиям Труди. Если ей удастся «отравить» кэш провайдера Алисы, запрос последней будет против ее воли направлен не к Бобу, а к злоумышленнице. Но наш протокол требует, чтобы получатель начального сообщения (то есть Труди) вернул открытый ключ, способный породить корректный кэш. Если Труди вернет открытый ключ Боба, Алиса не заметит подлога, но зашифрует свое следующее сообщение ключом Боба. Труди это сообщение получит, но расшифровать его не сможет. Следовательно, она не сможет расшифровать ни AES-ключ, ни нонс. В общем, худшее, что в этом случае можно сделать путем обмана DNS, это осуществить атаку типа DoS (отказ в обслуживании).

SSL — протокол защищенных сокетов

Ну что ж, защита имен ресурсов — это неплохое начало, однако этим не обеспечивается в полной мере безопасность во Всемирной паутине. Следующий шаг состоит в установлении безопасных соединений. Сейчас мы рассмотрим, как это делается.

Когда веб-технологии были впервые представлены широкой публике, они использовались для распространения статических страниц. Однако уже давным-давно некоторые компании задумались об использовании Паутины для выполнения финансовых транзакций, таких как покупка товаров по кредитным картам, онлайновые банковские операции, электронная торговля ценными бумагами. Для таких приложений требовалась организация защищенных соединений. В 1995 году тогдашний лидер среди производителей браузеров, корпорация Netscape Communications, в ответ на это представила систему безопасности под названием **SSL** (Secure Sockets Layer — протокол защищенных сокетов). Соответствующее программное обеспечение, как и сам протокол, в наше время используется очень широко (в том числе и программой Internet Explorer), поэтому стоит рассмотреть SSL более детально.

Итак, SSL создает защищенное соединение между двумя сокетами, позволяющее:

- клиенту и серверу договориться об используемых параметрах;
- клиенту и серверу произвести взаимную аутентификацию;
- организовать тайное общение;
- обеспечить защиту целостности данных.

Все перечисленные пункты нам уже знакомы, поэтому мы не будем их комментировать.

Расположение SSL в структуре обычного стека протоколов показано на рис. 8.45. По сути дела, между прикладным и транспортным уровнями появляется новый уровень, принимающий запросы от браузера и отсылающий их по TCP для передачи серверу. После установки защищенного соединения основная задача SSL заключается в поддержке сжатия и шифрования. Если поверх SSL используется HTTP, этот вариант называется **HTTPS** (Secure HTTP — защищенный HTTP) несмотря на то, что это обычный протокол HTTP. Впрочем, возможно и отличие: скажем, доступ может осуществляться через новый порт (443) вместо стандартного (80). Кстати говоря, область применения SSL не ограничивается исключительно веб-браузерами, но это наиболее распространенное применение.

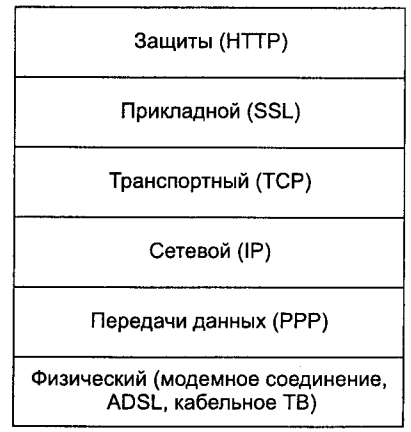


Рис. 8.45. Уровни (и протоколы), используемые обычным домашним браузером с SSL

Существует несколько версий протокола SSL. Далее мы будем обсуждать только версию 3, так она распространена наиболее широко. SSL поддерживает множество разных алгоритмов и может обладать разными дополнительными функциями, среди которых наличие или отсутствие сжатия, тот или иной алгоритм шифрования, а также некоторые вещи, связанные с ограничениями экспорта в криптографии. Последнее, в основном, предназначено для того, чтобы можно было

удостовериться, что оба конца соединения находятся в США. Иногда длину ключа ограничивают 40 битами, что шифровальщики воспринимают как своего рода шутку. Однако Netscape должен был ввести это ограничение, чтобы получить лицензию на экспорт от правительства США.

SSL состоит из двух субпротоколов, один из которых предназначен для установления защищенного соединения, а второй — для использования этого соединения. Начнем с рассмотрения вопроса установления соединения. Работа субпротокола, занимающегося этим, показана на рис. 8.46. Все начинается с сообщения 1, в котором Алиса посылает Бобу запрос на установку соединения. В нем указываются версия SSL, а также предпочтения Алисы относительно сжатия и алгоритмов шифрования. Также в нем содержится нонс R_A, который будет использован впоследствии.

Теперь наступает очередь Боба. В сообщении 2 он выбирает один из алгоритмов, поддерживаемых Алисой, и посылает собственный нонс R_B. В сообщении 3 он отсылает сертификат со своим открытым ключом. Если сертификат не подписан какой-нибудь уважаемой организацией, он также отправляет цепочку сертификатов, по которым Алиса может удостовериться в том, что сертификату Боба действительно можно доверять. Все браузеры, включая тот, что установлен у Алисы, изначально снабжаются примерно сотней открытых ключей, поэтому если среди присланных Бобом сертификатов встретится один из этих ключей, Алиса сможет по нему восстановить ключ Боба и проверить его. В этот момент Боб может прислать и другие сообщения (например, запрос на получение сетификата Алисы с ее открытым ключом). После окончания выполнения своей части протокола Боб посылает сообщение 4, в котором говорит, что настала очередь Алисы.

Алиса в ответ выбирает 384-разрядный **подготовительный ключ** и посылает его Бобу, зашифровав предварительно своим открытым ключом (сообщение 5). Настоящий ключ сеанса вычисляется при помощи подготовительного ключа и нонсов обеих сторон. Это довольно сложная процедура. После получения сообщения 5 и Алиса, и Боб могут вычислить ключ сеанса. Для этого Алиса просит Боба переключиться на новый шифр (сообщение 6), а также сообщает о том, что она считает субпротокол установления соединения оконченным (сообщение 7). Боб соглашается с ней (сообщения 8 и 9).

Однако несмотря на то, что Алиса знает, кто такой Боб, последний Алису не знает (если только у нее нет открытого ключа и сертификата к нему, что довольно необычно для обычного физического лица). Поэтому первым сообщением для Алисы запросто может оказаться просьба пройти регистрацию, используя полученные ранее имя пользователя и пароль. Впрочем, протокол регистрации в системе не выходит за область полномочий SSL. Так или иначе, по окончании этой серии запросов-подтверждений может начинаться передача данных.

Как уже говорилось, SSL поддерживает разнообразные криптографические алгоритмы. Наиболее сильный из них использует для шифрации тройной DES с тремя отдельными ключами и SHA-1 для обеспечения целостности данных. Такое сочетание алгоритмов работает довольно медленно, поэтому применяется в основном при выполнении банковских операций и в других приложениях, в которых

требуется высокий уровень защиты. В обычных приложениях электронной коммерции для шифрации применяется 128-разрядный ключ, а для аутентификации — MD5. В качестве исходных данных RC4 передается 128-разрядный ключ, который разрастается во много раз при работе алгоритма. Это внутреннее число используется для создания ключевого потока. Последний суммируется по модулю 2 с открытым текстом, в результате чего получается обычный потоковый шифр, как было показано на рис. 8.12. Экспортные версии алгоритма также работают с алгоритмом RC4 и 128-разрядным ключом, однако 88 из этих разрядов делаются открытыми, что позволяет довольно быстро взломать шифр.

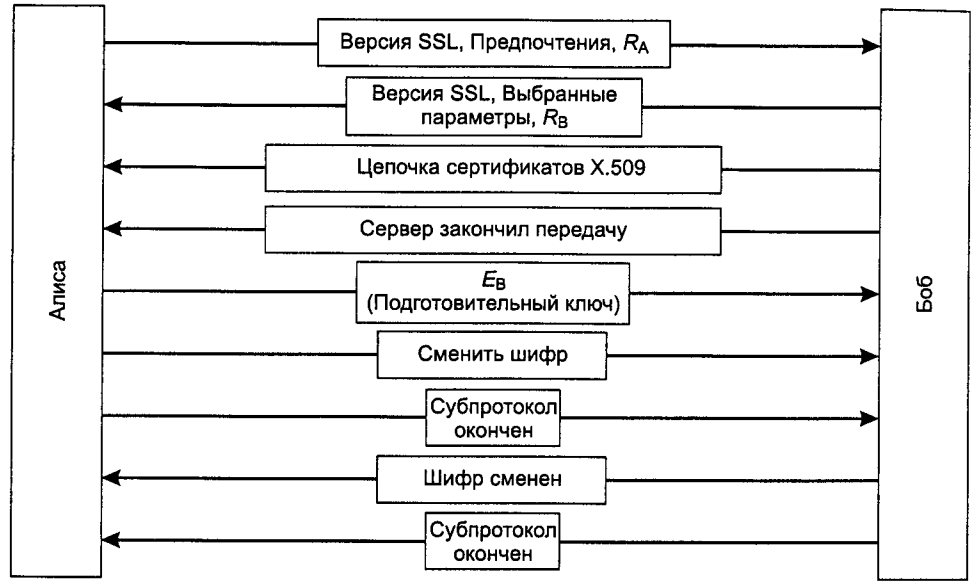


Рис. 8.46. Упрощенный вариант субпротокола SSL установления соединения

Для реальной передачи данных используется второй субпротокол, показанный на рис. 8.47. Сообщения, поступающие от браузера, разбиваются на единицы данных размером до 16 Кбайт. Если сжатие включено, каждая из этих единиц независимо сжимается. Затем по двум нонсам вычисляется закрытый ключ, подготовительный ключ объединяется со сжатым текстом и результат хэшируется по согласованному алгоритму (чаще всего MD5). Хэш добавляется к каждому фрагменту в виде MAC (Message Authetication Code — код аутентификации сообщения). Этот сжатый фрагмент вместе с MAC кодируется согласованным алгоритмом с симметричным ключом (обычно это суммирование по модулю 2 с ключевым потоком RC4). Наконец, присоединяется заголовок фрагмента, и фрагмент передается по TCP-соединению.

Следует остерегаться следующего подводного камня: уже говорилось о том, что RC4 имеет некоторые слабые ключи, которые довольно просто взламываются, поэтому SSL с RC4 — это довольно шаткая основа (Fluhrer и др., 2001). Браузеры, позволяющие пользователю выбирать тот или иной шифр, лучше всего

настраивать на постоянное использование тройного алгоритма DES со 168-разрядными ключами и SHA-1 невзирая на то, что такая комбинация работает еще медленнее, чем RC4 + MD5.

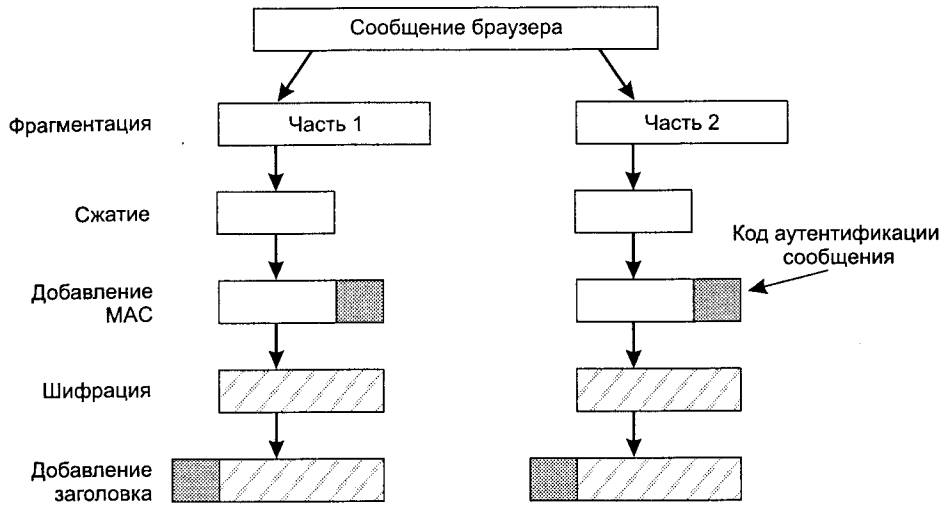


Рис. 8.47. Передача данных при использовании SSL

С SSL связана еще одна проблема: у принципалов может не быть сертификатов, а даже если они есть, далеко не всегда производится проверка соответствия ключей и сертификатов.

В 1996 году корпорация Netscape Communications направила SSL на стандартизацию в IETF. Результатом стал стандарт **TLS** (Transport Layer Security — защита транспортного уровня). Он описан в RFC 2246.

В SSL при создании стандарта TLS было внесено не так уж много изменений, однако их оказалось достаточно для того, чтобы SSL версии 3 и TLS стали несовместимыми. Например, в целях усиления ключа был изменен способ вычисления ключа сеанса по подготовительному ключу и нонсам. TLS также известен как SSL версии 3.1. Первая реализация появилась в 1999 году, однако до сих пор не очень понятно, заменит ли TLS SSL, даже учитывая то, что TLS несколько надежнее. Проблема с ключами RC4, между прочим, никуда не исчезла.

Защита переносимых программ

Именование ресурсов и соединения — это две области, которые, несомненно, тесно связаны с защитой информации во Всемирной паутине. Однако существуют и другие, не менее важные вопросы, связанные с той же темой. Поначалу веб-страницы представляли собой полностью статические HTML-файлы и не содержали исполняемый код. Теперь же на веб-страницах очень часто встречаются небольшие программы: Java-апплеты, управляющие элементы ActiveX, скрипты JavaScript. Загрузка и выполнение таких **переносимых программ**, очевидно, связаны

с большим риском возникновения массовых атак. Были разработаны различные методы, направленные на минимизацию этого риска. Далее мы обозначим некоторые вопросы, связанные с проблемами защиты переносимых программ.

Защита Java-апплетов

Java-апплеты — это небольшие программы на языке Java, откомпилированные в машинный язык со стековой организацией под названием **JVM** (Java Virtual Machine — виртуальная машина Java). Такие программы могут размещаться на веб-странице и загружаться вместе с ней. После загрузки страницы апплеты обрабатываются интерпретатором JVM в браузере, как показано на рис. 8.48.

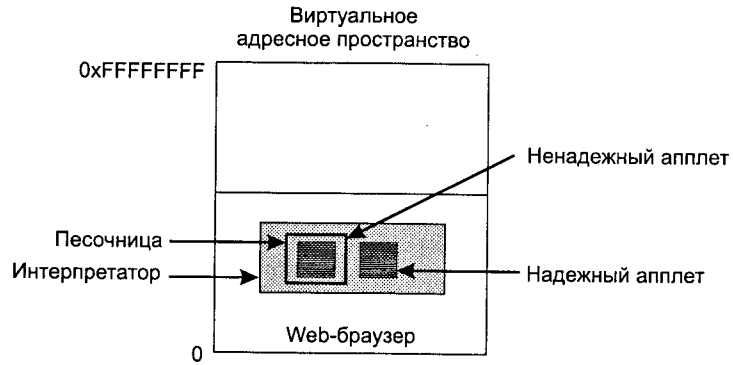


Рис. 8.48. Апплеты могут интерпретироваться веб-браузером

Преимущество интерпретируемого кода перед компилируемым состоит в том, что перед исполнением изучается каждая инструкция. Это дает интерпретатору возможность проверить состоятельность адреса инструкции. Кроме того, системные вызовы также перехватываются и интерпретируются. Как именно они обрабатываются, зависит от политики защиты информации. Например, если апплет надежный (например, он был создан на локальном диске), его системные вызовы могут обрабатываться без дополнительных проверок. Если же апплет не может считаться надежным (например, он был загружен из Интернета), его можно поместить в так называемую **песочницу**, регулирующую его поведение и пресекающую его попытки использовать системные ресурсы.

Если апплет пытается захватить системный ресурс, вызов передается монитору безопасности, который может разрешить или запретить данное действие. Монитор исследует вызов с точки зрения локальной политики защиты информации и затем принимает нужное решение. Таким образом, можно предоставить апплетам доступ к некоторым (но не ко всем) ресурсам. К сожалению, в реальной жизни такая модель работает плохо, в ней постоянно возникают ошибки.

ActiveX

Управляющие элементы ActiveX — это двоичные программы, рассчитанные на процессор Pentium, которые можно внедрять в веб-страницы. Когда на странице

встречается такая программа, производится проверка необходимости ее выполнения, и в случае положительного ответа она запускается. Эти программы не интерпретируются и не помещаются в песочницы, поэтому они обладают такими же возможностями, как обычные пользовательские программы, и, в принципе, могут нанести большой вред. Таким образом, вся защита информации в данном случае сводится к вопросу о том, стоит ли запускать управляющий элемент.

Для принятия таких решений корпорацией Microsoft был выбран метод, базирующийся на **подписях кода**. Суть в том, что каждый элемент ActiveX снабжается цифровой подписью, а именно хэшем кода, подписанным его создателем с использованием открытого ключа. Когда браузер встречает управляющий элемент, он вначале проверяет правильность подписи, убеждаясь в том, что код не был заменен по дороге. Если подпись корректна, браузер проверяет по своим внутренним таблицам, можно ли доверять создателю программы. Возможно, про самого создателя ничего не известно, но существует цепочка заверений, ведущая к какому-либо известному своей надежностью разработчику. Если создатель надежный, программа выполняется, в противном случае игнорируется. Система, созданная Microsoft для проверки управляющих элементов ActiveX, называется **Authenticode**.

Полезно противопоставлять друг другу подходы Java и ActiveX. В первом случае не производятся никакие попытки установить авторство апплета.

Вместо этого используется интерпретатор, который запрещает апплету совершать определенные нежелательные действия. Что касается метода подписания кода, то в этом случае, напротив, поведение программы во время ее выполнения никак не отслеживается. Если она была получена из проверенного источника и по дороге не была изменена, она просто запускается. Проверка самого кода не осуществляется. Если программист намеренно написал код, форматирующий жесткий диск и стирающий флэш-память компьютера, и при этом он считается проверенным и надежным программистом, то код выполнится и выведет из строя компьютер (если только в браузере не отключены управляющие элементы ActiveX).

Многие считают, что доверять неизвестным производителям программного обеспечения несколько легкомысленно. Чтобы доказать это, один программист из Сиэтла основал свою компанию и добился получения сертификата надежности, что не так уж сложно. После этого он написал управляющий элемент ActiveX, который всего-навсего выключал компьютер. Он распространил свою программу весьма широко, и она выключила не одну тысячу компьютеров. Впрочем, машины после этого можно было запросто включить заново, поэтому никакого ущерба такой управляющий элемент нанести не мог. Цель проекта состояла в том, чтобы указать миру на наличие проблемы. Официальная реакция выразилась в отзыве сертификата для данного конкретного управляющего элемента, и на этом инцидент был исчерпан. Но проблема-то решена не была, и нечистые на руку программисты могли продолжать использовать эту дыру в защите (Garfinkel и Spafford, 2002). Поскольку нет никакой возможности проследить за деятельностью всех компаний, пишущих переносимые программы, вскоре метод подписания кода может представлять собой довольно серьезную угрозу.

JavaScript

В JavaScript вообще отсутствует какая-либо официальная модель системы защиты информации, зато существует длинная история неудачных попыток ее внедрения. Каждый производитель пытается придумать что-нибудь свое. Например, в Netscape Navigator версии 2 было реализовано нечто подобное Java-модели, а уже в четвертой версии прослеживаются черты модели подписей кода.

Суть проблемы в том, что чужеродной программе разрешается выполнять какие-то действия. Это может привести к непредсказуемым последствиям. С точки зрения безопасности это то же самое, что позвать в гости вора и пытаться внимательно следить за тем, чтобы он не проник из кухни в гостиную. Если произойдет что-нибудь неожиданное, а вы в этот момент отвлечетесь, может случиться что угодно. Аргументом в защиту переносимых программ служит то, что с их помощью легко реализуются флэш-графика и быстрое взаимодействие с пользователем. Создатели веб-сайтов обычно считают, что это гораздо важнее, чем защита информации, особенно когда дело касается какого-нибудь чужого компьютера.

