Аутентификация, авторизация, аудит

БГА, РТФ
Кафедра ИБ

Зензин Александр Степанович, к.т.н.

Обзор

1. Понятие аутентификации
2. Авторизация доступа
3. Аудит
4. Строгая аутентификация на основе многоразового пароля в протоколе CHAP
5. Аутентификация на основе одноразового пароля
6. Аутентификация на основе сертификатов
7. Схема использования сертификатов
8. Сертифицирующие центры
9. Инфраструктура с открытым ключом
10. Аутентификация информации
11. Цифровая подпись
12. Аутентификация программных кодов

Понятие аутентификации

Аутентификация наряду с авторизацией (о которой рассказывается далее) представляет собой фундаментальный атрибут информационной безопасности.

Термин «аутентификация» (authentication) происходит от латинского слова authenticus, которое означает подлинный, достоверный, соответствующий самому себе. Аутентификация, или, другими словами, процедура установления подлинности, может быть применима как к людям, так и другим объектам, в частности к программам, устройствам, документам.

Аутентификация пользователя — *это процедура доказательства пользователем того, что он есть тот, за кого себя выдает*.

В частности, при выполнении логического входа в защищенную систему пользователь должен пройти процедуру аутентификации, то есть доказать, что именно ему принадлежит введенный им идентификатор (имя пользователя). Аутентификация предотвращает доступ к сети нежелательных лиц и разрешает вход для легальных пользователей.

В процедуре аутентификации участвуют две стороны: одна сторона доказывает свою аутентичность, предъявляя некоторые доказательства, другая сторона — аутентификатор — проверяет эти доказательства и принимает решение.

Понятие аутентификации

В качестве доказательства аутентичности применяются самые разнообразные приемы:

- аутентифицируемый может продемонстрировать знание некоего общего для обеих сторон секрета: слова (пароля) или факта (даты и места события, прозвища человека и т. п.);
- аутентифицируемый может продемонстрировать, что он владеет неким уникальным предметом (физическим ключом), в качестве которого может выступать, например, электронная магнитная карта;
- аутентифицируемый может доказать свою идентичность, используя собственные биохарактеристики: рисунок радужной оболочки глаза или отпечатки пальцев, которые предварительно были занесены в базу данных аутентификатора.

Сетевые службы аутентификации строятся на основе всех этих приемов, но чаще всего для доказательства идентичности пользователя применяют пароли. Простота и логическая ясность механизмов аутентификации на основе паролей в какой-то степени компенсирует известные слабости паролей. Это, во-первых, возможность раскрытия и разгадывания паролей, во-вторых, возможность «подслушивания» пароля путем анализа сетевого трафика. Для снижения уровня угрозы раскрытия паролей администраторы сети, как правило, применяют встроенные программные средства, служащие для формирования политики назначения и использования паролей: задание максимального и минимального сроков действия пароля, хранение списка уже использованных паролей, управление поведением системы после нескольких неудачных попыток логического входа и т. п.

Понятие аутентификации

ПРИМЕЧАНИЕ

Многие пользователи пренебрегают угрозами, которые несут в себе легко угадываемые пароли. Так, червь Mumu, поразивший компьютерные сети в 2003 году, искал свои жертвы, подбирая пароли из очень короткого списка: password, passwd, admin, pass, 123, 1234, 12345, 123456 и пустая строка. Такая на удивление примитивная стратегия дала прекрасные (с точки зрения атакующей стороны) результаты — множество компьютеров было взломано.

Легальность пользователя может устанавливаться по отношению к различным системам. Так, работая в сети, пользователь может проходить процедуру аутентификации и как локальный пользователь, который претендует на ресурсы только данного компьютера, и как пользователь сети, желающий получить доступ ко всем сетевым ресурсам. При локальной аутентификации пользователь вводит свои идентификатор и пароль, которые автономно обрабатываются операционной системой, установленной на данном компьютере. При логическом входе в сеть данные о пользователе (идентификатор и пароль) передаются на сервер, который хранит учетные записи всех пользователей сети. Однако такая упрощенная схема имеет большой изъян — при передаче пароля с клиентского компьютера на сервер, выполняющий процедуру аутентификации, этот пароль может быть перехвачен злоумышленником. Поэтому применяются разные приемы, чтобы избежать передачи пароля по сети в незащищенном виде.

Аутентификация, в процессе которой используются методы шифрования, а аугентификационная информация не передается по сети, называется **строгой**.

Понятие аутентификации

Многие приложения имеют собственные средства определения, является ли пользователь законным. И тогда пользователю приходится проходить дополнительные этапы проверки.

Как уже отмечалось, в качестве объектов, требующих аутентификации, могут выступать не только пользователи, но и различные приложения, устройства, текстовая и другая информация.

Так, пользователь, обращающийся с запросом к корпоративному веб-серверу, должен доказать ему свою легальность, но он также должен убедиться сам, что ведет диалог действительно с веб-сервером своего предприятия. Другими словами, сервер и клиент должны пройти процедуру взаимной аутентификации. Здесь мы имеем дело с *аутентификацией на уровне приложений*.

При установлении сеанса связи между двумя устройствами также часто предусматриваются процедуры взаимной аутентификации устройств на более низком, канальном, уровне (см. далее раздел «Строгая аутентификация на основе многоразового пароля в протоколе CHAP»).

Аутентификация данных означает доказательство целостности этих данных, а также то, что они поступили именно от того человека, который объявил об этом. Для этого используется механизм электронной подписи. Ранее мы уже узнали, как используется для аутентификации данных несимметричное шифрование.

Авторизация доступа

Термин авторизация (authorization) происходит от латинского слова auctoritas, показывающее уровень престижа человека в Древнем Риме и соответствующие этому уровню привилегии.

Авторизация — *это процедура контроля доступа легальных Пользователей к ресурсам системы и предоставление каждому из них именно тех прав; которые ему были определены администратором*.

В отличие от аутентификации, которая позволяет распознать легальных и нелегальных пользователей, авторизация имеет дело только с *легальными* пользователями, успешно прошедшими процедуру аутентификации. Помимо предоставления пользователям прав доступа к каталогам, файлам и принтерам, средства авторизации могут контролировать возможность выполнения пользователями различных системных функций, таких как локальный доступ к серверу, установка системного времени, создание резервных копий данных, выключение сервера и т. п.

Средства авторизации наделяют пользователя сети правами выполнять определен-ные действия по отношению к определенным ресурсам. Для этого могут применяться различные формы предоставления правил доступа, которые часто делят на два класса:

- **Избирательный доступ** наиболее широко используется в компьютерных сетях. При этом подходе определенные операции с определенным ресурсом разрешаются или запрещаются пользователям или группам пользователей, явно указанным своими идентификаторами, например: «пользователю User_T разрешено читать и записывать в файл File t».

Авторизация доступа

- **Мандатный подход** к определению прав доступа заключается в том, что вся информация делится на уровни в зависимости от степени секретности, а все пользователи сети также делятся на группы, образующие иерархию в соответствии с *уровнем допуска* к этой информации. Такой подход позволяет классифицировать данные на информацию для служебного пользования, а также секретную и совершенно секретную информацию. Пользователи этой информации в зависимости от определенного для них статуса получают разные формы допуска: первую, вторую или третью. В отличие от систем с избирательными правами доступа, в системах с мандатным подходом пользователи в принципе не имеют возможности изменить уровень доступности информации. Например, пользователь более высокого уровня не может разрешить читать данные из своего файла пользователю, относящемуся к более низкому уровню. Отсюда видно, что мандатный подход является более строгим.

Процедуры авторизации часто совмещаются с процедурами аутентификации и реализуются одними и теми же программными средствами, которые могут встраиваться в операционную систему или приложение, а также поставляться в виде отдельных программных продуктов.

Авторизация доступа

При этом программные системы аутентификации и авторизации могут строиться на базе двух схем :

- *Централизованная схема, базирующаяся на сервере*. В этой схеме сервер управляет процессом предоставления ресурсов сети пользователю. Главная цель таких систем — реализовать «принцип единого входа». В соответствии с централизованной схемой пользователь один раз логически входит в сеть и получает на все время работы некоторый набор разрешений по доступу к различным ресурсам сети. Система Kerberos с ее сервером безопасности и архитектурой клиент-сервер, а также более современная система Shibboleth, построенная в той же архитектуре, являются наиболее известными системами этого типа. Системы TACACS и RADIUS, часто применяемые совместно с системами удаленного доступа, также реализуют этот подход.
- *Децентрализованная схема, базирующаяся на рабочих станциях*. При этом подходе средства авторизации работают на каждой машине. Администратор должен отслеживать работу механизмов безопасности каждого отдельного приложения — электронной почты, справочной службы, локальных баз данных и т. п.

Подчеркнем, что системы аутентификации и авторизации совместно решают одну задачу — обеспечение контроля доступа, поэтому к ним необходимо предъявлять одинаковый уровень требований. Ненадежность одного звена здесь не может быть компенсирована надежностью другого.

Аудит

Аудит (auditing) — *это набор процедур мониторинга и учета всех событий, представляющих потенциальную угрозу для безопасности системы*.

Аудит позволяет «шпионить» за выбранными объектами и выдавать сообщения тревоги, когда, например, какой-либо рядовой пользователь попытается прочитать или модифицировать системный файл. Если кто-то пытается выполнить действия, выбранные системой безопасности для мониторинга, то система аудита пишет сообщение в журнал регистрации, идентифицируя пользователя. Системный менеджер может готовить отчеты безопасности, которые содержат информацию из журнала регистрации. Для «сверхбезопасных» систем предусматриваются аудио- и видеосигналы тревоги, устанавливаемые на машинах администраторов, отвечающих за безопасность.

Поскольку никакая система безопасности не гарантирует защиту на уровне 100 %, последним рубежом в борьбе с нарушениями оказывается система аудита. Действительно, после того как злоумышленнику удалось провести успешную атаку, пострадавшей стороне не остается ничего другого, как обратиться к службе аудита. Если при настройке службы аудита были правильно заданы события, которые требуется отслеживать, то подробный анализ записей в журнале может дать много полезной информации. Эта информация, возможно, позволит найти злоумышленника или, по крайней мере, предотвратить повторение подобных атак путем устранения уязвимых мест в системе защиты. Функции аудита встраиваются в различные средства обеспечения безопасности: сетевые экраны, системы обнаружения вторжений, антивирусные системы, сетевые мониторы.

Строгая аутентификация на основе многоразового пароля в протоколе CHAP

Протокол аутентификации по квитированию вызова (Challenge Handshake Authentication Protocol, CHAP) входит в семейство протоколов PPR В этом протоколе предусмотрено 4 типа сообщений: *Success* (успех), *Challenge* (вызов), *Response* (ответ), *Failure* (ошибка).

Этот протокол используется, например, при аутентификации удаленных пользователей, подключенных к Интернету по коммутируемому каналу Здесь аутентификатором является сервер провайдера, а аутентифицируемым — клиентский компьютер (рис. 1).

При заключении договора клиент получает от провайдера пароль (пусть, например, это будет слово parol). Этот пароль хранится в базе данных провайдера в виде дайджеста $Z = d(\text{parol})$, полученного путем применения к паролю односторонней хэш-функции MD5.

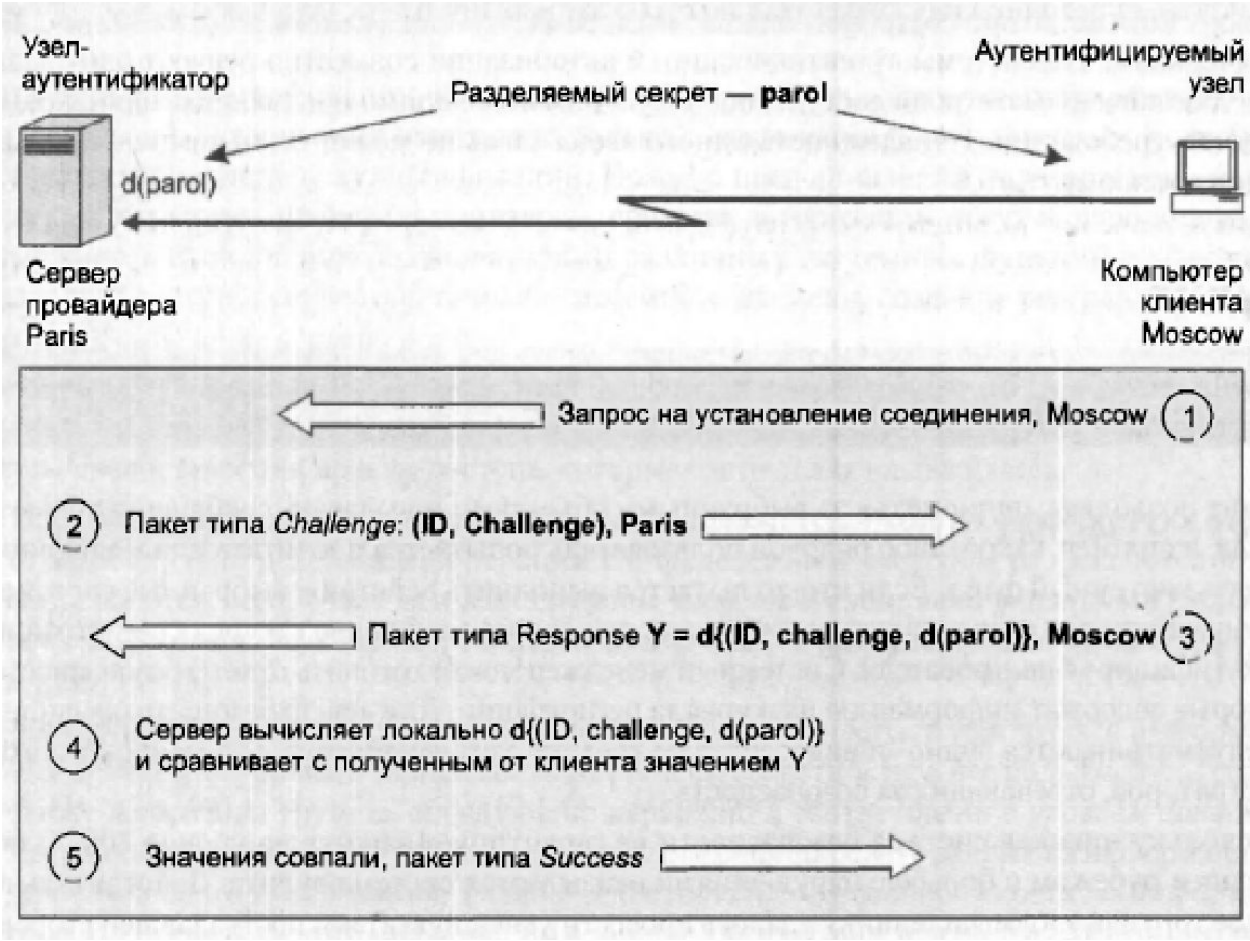

Рис. 1. Аутентификация по протоколу CHAP

Строгая аутентификация на основе многоразового пароля в протоколе CHAP

Аутентификация выполняется в следующей последовательности.

1. Пользователь-клиент активизирует программу (например, программу дозвона) удаленного доступа к серверу провайдера, вводя имя и назначенный ему пароль. Имя (на рисунке это "Moscow") передается по сети провайдеру в составе запроса на соединение, но пароль не передается в сеть ни в каком виде. То есть здесь мы имеем дело со строгой аутентификацией.
2. Сервер провайдера, получив запрос от клиента, генерирует псевдослучайное слово-вызов (пусть это будет слово «challenge») и передает его клиенту вместе со значением, идентифицирующем сообщение в рамках данного сеанса (ID), и собственным именем (здесь "Paris"). Это сообщение типа Challenge. (Для защиты от перехвата ответа аутентификатор должен использовать разные значения слова-вызова при каждой процедуре аутентификации).
3. Программа клиента, получив этот пакет, извлекает из него слово-вызов, добавляет к нему идентификатор и вычисленный локально дайджест Z = d(parol), а затем вычисляет с помощью все той же функции MD5 дайджест Y = d{(ID, challenge, J(parol)} от всех этих трех значений. Результат клиент посылает серверу провайдера в пакете Response.
4. Сервер провайдера сравнивает полученный по сети дайджест Ус тем значением, которое он получил, локально применив ту же хэш-функцию к набору аналогичных компонентов, хранящихся в его памяти.
5. Если результаты совпадают, то аутентификация считается успешной и аутентификатор посылает партнеру пакет Success.

Строгая аутентификация на основе многоразового пароля в протоколе CHAP

Аналогичный алгоритм аутентификации применяется в семействе ОС Windows. Там многоразовые пароли пользователей также хранятся в базе данных сервера в виде дайджестов, а по сети в открытом виде передается только слово-вызов. Кажется, что такой способ хранения паролей надежно защищает их от злоумышленника, даже если он сможет получить к ним доступ. Действительно, ведь даже теоретически нельзя восстановить исходное значение по дайджесту.