Вирусы

Вирусы — это своеобразная форма переносимого кода. Только в отличие от приведенных выше примеров, запускать такие программы никто не хочет. Основное отличие вирусов от обычных переносимых программ заключается в том, что они воспроизводят сами себя. Когда в систему проникает вирус (с веб-страницы, во вложении электронного письма или как-то еще), для начала он заражает исполняемые программы, хранящиеся на диске. При запуске какой-либо из этих программ управление передается вирусу, а тот обычно пытается распространить свое действие еще и на другие машины, например, рассылая самого себя по электронной почте всем адресатам из адресной книги жертвы. Некоторые вирусы заражают загрузочный сектор жесткого диска, поэтому вирус активируется при загрузке машин.

Вирусы в какой-то момент стали представлять собой крупномасштабную проблему для Интернета и принесли многомиллиардные убытки. Какого-либо простого решения проблемы не существует. Возможно, положение может спасти создание нового поколения операционных систем, базирующихся на защищенных микроядрах, и сплоченность пользователей, процессов и ресурсов.

Социальный аспект

Интернет и технологии защиты информации — это те области, в которых очень тесно сплелись социальные вопросы, государственная политика и технологии. Далее мы кратко рассмотрим три проблемы: конфиденциальность, свободу слова и авторские права. Совершенно очевидно, что в рамках этой книги мы сможем дать лишь поверхностное описание этой темы. Более подробную информацию следует искать в (Anderson, 2001; Garfinkel и Spafford, 2002; Schneier, 2000). Многие материалы можно прочитать в Интернете. Достаточно лишь набрать в поис-

ковой машине «конфиденциальность» (privacy — для получения информации на английском языке), «цензура» (censorship) или «авторские права» (copyright).

Конфиденциальность

Имеют ли люди право на секреты? Хороший вопрос. Четвертая поправка к конституции США запрещает правительственным организациям без особой нужды интересоваться намерениями граждан, их жильем и личными бумагами. Ограничен перечень обстоятельств, при которых этот запрет может быть нарушен. Таким образом, вопрос конфиденциальности стоит на повестке дня уже более 200 лет, по крайней мере, в США.

Что изменилось за последнее десятилетие? Правительство получило возможность с невиданной легкостью шпионить за гражданами, а граждане — с не меньшей легкостью предотвращать шпионаж. В XVIII веке для получения доступа к личным бумагам гражданина требовалось выслать к нему в имение полицейского, которому нужно было в любую непогоду доскакать на коне, претерпевая всевозможные лишения, которые нередки в долгом пути, — и все это для того, чтобы прочесть один чужой листок бумаги. В наши дни телефонные компании и поставщики услуг Интернета обеспечивают всех, кто может предъявить соответствующий ордер, подслушивающими устройствами. С их помощью задача полицейских сильно облегчается, к тому же нет риска выпасть из седла, заснув в пути.

Тем не менее, использование криптографии в значительной степени меняет дело. Любой желающий может озаботиться загрузкой и установкой PGP, генерированием хорошо защищенного, надежного ключа, и в результате он получит уверенность в том, что никто во Вселенной не сможет прочесть его электронную почту, независимо от наличия у него ордера на обыск. Правительства прекрасно это понимают, и им, разумеется, это сильно не нравится. В реальности конфиденциальность означает, что уполномоченным органам очень трудно следить за преступниками всех мастей, а также за журналистами и политическими оппонентами. Неудивительно, что многие правительства запрещают использование и экспорт криптографии. Во Франции, к примеру, до 1999 года любая негосударственная криптография была просто запрещена, если только государству не предоставлялись все используемые ключи.

Франция в этом деле не была одинока. В апреле 1993 года правительство США объявило о своем желании создать аппаратный криптопроцессор (clipper chip) и сделать его стандартным для применения в любых сетевых коммуникациях. Таким образом, как было заявлено, граждане получат гарантированную конфиденциальность. Вместе с тем, упоминалось о том, что правительство будет иметь возможность расшифровывать весь трафик таких криптопроцессоров при помощи специальной технологии, позволяющей правительству получать доступ ко всем ключам. Однако были даны обещания использовать эту возможность только при наличии соответствующей санкции. Понятно, что такое заявление вызвало большой фурор: сторонники конфиденциальности осуждали весь план от начала до конца, а чиновники, выступающие в поддержку этого начинания, были

восхищены предложением правительства. Тем не менее, правительство почему-то сдало позиции и отказалось от собственной идеи.

Огромное количество материалов, посвященных конфиденциальности цифровой информации, доступно на веб-сайте фонда Electronic Frontier (www.eff.org).

Анонимные рассылки

PGP, SSL и другие технологии позволяют устанавливать между двумя сторонами защищенные, аутентифицированные соединения, не подверженные вмешательству третьих сторон. Однако иногда конфиденциальность лучше всего обеспечивается как раз *отсутствием* аутентификации, то есть, по сути дела, установлением анонимных соединений. Анонимность востребована как при передаче сообщений между двумя пользователями, так и в сетевых телеконференциях.

Рассмотрим некоторые примеры. Во-первых, политические диссиденты, живущие при авторитарном режиме, могут захотеть во избежание репрессий общаться анонимно. Во-вторых, различные нарушения во многих коммерческих, образовательных, правительственных и других организациях зачастую выявляются не без помощи доносчиков, которые желают оставаться неизвестными. В-третьих, приверженцы нетрадиционных (а значит, как правило, порицаемых) социальных политических или религиозных убеждений видят одну из немногих возможностей общения в телеконференциях (или электронной почте), где они могут скрывать свои истинные имена. В-четвертых, многие предпочитают обсуждать алкоголизм, душевные заболевания, сексуальные проблемы, проблемы жестокого обращения с детьми или отношение к преследуемым меньшинствам в телеконференциях, где они могут оставаться анонимными. Кроме того, конечно, существует масса иных примеров.

Рассмотрим один конкретный пример. В 1990-х годах некоторые критики одной нетрадиционной религиозной секты опубликовали свои взгляды в конференции USENET с помощью **анонимной рассылки**. Сервер позволял пользователям создавать псевдонимы и посылать на него электронные письма, которые затем рассылались от имени выбранного псевдонима. В итоге не было возможности понять, кто является настоящим автором письма. Некоторые из этих статей были разоблачениями, в состав которых, по мнению представителей секты, входили коммерческие тайны и документы, защищенные авторским правом. В ответ на эти разоблачения секта подала в суд, жалуясь на раскрытие коммерческих тайн и нарушение закона об авторском праве. И то, и другое в том округе, где находился сервер, считалось преступлением. Последовал суд, и владельцев сервера вынудили раскрыть истинные имена тех, кто скрывался под псевдонимами и писал разоблачения (кстати, это был не первый прецедент, связанный с недовольством церкви раскрытием ее тайн: Уильям Тиндэйл (William Tyndale) был в 1536 году сожжен на столбе за перевод Библии на английский).

Значительная часть Интернет-сообщества была сильно возмущена таким грубым нарушением принципов конфиденциальности. Все были согласны и с тем, что владелец анонимной рассылки, хранившей таблицу соответствия настоящих электронных адресов и псевдонимов (это было названо анонимной рассылкой первого типа), был не прав. Этот случай стимулировал развитие анонимных рассылок, которые могли бы противостоять таким атакам со стороны суда.

Рассылки нового типа, часто называемые **шифрованными панковскими рассылками** (cypherpunk remailer), работают следующим образом. Пользователь создает электронное письмо с обычными заголовками RFC 822 (разумеется, отсутствует *From*), шифрует его открытым ключом рассылки и отправляет на сервер. Там от него отрезаются заголовки RFC 822, содержимое расшифровывается, и сообщение рассылается подписчикам. В рассылке нет никаких учетных записей, не ведутся никакие журналы, поэтому даже в случае конфискации сервера никаких следов прошедших через него писем обнаружено не будет.

Многие пользователи, которые особенно сильно озабочены проблемой собственной анонимности, прогоняют свои сообщения через цепочки анонимных рассылок, как показано на рис. 8.49. В данном примере Алиса хочет послать Бобу действительно очень-очень анонимное поздравление с днем св. Валентина. Для этого она использует три анонимные рассылки. Она сочиняет письмо M и вставляет заголовок, содержащий адрес электронной почты Боба. Затем все это сообщение шифруется открытым ключом рассылки 3, E_3 (показано горизонтальной штриховкой). К этому прибавляется заголовок с электронным адресом рассылки 3 (передается открытым текстом). В итоге получается сообщение, показанное между рассылками 2 и 3 на рисунке.

На этом история сообщения не заканчивается. Оно шифруется открытым ключом рассылки 2, E_2 (показано вертикальной штриховкой), и дополняется открытым заголовком, содержащим электронный адрес рассылки 2. Получившееся в итоге сообщение показано на рисунке между рассылками 1 и 2. Затем Алиса шифрует свое сообщение открытым ключом рассылки 1, E_1, добавляет адрес этой рассылки и, наконец, отправляет его. Конечное состояние сообщения показано на рисунке справа от Алисы.

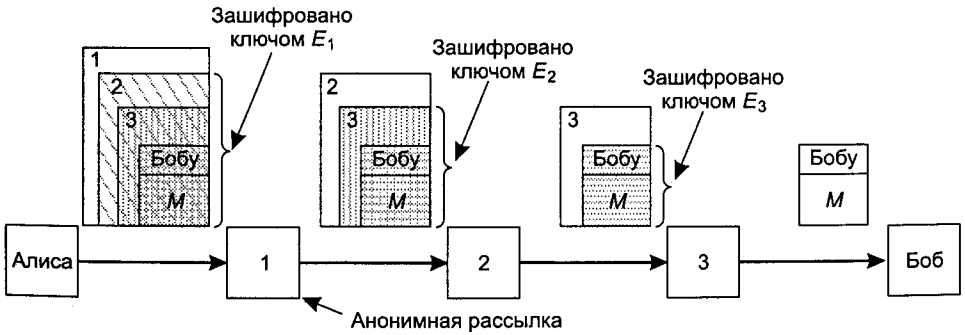


Рис. 8.49. Использование трех анонимных рассылок для передачи письма Алисы Бобу

Когда письмо Алисы достигает рассылки 1, от него отрезается внешний заголовок. Тело сообщения расшифровывается и пересылается в рассылку 2. Аналогичные шаги производятся и на двух других серверах.

Несмотря на то, что восстановить путь конечного сообщения до Алисы и так очень сложно, многие рассылки принимают еще и дополнительные меры предосторожности. Например, они могут задерживать сообщения на какой-то случайный промежуток времени, добавлять или удалять всякий мусор в конце сообщения, переставлять сообщения местами, в общем, делать все возможное для того, чтобы запутать тех, кто отслеживает трафик и пытается понять, кто является автором того или иного сообщения, прошедшего через анонимную рассылку. Описание системы, реализующей в настоящее время описанные ранее идеи, можно найти в (Mazi res и Kaashoek, 1998).

Сфера применения принципов анонимности не ограничивается одной электронной почтой. Существуют также услуги, позволяющие анонимно просматривать интернет-материалы. Пользователь может настроить свой браузер на использование такой услуги в качестве прокси. После этого все HTTP-запросы направляются по адресу, принадлежащему этому сервису, который производит запрос страницы за пользователя. Если на сервере, обеспечивающем анонимность, не хранятся журналы активности, никто не сможет определить, кто на самом деле запрашивал страницу.

Свобода слова

Конфиденциальность связана с проблемой сокрытия от посторонних глаз информации, не подлежащей обнародованию. Вторым ключевым социальным аспектом является, несомненно, свобода слова и ее противоположность — цензура. В этом случае правящие органы пытаются ограничить спектр информации, которую граждане могут читать и публиковать. Всемирная паутина с ее миллионами страниц — это настоящий рай для цензуры. В зависимости от типа и идеологии режима, в список запрещенных к просмотру материалов могут помещаться страницы, содержащие что-либо из перечисленного ниже:

1. Материалы, которые нельзя показывать детям и подросткам.
2. Материалы, пропагандирующие ненависть к каким-либо этническим, религиозным, сексуальным или другим группам.
3. Информацию о демократии и демократических ценностях.
4. Описания исторических событий, не совпадающие с официальной версией.
5. Руководства по взлому разного рода замков, созданию оружия, шифрованию сообщений и т. д.

Плохой, негодный сайт проще всего запретить к просмотру.

Иногда результаты такой политики оказываются неожиданными. Например, некоторые публичные библиотеки установили у себя веб-фильтры, не пропускающие порнографические сайты и таким образом делающие содержимое Паутины безопасным для просмотра детьми. Фильтры перед выводом страниц сверяют адреса со своими «черными списками», к тому же проверяют их на наличие бранных слов. Однажды в округе Лаудаун, штат Вирджиния, фильтр заблокировал поиск информации по раку молочной железы, так как запрос содержал слово «breast» (женская грудь, молочная железа). Клиент возбудил дело против правительства округа. В то же время, был и другой случай: в Ливерморе, штат Калифорния, один родитель подал в суд на публичную библиотеку за то, что там

не был установлен фильтр и он застал своего 12-летнего сына за просмотром порнографического сайта. Так что же библиотеке делать?

Многие никак не могут понять, что Всемирная паутина — действительно всемирная. Она охватывает весь земной шар. В разных странах существуют разные взгляды на то, что должно, а чего не должно быть в Сети. Например, в ноябре 2000 года французский суд постановил, что Yahoo, корпорация, находящаяся в Калифорнии, должна запретить доступ к аукциону памятных вещей нацистов для французских пользователей, так как обладание такой информацией идет вразрез с французским законодательством. Yahoo апеллировала к суду США, который решил спор в пользу корпорации, однако в целом проблема того, законы какой страны должны применяться в Интернете, остается актуальной.

Попробуйте представить себе, что случится, если какой-нибудь суд штата Юта вынесет решение о том, что Франция должна заблокировать сайты, посвященные винам, так как распространение подобной информации нарушает строгие законы штата, касающиеся алкогольной продукции? Точно так же Китай может сделать табуированными все сайты, на которых рассказывается про демократию, так как это не в интересах Поднебесной. Должны ли иранские законы о религии применяться в либеральной Швеции? Может ли Саудовская Аравия заблокировать сайты, защищающие права женщин? Становится понятно, что эта проблема подобна ящику Пандоры и способна породить множество вопросов.

Ценное замечание высказывает Джон Гилмор (John Gilmore): «Сеть воспринимает цензуру как разрушенный участок дороги и идет в обход». Конкретная реализация этой мысли называется **службой вечности** (Anderson, 1996). Ее цель — гарантировать, что однажды опубликованные материалы не исчезнут и не будут переписаны заново, как было принято в Советском Союзе во времена Сталина. Пользователь службы вечности должен лишь указать, в течение какого срока следует обеспечивать сохранность информации, заплатить пропорциональную сроку и объемам информации сумму и загрузить данные на сервер. После этого никто, включая самого пользователя, не сможет удалить или отредактировать размещенные на сервере службы вечности материалы.

Как такую услугу реализовать на практике? Проще всего организовать одноранговую систему, в которой документы будут размещаться на десятках серверов-участников проекта, каждый из которых будет получать свою долю вознаграждения, что послужит стимулом для их вступления в проект. Серверы должны располагаться в самых разных местах и под разной юрисдикцией, что обеспечит максимальную устойчивость системы. Списки 10 выбранных случайным образом серверов следует хранить в тайне в разных местах, чтобы в случае неудачи, произошедшей с одним из них, могли выжить другие. Любые государственные органы, помешанные на уничтожении неугодной информации, никогда не смогут быть до конца уверенными в том, что они нашли все копии. Кроме того, систему можно сделать самовосстанавливающейся, в том смысле, что в случае прихода известия об уничтожении каких-то экземпляров документов держатели остальных копий попытаются найти новые места хранения для замены выбывших из строя.

Служба вечности была первой попыткой противостояния цензуре в Сети. С тех пор было высказано много различных идей на эту тему, и некоторые из них даже нашли свое воплощение. Были добавлены новые возможности, такие как шифрование, анонимность, отказоустойчивость. Зачастую документы разбивают на несколько фрагментов и хранят их на нескольких серверах. Среди таких систем можно упомянуть Freenet (Clarke и др., 2002), PASIS (Wylie и др., 2000) и Publius (Waldman и др., 2000). Еще одна разработка описана в (Сержантов, 2002).

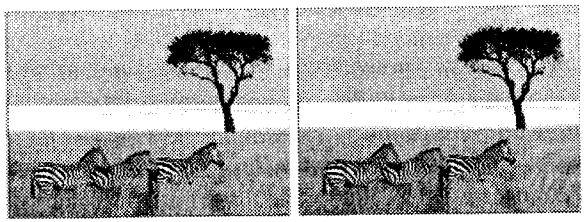
Все больше и больше стран пытаются контролировать экспорт таких неосязаемых вещей, как веб-сайты, программное обеспечение, научные документы, электронная почта, телефонные службы помощи и т. п. Даже в Великобритании, славящейся своими вековыми традициями поддержки свободы слова, появляются строгие законы, которые, к примеру, определяют техническую дискуссию между британским профессором и иностранным студентом в Кембриджском университете как предмет экспорта, подлежащий государственному лицензированию (Anderson, 2002). Очевидно, что такая политика крайне противоречива.

Стеганография

В странах, где цензура применяется особенно широко, всегда существуют диссиденты, использующие свои методы обхода этой цензуры. Криптография, конечно, позволяет (не всегда вполне законно) посылать секретные сообщения так, чтобы никто не смог узнать их смысл, однако если государство считает Алису своим врагом, один тот факт, что она общается с Бобом, может и его поставить в положение врага государства. Таким образом политики, обычно не слишком хорошо владеющие математикой, понимают и применяют принцип транзитивности. Выручить могут анонимные рассылки, но и их местное правительство может запретить, и тогда для отправки сообщения за границу понадобится экспортная лицензия. Таким образом, анонимные рассылки — это тоже не панацея. Однако Всемирная паутина всегда найдет выход из положения.

Люди, которым требуется секретное общение, зачастую пытаются скрыть сам факт общения. Наука, занимающаяся сокрытием сообщений, называется **стеганографией** (не путать со стенографией!), от греческого слова, которое можно перевести как «защищенное письмо». Сами древние греки первыми и начали использовать этот метод. Геродот описывал своеобразный метод секретного общения военачальников: посыльному брили волосы на голове, на затылке рисовали татуировку с секретным сообщением и ждали, пока волосы снова отрастут, после чего отправляли посыльного в путь. Современные технологии базируются на той же концепции, разве что пропускная способность стала выше, а задержки — ниже.

В качестве примера рассмотрим рис. 8.50, *а*. На этой фотографии, сделанной автором в Кении, изображены три зебры и акация. Однако она привлекательна не только с эстетической точки зрения. Дело в том, что рис. 8.50, *б* включает в себя внедренный полный текст пяти самых известных пьес Шекспира: «*Гамлет*», «*Король Лир*», «*Макбет*», «*Венецианский купец*» и «*Юлий Цезарь*» — все тексты вместе занимают более 700 Кбайт.



а *б*

Рис. 8.50. Три зебры и дерево (*а*); три зебры, дерево и полный текст пяти пьес Вильяма Шекспира (*б*)

Как же это сделано? Стеганографический канал работает следующим образом. Размер исходного изображения равен 1024×768 пикселов. Каждый пиксел представлен тремя 8-битными числами, каждое из которых отвечает за свою составляющую цвета (существуют красный, зеленый и синий каналы интенсивности цвета). Результирующий цвет получается линейной суперпозицией интенсивностей трех цветов. При методе стеганографического шифрования младший бит каждого из трех указанных чисел заменяется нужным значением секретного канала. Таким образом, в каждом пикселе помещаются три бита секретной информации: один в канале интенсивности красного цвета, один — зеленого и один — синего. В нашем изображении поместится $1024 \cdot 768 \cdot 3$ бит, или 294 912 байт секретной информации.