Однако создатель первого червя Роберт Моррис решил эту проблему. Он разработал довольно простую программу, которая генерировала возможные варианты паролей, как используя слова из словаря, так и путем последовательного перебора символов. Для каждого сгенерированного слова вычислялся дайджест и сравнивался с дайджестами из файла паролей. Удивительно, но такая стратегия оказалась весьма эффективной, и хакеру удалось завладеть несколькими паролями.

Аутентификация на основе одноразового пароля

Алгоритмы аутентификации, основанные на многоразовых паролях, не очень надежны. Пароли можно подсмотреть, разгадать или просто украсть. Более надежными оказываются схемы с одноразовыми паролями. К тому же одноразовые пароли намного дешевле и проще биометрических систем аутентификации, таких как сканеры сетчатки глаза или отпечатков пальцев. Все это делает системы, основанные на одноразовых паролях, очень перспективными. Следует иметь в виду, что, как правило, системы аутентификации на основе одноразовых паролей рассчитаны на проверку только удаленных, а не локальных пользователей.

Генерация одноразовых паролей может выполняться либо программно, либо аппаратно. Аппаратные реализации систем доступа на основе одноразовых паролей называют **аппаратными ключами**. Они представляют собой миниатюрные устройства со встроенным микропроцессором, похожие либо на обычные пластиковые карточки, используемые для доступа к банкоматам, либо на карманные калькуляторы, имеющие клавиатуру и маленькое дисплейное окно (рис. 2). Аппаратные ключи могут быть также реализованы в виде присоединяемого к разъему компьютера устройства.

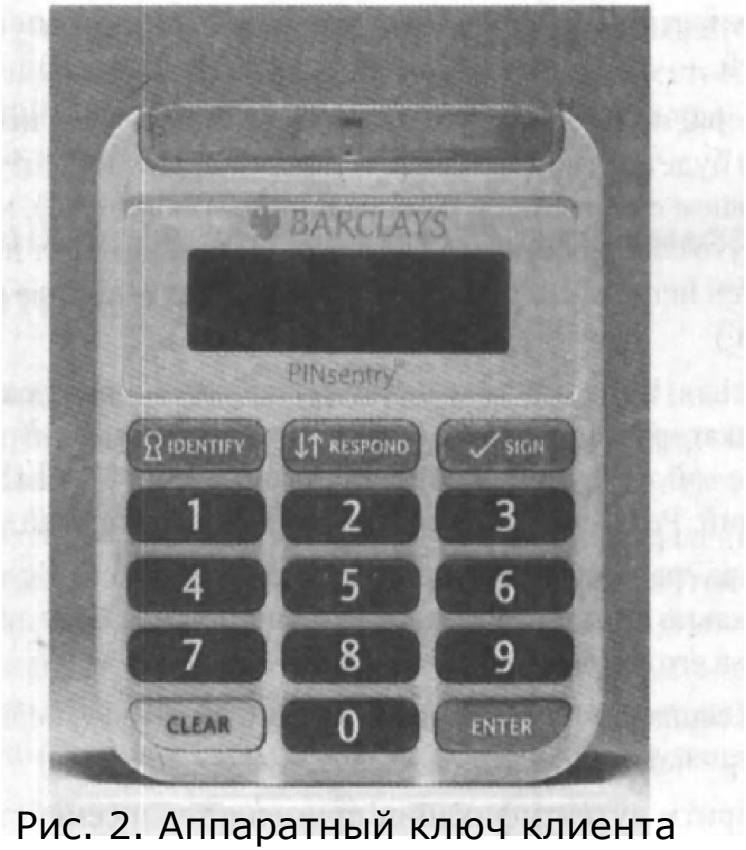

Рис. 2. Аппаратный ключ клиента банка для доступа к счетам

Аутентификация на основе одноразового пароля

Существуют и программные реализации средств аутентификации на основе одноразовых паролей — **программные ключи**. Программные ключи размещаются на сменном магнитном носителе в виде обычной программы, важной частью которой является генератор одноразовых паролей.

Независимо от того, какую реализацию системы аутентификации на основе одноразовых паролей выбирает пользователь, он, как и в системах аутентификации с применением многоразовых паролей, сообщает системе свой идентификатор, однако вместо того чтобы вводить каждый раз один и тот же пароль, он указывает последовательность цифр, сообщаемую ему аппаратным или программным ключом. Через определенный небольшой период времени генерируется другая последовательность — новый пароль. Сервер аутентификации проверяет введенную последовательность и разрешает пользователю осуществить логический вход. Сервер аутентификации может представлять собой отдельное устройство, выделенный компьютер или же программу, выполняемую на обычном сервере.

Рассмотрим схему использования аппаратных ключей, в основе которой лежит *синхронизация по времени*. Этот популярный алгоритм аутентификации был разработан компанией Security Dynamics.

Идея метода состоит в том, что аппаратный ключ и аутентифицирующий сервер вычисляют некоторое значение по одному и тому же алгоритму.

Аутентификация на основе одноразового пароля

Алгоритм имеет два параметра:

- разделяемый секретный ключ, представляющий собой 64-разрядное число, уникально назначаемое каждому пользователю и хранящееся как в аппаратном ключе, так и в базе данных сервера аутентификации;
- значение текущего времени.

Если вычисленные значения совпадают, то аутентификация считается успешной.

Итак, пусть удаленный пользователь пытается совершить логический вход в систему с персонального компьютера (рис. 3).

Аутентифицирующая программа предлагает ему ввести его личный персональный номер (PIN), состоящий из четырех десятичных цифр, а также 6 цифр случайного числа, отображаемого в тот момент на дисплее аппаратного ключа. На основе PIN-кода сервер извлекает из базы данных информацию о пользователе, а именно — его секретный ключ. Затем сервер выполняет вычисления по тому же алгоритму, который заложен в аппаратном ключе, используя в качестве параметров секретный ключ и значение текущего времени, проверяя, совпадает ли сгенерированное число с числом, которое ввел пользователь. Если они совпадают, то пользователю разрешается логический вход.

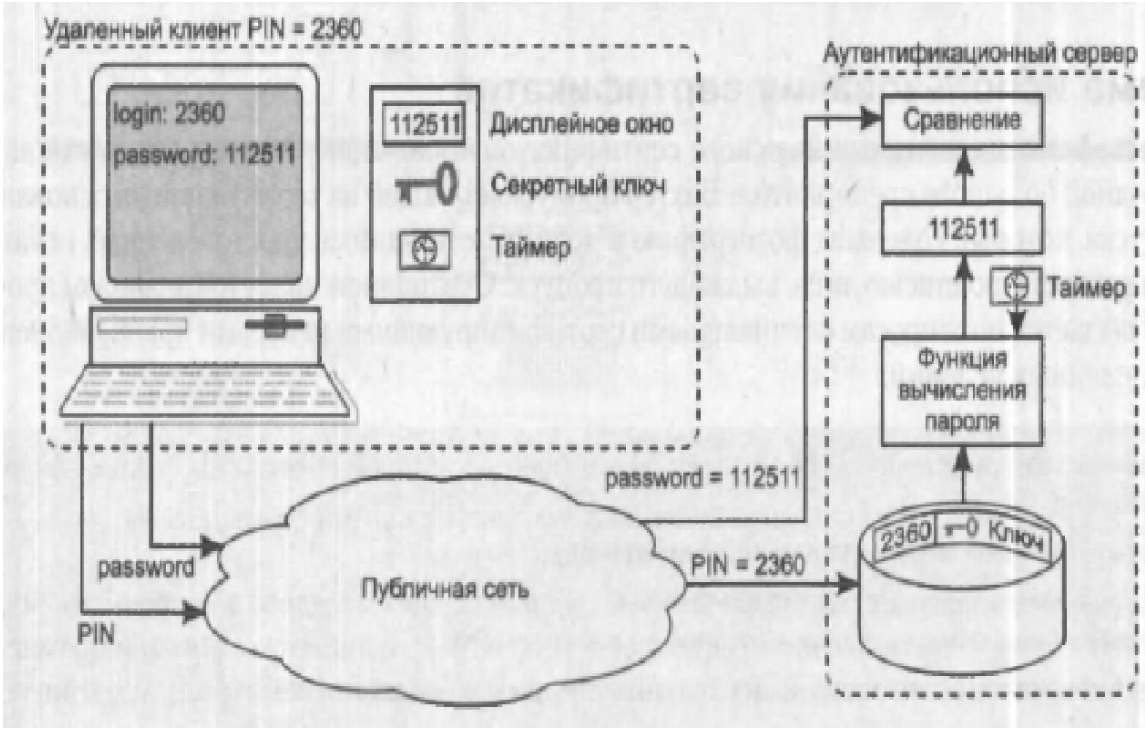

Рис. 3. Аутентификация, основанная на временной синхронизации

Аутентификация на основе одноразового пароля

Потенциальной проблемой этой схемы является временная синхронизация сервера и аппаратного ключа (ясно, что вопрос согласования часовых поясов решается просто). Гораздо сложнее обстоит дело с постепенным рассогласованием внутренних часов сервера и аппаратного ключа, тем более что потенциально аппаратный ключ может работать несколько лет. Компания Security Dynamics решает эту проблему двумя способами. Во-первых, при производстве аппаратного ключа измеряется отклонение частоты его таймера от номинала. Далее эта величина учитывается в виде параметра алгоритма сервера. Во-вторых, сервер отслеживает коды, генерируемые конкретным аппаратным ключом, и если таймер данного ключа постоянно спешит или отстает, то сервер динамически подстраивается под него.