Полный текст пяти пьес Шекспира с небольшим предисловием занимает 734 891 байт. Текст можно сжать до 274 Кбайт с помощью любого архиватора. Затем зашифровать эти сжатые данные с использованием алгоритма IDEA, после чего разместить данные в младших битах значений интенсивностей цветов. Как видите (вернее, это как раз остается невидимым), присутствие текстовой информации совершенно незаметно. На большой, полноцветной фотографии это также незаметно. Глаз человека просто не в состоянии отличить оттенки, выраженные 21-разрядным числом, от 24-разрядных оттенков.

Черно-белые фотографии, приведенные на рис. 8.50, возможно, дают неполное представление о возможностях стеганографии. Более эффектный вариант рис. 8.50, *б* продемонстрирован в полноцветной версии, которую можно найти на веб-сайте книги.

Для нужд незаметного обмена информацией диссиденты могут создать вебсайт, заполненный политкорректными фотографиями (например, Великого Вождя, местных спортивных команд, теле- и кинозвезд и т. п.). Разумеется, эти фотографии могут содержать стеганографические сообщения. Учитывая то, что эти сообщения перед помещением в графический файл сжимаются и шифруются, даже если кто-то и заподозрит их присутствие «внутри» фотографий, отличить их от белого шума, а тем более расшифровать их смысл будет очень и очень тяжело. Конечно, фотографии для стеганографической обработки должны быть отсканированы или сняты на цифровую камеру самостоятельно: если просто взять картинку из Интернета и записать в нее секретное сообщение, выявить «второе дно» простым побитовым сравнением будет гораздо проще.

Понятно, что стеганографические сообщения можно записывать не только в графические файлы. Очень хорошо для этих целей подходят звуковые файлы. А уж в видеофайлы можно записать действительно внушительные объемы данных. Надо сказать, что даже форматирование текста и расположение тегов в HTML-файле может нести определенную скрытую информацию.

Мы рассмотрели стеганографию в контексте проблемы свободы слова, однако у этой технологии есть масса иных применений. Очень часто авторы электронных изображений вшивают в файлы секретные сообщения, которые в случае необходимости смогут подтвердить их авторские права. Если кто-нибудь захочет украсть удачную фотографию и разместит ее на своем сайте без указания авторства, законный владелец с помощью стеганографической подписи сможет доказать в суде свою правоту. Такие подписи иногда называют **водяными знаками**. Эта тема обсуждается в (Piva и др., 2002).

Подробное рассмотрение проблем стеганографии можно найти в (Artz, 2001; Johnson и Jajoda, 1998; Katzenbeisser и Petitcolas, 2000; Wayner, 2002).

Защита авторских прав

Конфиденциальность и цензура — это те области, в которых сталкиваются технологические аспекты и общественные интересы. Третьей такой областью является защита авторских прав. **Авторское право** гарантирует свободу распоряжения **интеллектуальной собственностью** ее *создателям*, которыми могут быть, например, писатели, художники, композиторы, музыканты, фотографы, кинорежиссеры, хореографы и т. д. Авторское право выдается на определенный срок, который обычно равен сроку жизни автора плюс 50 лет (75 лет — в случае корпоративного авторского права). По окончании этого срока интеллектуальная собственность становится достоянием общества, и каждый волен распоряжаться ею, как хочет. Так, например, проект «Gutenberg» (www.promo.net/pg) разместил в Сети тысячи произведений, давно уже ставших общественным достоянием (работы Шекспира, Диккенса, Марка Твена). По просьбе Голливуда в 1998 году Конгресс США разрешил продлить срок действия авторских прав еще на 20 лет, утверждая, что без принятия этой меры больше никто ничего не станет создавать. В то же время, патенты на изобретения действуют в течение всего лишь 20 лет, и никто не жалуется — люди продолжают совершать открытия и делать изобретения.

Вопрос о защите авторских прав вышел на первый план, когда у службы Napster, незаконным образом распространявшей музыку, внезапно оказалось 50 миллионов пользователей. Несмотря на то, что записи системой Napster никуда не копировались, судом было предъявлено обвинение в хранении центральной базы данных с информацией о том, какие у кого из пользователей имеются записи. Таким образом, система побуждала пользователей совершать преступные действия, нарушающие закон об авторских правах. В общем-то, никто не жалуется на то, что идея авторских прав плоха (хотя многим не нравится то, что компании обладают намного большими привилегиями в этом плане, чем простые граждане), однако следующее поколение технологий распространения музыкальных записей уже поднимает вопросы этического характера.

Например, рассмотрим равноранговую сеть, члены которой занимаются вполне законным обменом файлами (записями, являющимися общественным достоянием, домашними видеозаписями, религиозными трактатами (не составляющими коммерческую тайну церквей) и т. д.). Возможно, какая-то небольшая часть этих файлов защищена законом об авторских правах. Допустим, все участники проекта находятся на постоянном соединении (пользуются ADSL или кабельным Интернетом). На каждой машине хранится список того, что есть на жестком диске, а также список всех машин сети. В поисках какого-либо файла можно обратиться к произвольно выбранной машине и запросить у нее список имеющихся материалов. Если нужная информация не находится, можно попробовать опросить все остальные машины из списка, а также машины из списков, хранящихся у других. Компьютеры здесь сильно облегчают задачу поиска некоторых редких материалов. Найдя нужный файл, пользователь просто копирует его себе.

Если найденная работа оказывается защищенной законом об авторских правах, пользователь, сам того не желая, становится нарушителем этого закона (тут еще, конечно же, играет роль то, в какой стране происходит дело и, соответственно, какие законы следует применять в каждом конкретном случае). А виноват ли поставщик информации? Является ли преступлением хранение у себя на жестком диске записей, за которые были заплачены деньги и которые были вполне легально загружены из Интернета, при условии, что к диску могут иметь доступ посторонние лица? Если в вашу лачугу, никогда не знавшую замков и засовов, проникает вор с ноутбуком и сканером, снимает копию с книги, защищенной законом об авторских правах, и уходит восвояси, разве *вы* виноваты в этом преступлении, разве *вы* должны защищать чужие авторские права?

Однако есть еще одна проблема, связанная с авторскими правами. Ведется ожесточенная борьба между Голливудом и компьютерной индустрией. Голливуд требует усилить защиту интеллектуальной собственности, а компьютерщики говорят, что они не обязаны сторожить голливудские ценности. В октябре 1998 года Конгресс принял **Акт об авторских правах в цифровых технологиях** (DCMA — Digital Millenium Copyright Act), в котором говорится о том, что взлом механизма защиты, присутствующего в работе, защищенной законом об авторских правах, а также сообщение другим технологии взлома является преступлением. Аналогичный законодательный акт был принят и в Евросоюзе. С одной стороны, все как-то забыли подумать о том, что для пиратов с Дальнего Востока такие акты указом не являются, а с другой стороны, многие считают, что новый закон нарушил баланс между интересами владельцев авторских прав и общественными интересами.

Взять хотя бы такой пример. В сентябре 2000 года консорциум, связанный с музыкальной индустрией, озабоченный созданием надежной онлайновой системы продажи аудиозаписей, организовал конкурс, пригласив всех желающих попробовать взломать систему (это действительно очень важный этап, необходимый при создании любой новой системы защиты). Группа ученых из различных университетов под руководством профессора Эдварда Фельтена (Edward Felten) из Принстона, специализирующихся в области защиты информации, приняли вызов и сломали систему. Затем была написана статья, описывающая выводы, сделанные в ходе исследования. Она была направлена на конференцию USENIX, посвященную проблемам защиты информации. Доклад был рассмотрен и принят на соответствующем уровне. Однако незадолго до конференции Фельтен получил уведомление от Ассоциации звукозаписи о том, что эта организация в случае опубликования статьи подаст на авторов в суд за нарушение акта DCMA.

Ученым ничего не оставалось делать, как послать запрос в федеральный судебный орган, пытаясь выяснить, является ли еще легальным опубликование научных статей, касающихся защиты информации. Опасаясь, что дальнейшее развитие событий будет отнюдь не в пользу звукозаписывающей индустрии, ее представители сняли свои претензии к Фельтену, и на этом инцидент был исчерпан. Несомненно, причиной недовольства звукозаписывающей индустрии была слабость предложенной ими системы. Получилось, что сначала пригласили людей, чтобы те попытались взломать защиту, а когда некоторым это реально удалось, на них чуть было не подали за это в суд. После того, как все конфликты были улажены, статья все-таки была напечатана (Craver и др., 2001). Почти очевидно, что таких конфронтаций впереди еще много.

С обсуждаемой темой тесно связана **доктрина законного использования**, ставшая результатом судебных решений во многих странах. Эта доктрина состоит в том, что покупатели продукции, защищенной законом об авторских правах, имеют сильно ограниченные права на копирование этой продукции, включая даже использование ее частей для научных целей (например, в качестве обучающего материала в школах и колледжах) и создание архивных резервных копий на тот случай, если что-нибудь случится с исходным носителем. Как проверить, является ли использование продукции законным? Показатели таковы: 1) коммерческое использование; 2) количество процентов скопированных данных; 3) влияние копирования на объем продаж. Так как DMCA и аналогичные законы, принятые Евросоюзом, запрещают взлом систем защиты от копирования, такие законы заодно запрещают легальное добросовестное использование. По сути, DMCA отбирает у потребителей историческое право активно поддерживать продавцов, у которых они приобрели продукцию. Провал этой идеи неизбежен.

Еще одно явление, затмевающее собой по уровню смещения баланса между обладателями авторских прав и потребителями даже DMCA, — это Альянс надежных вычислительных платформ (TCPA — Trusted Computing Platform Alliance). Разрабатывается этот проект совместными усилиями Intel и Microsoft. Идея состоит в том, чтобы процессор и операционная система зорко наблюдали за действиями пользователя (например, за воспроизведением скопированной незаконным образом звукозаписи) и запрещали совершать нежелательные действия. Такая система могла бы даже позволить обладателям авторских прав удаленно манипулировать персональными компьютерами пользователей, изменяя при необходимости определенные правила. Несомненно, общественный резонанс будет огромен. Конечно, это очень здорово, что индустрия наконец обратила внимание на проблемы защиты информации, однако нельзя не заметить с прискорбием, что большинство усилий направлено на усиление законов об авторских правах, а не на борьбу с вирусами, взломщиками, мошенниками и другими проблемами, волнующими большинство пользователей.

Короче говоря, авторам законов и юристам теперь предстоит в течение долгих лет пытаться урегулировать взаимоотношения владельцев авторских прав и потребителей. Киберпространство ничем не отличается от социума: и там, и там постоянно сталкиваются интересы различных групп, что приводит к ожесточенной борьбе и судебным разбирательствам, результатом которых рано или поздно становится нахождение какого-то компромисса. По крайней мере, стоит на это надеяться, как стоит надеяться на относительное затишье до появления новой противоречивой технологии.

Резюме

Криптография представляет собой инструмент, используемый для обеспечения конфиденциальности информации, проверки ее целостности и аутентичности. Все современные криптографические системы основаны на принципе Керкгофа, гласящем, что алгоритмы должны быть доступны всем желающим, а ключи — храниться в тайне. Многие алгоритмы при шифровании текста выполняют сложные преобразования, включающие в себя замены и перестановки. Тем не менее, если удастся реализовать на практике принципы квантовой криптографии, то с помощью одноразовых блокнотов можно будет создать действительно надежные криптосистемы.

Все криптографические алгоритмы можно разделить на два типа: с симметричными ключами и с открытыми ключами. Алгоритмы с симметричными ключами при шифрации искажают значения битов в последовательности итераций, параметризованных ключом. Наиболее популярные алгоритмы этого типа — тройной DES и Rijndael (AES). Алгоритмы с симметричным ключом могут работать в режиме электронного шифроблокнота, сцепления блоков шифра, потокового шифра и др.

Алгоритмы с открытыми ключами отличаются тем, что для шифрования и дешифрации используются разные ключи, причем ключ дешифрации невозможно вычислить по ключу шифрования. Эти свойства позволяют делать ключ открытым. Чаще всего применяется алгоритм RSA, основанный на сложности разложения больших чисел на простые сомножители.

Официальные, коммерческие и другие документы необходимо подписывать. Существуют различные методы генерации цифровых подписей, основанные на алгоритмах как с симметричными, так и с открытыми ключами. Обычно при помощи алгоритмов типа MD5 или SHA-1 вычисляется хэш-функция сообщения, которое необходимо подписать, и подпись ставится не на само сообщение, а на значение хэш-функции.

Управление открытыми ключами можно реализовать при помощи сертификатов, представляющих собой документы, связывающие между собой открытые ключи и обладающих ими принципалов. Сертификаты подписываются надежной организацией или кем-либо, способным предъявить цепочку сертификатов, ведущих в итоге к этой надежной организации. Начальное звено этой цепочки (центральное управление) должно быть известно заранее, для этого в браузеры вшиваются номера многих сертификатов центральных управлений.

Эти криптографические методы позволяют защитить сетевой трафик. На сетевом уровне для этого работает система IPsec, занимающаяся шифрацией пакетов, передающихся между хостами. Брандмауэры могут фильтровать как входящий, так и исходящий трафик, анализируя используемые протоколы и порты. Виртуальные частные сети могут стимулировать повышение степени защиты информации в старых сетях, построенных на основе выделенных линий. Наконец, в беспроводных сетях требуется довольно серьезная защита информации, и этого не может обеспечить WEP 802.11. Остается надеяться на улучшение ситуации с появлением 802.11i.

При установлении сеанса связи между двумя сторонами они должны идентифицировать себя и, возможно, определить общий ключ сеанса. Для этого существуют различные протоколы аутентификации, среди которых есть те, которые подразумевают наличие третьей, надежной стороны, а также протоколы Диффи—Хеллмана, «Керберос» и протоколы с использованием открытых ключей.

Защита информации в электронной почте может быть достигнута применением различных комбинаций тех методов, которые мы изучили в этой главе. PGP, например, сжимает сообщение, а потом шифрует его с помощью IDEA. Ключ IDEA шифруется при помощи открытого ключа получателя. Кроме того, вычисляется хэш-функция сообщения. Проверка целостности может быть выполнена за счет того, что на хэш перед отправкой ставится подпись.

Безопасность во Всемирной паутине — это тоже важная проблема, которая начинается с безопасного именования ресурсов. DNSsec позволяет предотвращать обманы DNS-серверов, а также создавать самозаверяющиеся имена. Большинство сайтов, посвященных электронной коммерции, для установления надежных аутентифицированных сеансов между клиентом и сервером используют SSL. Для борьбы с возможными проблемами, связанными с переносимыми программами, разработаны разные методы, среди которых помещение загружаемого кода в изолированную среду («песочницу») и подписание кодов.

В Интернете возникает множество вопросов, в которых технологические проблемы переплетаются с государственной политикой. Мы рассмотрели лишь некоторые из них: конфиденциальность, свободу слова и авторские права.

Вопросы

1. Расшифруйте следующее сообщение, составленное с помощью моноалфавитного шифра. Открытый текст, состоящий только из букв, представляет собой хорошо известный отрывок из поэмы Льюиса Кэррола.

```
kfd ktbd fzm eubd kfd pzyiom mztx ku kzyg ur bzha kfthcm
ur mfudm zhx mftnm zhx mdzythc pzq ur ezsszcdm zhx gthcm
zhx pfa kfd mdz tm sutythc fuk zhx pfdkfdi ntcm fzld pthcm
sok pztk z stk kfd uamkdim eitdx sdruid pd fzld uoi efzk
rui mubd ur om zid uok ur sidzkf zhx zyy ur om zid rzk
hu foiia mztx kfd ezindhkdi kfda kfzhgdx ftb boef rui kfzk
```

2. Взломайте следующий колоночный перестановочный шифр. Открытый текст взят из популярной книги о компьютерах, поэтому слово «computer» может

встретиться с большой вероятностью. Открытый текст состоит из одних букв (без пробелов). Зашифрованный текст разбит на блоки по пять символов для удобства чтения.

```
aauan cvlre rurnn dltme aeepb ytust iceat npmey iicgo gorch crsoc nntii imiha oofpa
gsivt tpsit lbolr otoex
```

3. Подберите 77-разрядный одноразовый блокнот, с помощью которого из шифра, показанного на рис. 8.3, можно получить текст «Donald Duck».
4. При использовании квантовой криптографии требуется наличие фотонной пушки, способной при необходимости испускать одиночный фотон, соответствующий одному биту. Подсчитайте, сколько фотонов переносит 1 бит по 100-гигабитной оптоволоконной линии связи. Длина фотона предполагается равной длине волны, то есть 1 мкм (для нужд данной задачи). Скорость света в оптоволокне равна 20 см/нс.
5. Если злоумышленнику удастся перехватить и регенерировать фотоны при использовании квантовой криптографии, некоторые значения будут приняты неверно, а значит, появятся ошибки и в одноразовом блокноте, принимаемом Бобом. Какая (в среднем) часть блокнота, принятого Бобом, будет содержать ошибки?
6. Фундаментальный принцип криптографии гласит, что все сообщения должны быть избыточными. Но мы знаем, что избыточность позволяет злоумышленнику понять, правильно ли он угадал секретный ключ. Рассмотрите два типа избыточности. В первом типе изначальные n бит открытого текста содержат известную последовательность. Во втором результирующие n бит сообщения содержат хэш. Эквивалентны ли эти типы избыточности с точки зрения безопасности? Ответ поясните.
7. На рис. 8.5 S-блоки и P-блоки сменяют друг друга. Сделано ли это из эстетических соображений, или такая организация шифратора дает более надежный код, чем в случае, когда сначала идут все P-блоки, а затем все S-блоки?
8. Разработайте атаку шифра DES, основываясь на информации, что открытый текст состоит исключительно из прописных ASCII-символов, пробелов, запятых, двоеточий, символов «точка с запятой», «перевод строки» и «возврат каретки». О битах четности, содержащихся в открытом тексте, ничего не известно.
9. В тексте было подсчитано, что машине для взлома шифра, снабженной миллиардом процессоров, способных обрабатывать 1 ключ за 1 пс, понадобится всего лишь 10^{10} лет для взлома 128-битной версии AES. Тем не менее, на сегодняшний день машины могут иметь не более 1024 процессоров и обрабатывать ключ за 1 мс, так что даже для достижения приведенных ранее результатов необходимо увеличить производительность в 10^{15} раз. Если закон Мура, утверждающий, что вычислительные мощности компьютеров удваиваются каждые 18 месяцев, будет продолжать соблюдаться всегда, сколько лет придется ждать указанного прироста производительности?
10. Ключи AES могут иметь длину 256 бит. Сколько вариантов ключей доступно в таком режиме? Сможете ли вы найти в какой-нибудь науке — например, физике, химии или астрономии — понятия, оперирующие подобными величинами? Найдите в Интернете материалы, посвященные большим числам. Сделайте выводы из этого исследования.

11. Предположим, что сообщение было зашифровано с помощью шифра DES в режиме сцепления блоков. В одном из блоков зашифрованного текста при передаче один бит изменился с 0 на 1. Какая часть текста будет при этом повреждена?
12. Снова рассмотрим шифрование со сцеплением блоков. На этот раз между блоками зашифрованного текста при передаче вставился один нулевой бит. Какая часть текста будет повреждена в этом случае?
13. Сравните режим шифрования со сцеплением блоков с режимом шифрованной обратной связи с точки зрения количества операций шифрования, необходимых для передачи большого файла. Какой режим более эффективен и насколько?
14. Используя алгоритм открытого ключа RSA, при $a = 1$, $b = 2$ и т. д.;

    1) перечислите пять допустимых значений d для $p = 7$ и $q = 11$;

    2) найдите e, если $p = 13$, $q = 31$ и $d = 7$;

    3) для $p = 5$, $q = 11$ и $d = 27$ найдите e и зашифруйте «abcdefghij».
15. Предположим, что Мария внезапно обнаруживает, что ее закрытый ключ RSA (d 1, n 1) совпадает с открытым ключом RSA (e 2, n 2) другого пользователя, Франчески. Другими словами, d 1 = e 2 и n 1 = n 2. Следует ли Марии задуматься о смене открытого и закрытого ключей? Ответ поясните.
16. Рассмотрите режим счетчика, показанный на рис. 8.13, но со значением вектора инициализации, равным 0. Угрожает ли использование нуля надежности шифра в целом?
17. Протокол генерации подписей, показанный на рис. 8.15, обладает одним недостатком. Если с компьютером Боба что-нибудь случится, содержимое оперативной памяти будет утеряно. Какие проблемы это может вызвать, и как с ними бороться?
18. На рис. 8.17 мы видим, как Алиса передает Бобу подписанное сообщение. Если Труди заменит P, Боб заметит это. А что будет, если Труди заменит и P, и подпись?
19. Цифровые подписи имеют один недостаток: их могут использовать лентяи. После составления договора в электронной коммерции обычно требуется, чтобы клиент подписал его своим хэшем SHA-1. Если пользователь не озаботится проверкой соответствия хэша и контракта, он имеет шанс случайно подписать другой договор. Допустим, вездесущая и бессмертная мафия решила сделать на этом деньги. Бандиты создают платный веб-сайт (содержащий, например, порнографию, азартные игры и т. п.) и спрашивают у новых клиентов номера кредитных карт. Затем пользователю отсылается договор, в котором говорится, что он желает воспользоваться услугами и оплатить их с помощью кредитной карты. Договор следует подписать, однако владельцы

сайта знают, что вряд ли кто-то станет сверять хэш с контрактом. Покажите, каким образом злоумышленники смогут купить бриллианты в легальном ювелирном интернет-магазине за счет ничего не подозревающих клиентов.

20. В математическом классе 20 учащихся. Какова вероятность того, что, по крайней мере, у двух из них совпадают дни рождения? Допустим, что ни у кого из них день рождения не приходится на 29 февраля, поэтому общее число рассматриваемых вариантов дней рождения равно 365.
21. После того как Элен созналась Мэрилин в своем обмане в деле с предоставлением должности Тому, Мэрилин решила устранить проблему, записывая содержимое своих сообщений на диктофон, с тем, чтобы ее новая секретарша просто набирала соответствующий текст. Затем Мэрилин собирается изучать набранные сообщения на своем терминале, чтобы проверить, что они содержат точные ее слова. Может ли новая секретарша по-прежнему воспользоваться атакой, основанной на задаче о днях рождения, чтобы фальсифицировать сообщение, и если да, то как? *Подсказка*: может.
22. Рассмотрите пример неудачной попытки получения Алисой открытого ключа Боба (см. рис. 8.20). Допустим, они уже установили общий закрытый ключ, однако Алиса все же хочет узнать открытый ключ Боба. Существует ли в данной ситуации безопасный способ его получения? Если да, то какой?
23. Алиса желает общаться с Бобом, используя шифрование с открытым ключом. Она устанавливает соединение с кем-то, кто, по ее предположению, является Бобом. Она спрашивает у него открытый ключ, и тот отсылает его вместе с сертификатом X.509, подписанным Центральным управлением. У Алисы уже есть открытый ключ ЦУ. Что должна теперь сделать Алиса, чтобы удостовериться, что она действительно общается с Бобом? Допустим, Бобу безразлично, с кем он общается (то есть под именем Боба мы здесь понимаем, например, какую-нибудь общедоступную службу).
24. Допустим, система использует PKI, базирующийся на древовидной иерархии Управлений сертификации. Алиса желает пообщаться с Бобом и после установления соединения получает от него сертификат, подписанный УС *X*. Допустим, Алиса никогда не слышала про *X*. Что ей нужно сделать, чтобы удостовериться, что на той стороне канала связи находится именно Боб?
25. Может ли IPsec с заголовком AH использоваться в транспортном режиме, если одна из машин находится за NAT-блоком? Ответ поясните.
26. В чем заключается одно из преимуществ использования HMAC над использованием RSA при создании подписей хэшей SHA-1?
27. Назовите одну причину, по которой брандмауэры следует настраивать на анализ входящего трафика. Теперь назовите одну причину, по которой брандмауэры следует настраивать на анализ исходящего трафика. Как вы думаете, насколько успешен такой анализ?
28. Формат WEP-пакета показан на рис. 8.27. Допустим, используется 32-битная контрольная сумма, вычисляемая путем суммирования по модулю 2 всех 32-битных слов заголовка. Предположим, что проблема RC4 решилась его заменой на потоковый шифр без изъянов, а вектор инициализации расширен до 128 бит. Может ли злоумышленник каким-то образом незаметно заниматься шпионажем или изменять сообщения?

29. Допустим, организация установила виртуальную частную сеть для обеспечения безопасной связи между своими сайтами в Интернете. Имеет ли смысл Джиму, сотруднику этой организации, для общения с Мэри, сотрудницей той же организации, применять шифрование или какие-то еще механизмы защиты информации?
30. Измените одно сообщение в протоколе, изображенном на рис. 8.30, так, чтобы протокол стал устойчивым к зеркальной атаке. Объясните суть ваших изменений.
31. Для установки секретного ключа между Алисой и Бобом используется алгоритм Диффи—Хеллмана. Алиса посылает Бобу (719, 3, 191). Боб отвечает (543). Секретное число Алисы $x = 16$. Чему равен секретный ключ?
32. Если Алиса никогда не встречалась с Бобом, если они не пользовались общим ключом, не имеют сертификатов, они все равно могут установить общий закрытый ключ при помощи алгоритма Диффи—Хеллмана. Объясните, почему так тяжело бороться с атакой типа «человек посередине».
33. Почему в протоколе, изображенном на рис. 8.35, *A* посылается открытым текстом вместе с зашифрованным ключом сеанса?
34. В протоколе, изображенном на рис. 8.35, мы отмечали, что начинать каждое сообщение с 32 нулевых битов рискованно с точки зрения безопасности. Предположим, что каждое сообщение начинается со случайного числа или второго секретного ключа, известного только пользователю и центру распространения ключей. Защищает ли это от атаки методом известного открытого текста?
35. В протоколе Нидхэма—Шрёдера (Needham—Schroeder) Алиса формирует два оклика, R_A и R_{A2}. Не достаточно ли будет использовать один оклик?
36. Предположим, что организация для аутентификации применяет метод Kerberos. Как в терминах безопасности и работоспособности повлияет на систему выход из строя сервера аутентификации или сервера выдачи билетов?
37. В протоколе аутентификации с открытым ключом, показанном на рис. 8.39, в сообщении 7 случайное число R_B зашифровано ключом K_S. Необходимо ли это шифрование или допустимо отправить это число обратно открытым текстом? Ответ поясните.
38. У кассовых аппаратов, использующих магнитные карты и PIN-коды, есть существенный недостаток: кассир-злоумышленник может модифицировать считывающее устройство своего кассового аппарата, чтобы считать и сохранить всю информацию, хранящуюся на карте, включая PIN-код, с целью послать дополнительные (поддельные) транзакции в будущем. В следующем поколении кассовых терминалов будут использоваться карты с полноценным центральным процессором, клавиатурой и небольшим дисплеем. Разработайте для этой системы протокол, который не сможет взломать кассир-злоумышленник.

39. Назовите *две* причины, по которым PGP использует сжатие сообщений.
40. Предположим, что повсеместно в Интернете используется PGP. Можно ли в этом случае послать PGP-сообщение по произвольному интернет-адресу и быть полностью уверенным в том, что на приемной стороне оно будет корректно расшифровано? Обсудите свой ответ.
41. На атаке, показанной на рис. 8.43, пропущен один шаг. Он не является необходимым для того, чтобы атака была удачной, однако его наличие может помочь уменьшить возможные подозрения. Итак, что же пропущено?
42. Однажды было предложено в целях противостояния атакам DNS-серверов использовать предсказание идентификаторов (то есть ввести случайные, а не последовательные идентификаторы). Обсудите это предложение с точки зрения защиты информации.
43. Протокол передачи данных SSL подразумевает наличие двух нонсов и подготовительного ключа. Какую роль играют нонсы (если они вообще играют какую-либо роль)?
44. Изображение на рис. 8.50, *б* содержит ASCII-текст пяти пьес Шекспира (хотя по фотографии этого не скажешь). Можно ли спрятать музыку, а не текст, между зебрами? Если да, то как и сколько? Если нет, то почему?
45. Алиса была постоянным пользователем анонимной рассылки первого типа. Она могла помещать сколько угодно сообщений в свою любимую конференцию *alt.fanclub.alice*, но все знали, что их автором является Алиса, так как все письма были подписаны одним и тем же псевдонимом. Если предполагать, что рассылка работает правильно, у Труди нет возможности писать от имени Алисы. После того как все анонимные рассылки первого типа были закрыты, Алиса перешла на шифрованные панковские рассылки и начала обсуждать новую тему в своей конференции. Предложите способ защиты в новых условиях от Труди, пытающейся писать письма от чужого имени.
46. Найдите в Интернете описание какого-нибудь интересного случая, касающегося конфиденциальности, и напишите небольшой отчет на эту тему.
47. Найдите в Интернете описание очередного случая противостояния закона об авторских правах добросовестному использованию продукции и напишите небольшой отчет о том, что вы узнали.
48. Напишите программу, шифрующую входные данные путем суммирования по модулю 2 с ключевым потоком. Найдите или напишите сами хороший генератор случайных чисел для создания ключевого потока. Программа должна работать подобно фильтру, принимая на входе открытый текст и выдавая на выходе на стандартное устройство вывода шифр (и наоборот). У программы должен быть один параметр: ключ, запускающий генератор случайных чисел.
49. Напишите процедуру для вычисления хэша SHA-1 блока данных. У процедуры должно быть два параметра: указатель на входной буфер и указатель на 20-байтовый выходной буфер. Чтобы найти спецификацию SHA-1, поищите в Интернете FIPS 180-1, в этом документе содержится полное описание.

Глава 9
Библиография

- Литература для дальнейшего чтения
- Алфавитный список литературы

Мы закончили изучение компьютерных сетей, но все это — только начало. Многие интересные темы не были рассмотрены во всей полноте и со всеми подробностями, а многие вопросы были вообще опущены из-за отсутствия места. Эта глава содержит список дополнительной литературы для читателей, желающих продолжить изучение компьютерных сетей.

Литература для дальнейшего чтения

Существует большое количество книг, касающихся всех аспектов компьютерных сетей и распределенных систем. Среди журналов, часто публикующих статьи по этой теме, стоит выделить следующие три: *IEEE Transactions on Communications*, *IEEE Journal on Selected Areas in Communications* и *Computer Communication Review*. Многие другие журналы также иногда публикуют статьи по теме компьютерных сетей.

Кроме того, Институт инженеров по электротехнике и электронике IEEE выпускает еще три журнала: *IEEE Internet Computing*, *IEEE Network Magazine* и *IEEE Communications Magazine* — в них содержатся обзоры, учебные статьи и информация об исследованиях, связанные с компьютерными сетями. Первые два в основном посвящены архитектуре, стандартам и программному обеспечению, тогда как журнал *IEEE Communications Magazine* большей частью занят освещением технологий коммуникаций (оптоволоконной, спутниковой связи и т. д.).

Помимо этого ежегодно или раз в два года проводятся несколько конференций, которые освещаются в многочисленных статьях, посвященных сетям и распределенным системам, в частности, ежегодная конференция *SIGCOMM, The*

International Conference on Distributed Computer Systems и *The Symposium on Operating Systems Principles.*

Далее мы перечислим дополнительную литературу, сгруппированную по главам этой книги. Большинство книг представляют собой самоучители либо обзоры. Иногда даются ссылки на главы из руководств.

Введение и неспециализированная литература

Bi и др., «Wireless Mobile Communications at the Start of the 21st Century» Новый век, новые технологии. Звучит здорово. Приводится история беспроводной связи, а также рассматриваются основные вопросы, включая стандарты, применения, Интернет и технологии.

Comer, «The Internet Book» Сюда стоит заглянуть всем, кто ищет простое и понятное описание Интернета. В этой книге в доступной даже для новичка форме рассказывается об истории, развитии, технологиях, протоколах и службах Интернета. Тем не менее, эта книга будет интересна и более подготовленным читателям благодаря большому количеству содержащегося в ней материала.

Garber, «Will 3G Really Be the Next Big Wireless Technology?» Предполагается, что третье поколение мобильных телефонов будет сочетать в себе возможности передачи как голоса, так и данных со скоростью до 2 Мбит/с. Движение в эту сторону уже началось. В этой статье, которая читается очень легко, дан обзор возможностей, подводных камней, технологий, политических и экономических аспектов применения широкополосных беспроводных коммуникаций.

IEEE Internet Computing, Jan.-Feb. 2000 В первом выпуске нового тысячелетия журнал *IEEE Internet Computing* представил то, что и ожидалось: размышления людей, участвовавших в создании Интернета, о том, каким он будет в новом веке. В обсуждении участвуют такие эксперты, как Поль Бэрэн (Paul Baran), Лоуренс Робертс (Lawrence Roberts), Леонард Кляйнрок (Leonard Kleinrock), Стефан Крокер (Stephen Crocker), Дэнни Коэн (Danny Cohen), Боб Меткалф (Bob Metcalfe), Билл Гейтс (Bill Gates), Билли Джой (Billy Joy) и др. Советую подождать лет 500 и только потом перечитать эти предсказания.

Kipnis, «Beating the System: Abuses of the Standards Adoption Process» Комитеты по стандартизации пытаются работать максимально добросовестно и независимо от разработчиков, но, к сожалению, некоторые компании пытаются нарушить эту систему. Например, уже не раз случалось так, что компания помогает в разработке стандарта, а после его утверждения заявляет, что стандарт основывается на принадлежащем ей патенте и что вопросы выдачи лицензий и цен на них компания будет решать сама. Данный материал будет полезен тем, кто хочет узнать о нелицеприятной стороне стандартизации.

Kyas и Crawford, «ATM Networks» Стандарт ATM был когда-то разрекламирован как сетевой протокол будущего, и он до сих пор остается весьма значимым в телефонной системе. Эта книга отражает нынешнее состояние дел ATM, содержит детальное описание протоколов ATM и их интеграции с сетями на базе IP.

Kwok, «A Vision for Residential Broadband Service» Если вам интересно, что думала корпорация Microsoft об организации видео по заказу в 1995 году, эта статья для вас. Спустя пять лет эта точка зрения безнадежно устарела. Назначение этой статьи состоит в том, чтобы показать, что даже высокообразованные и подкованные эксперты иногда не способны предвидеть развитие ситуации даже на пять лет вперед. Это должно послужить хорошим уроком для всех нас.

Naughton, «A Brief History of the Future» Кто же все-таки изобрел Интернет? Многие хотят, чтобы их включили в число изобретателей. И некоторые из них по праву этого заслуживают. Правдивая история Интернета, рассказанная остроумно и завораживающе, перемежающаяся историческими анекдотами (например, о том, как компания AT&T многократно разъясняла всем и искренне верила сама, что у цифровых коммуникаций нет будущего) — вот суть этой книги.

Perkins, «Mobile Networking in the Internet» Здесь вы найдете качественное описание всех уровней протоколов мобильных сетей, от физического до транспортного, включая такие вопросы как защита информации и специализированные сети.

Teger и Waks, «End-User Perspectives on Home Networking» Домашние сети не похожи на корпоративные. Отличаются и приложения (в домашних сетях более интенсивно используется мультимедиа), и оборудование, и сами пользователи, которые обычно мало искушены в технических вопросах и не понимают, что делать при возникновении каких бы то ни было неполадок. Более подробную информацию вы найдете в этой книге.

Varshney и Vetter, «Emerging Mobile and Wireless Networks» Еще одна книга, посвященная основам беспроводной связи. Здесь рассматриваются беспроводные ЛВС, местные линии связи, спутниковые сети, а также некоторые программные продукты и приложения.

Wetteroth, «OSI Reference Model for Telecommunications» Несмотря на то, что сами протоколы OSI в чистом виде сейчас не используются, семиуровневая модель стала очень известной. В этой книге не только подробно рассказывается об OSI, но и описывается связь этой модели с телефонными (то есть *не* компьютерными) сетями. Показано, как обычная телефония и другие голосовые протоколы вписываются в стек сетевых протоколов.

Физический уровень

Abramson, «Internet Access Using VSATs» В развитых странах маленькие наземные станции широко применяются как для организации телефонной связи в сельской местности, так и для корпоративного доступа в Интернет. Однако природа трафика в этих двух случаях очень разная, соответственно, нужны разные протоколы. В этой статье изобретатель системы ALOHA описывает методы выделения каналов, которые могут использоваться в системах VSAT.

Alkhatib и др., «Wireless Data Networks: Reaching the Extra Mile» Небольшое введение в технологии (включая расширенный спектр) и терминологию беспроводных сетей, которое можно использовать в качестве самоучителя.

Azzam и Ransom, «Broadband Access Technologies» В качестве технологий сетевого доступа здесь рассматриваются телефонные системы, оптоволоконные сети, ADSL, кабельные сети, спутники и даже линии электропередач. Разговор идет также о домашних сетях, службах, производительности и стандартах. В конце книги приведены биографии наиболее значимых телекоммуникационных и сетевых компаний, однако, учитывая быстро меняющуюся конъюнктуру этого рынка, можно предположить, что эта последняя глава устареет быстрее, чем остальные.

Bellamy, «Digital Telephony» В этом солидном издании содержится все, что вы когда-либо хотели узнать о телефонной системе, и даже более того. Особый интерес представляют главы, посвященные передаче данных и мультиплексированию, цифровой коммутации, волоконной оптике, мобильной телефонии и DSL.

Berezdivin и др., «Next-Generation Wireless Communications Concepts and Technologies» Авторы этой книги опередили всех остальных, так как рассказывают о четвертом поколении беспроводных сетей. Ожидается, что эти сети будут повсеместно предоставлять услуги IP, а значит, и доступ в Интернет, обеспечивая высокую пропускную способность и отличное качество обслуживания. Этого можно добиться грамотным распределением спектра, динамическим управлением ресурсами, адаптивным обслуживанием. Все это звучит несколько утопически сейчас, однако не менее утопически звучали слова о мобильной телефонии в 1995 году.

Dutta-Roy, «An Overview of Cable Modem Technology and Market Perspectives» Кабельное телевидение уже давно перестало быть службой передачи телевизионного сигнала и превратилось в сложную распределенную систему, совмещающую телевидение, Интернет и телефонию. Эти изменения заметно повлияли на инфраструктуру кабельных систем. Статья посвящена обсуждению кабельных участков сетей связи, стандартам, а также маркетинговым аспектам, особый упор делается на DOCSIS.

Farserotu и Prasad, «A Survey of Future Broadband Multimedia Sattelite Systems, Issues, and Trends» В небе над нами летает множество спутников передачи данных, среди которых Astrolink, Cyberstar, Spaceway, Skybridge, Teledisc и iSky. Немало спутниковых программ в настоящее время еще только разрабатываются. В них применяются различные технологии, включая методы «трубы» и коммутацию спутников. В предлагаемом вашему вниманию материале дается обзор различных спутниковых систем и технологий.

Hu и Li, «Sattelite-Based Internet: A Tutorial» Доступ в Интернет через спутник отличается от доступа с помощью наземных линий связи. Здесь должны учитываться не только задержки, но и маршрутизация, а также коммутация. Авторы рассматривают некоторые проблемы применения спутниковых систем для доступа в Интернет.

Joel, «Telecommunications and the IEEE Communications Society» Здесь в очень сжатой, но удивительно понятной форме описывается история телекоммуникаций, начиная с телеграфа и заканчивая сетями стандарта 802.11. Вы найдете разделы, посвященные радио, телефонии, аналоговой и цифровой коммутации, подводным кабелям, цифровой передаче данных, ATM, телевизионному вещанию, спутникам, кабельному TB, оптическим линиям связи, мобильным телефонам, пакетной коммутации, ARPANET и, конечно же, Интернету.

Metcalfe, «Computer/Network Interface Design: Lessons from Arpanet & Ethernet» Хотя инженеры занимаются созданием сетей уже несколько десятков лет, иногда задаешься вопросом, научились ли они чему-либо за это время? В этой статье разработчик сетей Ethernet рассказывает, как построить сетевой интерфейс и что с ним делать после этого. Автор откровенно сообщает, где он ошибался, а где был прав.

Palais, «Fiber Optic Communications», 3-е издание Обычно книги по оптоволоконной технологии предназначаются для специалистов, но эта книга написана более доступным языком. В этой книге рассказывается про волноводы, источники света, детекторы света, соединительные муфты, модуляцию, шум и др.