Существует еще одна проблема, связанная со схемой временной синхронизации. Одноразовый пароль, генерируемый аппаратным ключом, действителен в течение некоторого интервала времени (от нескольких десятков секунд до нескольких десятков минут), то есть в течение этого времени одноразовый пароль, в сущности, является многоразовым. Поэтому теоретически возможно, что очень проворный хакер сможет перехватить PIN-код и одноразовый пароль с тем, чтобы также получить доступ в сеть в течение этого интервала.

Аутентификация на основе сертификатов

Аутентификация с применением цифровых сертификатов является альтернативой применению паролей и представляется естественным решением в условиях, когда число пользователей сети (пусть и потенциальных) измеряется миллионами. В таких обстоятельствах процедура предварительной регистрации пользователей, связанная с назначением и хранением их паролей, становится крайне обременительной, опасной, а иногда и просто нереализуемой. При наличии сертификатов сеть, которая дает пользователю доступ к своим ресурсам, не хранит никакой информации о своих пользователях — они ее предоставляют сами в своих запросах в виде сертификатов, удостоверяющих личность пользователей. Сертификаты выдаются специальными уполномоченными организациями — центрами сертификации (Certificate Authority, CA). Поэтому задача хранения секретной информации (закрытых ключей) возлагается на самих пользователей, что делает это решение гораздо более масштабируемым, чем вариант с централизованной базой паролей.

Схема использования сертификатов

Аутентификация личности на основе сертификатов происходит примерно так же, как на проходной большого предприятия. Вахтер пропускает людей на территорию на основании пропуска, который содержит фотографию и подпись сотрудника, удостоверенных печатью предприятия и подписью лица, выдавшего пропуск. Сертификат является аналогом пропуска и выдается по запросам специальными сертифицирующими центрами при выполнении определенных условий.

Схема использования сертификатов

Сертификат представляет собой электронную форму, в которой содержится следующая информация:

- открытый ключ владельца данного сертификата;
- сведения о владельце сертификата, такие, например, как имя, адрес электронной почты, наименование организации, в которой он работает и т. п.;
- наименование сертифицирующей организации, выдавшей данный сертификат;
- электронная подпись сертифицирующей организации, то есть зашифрованные закрытым ключом этой организации данные, содержащиеся в сертификате.

Использование сертификатов основано на предположении, что сертифицирующих организаций немного и их открытые ключи широко доступны, например, из публикаций в журналах.

Когда пользователь хочет подтвердить свою личность, он предъявляет свой сертификат в двух формах: открытой (то есть такой, в которой он получил его в сертифицирующей организации) и зашифрованной с применением своего закрытого ключа (рис. 4). **Сторона, проводящая аутентификацию, берет из незашифрованного сертификата *открытый ключ пользователя* и расшифровывает с его помощью зашифрованный сертификат**. Совпадение результата с открытым сертификатом подтверждает, что предъявитель действительно является владельцем закрытого ключа, соответствующего указанному открытому.

Затем с помощью известного открытого ключа указанной в сертификате организации проводится расшифровка подписи этой организации в сертификате.

Схема использования сертификатов

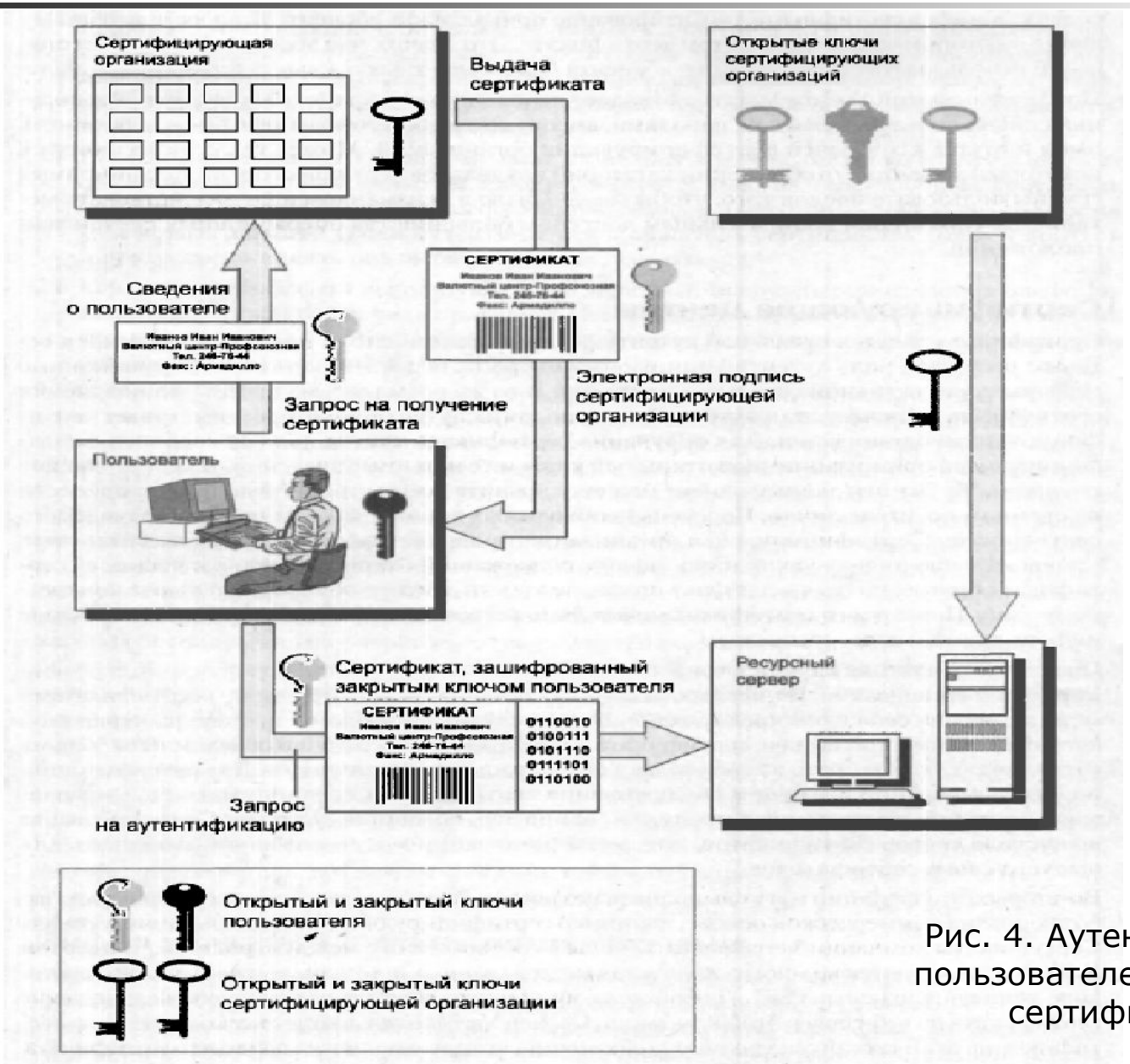

Рис. 4. Аутентификация пользователей на основе сертификатов

Схема использования сертификатов

Если в результате получается тот же сертификат с тем же именем пользователя и его открытым ключом, значит, он действительно прошел регистрацию в сертификационном центре, является тем, за кого себя выдает, и указанный в сертификате открытый ключ действительно принадлежит ему.