Pandya, «Emerging Mobile and Personal Communication Systems» Эта статья представляет собой краткое и забавное введение в портативные системы связи. Одна из девяти страниц статьи содержит список из 70 сокращений, используемых на остальных восьми страницах.

Sarikaya, «Packet Mode in Wireless Networks: Overview of Transition to Third Generation» Суть сотовых систем третьего поколения состоит в беспроводной передаче данных. Обзор того, как передача данных осуществляется в сетях второго поколения и какие изменения ждут нас с приходом третьего поколения, изложен в этой книге. Среди обсуждаемых тем GPRS, IS-95B, WCDMA и CDMA2000.

Уровень передачи данных

Carlson, PPP Design, Implementation and Debugging», 2-е издание Если вы хотите узнать о подробностях реализации протоколов, входящих в PPP, включая CCP (Сжатие) и ECP (Шифрование), эта книга послужит хорошим подспорьем. Много внимания, в частности, уделяется одной из популярных реализаций PPP, а именно ANU PPP-2.3.

Gravano, «Introduction to Error Control Codes» Ошибки проникают практически во все цифровые системы связи, поэтому было разработано много типов кодов обнаружения и исправления ошибок. В этой книге рассказывается о наиболее важных из них, начиная от простых линейных кодов Хэмминга и заканчивая кодами, в основе которых лежит теория полей Галуа, а также сверточными кодами. Разумеется, автор оперирует лишь минимально необходимыми алгебраическими понятиями, но и это может показаться слишком сложным человеку, не искушенному в специальных разделах высшей математики.

Holtzman, «Design and Validation of Computer Protocols» Читателям, интересующимся более формальным аспектами протоколов передачи данных (и подобным им), следует прочитать эту книгу. Здесь подробно описываются спецификация, моделирование и тестирование таких протоколов.

Peterson и Davie, «Computer Networks: A System Approach» В главе 2 этой книги содержится материал, касающийся проблем уровня передачи данных, включая формирование кадров, обнаружение ошибок, протоколы с ожиданием, протоколы скользящего окна и локальные сети IEEE 802.

Stallings, «Data and Computer Communications» В главе 7 описывается уровень передачи данных и связанные с ним вопросы управления потоком и обнаружения ошибок, а также базовые протоколы этого уровня, включая протокол с ожиданием и возвратом на *n*. Описаны и протоколы типа HDLC.

Подуровень управления доступом к носителю

Bhagwat, «Bluetooth: Technology for Short-Range Wireless Apps» Простое и понятное введение в систему Bluetooth. Основные протоколы и профили, радиосвязь, пикосети, связи, а также основы различных протоколов — все это составляет книгу.

Bisdikian, «An Overview of the Bluetooth Wireless Technology» Как и представленный выше материал, это хорошая отправная точка для тех, кто хочет узнать больше о системе Bluetooth. Кроме всего прочего, рассматриваются пикосети, стек протоколов и профили.

Crow и др., «IEEE 802.11 Wireless Local Area Networks» Введение в технологии и протоколы 802.11, написанное довольно простым языком. Упор делается на подуровень управления доступом к носителю (MAC-подуровень). Обсуждается как распределенное, так и централизованное управление. В конце приводится анализ производительности 802.11 при различных условиях.

Eklund и др., «IEEE Standard 802.16: A Technical Overview of the Wireless MAN Air Interface for Broadband Wireless Access» Беспроводные местные линии связи, стандартизованные IEEE в 2002 году (стандарт 802.16), могут совершить настоящую революцию на рынке услуг телефонной связи, позволив провести широкополосные линии в каждый дом. В данном обзоре автор поясняет основные технологические вопросы, связанные с этим стандартом.

Kapp, «802.11: Leaving the Wire Behind» Краткое введение в 802.11 включает в себя описание основ, протоколов и наиболее значимых стандартов.

Kleinrock, «On Some Principles of Nomadic Computing and Multi-Access Communications» Беспроводный доступ по общему разделяемому каналу — это более сложный вопрос, чем разделение обычного проводного канала. Кроме всего прочего, возникают проблемы динамических топологий, маршрутизации и управления питанием. В этой статье описываются эти и другие вопросы доступа к каналу со стороны мобильных беспроводных устройств.

Miller и Cummins, «LAN Technologies Explained» Хотите узнать больше о технологиях, применяемых в локальных сетях? В этой книге описано большинство из них, включая FDDI и маркерное кольцо и, разумеется, Ethernet. Сети первых двух типов сейчас устанавливают довольно редко, однако эти технологии широко применяются в уже существующих сетях, а кольцевая организация и сейчас очень популярна (взять хотя бы SONET).

Perlman, «Interconnections», 2-е издание Это заслуживающая доверия и в то же время увлекательно написанная книга о мостах, маршрутизаторах и маршрутизации в целом. Автор книги сам участвовал в разработке алгоритмов, применяемых в мосте связующего дерева в сетях стандарта IEEE 802, так что он, несомненно, является одним из ведущих экспертов в области различных аспектов сетевых технологий.

Webb, «Broadband Fixed Wireless Access» Здесь вы найдете ответы на вопросы, как и зачем функционируют стационарные беспроводные широкополосные сети. В разделе «Зачем?» приводится следующий аргумент: людям надоедает иметь разные домашний и рабочий электронные адреса, отдельные номера домашнего телефона, рабочего и мобильного, а также центра диалогового обмена сообщениями, да к тому же еще один-два номера факса. Хочется иметь единую интегрированную систему, которая работала бы везде. В разделе, посвященном технологиям («Как?»), упор делается на физический уровень, здесь вы найдете сравнения TDD и FDD, адаптивной и постоянной модуляций, а также большого числа носителей.

Сетевой уровень

Bhatti и Crowcroft, «QoS Sensitive Flows: Issues in IP Packet Handling» Один из способов достижения высокого качества обслуживания в сетях заключается в тщательном составлении расписания отправки пакетов с каждого маршрутизатора. В этой статье более или менее детально рассматриваются алгоритмы планирования и смежные вопросы.

Chakrabarti, «QoS Issues in Ad Hoc Wireless Networks» Маршрутизация в специализированных сетях, составленных из ноутбуков, оказавшихся рядом, достаточно сложна, даже если не задумываться о качестве обслуживания. Однако людям все же важно качество обслуживания, поэтому на данный вопрос приходится обращать внимание. В этой статье обсуждаются природа специализированных сетей и некоторые проблемы маршрутизации и качества обслуживания.

Comer, «Internetworking with TCP/IP», том 1, 4-е издание Автор представил наиболее полный труд о наборе протоколов TCP/IP. Главы с 4-й по 11-ю посвящены протоколу IP и родственным протоколам сетевого уровня. В остальных главах, также заслуживающих внимания, описываются в основном более высокие уровни.

Hiutema, «Routing in the Internet» Если вы хотите знать абсолютно все о маршрутизации в Интернете, то эта книга для вас. В ней детально описываются алгоритмы как с произносимыми названиями (например, RIP, CIDR и MBONE), так

и с непроизносимыми (OSPF, IGRP, EGP и BGP). В книге рассказывается также о новых возможностях, таких как многоадресная рассылка, мобильный IP-протокол и резервирование от источника.

Malhotra, «IP Routing» Детальное описание IP-маршрутизации. Среди рассматриваемых протоколов — RIP, RIP-2, IGRP, EIGRP, OSPF и BGP-4.

Metz, «Differentiated Services» Гарантии качества обслуживания важны для многих мультимедийных приложений. Интегрированные и дифференцированные услуги представляют собой два возможных подхода к обеспечению качества обслуживания. Здесь обсуждается и то, и другое, однако упор делается на дифференцированное обслуживание.

Metz, «IP routers: New Tool for Gigabit Networking» Большинство литературы, посвященной темам, обсуждаемым в главе 5 нашей книги, связано с алгоритмами маршрутизации. Здесь же рассказывается о том, как реально работают маршрутизаторы. Они прошли в своем развитии долгий путь от рабочих станций общего назначения до узкоспециализированных устройств маршрутизации. Если вы хотите познакомиться с маршрутизаторами поближе, советуем вам для начала прочесть эту статью.

Nemeth и др., «UNIX System Administration Handbook» Надо сказать, что глава 13 этой книги представляет собой наиболее практическое руководство по сравнению со всеми остальными ссылками. Здесь не описываются абстрактные концепции, а даются практические советы о том, что делать в той или иной ситуации, возникающей при управлении реальной сетью.

Perkins, «Mobile Networking through Mobile IP» Мобильные вычислительные устройства становятся все популярнее, а с ними и протокол Mobile IP. Этот краткий самоучитель представляет собой хорошее введение в курс дела.

Perlman, «Interconnections: Bridges and Routers», 2-е издание В главах с 12-й по 15-ю автор описывает многочисленные аспекты разработки алгоритмов одноадресной и групповой рассылки, как для глобальных, так и для локальных сетей, и их реализацию в различных протоколах. Однако интереснее всего читать главу 18, в которой автор делится своими личными впечатлениями о работе с сетевыми протоколами. Эта глава и информативна, и весьма забавна.

Puzmanova, «Routing and Switching: Time of Convergence?» В конце 1990-х некоторые производители оборудования стали называть коммутаторами все подряд, а менеджеры многих крупных сетей стали говорить, что они «переходят с маршрутизаторов на коммутаторы». Как следует из названия книги («Маршрутизация и коммутация: пора сближения?»), автор пытается угадать будущее обоих типов устройств и проанализировать реальные возможности их совмещения.

Royer и Toh, «A Review of Current Routing Protocols for Ad-Hoc Mobile Wireless Networks» Специализированный алгоритм маршрутизации AODV, который мы рассматривали в главе 5 нашей книги, разумеется, далеко не единственный. Среди прочих стоит назвать DSDV, CGSR, WRP, DSR, TORA, ABR, DRP и SRP. Все они рассматриваются и сравниваются между собой в этой работе. Кажется, уже сейчас вам должен быть очевиден первый шаг, необходимый для разработки любого специализированного алгоритма маршрутизации: придумывание какого-нибудь ужасного трех- или четырехбуквенного сокращения.

Stevens, «TCP/IP Illustrated», том 1 Главы с 3-й по 10-ю содержат доступное описание протокола IP и родственных ему протоколов (ARP, RARP и ICMP), дополненное примерами.

Streigel и Manimaran, «A Survey of QoS Multicasting Issues» Многоадресная рассылка и качество обслуживания — это те основные проблемы, которые возникают в условиях растущей популярности радио и телевидения в Интернете. Что касается данного обзора, здесь авторы обсуждают то, как в алгоритмах маршрутизации следует учитывать оба этих вопроса.

Yang и Reddy, «A Taxonomy for Congestion Control Algorithms in Packet Switching Networks» Авторы систематизировали алгоритмы борьбы с перегрузкой. Основными категориями являются алгоритмы без обратной связи с управлением от источника и от приемника, а также алгоритмы с явной и неявной обратной связью. В книге классифицируются и описываются 23 существующих алгоритма.

Транспортный уровень

Comer, Internetworking with TCP/IP, том 1, 4-е издание Как уже говорилось ранее, автор написал наиболее полный труд о наборе протоколов TCP/IP. Глава 12 посвящена протоколу UDP, а глава 13 — протоколу TCP.

Hall и Cerf, «Internet Core Protocols: The Definitive Guide» Если вы предпочитаете получать информацию из первых рук и вам хочется узнать побольше о TCP, эта книга для вас. Один из авторов, Cerf, как-никак был одним из разработчиков TCP. В главе 7 вы найдете качественное описание TCP, показывающее, как интерпретировать информацию, получаемую в результате анализа протокола и с помощью инструментария управления сетью. В остальных главах рассказывается про UDP, IGMP, ICMP и ARP.

Kurose и Ross, «Computer Networking: A Top-Down Approach Featuring the Internet» Глава 3 этой книги посвящена транспортному уровню и содержит много информации про UDP и TCP. Также обсуждаются протоколы с ожиданием и возвратом на *n*, описанные в главе 3 нашей книги.

Mogul, «IP Network Performance» Несмотря на свое название, эта статья скорее о производительности протокола TCP и производительности сети вообще. Книга содержит большое количество полезных указаний и практических советов.

Peterson и Davie, «Computer Networks: A System Approach» Глава 5 посвящена UDP, TCP и некоторым смежным протоколам. Также вкратце рассказывается о производительности сетей.

Stevens, «TCP/IP Illustrated», том 1 Главы с 17-й по 24-ю содержат доступное описание протокола TCP, дополненное примерами.

Прикладной уровень

Begholz, «Extending Your Markup: An XML Tutorial» Короткое и понятное введение в XML для начинающих.

Cardellini и др., «The State-of-the-Art in Locally Dustributed Web-Server Systems» По мере роста популярности Всемирной паутины некоторым сайтам требуются все большие масштабы серверных ферм для обработки трафика. Одной из тяжелейших задач построения серверных ферм является распределение нагрузки между машинами. В данной работе очень подробно обсуждается эта проблема.

Berners-Lee и др., «The World Wide Web» Взгляд на Паутину и на пути ее развития со стороны человека, который ее придумал, и его коллег по CERN. Статья посвящена архитектуре Паутины, унифицированным указателям (URL), протоколу HTTP, языку HTML, а также перспективам на будущее. Приводится сравнение с другими распределенными информационными средами.

Choudbury и др., «Copyright Protection for Electronic Publishing on Computer Networks» Об алгоритмах шифрования написано огромное количество книг и статей. Однако мало где можно найти сведения о том, как использовать эти алгоритмы, чтобы предотвратить дальнейшее распространение пользователями документов, которые им разрешено расшифровывать. В этой статье описывается несколько способов, полезных для защиты авторских прав в эру электроники.

Collins, «Carrier Grade Voice over IP» Если вы уже прочитали работу Varshney и др. и теперь хотите узнать все подробности передачи голоса поверх IP с использованием H.323, вам следует обратить внимание на этот труд. Несмотря на то, что книга толстая и подробная, по сути своей она является самоучителем и даже, в общем-то, не требует предварительных знаний о телефонных системах.

Davidson, «A Web Chaching Primer» По мере роста Всемирной паутины кэширование становится все более критичным вопросом в деле обеспечения хорошей производительности. Краткое введение, описывающее веб-кэширование, составляет суть этой работы.

Krishnamurthy и Redfox, «Web Protocols and Practice» Тяжело найти более понятную книгу, которая при этом охватывала бы все аспекты Всемирной паутины. Здесь, разумеется, рассказывается о клиентах, серверах, прокси и кэшировании. Однако вряд ли вы могли бы предположить, что есть в этой книге разделы, посвященные трафику и его измерению в Паутине, а также текущим исследованиям и направлениям развития Веб.

Rabinovich и Spatscheck, «Web Caching and Replication» Здесь весьма просто объясняются принципы кэширования и репликации во Всемирной паутине. Очень подробно описываются прокси, кэширование, сети доставки содержимого, выбор сервера и многое другое.

Shahabi и др., «Yima: A Second-Generation Continuous Media Server» Серверы мультимедиа представляют собой сложные системы, которым приходится заниматься планированием загрузки ЦП, размещением файлов на диске, синхронизацией потоков и многим другим. Со временем люди учатся реализовывать их все лучше и лучше. В этой работе представлен обзор архитектуры одной из недавно появившихся систем.

Tittel и др., «Mastering XHTML» Две книги, описывающие новый стандарт языка описания веб-страниц, собраны под одной обложкой. В начале находится текст, описывающий XHTML, в основном в сравнении с обычным HTML. Далее следует удобный справочник по тегам, кодам и спецсимволам, использующимся в XHTML 1.0.

Varshney и др., «Voice over IP» Как работает система передачи голоса поверх IP? Заменит ли она обычную коммутируемую телефонную сеть общего доступа? Об этом вы узнаете, прочитав эту книгу.

Безопасность в сетях

Anderson, R., «Why Cryptosystems Fail» Согласно Андерсону, безопасность банковских систем очень плоха, но не потому, что хитроумные злоумышленники взламывают шифр DES на своих персональных компьютерах. Настоящие проблемы варьируются от нечестных банковских служащих (изменяющих почтовый адрес клиента на свой, чтобы перехватить пластиковую карту и номер PIN) до ошибок в программах (всем клиентам выдается один и тот же номер PIN). Особенно интересна реакция банков на случаи ошибок: наши системы безупречны, так что все подобные случаи происходят в результате ошибок или мошенничества самих клиентов.

Anderson, «Security Engineering» Можно сказать, что это 600-страничная версия описанной ранее книги того же автора. В ней описываются более технические вещи, нежели в «*Secrets and Lies*», однако менее технические, чем в «*Network Security*» (см. далее). Вначале даются основы методов защиты информации, затем следуют целые главы, посвященные разным приложениям, включая банковские системы, системы управления атомной энергетикой, безопасную печать, биометрию, физическую защиту, электронные войны, защиту в телекоммуникациях, электронную коммерцию и защиту авторских прав. Третья часть книги посвящена политике, управлению и оценке систем.

Brands, «Rethinking Public Key Infrastructures and Digital Certificates» Это больше, чем просто хорошее введение в технологии цифровых сертификатов, — это еще и мощная пропагандистская работа. Автор уверен, что нынешние бумажные системы верификации изжили себя и являются неэффективными, и утверждает, что цифровые сертификаты очень хорошо подходят для таких приложений, как цифровые голосования, управление предоставлением прав и даже в качестве замены бумажных денег. Кроме того, он предупреждает, что без PKI и шифрования Интернет может стать огромной системой слежения.

Kaufman и др., «Network Security», 2-е издание Эту заслуживающую доверия и часто остроумную книгу следует читать в первую очередь, если вас интересует дополнительная информация о техническом аспекте алгоритмов и протоколов безопасности сетей. В ней подробно освещаются алгоритмы и протоколы с секретным и открытым ключом, хэширование сообщений, протокол Kerberos, PKI, IPsec, SSL/TLS и электронная почта. Все темы снабжены примерами. Глава 26, посвященная фольклору на тему защиты информации, — это настоящий шедевр. В деле обеспечения безопасности важны все детали. Если вы собираетесь разработать действительно полезную систему защиты, эта глава подскажет вам в виде примеров из реальной жизни много интересного.

Pohlmann, «Firewall Systems» Брандмауэры — это начало и конец системы защиты от взломщиков. В этой книге поясняется, как работают брандмауэры и чем они занимаются. Начинается описание с простейших программных систем, предназначенных для защиты отдельных ПК, и заканчивается сложными аппаратами, располагающимися между частными сетями и Интернетом.

Schneier, «Applied Cryptography», 2-е издание Этот монументальный сборник является самым страшным кошмаром Агентства Национальной Безопасности США: под одной обложкой собраны описания всех известных криптографических алгоритмов. Мало того, большинство алгоритмов приводятся в этой книге в виде действующих программ на языке C. Также в книге содержится более 1600 ссылок на криптографическую литературу. Если вы *всерьез* решили сохранить содержимое ваших файлов в секрете, прочитайте эту книгу.

Schneier, «Secrets and Lies» Прочитав «*Applied Cryptography*» от корки до корки, вы станете знатоком криптографических алгоритмов. Если же вы после этого не менее внимательно прочтете «*Secrets and Lies*» (что займет уже гораздо меньше времени), то поймете, что одними алгоритмами дело не ограничивается. Слабость большинства систем защиты связана не с плохими алгоритмами или слишком короткими ключами, а с пороками в окружающей эти системы среде. Приведено бессчетное множество примеров возможных угроз, атак, защит от них, контратак и т. д. Эта книга является прекрасным нетехническим описанием систем безопасности, рассматривающим проблему в самом широком смысле.

Skoudis, «Counter Hack» Как остановить взломщика? Надо думать так же, как он. В этой книге показан взгляд на сеть со стороны взломщика; автор утверждает, что защита информации должна быть одной из функций всей сетевой системы в целом, она не должна додумываться и прикручиваться в виде специальной технологии к уже существующим сетям. Рассматриваются почти все типы наиболее распространенных атак, включая «социальный инжиниринг» — тип атаки, рассчитанный на незнание пользователем систем электронной безопасности.