Сертификаты можно использовать не только для аутентификации, но и для предоставления избирательных прав доступа. Для этого в сертификат могут вводиться дополнительные поля, в которых указывается принадлежность его владельцев к той или иной категории пользователей. Эта категория назначается сертифицирующей организацией в зависимости от условий, на которых выдается сертификат. Например, организация, поставляющая через Интернет на коммерческой основе информацию, может выдавать сертификаты определенной категории пользователям, оплатившим годовую подписку на некоторый бюллетень, тогда веб-сервер будет предоставлять доступ к страницам бюллетеня только пользователям, предъявившим сертификат данной категории.

Подчеркнем тесную связь открытых ключей с сертификатами. Сертификат является удостоверением не только личности, но и принадлежности открытого ключа. Цифровой сертификат устанавливает и гарантирует соответствие между открытым ключом и его владельцем. Это предотвращает угрозу подмены открытого ключа.

Схема использования сертификатов

Если некоторый абонент А получает по сети сертификат от абонента Б, то он может быть уверен, что открытый ключ, содержащийся в сертификате, гарантированно принадлежит абоненту Б, адрес и другие сведения о котором содержатся в этом сертификате. Это значит, что абонент А может без опасений использовать открытый ключ абонента Б для секретных посланий в адрес последнего.

При использовании сертификатов отпадает необходимость хранить на серверах корпораций списки пользователей с их паролями, вместо этого достаточно иметь на сервере список имен и открытых ключей сертифицирующих организаций. Может также понадобиться некоторый механизм отображений категорий владельцев сертификатов на традиционные группы пользователей для того, чтобы можно было в неизменном виде задействовать механизмы управления избирательным доступом большинства операционных систем или приложений.

Сертифицирующие центры

Сертификат является средством аутентификации пользователя при его обращении к сетевым ресурсам, роль аутентифицирующей стороны играют при этом информационные серверы корпоративной сети или Интернета. В то же время и сама процедура получения сертификата включает этап аутентификации, когда аутентификатором выступает сертифицирующая организация. Для получения сертификата клиент должен сообщить сертифицирующей организации свой открытый ключ и те или иные сведения, удостоверяющие его личность. Все эти данные клиент может отправить по электронной почте или принести на съемном носителе лично. Перечень необходимых данных зависит от типа получаемого сертификата. Сертифицирующая организация проверяет доказательства подлинности, помещает свою цифровую подпись в файл, содержащий открытый ключ, и посылает сертификат обратно, подтверждая факт принадлежности данного конкретного ключа конкретному лицу. После этого сертификат может быть встроен в любой запрос на использование информационных ресурсов сети.

Практически важным является вопрос о том, кто имеет право выполнять функции сертифицирующей организации. Во-первых, задачу обеспечения своих сотрудников сертификатами может взять на себя само предприятие. В этом случае упрощается процедура первичной аутентификации при выдаче сертификата. Предприятия достаточно осведомлены о своих сотрудниках, чтобы брать на себя задачу подтверждения их личности. Для автоматизации процесса генерации, выдачи и обслуживания сертификатов предприятия могут использовать готовые программные продукты, например, компания Netscape Communications выпустила сервер сертификатов, который организации могут у себя устанавливать для выпуска своих сертификатов.

Сертифицирующие центры

Во-вторых, эти функции могут выполнять независимые центры по выдаче сертификатов, работающие на коммерческой основе, например сертифицирующий центр компании Verisign. Сертификаты компании Verisign выполнены в соответствии с международным стандартом X.509 и используются во многих продуктах защиты данных, в том числе в популярном протоколе защищенного канала SSL. Любой желающий может обратиться с запросом на получение сертификата на веб-сервер этой компании. Сервер Verisign предлагает несколько типов сертификатов с разными уровнями полномочий:

- сертификаты класса 1 предоставляют пользователю самый низкий уровень полномочий. Они могут применяться при отправке и получении шифрованной электронной почты через Интернет. Чтобы получить сертификат этого класса, пользователь должен сообщить серверу Verisign свой адрес электронной почты или свое уникальное имя;
- сертификаты класса 2 дают возможность его владельцу пользоваться внутрикорпоративной электронной почтой и принимать участие в подписных интерактивных службах. Чтобы получить сертификат этого более высокого уровня, пользователь должен организовать подтверждение своей личности сторонним лицом, например своим работодателем. Такой сертификат с информацией от работодателя может эффективно применяться при деловой переписке;
- сертификаты класса 3 предоставляют владельцу все те возможности, которые имеет обладатель сертификата класса 2, плюс возможность участия в электронных банковских операциях, электронных сделках по покупке товаров и некоторые другие возможности. Для доказательства аудентичности соискателю необходимо лично явится для представления подтверждающих документов;

Сертифицирующие центры

- сертификаты класса 4 используются при выполнении крупных финансовых операций. Поскольку такой сертификат наделяет владельца самым высоким уровнем доверия, сертифицирующий центр Verisign проводит тщательное изучение частного лица или организации, запрашивающей сертификат.

Механизм получения пользователем сертификата хорошо автоматизируется в сети в модели клиент-сервер, когда браузер исполняет роль клиента, а в сертифицирующей организации установлен специальный сервер выдачи сертификатов. Браузер генерирует для пользователя пару ключей, оставляет закрытый ключ у себя и передает частично заполненную форму сертификата серверу. Для того чтобы неподписанный еще сертификат нельзя было подменить при передаче по сети, браузер ставит свою электронную подпись, зашифровывая сертификат выработанным закрытым ключом. Сервер сертификатов подписывает полученный сертификат, фиксирует его в своей базе данных и возвращает его каким-либо способом владельцу. Очевидно, что при этом может выполняться еще и неформальная процедура подтверждения пользователем своей личности и права на получение сертификата, требующая участия оператора сервера сертификатов. Это могут быть доказательства оплаты услуги, доказательства принадлежности к той или иной организации — все случаи жизни предусмотреть и автоматизировать нельзя. После получения сертификата браузер сохраняет его вместе с закрытым ключом и использует при аутентификации на тех серверах, которые поддерживают такой процесс.

В настоящее время существует большое количество протоколов и продуктов, использующих сертификаты. Например, компания Microsoft реализовала поддержку сертификатов и в своем браузере Internet Explorer, и в сервере Internet Information Server, разработала собственный сервер сертификатов, а также продукты, которые позволяют хранить сертификаты пользователя, его закрытые ключи и пароли защищенным образом.

Инфраструктура с открытыми ключами

Несмотря на активное использование технологии цифровых сертификатов во многих системах безопасности, эта технология еще не решила целый ряд серьезных проблем. Это, прежде всего, поддержание базы данных о выпущенных сертификатах. Сертификат выдается не навсегда, а на некоторый вполне определенный срок. По истечении срока годности сертификат должен либо обновляться, либо аннулироваться. Кроме того, необходимо предусмотреть возможность досрочного прекращения полномочий сертификата. Все заинтересованные участники информационного процесса должны быть вовремя оповещены о том, что некоторый сертификат уже недействителен. Для этого сертифицирующая организация должна оперативно поддерживать список аннулированных сертификатов.

Имеется также ряд проблем, связанных с тем, что сертифицирующие организации существуют не в единственном числе. Все они выпускают сертификаты, но даже если эти сертификаты соответствуют единому стандарту (сейчас это, как правило, стандарт X.509), все равно остаются нерешенными многие вопросы. Все ли сертифицирующие центры заслуживают доверия? Каким образом можно проверить полномочия того или иного сертифицирующего центра? Можно ли создать иерархию сертифицирующих центров, когда сертифицирующий центр, стоящий выше, мог бы сертифицировать центры, расположенные ниже в иерархии? Как организовать совместное использование сертификатов, выпущенных разными сертифицирующими организациями?

Для решения этих и многих других проблем, возникающих в системах, использующих технологии шифрования с открытыми ключами, оказывается необходимым комплекс программных средств и методик, называемый **инфраструктурой с открытыми ключами** (Public Key Infrastructure, PKI).

Инфраструктура с открытыми ключами

Информационные системы больших предприятий нуждаются в специальных средствах администрирования и управления цифровыми сертификатами, парами открытых/закрытых ключей, а также приложениями, функционирующими в среде с открытыми ключами.

В настоящее время любой пользователь имеет возможность, загрузив широко доступное программное обеспечение, абсолютно бесконтрольно сгенерировать себе пару открытый/закрытый ключ. Затем он может также совершенно независимо от администрации вести шифрованную переписку со своими внешними абонентами. Такая «свобода» пользователя часто не соответствует принятой на предприятии политике безопасности. Для более надежной защиты корпоративной информации желательно реализовать централизованную службу генерации и распределения ключей. Для администрации предприятия важно иметь возможность получить копии закрытых ключей каждого пользователя сети, чтобы в случае увольнения пользователя или потери пользователем его закрытого ключа сохранить доступ к зашифрованным данным этого пользователя. В противном случае резко ухудшается одна из трех характеристик безопасной системы — доступность данных.

Процедура, позволяющая получать копии закрытых ключей, называется **восстановлением ключей**. Вопрос, включать ли в продукты безопасности средства восстановления ключей, в последние годы приобрел политический оттенок. В США прошли бурные дебаты, тему которых можно примерно сформулировать так: обладает ли правительство правом доступа к любой частной информации при условии, что на это есть постановление суда?

Инфраструктура с открытыми ключами

И хотя в такой широкой постановке проблема восстановления ключей все еще не решена, необходимость включения средств восстановления в корпоративные продукты ни у кого сомнений не вызывает. Принцип доступности данных не должен нарушаться из-за волюнтаризма сотрудников, монопольно владеющих своими закрытыми ключами. Ключ может быть восстановлен при выполнении некоторых условий, которые должны быть четко определены в политике безопасности предприятия.

Как только принимается решение о включении в систему безопасности средств восстановления, возникает вопрос, как же быть с надежностью защиты данных, как убедить пользователя в том, что его закрытый ключ не употребляется с какими-либо другими целями, не имеющими отношения к резервированию? Некоторую уверенность в секретности хранения закрытых ключей может дать технология депонирования ключей. Депонирование ключей — это предоставление закрытых ключей на хранение третьей стороне, надежность которой не вызывает сомнений. Этой третьей стороной может быть правительственная организация или группа уполномоченных на это сотрудников предприятия, которым оказывается полное доверие.

Аудентификация информации

Под аутентификацией информации в компьютерных системах понимают установление подлинности полученных по сети данных исключительно на основе информации, содержащейся в полученном сообщении.

Если конечной целью шифрования информации является защита от несанкционированного ознакомления с этой информацией, то конечной целью аутентификации информации является защита участников информационного обмена от навязывания ложной информации. Концепция аутентификации в широком смысле предусматривает установление подлинности информации как при наличии взаимного доверия между участниками обмена, так и при его отсутствии.

В компьютерных системах выделяют два вида аутентификации информации:

- аутентификация хранящихся массивов данных и программ — установление факта того, что данные не подвергались модификации;
- аутентификация сообщений — установление подлинности полученного сообщения, в том числе решение вопроса об авторстве этого сообщения и установление факта приема.

Цифровая подпись

Для решения задачи аутентификации информации используется концепция **цифровой**, или **электронной, подписи**. Согласно терминологии, утвержденной Международной организацией по стандартизации (ISO), под термином «цифровая подпись» понимаются методы, позволяющие устанавливать подлинность автора сообщения (документа) при возникновении спора относительно авторства. Основная область применения цифровой подписи — это финансовые документы, сопровождающие электронные сделки, документы, фиксирующие международные договоренности и т. п.

До настоящего времени чаще всего для построения схемы цифровой подписи использовался алгоритм RSA. Как уже отмечалось (см. раздел «Алгоритм RSA»), в основе этого алгоритма лежит концепция Диффи—Хеллмана. Она заключается в том, что каждый пользователь сети имеет свой закрытый ключ, необходимый для формирования подписи, а соответствующий этому секретному ключу открытый ключ, предназначенный для проверки подписи, известен всем другим пользователям сети.

На рис. 5 показана схема формирования цифровой подписи по алгоритму RSA. Подписанное сообщение состоит из двух частей: незашифрованной части, в которой содержится исходный текст T, и зашифрованной части, представляющей собой цифровую подпись. Цифровая подпись S вычисляется с использованием закрытого ключа (D, n) по формуле: $S = T^d \bmod n$.

Рис. 5. Схема формирования ЦП по алгоритму RSA

Цифровая подпись

Сообщение посылается в виде пары (T, S). Каждый пользователь, имеющий соответствующий открытый ключ (E, n), получив сообщение, отделяет открытую часть *T*, расшифровывает цифровую подпись *S* и проверяет равенство: $T = S^E \bmod n$.

Если результат расшифровки цифровой подписи совпадает с открытой частью сообщения, считается, что документ подлинный, не претерпел никаких изменений в процессе передачи, а автором его является именно тот человек, который передал свой открытый ключ получателю. Если сообщение снабжено цифровой подписью, то получатель может быть уверен, что оно не было изменено или подделано по пути. Такие схемы аутентификации называются асимметричными. К недостаткам данного алгоритма можно отнести то, что длина подписи в этом случае равна длине сообщения, что не всегда удобно.

Если помимо проверки целостности документа, обеспечиваемой цифровой подписью, надо обеспечить его конфиденциальность, то после применения к тексту цифровой подписи выполняют шифрование и исходного текста, и цифровой подписи (рис. 6).

Цифровая подпись

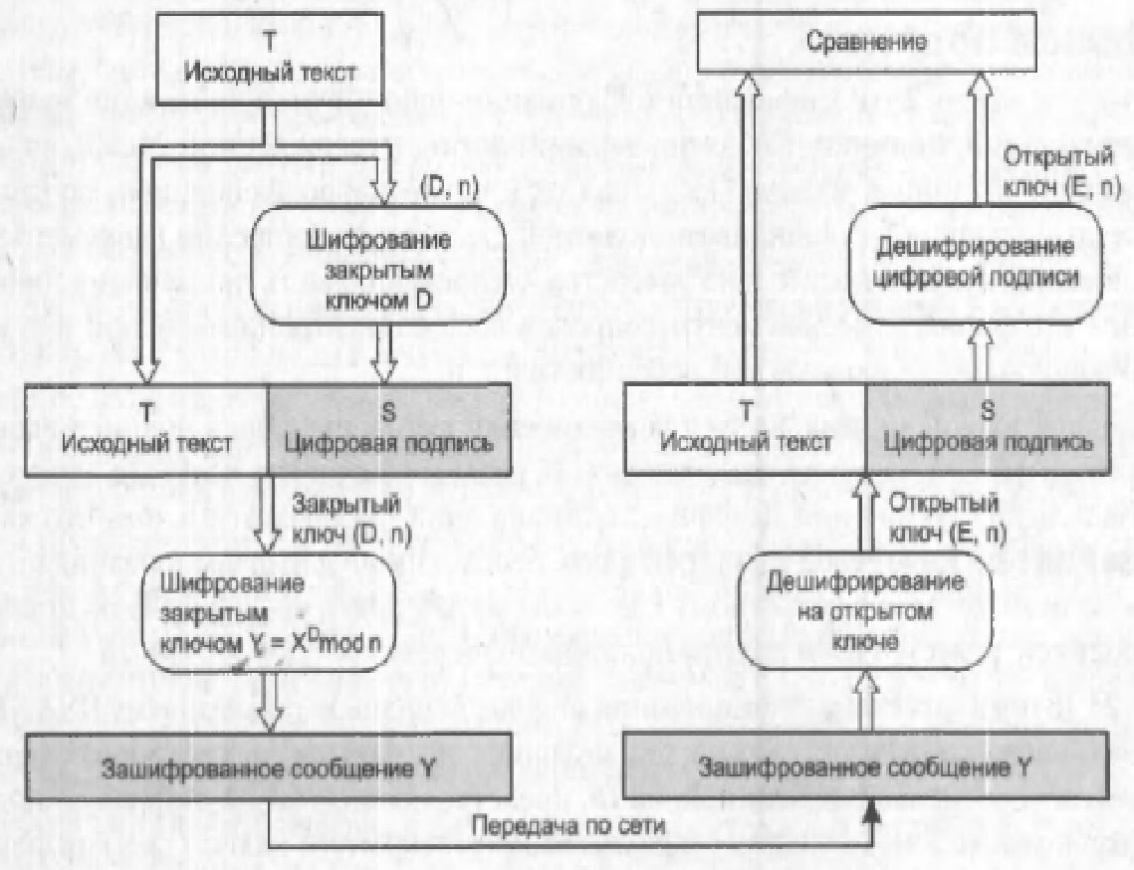

Рис. 6. Обеспечение конфиденциальности документа с цифровой подписью

Другие методы цифровой подписи основаны на формировании соответствующей сообщению контрольной комбинации с помощью симметричных алгоритмов типа DES. Учитывая более высокую производительность алгоритма DES по сравнению с RSA, он более эффективен для подтверждения аутентичности больших объемов информации. А для коротких сообщений типа платежных поручений или квитанций подтверждения приема, наверное, лучше подходит алгоритм RSA.