Алфавитный список литературы

Abramson, N.: «Internet Access Using VSATs», IEEE Commun. Magazine, vol. 38, pp. 60–68, July 2000.

Abramson, N.: «Development of the ALOHANET», IEEE Trans, on Information Theory, vol. IT-31, pp. 119–123, March 1985.

Adams, M., and Dulchinos, D.: «OpenCable», IEEE Commun. Magazine, vol. 39, pp. 98–105, June 2001.

Alkhatib, H. S., Bailey, C., Gerla, M., and McRae, J.: «Wireless Data Networks: Reaching the Extra Mile», Computer, vol. 30, pp. 59–62, Dec. 1997.

Anderson, R.J.: «Free Speech Online and Office», Computer, vol. 25, pp. 28–30, June 2002.

Anderson, R.J.: «Security Engineering», New York: Wiley, 2001.

Anderson, R.J.: «The Eternity Service», Proc. First Int'l Conf. on Theory and Appl. of Cryptology, CTU Publishing House, 1996.

Anderson, R.J.: «Why Cryptosystems Fail», Commun. of the ACM, vol. 37, pp. 32–40, Nov. 1994.

Artz, D.: «Digital Steganography», IEEE Internet Computing, vol. 5, pp. 75–80, 2001.

Azzam, A. A., and Ransom, N.: «Broadband Access Technologies», New York: McGraw-Hill, 1999.

Bakne, A., and Badrinath, B. R.: «I-TCP: Indirect TCP for Mobile Hosts», Proc. 15th Int'l Conf. on Distr. Computer Systems, IEEE, pp. 136–143, 1995.

Balakrishnan, H., Seshan, S., and Katz, R. H.: «Improving Reliable Transport and Handoff Performance in Cellular Wireless Networks», Proc. ACM Mobile Computing and Networking Conf., ACM, pp. 2–11, 1995.

Ballardie, T., Francis, P., and Crowcroft, J.: «Core Based Trees (CBT)», Proc. SIGCOMM '93 Conf., ACM, pp. 85–95, 1993.

Barakat, C., Altman, E., and Dabbous, W.: «On TCP Performance in a Heterogeneous Network: A Survey», IEEE Commun. Magazine, vol. 38, pp. 40–46, Jan. 2000.

Bellamy, J.: Digital Telephony, 3rd ed., New Ytfrk: Wiley, 2000.

Bellman, R. E.: Dynamic Programming, Princeton, NJ: Princeton University Press, 1957.

Belsnes, D.: «Flow Control in the Packet Switching Networks», Communications Networks, Uxbridge, England: Online, pp. 349–361, 1975.

Bennet, C. H., and Brassard, G.: «Quantum Cryptography: Public Key Distribution and Coin Tossing», Int'l Conf. on Computer Systems and Signal Processing, pp. 175–179, 1984.

Berezdivin, R., Breinig, R., and Topp, R.: «Next-Generation Wireless Communication Concepts and Technologies», IEEE Commun. Magazine, vol. 40, pp. 108–116, March 2002.

Berghel, H. L.: «Cyber Privacy in the New Millennium», Computer, vol. 34, pp. 132–134, Jan. 2001.

Bergholz, A.: «Extending Your Markup: An XML Tutorial», IEEE Internet Computing, vol. 4, pp. 74–79, July-Aug. 2JX)0.

Berners-Lee, T., Cailliau, A., Loutonen, A., Nielsen, H. F., and Secret, A.: «The World Wide Web», Commun. of the ACM, vol. 37, pp. 76–82, Aug. 1994.

Bertsekas, D., and Gallager, R.: «Data Networks», 2nd ed., Englewood Cliffs, NJ: Prentice Hall, 1992.

Bhagwat, P.: «Bluetooth: Technology for Short-Range Wireless Apps», IEEE Internet Computing, vol. 5, pp. 96–103, May-June 2001.

Bharghavan, V., Demers, A., Shenker, S., and Zhang, L.: «MACAW: A Media Access Protocol for Wireless LANs», Proc. SIGCOMM '94 Conf., ACM, pp. 212–225, 1994.

Bhatti, S. N., and Crowcroft, J.: «QoS Sensitive Flows: Issues in IP Packet Handling», IEEE Internet Computing, vol. 4, pp. 48–57, July-Aug. 2000.

Bi, Q., Zysman, G. I., and Menkes, H.: «Wireless Mobile Communications at the Start of the 21st Century», IEEE Commun. Magazine, vol. 39, pp. 110–116, Jan, 2001.

Biham, E., and Shamir, A.: «Differential Cryptanalysis of the Data Encryption Standard», Proc. 17th Ann. Int'l Cryptology Conf., Berlin: Springer-Verlag LNCS 1294, pp. 513–525, 1997.

Bird, R., Gopal, I., Herzberg, A., Janson, P. A., Kutten, S., Molva, R, and Yung, M.: «Systematic Design of a Family of Attack-Resistant Authentication Protocols», IEEE J. on Selected Areas in Commun., vol. 11, pp. 679–693, June 1993.

Birrell, A. D., and Nelson, B. J.: «Implementing Remote Procedure Calls», ACM Trans, on Computer Systems, vol. 2, pp. 39–59, Feb. 1984.

Biryukov, A., Shamir, A., and Wagner, D.: «Real Time Cryptanalysis of A5/1 on a PC», Proc. Seventh Int'l Workshop on Fast Software Encryption, Berlin: Springer-Verlag LNCS 1978, 2000.

Bisdikian, C.: «An Overview of the Bluetooth Wireless Technology», IEEE Commun. Magazine, vol. 39, pp. 86–94, Dec. 2001.

Blaze, M.: «Protocol Failure in the Escrowed Encryption Standard», Proc. Second ACM Conf. on Computer and Commun. Security, ACM, pp. 59–67, 1994.

Blaze, M., and Bellovin, S.: «Tapping on My Network Door», Commun. of the ACM, vol. 43, p. 136, Oct. 2000.

Bogineni, K., Sivalingam, K. M., and Dowd, P. W.: «Low-Complexity Multiple Access Protocols for Wavelength-Division Multiplexed Photonic Networks», IEEE Journal on Selected Areas in Commun., vol. 11, pp. 590–604, May 1993.

Bolcskei, H., Paulraj, A. J., Hari, K. V. S., and Nabar, R. U.: «Fixed Broadband Wireless Access: State of the Art, Challenges, and Future Directions», IEEE Commun. Magazine, vol. 39, pp. 100–108, Jan. 2001.

Borisov, N., Goldberg, I., and Wagner, D.: «Intercepting Mobile Communications: The Insecurity of 802.11», Seventh Int'l Conf. on Mobile Computing and Networking, ACM, pp. 180–188, 2001.

Brands, S.: «Rethinking Public Key Infrastructures and Digital Certificates», Cambridge, MA: M.I.T. Press, 2000.

Bray, J., and Sturman, C. F.: «Bluetooth 1.1: Connect without Cables», 2nd ed., Upper Saddle River, NJ: Prentice Hall, 2002.

Breyer, R., and Riley, S.: «Switched, Fast, and Gigabit Ethernet», Indianapolis, IN: New Riders, 1999.

Brown, S.: «Implementing Virtual Private Networks», New York: McGraw-Hill, 1999.

Brown, L., Kwan, M., Pieprzyk, J., and Seberry, J.: «Improving Resistance to Differential Cryptanalysis and the Redesign of LOKI», ASIACRYPT '91 Abstracts, pp. 25–30, 1991.

Burnett, S., and Paine, S.: «RSA Security's Official Guide to Cryptography», Berkeley, CA: Osborne/McGraw-Hill, 2001.

Capetanakis, J. I.: «Tree Algorithms for Packet Broadcast Channels», IEEE Trans, on Information Theory, vol. IT-25, pp. 505–515, Sept. 1979.

Cardellini, V., Casalicchio, E., Colajanni, M., and Yu, P. S.: «The State-of-the-Art in Locally Distributed Web-Server Systems», ACM Computing Surveys, vol. 34, pp. 263–311, June 2002.

Carlson, J.: «PPP Design, Implementation and Debugging», 2nd ed., Boston: Addison-Wesley, 2001.

Cere, V., and Kahn, R.: «A Protocol for Packet Network Interconnection», IEEE Trans, on Commun., vol. COM-22, pp. 637–648, May 1974.

Chakrabarti, S.: «QoS Issues in Ad Hoc Wireless Networks», IEEE Commun. Magazine, vol. 39, pp. 142–148, Feb. 2001.

Chase, J. S., Gallatin, A. J., and Yocum, K. G.: «End System Optimizations for High-Speed TCP», IEEE Commun. Magazine, vol. 39, pp. 68–75, April 2001.

Chen, B., Jamieson, K., Balakrishnan, H., and Morris, R.: «Span: An Energy-Efficient Coordination Algorithm for Topology Maintenance in Ad Hoc Wireless Networks», ACM Wireless Networks, vol. 8, Sept. 2002.

Chen, K.-C.: «Medium Access Control of Wireless LANs for Mobile Computing», IEEE Network Magazine, vol. 8, pp. 50–63, Sept.-Oct. 1994.

Choudbury, A. K., Maxemchuk, N. F., Paul, S., and Schulzrinne, H. G.: «Copyright Protection for Electronic Publishing on Computer Networks», IEEE Network Magazine, vol. 9, pp. 12–20, May-June 1995.

Chu, Y., Rao, S. G., and Zhang, H.: «A Case for End System Multicast», Proc. Int'l Conf. on Measurements and Modeling of Computer Syst., ACM, pp. 1–12, 2000.

Clark, D. D.: «The Design Philosophy of the DARPA Internet Protocols», Proc. SIGCOMM '88 Conf., ACM, pp. 106–114, 1988.

Clark, D. D.: «Window and Acknowledgement Strategy in TCP», RFC 813, July 1982.

Clark, D. D., Da Vie, B. S., Farber, D. J., Gopal, I. S., Kadaba, B. K., Sincoskie, W. D., Smith, J. M., and Tennenhouse, D. L.: «The Aurora Gigabit Testbed», Computer Networks and ISDN Systems, vol. 25, pp. 599–621, Jan. 1993.

Clark, D. D., Jacobson, V., Romkey, J., and Sal Wen, H.: «An Analysis of TCP Processing Overhead», IEEE Commun. Magazine, vol. 27, pp. 23–29, June 1989.

Clark, D. D., Lambert, M., and Zhang, L.: «NETBLT: A High Throughput Transport Protocol», Proc. SIGCOMM '87 Conf., ACM, pp. 353–359, 1987.

Clarke, A. C.: «Extra-Terrestrial Relays», Wireless World, 1945.

Clarke, I., Miller, S. G., Hong, T. W., Sandberg, O., and Wiley, B.: «Protecting Free Expression Online with Freenet», IEEE Internet Computing, vol. 6, pp. 40–49, Jan.-Feb. 2002.

Collins, D.: «Carrier Grade Voice over IP», New York: McGraw-Hill, 2001.

Collins, D., and Smith, C.: «3G Wireless Networks», New York: McGraw-Hill, 2001.

Comer, D. E.: «The Internet Book», Englewood Cliffs, NJ: Prentice Hall, 1995.

Comer, D. E.: «Internetworking with TCP/IP», vol. 1, 4th ed., Englewood Cliffs, NJ: Prentice Hall, 2000.

Costa, L. H. M. K., Fdida, S., and Duarte, O. C. M. B.: «Hop by Hop Multicast Routing Protocol», Proc. 2001 Conf. on Applications, Technologies, Architectures, and Protocols for Computer Commun., ACM, pp. 249–259, 2001.

Craver, S. A., Wu, M., Liu, B., Stubblefield, A., Swartzlander, B., Wallach, D. W., Dean, D., and Felten, E. W.: «Reading Between the Lines: Lessons from the SDMI Challenge», Proc. 10th USENIX Security Symp., USENIX, 2001.

Crespo, P. M., Honig, M. L., and Salehi, J. A.: «Spread-Time Code-Division Multiple Access», IEEE Trans, on Commun., vol. 43, pp. 2139–2148, June 1995.

Crow, B. P., Widjaja, I, Kim, J. G., and Sakai, P. T.: «IEEE 802.11 Wireless Local Area Networks», IEEE Commun. Magazine, vol. 35, pp. 116–126, Sept. 1997.

Crowcroft, J., Wang, Z., Smith, A., and Adams, J.: «A Rough Comparison of the IETF and ATM Service Models», IEEE Network Magazine, vol. 9, pp. 12–16, Nov.-Dec. 1995.

Dabek, F., Brunskill, E., Kaashoek, M. F., Karger, D., Morris, R., Stoica, R., and Balakrishnan, H.: «Building Peer-to-Peer Systems With Chord, a Distributed Lookup Service», Proc. 8th Workshop on Hot Topics in Operating Systems, IEEE, pp. 71–76, 2001a.

Dabek, F., Kaashoek, M. F., Karger, D., Morris, R., and Stoica, I.: «Wide-Area Cooperative Storage with CFS», Proc. 18th Symp. on Operating Systems Prin., ACM, pp. 202–215, 2001b.

Daemen, J., and Rijmen, V.: «The Design of Rijndael», Berlin: Springer-Verlag, 2002.

Danthine, A. A. S.: «Protocol Representation with Finite-State Models», IEEE Trans, on Commun., vol. COM-28, pp. 632–643, April 1980.

Davidson, J., and Peters, J.: «Voice over IP Fundamentals», Indianapolis, IN: Cisco Press, 2000.

Davie, B., and Rekhter, Y.: «MPLS Technology and Applications», San Francisco: Morgan Kaufmann, 2000.

Davis, P. T., and McGuffin, C. R.: «Wireless Local Area Networks», New York: McGraw-Hill, 1995.

Davison, B. D.: «A Web Caching Primer», IEEE Internet Computing, vol. 5, pp. 38–45, July-Aug. 2001.

Day, J. D.: «The (Un)Revised OSI Reference Model», Computer Commun. Rev., vol. 25, pp. 39–55, Oct. 1995.

Day, J. D., and Zimmermann, H.: «The OSI Reference Model», Proc. of the IEEE, vol. 71, pp. 1334–1340, Dec. 1983.

De Vriendt, J., Laine, P., Lerouge, C, and Xu, X.: «Mobile Network Evolution: A Revolution on the Move», IEEE Commun. Magazine, vol. 40, pp. 104–111, April 2002.

Deering, S. E.: «SIP: Simple Internet Protocol», IEEE Network Magazine, vol. 7, pp. 16–28, May-June 1993.

Demers, A., Keshav, S., and Shenker, S.: «Analysis and Simulation of a Fair Queue-ing Algorithm», Internetwork: Research and Experience, vol. 1, pp. 3–26, Sept. 1990.

Denning, D. E., and Sacco, G. M.: «Timestamps in Key Distribution Protocols», Commun. of the ACM, vol. 24, pp. 533–536, Aug. 1981.

Diffie, W., and Hellman, M. E.: «Exhaustive Cryptanalysis of the NBS Data Encryption Standard», Computer, vol. 10, pp. 74–84, June 1977.

Diffie, W., and Hellman, M. E.: «New Directions in Cryptography», IEEE Trans, on Information Theory, vol. IT-22, pp. 644–654, Nov. 1976.

Dijkstra, E. W.: «A Note on Two Problems in Connexion with Graphs», Numer. Math., vol. 1, pp. 269–271, Oct. 1959.

Dobrowski, G., and Grise, D.: «ATM and SONET Basics», Fuquay-Varina, NC: APDG Telecom Books, 2001.

Donaldson, G., and Jones, D.: «Cable Television Broadband Network Architectures», IEEE Commun. Magazine, vol. 39, pp. 122–126, June 2001.

Dorfman, R.: «Detection of Defective Members of a Large Population», Annals Math. Statistics, vol. 14, pp. 436–440, 1943.

Doufexi, A., Armour, S., Butler, M., Nix, A., Bull, D., McGeehan, J., and Karlsson, P.: «A Comparison of the HIPERLAN/2 and IEEE 802.11 A Wireless LAN Standards», IEEE Commun. Magazine, vol. 40, pp. 172–180, May 2002.

Durand, A.: «Deploying IPv6», IEEE Internet Computing, vol. 5, pp. 79–81, Jan.-Feb. 2001.

Dutcher, B.: «The NAT Handbook», New York: Wiley, 2001.

Dutta-Roy, A.: «An Overview of Cable Modem Technology and Market Perspectives», IEEE Commun. Magazine, vol. 39, pp. 81–88, June 2001.

Easttom, C.: «Learn JavaScript», Ashburton, U.K.: Wordware Publishing, 2001.

El Gamal, T.: «A Public-Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms», IEEE Trans, on Information Theory, vol. IT-31, pp. 469–472, July 1985.

Elhanany, I., Kahane, M., and Sadot, D.: «Packet Scheduling in Next-Generation Multiterabit Networks», Computer, vol. 34, pp. 104–106, April 2001.

Elmirghani, J. M. H., and Mouftah, H. T.: «Technologies and Architectures for Scalable Dynamic Dense WDM Networks», IEEE Commun. Magazine, vol. 38, pp. 58–66, Feb. 2000.

Farserotu, J., and Prasad, R.: «A Survey of Future Broadband Multimedia Satellite Systems, Issues, and Trends», IEEE Commun. Magazine, vol. 38, pp. 128–133, June 2000.

Fiorini, D., Chiani, M., Tralli, V., and Salati., C.: «Can we Trust HDLC?», Computer Commun. Rev., vol. 24, pp. 61–80, Oct. 1994.

Floyd, S., and Jacobson, V.: «Random Early Detection for Congestion Avoidance», IEEE/ACM Trans, on Networking, vol. 1, pp. 397–413, Aug. 1993.

Fluhrer, S., Mantin, I., and Shamir, A.: «Weakness in the Key Scheduling Algorithm of RC4», Proc. Eighth Ann. Workshop on Selected Areas in Cryptography, 2001.

Ford, L. R., Jr., and Fulkerson, D. R.: «Flows in Networks», Princeton, NJ: Princeton University Press, 1962.

Ford, W., and Baum, M. S.: «Secure Electronic Commerce», Upper Saddle River, NJ: Prentice Hall, 2000.

Forman, G. H., and Zahorjan, J.: «The Challenges of Mobile Computing», Computer, vol. 27, pp. 38–47, April 1994.

Francis, P.: «A Near-Term Architecture for Deploying Pip», IEEE Network Magazine, vol. 7, pp. 30–37, May-June 1993.

Fraser, A. G.: «Towards a Universal Data Transport System», in Advances in Local Area Networks, Kummerle, K., Tobagi, F., and Limb, J.O. (Eds.), New York: IEEE Press, 1987.

Frengle, N.: «I-Mode: A Primer», New York: Hungry Minds, 2002.

Gadecki, C., and Heckert, C.: «ATM for Dummies», New York: Hungry Minds, 1997.

Gareer, L.: «Will 3G Really Be the Next Big Wireless Technology?», Computer, vol. 35, pp. 26–32, Jan. 2002.

Garfinkel, S., with Spafford, G.: «Web Security, Privacy, and Commerce», Sebastopol, CA: O'Reilly, 2002.

Geier, J.: «Wireless LANs», 2nd ed., Indianapolis, IN: Sams, 2002.

Gevros, P., Crowcroft, J., Kirstein, P., and Bhatti, S.: «Congestion Control Mechanisms and the Best Effort Service Model», IEEE Network Magazine, vol. 15, pp. 16–25, May-June 2001.

Ghani, N., and Dixit, S.: «TCP/IP Enhancements for Satellite Networks», IEEE Commun. Magazine, vol. 37, pp. 64–72, 1999.

Ginsburg, D.: «ATM: Solutions for Enterprise Networking», Boston: Addison-Wesley, 1996.

Goodman, D. J.: «The Wireless Internet: Promises and Challenges», Computer, vol. 33, pp. 36–41, July 2000.

Goralski, W. J.: «Optical Networking and WDM», New York: McGraw-Hill, 2001.

Goralski, W. J.: «SONET», 2nd ed., New York: McGraw-Hill, 2000.

Goralski, W. J.: «Introduction to ATM Networking», New York: McGraw-Hill, 1995.