Аутентификация программных кодов

Компания Microsoft разработала средства для доказательства аутентичности программных кодов, распространяемых через Интернет. Пользователю важно иметь доказательства, что программа, которую он загрузил с какого-либо сервера, действительно содержит коды, разработанные определенной компанией. Протоколы защищенного канала (см. далее) типа SSL помочь здесь не могут, так как позволяют удостоверить только аутентичность сервера. Суть технологии аутентикода (authenticode), разработанной Microsoft, состоит в следующем.

Организация, желающая подтвердить свое авторство на программу, должна встроить в распространяемый код так называемый подписывающий блок (рис. 7). Этот блок состоит из двух частей. Первая часть — сертификат этой организации, полученный обычным образом от какого-либо сертифицирующего центра. Вторую часть образует зашифрованный дайджест, полученный в результате применения односторонней функции к распространяемому коду. Шифрование дайджеста выполняется с помощью закрытого ключа организации.

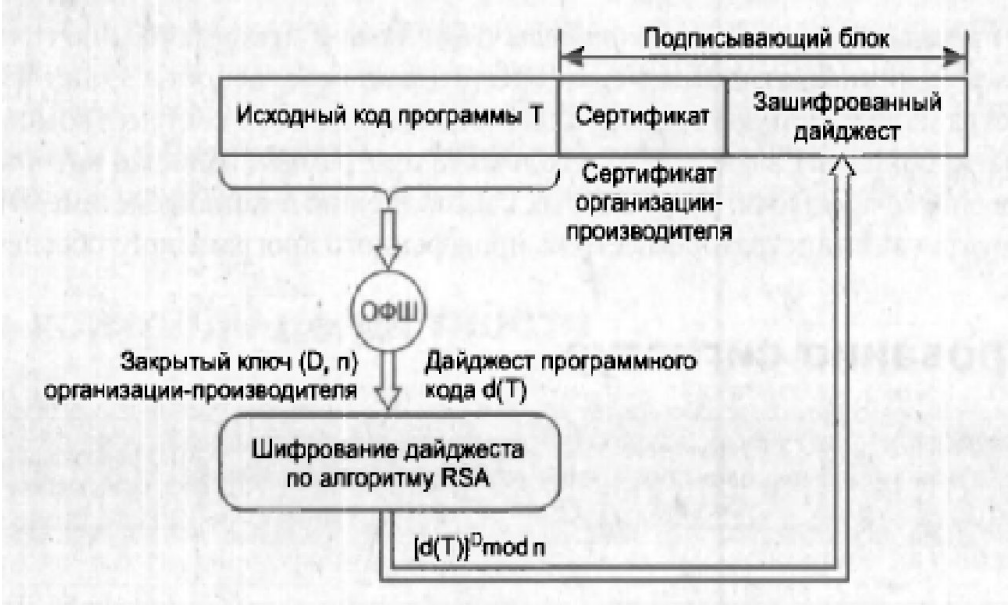

Рис. 7. Схема получения аутентикода

1. Аутентификационные технологии

Существует некоторое число различных классов аутентификации, начиная с полного ее отсутствия до очень строгих механизмов контроля. Для различных целей могут использоваться разные виды аутентификации.

1.1. Отсутствие аутентификации

Простейшая аутентификационная система не имеет аутентификации вовсе. Изолированная от сети частная персональная ЭВМ является примером, где аутентификация не нужна. Другим примером может служить автономная общедоступная рабочая станция, обслуживающая некоторые конференции, где раскрытие информации или ее модификация не являются критическими.

1.2. Аутентификационные механизмы, уязвимые для пассивных атак

Простая проверка пароля является наиболее общей формой аутентификации. Простые аутентификационные проверки имеют различные формы: ключ может быть паролем, запомненным пользователем, он может быть физическим или электронным объектом, принадлежащим пользователю, он может быть уникальной биологической характеристикой. Простые аутентификационные системы считаются "**раскрывающими**", так как, *если ключ передается по сети*, он может быть перехвачен злоумышленником. Имеются сообщения об успешных пассивных атаках в Интернет с помощью “расколотых” уже ЭВМ [CERT94]. Механизмы раскрывающей аутентификации уязвимы для атак “воспроизведения” (т.е. прослушивания сети). Ключи доступа могут быть запомнены в атакуемой машине и при наличии бреши в системе безопасности можно получить доступ ко всем паролям. Обычно форма хранения паролей допускает их сверку, но не чтение.

1.3. Аутентификационные механизмы, уязвимые для активных атак

Не раскрывающие парольные системы созданы для предотвращения атак воспроизведения. Разработано несколько систем для генерации не раскрываемых паролей. Система аутентификации S/Key (TM), разработанная в Bellcore, генерирует много одноразовых паролей из одного секретного ключа [Haller94]. В системах одноразовых паролей они могут меняться один раз в минуту или даже чаще, такие системы часто используют аппаратные средства. Bellcore не использует физических объектов (token), поэтому удобна для аутентификации машина-машина. Аутентификация S/Key не требует запоминания секретного ключа пользователя, что является преимуществом при работе с ненадежными вычислительными системами. В ее сегодняшнем виде система S/Key уязвима для переборных атак со словарем в случае неудачного выбора пароля. Система CHAP протокола PPP не является раскрывающей, но применима только локально [LS92, Simpson93].

1.4. Аутентификационные механизмы, не уязвимые для пассивных атак

По мере расширения применения сетей растет необходимость более жесткой аутентификации. В открытых сетях большое число пользователей могут получить доступ к информации, переносимой по сети. При желании пользователь может сымитировать ситуацию, при которой посланная им информация будет восприниматься, как посланная другим сетевым объектом/

Более мощные аутентификационные системы используют вычислительные возможности партнеров, участвующих в процессе аутентификации. Аутентификация может быть однонаправленной, например аутентификация пользователей в вычислительной системе, или она может быть взаимной, когда оба партнера должны идентифицировать друг друга.

Дополнительные материалы для изучения

Аутентификация в Internet

Некоторые системы аутентификации используют криптографические методы и формируют общий секретный код (например, ключ сессии), который может быть использован при последующем обмене. Например, пользователю после завершения процесса аутентификации может быть предоставлен аутентификационный билет, который может быть использован для получения других услуг без дополнительной аутентификации. Эти системы аутентификации могут также обеспечить, когда требуется, конфиденциальность (используя шифрование) при передаче данных по незащищенным сетям.

Технология LDAP

Популярность технологии каталожных сервисов **LDAP** продолжает расти. LDAP была разработана в 1993 году в университете Мичигана (смотри RFC-2849, -2891, -3062, -3296, -3671-74, -3687, -3703, -3727, -3866, -3876, -3909, -3928,-4510-33 и -5020). Данный список включает только основные документы RFC, посвященные LDAP. Смотри также Протокол LDAP (Lightweight Directory Access Protocol)
-(протокол легкого доступа к директории).

- **RFC-4511** - Описание протокола
- **RFC-4512** - Информационная модель каталогов
- **RFC-4513** - Методы аутентификации и механизмы безопасности
- **RFC-4514** - Представление строк для уникальных имен (DN)
- **RFC-4515** - Представление строк для поисковых фильтров
- **RFC-4516** - URL
- **RFC-4517** - Синтаксис и правила согласования
- **RFC-4518** - Подготовка интернационализованных строк
- **RFC-4519** - Схема пользовательских приложений

Дополнительные материалы для изучения

Аутентификация в Internet

Эта техника явилась исходной моделью для разработки Microsoft Active Directory и **MSDN** (Microsoft Developer Network), она поддерживает современные технологии аутентификации, такие как карты доступа и биометрические устройства контроля. LDAP является **стандартным Интернет протоколом для каталогов**, где хранятся данные об аккаунте пользователя. Первоначально эта разработка служила для взаимодействия со стандартом каталогов X.500 (ITU - Международный Телекоммуникационный союз)). Но к 1997 году, когда появились уже версии LDAPv2 и v3, функциональность была существенно расширена (были включены сервисы аутентификации и безопасности TLS и SSL). LDAP работает как в среде LINUX, так и в Windows. В рамках LDAP каждому пользователю присваивается уникальное имя **DN**. Каждое DN имеет несколько атрибутов, которые детально определяют доступ пользователя к дереву каталогов. DN является объектом, который может быть использован при написании программы управления на объектно - ориентированных языках. DN может быть встраиваться в URL и доступен через DNS-сервисы. На его основе были созданы Интернет протоколы **XED** (XML Enabled Directory) и **DSML** (Directory Service Markup Language). LDAP поддерживает сервисы динамических каталогов. Последние улучшения были связаны в основном с разработкой разнообразных GUI.