Gossain, H., De Morais Cordeiro, and Agrawal, D. P.: «Multicast: Wired to Wireless», IEEE Commun. Mag., vol. 40, pp. 116–123, June 2002.

Gravano, S.: «Introduction to Error Control fades», Oxford, U.K.: Oxford University Press, 2001.

Guo, Y., and Chaskar, H.: «Class-Based Quality of Service over Air Interfaces in 4G Mobile Networks», IEEE Commun. Magazine, vol. 40, pp. 132–137, March 2002.

Haartsen, J.: «The Bluetooth Radio System», IEEE Personal Commun. Magazine, vol. 7, pp. 28–36, Feb. 2000.

Hac, A.: «Wireless and Cellular Architecture and Services», IEEE Commun. Magazine, vol. 33, pp. 98–104, Nov. 1995.

Hac, A., and Guo, L.: «A Scalable Mobile Host Protocol for the Internet», Int'l J. of Network Mgmt, vol. 10, pp. 115–134, May-June, 2000.

Hall, E., and Cerf, V.: «Internet Core Protocols: The Definitive Guide», Sebastopol, CA: O'Reilly, 2000.

Hamming, R. W.: «Error Detecting and Error Correcting Codes», Bell System Tech. J., vol. 29, pp. 147–160, April 1950.

Hanegan, K.: «Custom CGI Scripting with Perl», New York: Wiley, 2001.

Harris, A.: «JavaScript Programming for the Absolute Beginner», Premier Press, 2001.

Harte, L., Kellogg, S., Dreher, R., and Schaffnit, T.: «The Comprehensive Guide to Wireless Technology», Fuquay-Varina, NC: APDG Publishing, 2000.

Harte, L., Levine, R., and Kikta, R.: «3G Wireless Demystified», New York: McGraw-Hill, 2002.

Hawley, G. T.: «Historical Perspectives on the U.S. Telephone System», IEEE Commun. Magazine, vol. 29, pp. 24–28, March 1991.

Hecht, J.: «Understanding Fiber Optics», Upper Saddle River, NJ: Prentice Hall, 2001.

Heegard, C., Coffey, J. T., Gummadi, S., Murphy, P. A., Provencio, R., Rossin, E. J., Schrum, S., and Shoemaker, M. B.: «High-Performance Wireless Ethernet», IEEE Commun. Magazine, vol. 39, pp. 64–73, Nov. 2001.

Held, G.: «The Complete Modem Reference», 2nd ed., New York: Wiley, 1994.

Hellman, M. E.: «A Cryptanalytic Time-Memory Tradeoff», IEEE Trans, on Information Theory, vol. IT-26, pp. 401–406, July 1980.

Hills, A.: «Large-Scale Wireless LAN Design», IEEE Commun. Magazine, vol. 39, pp. 98–104, Nov. 2001.

Holzmann, G.J.: «Design and Validation of Computer Protocols», Englewood Cliffs, NJ: Prentice Hall, 1991.

Hu, Y., and Li, V. O. K.: «Satellite-Based Internet Access», IEEE Commun. Magazine, vol. 39, pp. 155–162, March 2001.

Hu, Y.-C., and Johnson, D. B.: «Implicit Source Routes for On-Demand Ad Hoc Network Routing», Proc. ACM Int'l Symp. on Mobile Ad Hoc Networking & Computing, ACM, pp. 1–10, 2001.

Huang, V., and Zhuang, W.: «QoS-Oriented Access Control for 4G Mobile Multimedia CDMA Communications», IEEE Commun. Magazine, vol. 40, pp. 118–125, March 2002.

Huber, J. F., Weiler, D., and Brand, H.: «UMTS, the Mobile Multimedia Vision for IMT-2000: A Focus on Standardization», IEEE Commun. Magazine, vol. 38, pp. 129–136, Sept. 2000.

Hui, J.: «A Broadband Packet Switch for Multi-rate Services», Proc. Int'l Conf. on Com-mun., IEEE, pp. 782–788, 1987.

Huitema, C.: «Routing in the Internet, Englewood Cliffs», NJ: Prentice Hall, 1995.

Hull, S.: «Content Delivery Networks», Berkeley, CA: Osborne/McGraw-Hill, 2002.

Humblet, P. A., Ramaswami, R., and Sivarajan, K. N.: «An Efficient Communication Protocol for High-Speed Packet-Switched Multichannel Networks», Proc. SIGCOMM '92 Conf., ACM, pp. 2–13, 1992.

Hunter, D. K., and Andonovic, I.: «Approaches to Optical Internet Packet Switching», IEEE Commun. Magazine, vol. 38, pp. 116–122, Sept. 2000.

Huston, G.: «TCP in a Wireless World», IEEE Internet Computing, vol. 5, pp. 82–84, March-April, 2001.

Ibe, O. C.: «Essentials of ATM Networks and Services», Boston: Addison-Wesley, 1997.

Irmer, T.: «Shaping Future Telecommunications: The Challenge of Global Standard-ization», IEEE Commun. Magazine, vol. 32, pp. 20–28, Jan. 1994.

Izzo, P.: «Gigabit Networks», New York: Wiley, 2000.

Jacobson, V.: «Congestion Avoidance and Control», Proc. SIGCOMM '88 Conf., ACM, pp. 314–329, 1988.

Jain, R.: «Congestion Control and Traffic Management in ATM Networks: Recent Advances and a Survey», Computer Networks and ISDN Systems, vol. 27, Nov. 1995.

Jain, R.: «FDDI Handbook — High-Speed Networking Using Fiber and Other Me-dia», Boston: Addison-Wesley, 1994.

Jain, R.: «Congestion Control in Computer Networks: Issues and Trends», IEEE Net-work Magazine, vol. 4, pp. 24–30, May-June 1990.

Jakobsson, M., and Wetzel, S.: «Security Weaknesses in Bluetooth», Topics in Cryp-tology: CT-RSA 2001, Berlin: Springer-Verlag LNCS 2020, pp. 176–191, 2001.

Joel, A.: «Telecommunications and the IEEE Communications Society», IEEE Commun. Magazine, 50th Anniversary Issue, pp. 6–14 and 162–167, May 2002.

Johansson, P., Kazantzidis, M., Kapoor, R., and Gerla, M.: «Bluetooth: An Enabler for Personal Area Networking», IEEE Network Magazine, vol. 15, pp. 28–37, Sept.-Oct. 2001.

Johnson, D. B.: «Scalable Support for Transparent Mobile Host Internetworking», Wireless Networks, vol. 1, pp. 311–321, Oct. 1995.

Johnson, H. W.: «Fast Ethernet — Dawn of a New Network», Englewood Cliffs, NJ: Prentice Hall, 1996.

Johnson, N. F., and Jajoda, S.: «Exploring Steganography: Seeing the Unseen», Com-puter, vol. 31, pp. 26–34, Feb. 1998.

Kahn, D.: «Cryptology Goes Public», IEEE Commun. Magazine, vol. 18, pp. 19–28, March 1980.

Kahn, D.: «The Codebreakers», 2nd ed., New York: Macmillan, 1995.

Kamoun, F., and Kleinrock, L.: «Stochastic Performance Evaluation of Hierarchical Routing for Large Networks», Computer Networks, vol. 3, pp. 337–353, Nov. 1979.

Kapp, S.: «802.11: Leaving the Wire Behind», IEEE Internet Computing, vol. 6, pp. 82–85, Jan.-Feb. 2002.

Karn, P.: «MACA — A New Channel Access Protocol for Packet Radio», ARRL/ CRRL Amateur Radio Ninth Computer Networking Conf., pp. 134–140, 1990.

Kartalopoulos, S.: «Introduction to DWDM Technology: Data in a Rainbow», New York, NY: IEEE Communications Society, 1999.

Kasera, S. K., Hjalmtysson, G., Towlsey, D. F., and Kurose, J. F.: «Scalable Reliable Multicast Using Multiple Multicast Channels», IEEE/ACM Trans, on Net-working, vol. 8, pp. 294–310, 2000.

Katz, D., and Ford, P. S.: «TUBA: Replacing IP with CLNP», IEEE Network Maga-zine, vol. 7, pp. 38–47, May-June 1993.

Katzenbeisser, S., and Petitcolas, F. A. P.: «Information Hiding Techniques for Steganography and Digital Watermarking», London, Artech House, 2000.

Kaufman, C., Perlman, R., and Speciner, M.: «Network Security», 2nd ed., Engle-wood Cliffs, NJ: Prentice Hall, 2002.

Kellerer, W., Vogel, H.-J., and Steinberg, K.-E.: «A Communication Gateway for Infra-structure-Independent 4G Wireless Access», IEEE Commun. Magazine, vol. 40, pp. 126–131, March 2002.

Kerckhoff, A.: «La Cryptographic Militaire», J. des Sciences Militaires, vol. 9, pp. 5–38, Jan. 1883 and pp. 161–191, Feb. 1883.

Kim, J. B., Suda, T., and Yoshimura, M.: «International Standardization of B-ISDN», Computer Networks and ISDN Systems, vol. 27, pp. 5–27, Oct. 1994.

Kipnis, J.: «Beating the System: Abuses of the Standards Adoptions Process», IEEE Commun. Magazine, vol. 38, pp. 102–105, July 2000.

Kleinrock, L.: «On Some Principles of Nomadic Computing and Multi-Access Com-munications», IEEE Commun. Magazine, vol. 38, pp. 46–50, July 2000.

Kleinrock, L., and Tobagi, F.: «Random Access Techniques for Data Transmission over Packet-Switched Radio Channels», Proc. Nat. Computer Conf., pp. 187–201, 1975.

Krishnamurthy, B., and Rexford, J.: «Web Protocols and Practice», Boston: Addison-Wesley, 2001.

Kumar, V., Korpi, M., and Sengodan, S.: «IP Telephony with H.323», New York: Wiley, 2001.

Kurose, J. F., and Ross, K. W.: «Computer Networking: A Top-Down Approach Fea-turing the Internet», Boston: Addison-Wesley, 2001.

Kwok, T.: «A Vision for Residential Broadband Service: ATM to the Home», IEEE Network Magazine, vol. 9, pp. 14–28, Sept.-Oct. 1995.

Kyas, O., and Crawford, G.: «ATM Networks», Upper Saddle River, NJ: Prentice Hall, 2002.

Lam, C. K. M., and Tan, B. C. Y.: «The Internet Is Changing the Music Industry», Commun. of the ACM, vol. 44, pp. 62–66, Aug. 2001.

Lansford, J., Stephens, A, and Nevo, R.: «Wi-Fi (802.1 Ib) and Bluetooth: Enabling Coexistence», IEEE Network Magazine, vol. 15, pp. 20–27, Sept.-Oct. 2001.

Lash, D. A.: «The Web Wizard's Guide to Perl and CGI», Boston: Addison-Wesley, 2002.

Laubach, M. E., Farber, D. J., and Dukes, S. D.: «Delivering Internet Connections over Cable», New York: Wiley, 2001.

Lee, J. S., and Miller, L. E.: «CDMA Systems Engineering Handbook», London: Artech House, 1998.

Leeper, D. G.: «A Long-Term View of Short-Range Wireless», Computer, vol. 34, pp. 39–44, June 2001.

Leiner, B. M., Cole, R., Postel, J., and Mills, D.: «The DARPA Internet Protocol Suite», IEEE Commun. Magazine, vol. 23, pp. 29–34, March 1985.

Levine, D. A., and Akyildiz, I. A.: «PROTON: A Media Access Control Protocol for Optical Networks with Star Topology», IEEE/ACM Trans, on Networking, vol. 3, pp. 158–168, April 1995.

Levy, S.: «Crypto Rebels», Wired, pp. 54–61, May-June 1993.

Li, J., Blake, C., De Couto, D. S. J., Lee, H. I., and Morris, R.: «Capacity of Ad Hoc Wireless Networks», Proc. 7th Int'l Conf. on Mobile Computing and Networking, ACM, pp. 61–69, 2001.

Lin, F., Chu, P., and Liu, M.: «Protocol Verification Using Reachability Analysis: The State Space Explosion Problem and Relief Strategies», Proc. SIGCOMM '87 Conf., ACM, pp. 126–135, 1987.

Lin, Y.-D., Hsu, N.-B., and Hwang, R.-H.: «QoS Routing Granularity in MPLS Networks», IEEE Commun. Magazine, vol. 40, pp. 58–65, June 2002.

Listani, M., Eramo, V., and Sabella, R.: «Architectural and Technological Issues for Future Optical Internet Networks», IEEE Commun. Magazine, vol. 38, pp. 82–92, Sept. 2000.

Liu, C. L., and Layland, J. W.: «Scheduling Algorithms for Multiprogramming in a Hard Real-Time Environment», Journal of the ACM, vol. 20, pp. 46–61, Jan. 1973.

Metcalfe, R. M., and Boggs, D. R.: «Ethernet: Distributed Packet Switching for Local Computer Networks», Commun. of the ACM, vol. 19, pp. 395–404, July 1976.

Metz, C.: «Interconnecting ISP Networks», IEEE Internet Computing, vol. 5, pp. 74–80, March-April 2001.

Metz, C.: «Differentiated Services», IEEE Multimedia Magazine, vol. 7, pp. 84–90, July-Sept. 2000.

Metz, C.: «IP Routers: New Tool for Gigabit Networking», IEEE Internet Computing, vol. 2, pp. 14–18, Nov.-Dec. 1998.

Miller, B. A., and Bisdikian, C.: «Bluetooth Revealed», Upper Saddle River, NJ: Prentice Hall, 2001.

Miller, P., and Cummins, M.: «LAN Technologies Explained», Woburn, MA: Butterworth-Heinemann, 2000.

Minoli, D.: «Video Dialtone Technology», New York: McGraw-Hill, 1995.

Minoli, D., and Vitella, M.: «ATM & Cell Relay for Corporate Environments», New York: McGraw-Hill, 1994.

Mishra, P. P., and Kanakia, H.: «A Hop by Hop Rate-Based Congestion Control Scheme», Proc. SIGCOMM '92 Conf., ACM, pp. 112–123, 1992.

Misra, A., Das, S., Dutta, A., McAuley, A., and DAS, S.: «IDMP-Based Fast Hand-offs and Paging in IP-Based 4G Mobile Networks», IEEE Commun. Magazine, vol. 40, pp. 138–145, March 2002.

Mogul, J. C.: «IP Network Performance», in Internet System Handbook, Lynch, D.C. and Rose, M.T. (eds.), Boston: Addison-Wesley, pp. 575–675, 1993.

Mok, A. K., and Ward, S. A.: «Distributed Broadcast Channel Access», Computer Networks, vol. 3, pp. 327–335, Nov. 1979.

Moy, J.: «Multicast Routing Extensions», Commun. of the ACM, vol. 37, pp. 61–66, Aug. 1994.

Mullins, J.: «Making Unbreakable Code», IEEE Spectrum, pp. 40–45, May 2002.

Nagle, J.: «On Packet Switches with Infinite Storage», IEEE Trans, on Commun., vol. COM-35, pp. 435–438, April 1987. '

Nagle, J.: «Congestion Control in TCP/IP Internetworks», Computer Commun. Rev., vol. 14, pp. 11–17, Oct. 1984.

Narayanaswami, C., Kamijoh, N., Raghunath, M., Inoue, T., Cipolla, T., Sanford, J., Schlig, E., Ventkiteswaran, S., Guniguntala, D., Kulkarni, V., and Yamazaki, K.: «IBM's Linux Watch: The Challenge of Miniaturization», Computer, vol. 35, pp. 33–41, Jan. 2002.

Naughton, J.: «A Brief History of the Future», Woodstock, NY: Overlook Press, 2000.

Needham, R. M., and Schroeder, M. D.: «Authentication Revisited», Operating Systems Rev., vol. 21, p. 7, Jan. 1987.

Needham, R. M., and Schroeder, M. D.: «Using Encryption for Authentication in Large Networks of Computers», Commun. of the ACM, vol. 21, pp. 993–999, Dec. 1978.

Nelakuditi, S., and Zhang, Z.-L.: «A Localized Adaptive Proportioning Approach to QoS Routing», IEEE Commun. Magazine vol. 40, pp. 66–71, June 2002.

Nemeth, E., Snyder, G., Seebass, S., and Hein, T. R.: «UNIX System Administration Handbook», 3rd ed., Englewood Cliffs, NJ: Prentice Hall, 2000.

Nichols, R. K., and Lekkas, P. C.: «Wireless Security», New York: McGraw-Hill, 2002.

Nist: «Secure Hash Algorithm», U.S. Government Federal Information Processing Standardise, 1993.

O'Hara, B., and Petrick, A.: «802.11 Handbook: A Designer's Companion», New York: IEEE Press, 1999.

Otway, D., and Rees, O.: «Efficient and Timely Mutual Authentication», Operating Systems Rev., pp. 8–10, Jan. 1987.

Ovadia, S.: «Broadband Cable TV Access Networks: from Technologies to Applications», Upper Saddle River, NJ: Prentice Hall, 2001.

Palais, J. C.: «Fiber Optic Commun.», 3rd ed., Englewood Cliffs, NJ: Prentice Hall, 1992.

Pan, D.: «A Tutorial on MPEG/Audio Compression», IEEE Multimedia Magazine, vol. 2, pp.60–74, Summer 1995.

Pandya, R.: «Emerging Mobile and Personal Communication Systems», IEEE Commun. Magazine, vol. 33, pp. 44–52, June 1995.

Parameswaran, M., Susarla, A., and Whinston, A. B.: «P2P Networking: An Information-Sharing Alternative», Computer, vol. 34, pp. 31–38, July 2001.

Park, J. S., and Sandhu, R.: «Secure Cookies on the Web», IEEE Internet Computing, vol. 4, pp. 36–44, July-Aug. 2000.

Partridge, C., Hughes, J., and Stone, J.: «Performance of Checksums and CRCs over Real Data», Proc. SIGCOMM '95 Conf., ACM, pp. 68–76, 1995.

Paxson, V.: «Growth Trends in Wide-Area TCP Connections», IEEE Network Magazine, vol. 8, pp. 8–17, July-Aug. 1994.

Paxson, V., and Floyd, S.: «Wide-Area Traffic: The Failure of Poisson Modeling», Proc. SIGCOMM '94 Conf., ACM, pp. 257–268, 1995.

Pepelnjak, I., and Guichard, J.: «MPLS and VPN Architectures», Indianapolis, IN: Cisco Press, 2001.

Perkins, C. E.: «RTP: Audio and Video for the Internet», Boston: Addison-Wesley, 2002. *Perkins, C. E. (ed.)*: «Ad Hoc Networking», Boston: Addison-Wesley, 2001.

Perkins, C. E.: «Mobile IP Design Principles and Practices», Upper Saddle River, NJ: Prentice Hall, 1998a.

Perkins, C. E.: «Mobile Networking in the Internet», Mobile Networks and Applications, vol. 3, pp. 319–334, 1998b.

Perkins, C. E.: «Mobile Networking through Mobile IP», IEEE Internet Computing, vol. 2, pp. 58–69, Jan.-Feb. 1998c.

Perkins, C. E., and Royer, E.: «The Ad Hoc On-Demand Distance-Vector Protocol», in Ad Hoc Networking, edited by C. Perkins, Boston: Addison-Wesley, 2001.

Perkins, C. E., and Royer, E.: «Ad-hoc On-Demand Distance Vector Routing», Proc. Second Ann. IEEE Workshop on Mobile Computing Systems and Applications, IEEE, pp. 90–100, 1999.

Perlman, R.: «Interconnections», 2nd ed., Boston: Addison-Wesley, 2000.

Perlman, R.: «Network Layer Protocols with Byzantine Robustness», Ph.D. thesis, M.I.T., 1988.

Perlman, R., and Kaufman, C.: «Key Exchange in IPsec», IEEE Internet Computing, vol. 4, pp. 50–56, Nov.-Dec. 2000.

Peterson, L. L., and Da Vie, B. S.: «Computer Networks: A Systems Approach», San Francisco: Morgan Kaufmann, 2000.

Peterson, W. W., and Brown, D. T.: «Cyclic Codes for Error Detection», Proc. IRE, vol. 49, pp. 228–235, Jan. 1961.

Pickholtz, R. L., Schilling, D. L., and Milstein, L. B.: «Theory of Spread Spectrum Communication — A Tutorial», IEEE Trans, on Commun., vol. COM-30, pp. 855–884, May 1982.

Pierre, G., Kuz, I., van Steen, M., Tanenbaum, A. S.: «Differentiated Strategies for Replicating Web Documents», Computer Commun., vol. 24, pp. 232–240, Feb. 2001.

Pierre, G., van Steen, M., and Tanenbaum, A. S.: «Dynamically Selecting Optimal Distribution Strategies for Web Documents», IEEE Trans, on Computers, vol. 51, June 2002.

Piscitello, D. M., and Chapin, A. L.: «Open Systems Networking: TCP/IP and OSI», Boston: Addison-Wesley, 1993.

Pitt, D. A.: «Bridging — The Double Standard», IEEE Network Magazine, vol. 2, pp. 94–95, Jan. 1988.

Piva, A., Bartolini, F., and Barni, M.: «Managing Copyrights in Open Networks», IEEE Internet Computing, vol. 6, pp. 18–26, May-June 2002.

Pohlmann, N.: «Firewall Systems», Bonn, Germany: MITP-Verlag, 2001.

Puzmanova, R.: «Routing and Switching: Time of Convergence?», London: Addison-Wesley, 2002.

Rabinovich, M., and Spatscheck, O.: «Web Caching and Replication», Boston: Addison-Wesley, 2002.

Raju, J., and Garcia-Luna-Aceves, J. J.: «Scenario-based Comparison of Source-Tracing and Dynamic Source Routing Protocols for Ad-Hoc Networks», ACM Computer Communications Review, vol. 31, October 2001.

Ramanathan, R., and Redi, J.: «A Brief Overview of Ad Hoc Networks: Challenges and Directions», IEEE Communf Magazine, 50th Anniversary Issue, pp. 20–22, May 2002.

Ratnasamy, S., Francis, P., Handley, M., Karp, R., and Shenker, S.: «A Scalable Content-Addressable Network», Proc. SIGCOMM '01 Conf., ACM, pp. 161–172, 2001.

Rivest, R. L.: «The MD5 Message-Digest Algorithm», RFC 1320, April 1992.

Rivest, R. L., and Shamir, A.: «How to Expose an Eavesdropper», Commun. of the ACM, vol. 27, pp. 393–395, April 1984.

Rivest, R. L., Shamir, A., and Adleman, L.: «On a Method for Obtaining Digital Signatures and Public Key Cryptosystems», Commun. of the ACM, vol. 21, pp. 120–126, Feb. 1978.

Roberts, L. G.: «Dynamic Allocation of Satellite Capacity through Packet Reservation», Proc. NCC, AFIPS, pp. 711–716, 1973.

Roberts, L. G.: «Extensions of Packet Communication Technology to a Hand Held Personal Terminal», Proc. Spring Joint Computer Conference, AFIPS, pp. 295–298, 1972.

Roberts, L. G.: «Multiple Computer Networks and Intercomputer Communication», Proc. First Symp. on Operating Systems Prin., ACM, 1967.

Rose, M. T.: «The Simple Book», Englewood Cliffs, NJ: Prentice Hall, 1994.

Rose, M. T.: «The Internet Message», Englewood Cliffs, NJ: Prentice Hall, 1993.

Rose, M. T., and McCloghrie, K.: «How to Manage Your Network Using SNMP», Englewood Cliffs, NJ: Prentice Hall, 1995.

Rowstron, A., and Druschel, P.: «Storage Management and Caching in PAST, a Large-Scale, Persistent Peer-to-Peer Storage Utility», Proc. 18th Symp. on Operating Systems Prin., ACM, pp. 188–201, 2001a.

Rowstron, A., and Druschel, P.: «Pastry: Scalable, Distributed Object Location and Routing for Large-Scale Peer-to-Peer Storage Utility», Proc. 18th Int'l Conf. on Distributed Systems Platforms, ACM/1FIP, 2001b.

Royer, E. M., and Toh, C.-K.: «A Review of Current Routing Protocols for Ad-Hoc Mobile Wireless Networks», IEEE Personal Commun. Magazine, vol. 6, pp. 46–55, April 1999.

Ruiz-Sanchez, M. A., Biersack, E. W., and Dabbous, W.: «Survey and Taxonomy of IP Address Lookup Algorithms», IEEE Network Magazine, vol. 15, pp. 8–23, March-April 2001.

Sairam, K. V. S. S. S. S., Gunasekaran, N., and Reddy, S. R.: «Bluetooth in Wireless Communication», IEEE Commun. Mag., vol. 40, pp. 90–96, June 2002.

Saltzer, J. H., Reed, D. P., and Clark, D. D.: «End-to-End Arguments in System Design», ACM Trans, on Computer Systems, vol. 2, pp. 277–288, Nov. 1984.

Sanderson, D. W., and Dougherty, D.: «Smileys», Sebastopol, CA: O'Reilly, 1993.

Sari, H., Vanhaverbeke, F., and Moeneclaey, M.: «Extending the Capacity of Multiple Access Channels», IEEE Commun. Magazine, vol. 38, pp. 74–82, Jan. 2000.

Sarikaya, B.: «Packet Mode in Wireless Networks: Overview of Transition to Third Generation», IEEE Commun. Magazine, vol. 38, pp. 164–172, Sept. 2000.

Schneier, B.: «Secrets and Lies», New York: Wiley, 2000.

Schneier, B.: «Applied Cryptography», 2nd ed., New York: Wiley, 1996.

Schneier, B.: «E-Mail Security», New York: Wiley, 1995.

Schneier, B.: «Description of a New Variable-Length Key, 64-Bit Block Cipher [Blowfish]», Proc. of the Cambridge Security Workshop, Berlin: Springer-Verlag LNCS809, pp. 191–204, 1994.

Schnorr, C. P.: «Efficient Signature Generation for Smart Cards», Journal of Cryptology, vol. 4, pp. 161–174, 1991.

Scholtz, R. A.: «The Origins of Spread-Spectrum Communications», IEEE Trans, on Commun., vol. COM-30, pp. 822–854, May 1982.

Scott, R.: «Wide Open Encryption Design Offers Flexible Implementations», Cryptologia, vol. 9, pp. 75–90, Jan. 1985.

Seifert, R.: «The Switch Book», Boston: Addison-Wesley, 2000.

Seifert, R.: «Gigabit Ethernet», Boston: Addison-Wesley, 1998.

Senn, J. A.: «The Emergence of M-Commerce», Computer, vol. 33, pp. 148–150, Dec. 2000.

Serjantov, A.: «Anonymizing Censorship'Resistant Systems», Proc. First Int'l Workshop on Peer-to-Peer Systems, Berlin: Springer-Verlag LNCS, 2002.

Severance, C.: «IEEE 802.11: Wireless Is Coming Home», Computer, vol. 32, pp. 126–127, Nov. 1999.

Shahabi, C., Zimmermann, R., Fu, K., and Yao, S.-Y. D.: «YIMA: A Second-Generation Continuous Media Server», Computer, vol. 35, pp. 56–64, June 2002.

Shannon, C.: «A Mathematical Theory of Communication», Bell System Tech. J., vol. 27, pp. 379–423, July 1948; and pp. 623–656, Oct. 1948.

Shepard, S.: «SONET/SDH Demystified», New York: McGraw-Hill, 2001.

Shreedhar, M., and Varghese, G.: «Efficient Fair Queueing Using Deficit Round Robin», Proc. SIGCOMM '95 Conf., ACM, pp. 231–243, 1995.

Skoudis, E.: «Counter Hack, Upper Saddle River», NJ: Prentice Hall, 2002.

Smith, O. K., and Alexander, R. C.: «Fumbling the Future», New York: William Morrow, 1988.

Smith, R. W.: «Broadband Internet Connections», Boston: Addison Wesley, 2002.

Snoeren, A. C., and Balakrishnan, H.: «An End-to-End Approach to Host Mobility», Intel Conf. on Mobile Computing and Networking, ACM, pp. 155–166, 2000.

Sobel, D. L.: «Will Carnivore Devour Online Privacy», Computer, vol. 34, pp. 87–88, May 2001.

Solomon, J. D.: «Mobile IP: The Internet Unplugged», Upper Saddle River, NJ: Prentice Hall, 1998.

Spohn, M., and Garcia-Luna-Aceves, J. J.: «Neighborhood Aware Source Routing», Proc. ACMMobiHoc 2001, ACM, p. 2001.

Spurgeon, C. E.: «Ethernet: The Definitive Guide», Sebastopol, CA: O'Reilly, 2000.

Stallings, W.: «Data and Computer Communications», 6th ed., Upper Saddle River, NJ: Prentice Hall, 2000.

Steinmetz, R., and Nahrstedt, K.: «Multimedia Fundamentals». Vol. 1: «Media Coding and Content Processing», Upper Saddle River, NJ: Prentice Hall, 2002.

Steinmetz, R., and Nahrstedt, K.: «Multimedia Fundamentals». Vol. 2: «Media Processing and Communications», Upper Saddle River, NJ: Prentice Hall, 2003a.

Steinmetz, R., and Nahrstedt, K.: «Multimedia Fundamentals». Vol. 3: «Documents, Security, and Applications», Upper Saddle River, NJ: Prentice Hall, 2003b.

Steiner, J. G., Neuman, B. C., and Schiller, J. I.: «Kerberos: An Authentication Service for Open Network Systems», Proc. Winter USENIX Conf., USENIX, pp. 191–201, 1988.

Stevens, W. R.: «UNIX Network Programming», Volume 1: «Networking APIs — Sockets and XTI», Upper Saddle River, NJ: Prentice Hall, 1997.

Stevens, W. R.: «TCP/IP Illustrated», Boston: Addison-Wesley, 1994.

Stewart, R., and Metz, C.: «SCTP: New Transport Protocol for TCP/IP», IEEE Internet Computing, vol. 5, pp. 64–69, Nov.-Dec. 2001.

Stinson, D. R.: «Cryptography Theory and Practice», 2nd ed., Boca Raton, FL: CRC Press, 2002.

Stoica, I., Morris, R., Karger, D., Kaashoek, M. F., and Balakrishnan, H.: «Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications», Proc. SIGCOMM '01 Conf., ACM, pp. 149–160, 2001.

Striegel, A., and Manimaran, G.: «A Survey of QoS Multicasting Issues», IEEE Commun. Mag., vol. 40, pp. 82–87, June 2002.

Stubblefield, A., Ioannidis, J., and Rubin, A. D.: «Using the Fluhrer, Mantin, and Shamir Attack to Break WEP», Proc Network and Distributed Systems Security Symp., SOC, pp. 1–11, 2002.

Summers, C. K.: «ADSL: Standards, Implementation, and Architecture», Boca Raton, FL: CRC Press, 1999.

Sunshine, C. A., and Dalal, Y. K.: «Connection Management in Transport Protocols», Computer Networks, vol. 2, pp. 454–473, 1978.

Tanenbaum, A. S.: «Modern Operating Systems», Upper Saddle River, NJ: Prentice Hall, 2001.

Tanenbaum, A. S., and van Steen, M.: «Distributed Systems: Principles and Paradigms», Upper Saddle River, NJ: Prentice Hall, 2002.

Teger, S., and Waks, D. J.: «End-User Perspectives on Home Networking», IEEE Commun. Magazine, vol. 40, pp. 114–119, April 2002.

Thyagarajan, A. S., and Deering, S. E.: «Hierarchical Distance-Vector Multicast Routing for the MBone», Proc. SIGCOMM '95 Conf., ACM, pp. 60–66, 1995.

Tittel, E., Valentine, C., Burmeister, M., and Dykes, L.: «Mastering XHTML», Alameda, CA: Sybex, 2001.

Tokoro, M., and Tamaru, K.: «Acknowledging Ethernet», Compcon, IEEE, pp. 320–325, Fall 1977.

Tomlinson, R. S.: «Selecting Sequence Numbers», Proc. SIGCOMM/SIGOPS Interprocess Commun. Workshop, ACM, pp. 11–23, 1975.

Tseng, Y.-C., Wu, S.-L., Liao, W.-H., and Chad, C.-M.: «Location Awareness in Ad Hoc Wireless Mobile Networks», Computer, vol. 34, pp. 46–51, 2001.

Tuchman, W.: «Hellman Presents No Shortcut Solutions to DES», IEEE Spectrum, vol. 16, pp. 40–41, July 1979.

Turner, J. S.: «New Directions in Communications (or Which Way to the Information Age)», IEEE Commun. Magazine, vol. 24, pp. 8–15, Oct. 1986.

Vacca, J. R.: «I-Mode Crash Course», New York: McGraw-Hill, 2002.

Valade, J.: «PHP & MySQL for Dummies», New York: Hungry Minds, 2002.

Varghese, G., and Lauck, T.: «Hashed and Hierarchical Timing Wheels: Data Structures for the Efficient Implementation of a Timer Facility», Proc. llth Symp. on Operating Systems Prin., ACM, pp. 25–38, 1987.

Varshney, U., Snow, A., McGivern, M., and Howard, C.: «Voice over IP», Commun. of the ACM, vol. 45, pp. 89–96, 2002.

Varshney, U., and Vetter, R.: «Emerging Mobile and Wireless Networks», Commun. of the ACM, vol. 43, pp. 73–81, June 2000.

Vetter, P., Goderis, D., Verpooten, L., and Granger, A.: «Systems Aspects of APON/VDSL Deployment», IEEE Commun. Magazine, vol. 38, pp. 66–72, May 2000.

Waddington, D. G., and Chang, F.: «Realizing the Transition to IPv6», IEEE Commun. Mag., vol. 40, pp. 138–148, July 2002.

Waldman, M., Rubin, A. D., and Cranor, L. F.: «Publius: A Robust, Tamper-Evident, Censorship-Resistant, Web Publishing System», Proc. Ninth USENIX Security Symp., USENIX, pp. 59–72, 2000.

Wang, Y., and Chen, W.: «Supporting IP Multicast for Mobile Hosts», Mobile Networks and Applications, vol. 6, pp. 57–66, Jan.-Feb. 2001.

Wang, Z.: «Internet QoS», San Francisco: Morgan Kaufmann, 2001.

Warneke, B., Last, M., Liebowitz, B., and Pister, K. S. J.: «Smart Dust: Communicating with a Cubic Millimeter Computer», Computer, vol. 34, pp. 44–51, Jan. 2001.

Wayner, P.: «Disappearing Cryptography: Information Hiding, Steganography, and Watermarking», 2nd ed., San Francisco: Morgan Kaufmann, 2002.

Webb, W.: «Broadband Fixed Wireless Access as a Key Component of the Future Integrated Communications Environment», IEEE Commun. Magazine, vol. 39, pp. 115–121, Sept. 2001.

Weiser, M.: «Whatever Happened to the Next Generation Internet?», Commun. of the ACM, vol. 44, pp. 61–68, Sept. 2001.

Weltman, R., and Dahbura, T.: «LDAP Programming with Java», Boston: Addison-Wesley, 2000.

Wessels, D.: «Web Caching», Sebastopol, CA: O'Reilly, 2001.

Wetteroth, D.: «OSI Reference Model for Telecommunications», New York: McGraw-Hill, 2001.

Wiljakka, J.: «Transition to IPv6 in GPRS and WCDMA Mobile Networks», IEEE Commun. Magazine, vol. 40, pp. 134–140, April 2002.

Williamson, H.: «XML: The Complete Reference», New York: McGraw-Hill, 2001.

Willinger, W., Taqqu, M. S., Sherman, R., and Wilson, D. V.: «Self-Similarity through High Variability: Statistical Analysis of Ethernet LAN Traffic at the Source Level», Proc. SIGCOMM '95 Conf., ACM, pp. 100–113, 1995.

Wright, D. J.: «Voice over Packet Networks», New York: Wiley, 2001.

Wylie, J., Bigrigg, M. W., Strunk, J. D., Ganger, G. R., Kiliccote, H., and Khosla, P. K.: «Survivable Information Storage Systems», Computer, vol. 33, pp. 61–68, Aug. 2000.

Xylomenos, G., Polyzos, G. C., Mahonen, P., and Saaranen, M.: «TCP Performance Issues over Wireless Links», IEEE Commun. Magazine, vol. 39, pp. 52–58, April 2001.

Yang, C.-Q., and Reddy, A. V. S.: «A Taxonomy for Congestion Control Algorithms in Packet Switching Networks», IEEE Network Magazine, vol. 9, pp. 34–45, July-Aug. 1995.

Yuval, G.: «How to Swindle Rabin», Cryptologia, vol. 3, pp. 187–190, July 1979.

Zacks, M.: «Antiterrorist Legislation Expands Electronic Snooping», IEEE Internet Computing, vol. 5, pp. 8–9, Nov.-Dec. 2001.

Zadeh, A. N., Jabbari, B., Pickholtz, R., and Vojcic, B.: «Self-Organizing Packet Radio Ad Hoc Networks with Overlay (SOPRANO)», IEEE Commun. Mag., vol. 40, pp. 149–157, June 2002.

Zhang, L.: «Comparison of Two Bridge Routing Approaches», IEEE Network Magazine, vol. 2, pp. 44–48, Jan.-Feb. 1988.

Zhang, L.: «RSVP: A New Resource ReSerVation Protocol», IEEE Network Magazine, vol. 7, pp. 8–18, Sept.-Oct. 1993.

Zhang, Y., and Ryu, B.: «Mobile and Multicast IP Services in PACS: System Architecture, Prototype, and Performance», Mobile Networks and Applications, vol. 6, pp. 81–94, Jan.-Feb. 2001.

Zimmermann, P. R.: «The Official PGP User's Guide», Cambridge, MA: M.I.T. Press, 1995a.

Zimmermann, P. R.: «PGP: Source Code and Internals», Cambridge, MA: M.I.T. Press, 1995b.

Zipf, G. K.: «Human Behavior and the Principle of Least Effort: An Introduction to Human Ecology», Boston: Addison-Wesley, 1949.

Ziv, J., and Lempel, Z.: «A Universal Algorithm for Sequential Data Compression», IEEE Trans, on Information Theory, vol. IT-23, pp. 337–343, May 1977.

Алфавитный указатель

C

D

E

F

G

H

I

J

K

Л

М

Н

О

П

P

C

Т

У

Ф

Х

Ц

Ч

Ш

Э

Я

Э. Таненбаум

Компьютерные сети

4-е издание

Перевел с английского В. Шрага

Главный редактор	*Е. Строганова*
Заведующий редакцией	*И. Корнеев*
Руководитель проекта	*А. Васильев*
Научный редактор	*С. Орлов*
Литературные редакторы	*Т. Маслова, В. Шрага*
Художник	*Н. Биржаков*
Иллюстрации	*В. Шендерова*
Корректор	*Н. Рощина*
Верстка	*С. Панич*

Лицензия ИД № 05784 от 07.09.01.
Подписано в печать 15.07.03. Формат 70×100/16. Усл. п. л. 79,98.
Тираж 4000 экз. Заказ № 258.
ООО «Питер Принт». 196105, Санкт-Петербург, ул. Благодатная, д. 67в.
Налоговая льгота – общероссийский классификатор продукции
ОК 005-93, том 2; 953005 – литература учебная.
Отпечатано с готовых диапозитивов в ФГУП «Печатный двор» им. А. М. Горького
Министерства РФ по делам печати, телерадиовещания и средств массовых коммуникаций.
197110, Санкт-Петербург, Чкаловский пр., 15.

COMPUTER NETWORKS

4th edition

Andrew S. Tanenbaum

PH PTR

Prentice Hall PTR
Upper Saddle River, New Jersey 07458
www.phptr.com

КЛАССИКА COMPUTER SCIENCE

Э. ТАНЕНБАУМ

КОМПЬЮТЕРНЫЕ СЕТИ

4-Е ИЗДАНИЕ

ПИТЕР®

Москва · Санкт-Петербург · Нижний Новгород · Воронеж
Ростов-на-Дону · Екатеринбург · Самара
Киев · Харьков · Минск
2003