2. Криптография

Криптографические механизмы широко используются для осуществления аутентификации в современных сетях. Существует два базовых вида криптографии (симметричная и асимметричная). Одной из фундаментальных проблем для криптографии является транспортировка секретных ключей. Для этой цели может использоваться и *квантовая криптография*.

2.1. Симметричная криптография

Симметричная криптография включает в себя все системы, которые используют один и тот же ключ для шифрования и дешифрования. Таким образом, если кто-либо получит ключ, он сможет дешифровать и читать информацию, зашифрованную с его помощью. Такое лицо сможет шифровать и посылать любые данные, выдавая их за информацию, посланную легальным владельцем этого секретного ключа. Это означает, что знание ключа нежелательным третьим лицом полностью компрометирует конфиденциальность системы. Следовательно, используемые ключи должны доставляться безопасным способом, либо курьером, либо с применением специального протокола пересылки ключей, лучшим из которых является алгоритм Нидхэма-Шрёдера [NS78, NS87]. Широко используется алгоритм **DES** (Data Encryption Standard), который был стандартизован для защиты правительственной информации в США. Он представляет собой один из лучших симметричных алгоритмов шифрования [NBS77].

Хорошо известной системой, работающей в открытых сетях, является система аутентификации **Kerberos** (TM), которая была разработана в рамках проекта Athena в MIT. Система Kerberos базируется на алгоритме DES и использует специальный сервер, который хранит секретные ключи всех пользователей и услуг. Он может генерировать коды, которые позволяют пользователям и процессам идентифицировать себя в других системах.

Как в любой схеме с распределенной аутентификацией, эти верительные коды работают в пределах местного административного домена. Следовательно, если пароль пользователя раскрыт, злоумышленник будет способен маскироваться под этого пользователя и проникнуть в любую систему, обслуживаемую Kerberos. Так как сервер Kerberos знает все секретные ключи, он должен быть достаточно безопасным. Ключи сессии Kerberos могут использоваться для обеспечения конфиденциальности при обмене между любыми объектами в пределах зоны действия сервера.

2.2. Асимметричная криптография

В конце 1970, главным прорывом в криптографии стала разработка асимметричной криптографии. Здесь для шифрования и дешифрования используются разные ключи, которые генерируются совместно. Наилучшая асимметричная система базируется на алгоритме, предложенном Rivest, Shamir и Adleman, и называется по инициалам авторов **RSA** [RSA78].

SPX представляет собой экспериментальную систему, которая преодолевает ограничения системы Kerberos путем применения криптографии с общедоступным ключом RSA [TA91]. SPX предполагает глобальную иерархию сертифицирующих узлов по одному или более для каждого из партнеров. Она использует цифровую подпись, которая состоит из строки кодов, зашифрованных секретным ключом отправителя, и которая может быть проверена с помощью соответствующего общедоступного ключа. Общедоступные ключи предполагаются правильными, так как получены с сертифицирующей подписью. Критические секции аутентификационного обмена шифруются посредством общедоступных ключей получателя, что препятствует атаке воспроизведения.

2.3. Криптографические контрольные суммы

Криптографические контрольные суммы являются одним из наиболее важных средств разработчиков криптографических протоколов. Криптографическая контрольная сумма или MIC (Message Integrity Checksum) служат для контроля целостности сообщений и аутентификации. Например, Secure SNMP и SNMPv2 вычисляют криптографическую контрольную сумму MD5 для совместного секретного блока данных и информации, которая должна быть аутентифицирована [Rivest92, GM93]. Это служит для того, чтобы аутентифицировать источник данных при этом предполагается, что эту сумму крайне трудно фальсифицировать. Она не указывает на то, что сами посланные данные корректны, а лишь на то, что они посланы именно данным отправителем. Криптографические контрольные суммы могут использоваться для получения относительно эффективной аутентификации, и особенно полезны при обмене ЭВМ-ЭВМ. Главная трудность реализации - передача ключей.

2.4. Цифровые подписи (сигнатуры)

Цифровая подпись представляет собой криптографический механизм, который является аналогом рукописной подписи. Она служит для аутентификации блока данных и подтверждает то, что она получена от отправителя. Цифровая подпись, использующая асимметричную криптографию (общедоступные ключи) может быть полезной для определения источника сообщения даже в случае, когда отправитель отрицает свое авторство. Цифровая подпись обеспечивает аутентификацию без конфиденциальности, так как текст самого сообщения не шифруется. Цифровая подпись использована в системе конфиденциальной почты **PEM** (Privacy Enhanced Mail) [Linn93, Kent93, Balenson93, Kaliski93].

3. Аутентификация пользователя на ЭВМ

Существует много различных подходов к проблеме аутентификации пользователя в *удаленных ЭВМ*. Имеется две угрозы при доступе к удаленной ЭВМ. Во-первых, злоумышленник может перехватить идентификатор и пароль пользователя и позднее воспользоваться ими при атаке "воспроизведения". Во-вторых, сама форма пароля позволяет хакеру попытаться его угадать.

В настоящее время большинство систем используют открытый текст для передачи паролей по сетевым каналам, что сильно упрощает их перехват [Anderson84, Kantor91]. Такая система не обеспечивает адекватной защиты от атак воспроизведения, когда злоумышленник сумел заполучить идентификатор и пароль удаленного пользователя.

Современные системы аутентификации часто являются многофакторными:

- Шифрованное имя (login).
- Шифрованный пароль.
- Карта доступа (например, USB с сертификатом и одноразовыми параметрами доступа, например, SSO). Такая карта может быть также и ключом доступа в определенные помещения.
- Биомерия (голос, отпечаток пальца, ладони или радужка глаза).

Сегодня формируются базы данных отпечатков пальцев и радужек глаз. Только в Индии сформирована такая БД на 200 млн. людей.

3.1. Защита против пассивной атаки является необходимой

Отсутствие, по крайней мере, не раскрывающей парольной системы, означает предоставление неограниченного доступа любому, кто имеет физический доступ к сети. Например, всякий кто имеет доступ к кабелю Ethernet, может имитировать работу любого пользователя данного сегмента сети. Таким образом, когда кто-то имеет пароль, передаваемый по Ethernet открытым текстом, реализуется первичная система безопасности. Когда размер локальной системы невелик (а это справедливо не только для Ethernet, но и для FDDI или Token-Ring LAN), данная система может еще рассматриваться как приемлемая, но это совершенно не так для сетей Интернет [CERT94].

Минимальной защитой против пассивных атак, таких как прослушивание сети, является применение не раскрывающей системы паролей. Такая система может функционировать на пассивном терминале или в качестве коммуникационной программы (напр., Crosstalk или PROCOMM), которая эмулирует пассивный терминал на персональной ЭВМ. Использование более строгой аутентификационной системы защитит против пассивных атак со стороны удаленных систем за счет ограничения использования простых терминалов. Разумно ожидать, что производители коммуникационных программ и не программируемых пользователем терминалов (таких как X-терминалы) встроят систему не раскрываемых паролей или более строгие аутентификационные системы. Одним из преимуществ Kerberos является то, что при правильном использовании пароль пользователя никогда не покидает пределов рабочей станции. Вместо этого они используются для расшифровки билетов пользователя Kerberos.

3.2. Защита периметра

Защита периметра применяется все шире. В этих системах пользователь сначала осуществляет аутентификацию в определенном объекте сети, например, в ЭВМ **"firewall"**, используя систему не раскрываемых паролей. Пользователь затем использует вторую систему для аутентификации в каждой ЭВМ или в группе ЭВМ, где он хотел бы получить доступ к определенным услугам.

В защите периметра существует несколько недостатков, по этой причине эту систему следует рассматривать как временное решение. Сетевой шлюз не прозрачен на IP-уровне и по этой причине работа с каждым видом сервиса должна производиться независимо. Использование двойной аутентификации является трудно осуществимым или невозможным для связи ЭВМ-ЭВМ. Протоколы точка-точка, которые являются обычными для Интернет механизмов без установления связи, легко уязвимы. Защита периметра должна быть плотной и полной, т.к. при ее прорыве внутренняя защита оказывается легко преодолимой.

Частой формой защиты периметра является передача приложений. Так как эти передачи являются протокольно зависимыми, IP-коннективность ЭВМ в пределах периметра с внешним миром оказывается нарушенной и часть преимуществ Интернет пропадает. Административное преимущество защиты периметра заключается в том, что число ЭВМ, которые могут быть подвергнуты атаке, достаточно мало. Эти машины могут быть тщательно проверены с точки зрения угроз безопасности. Но достаточно трудно или даже невозможно создать достаточно "герметичную" систему. Безопасность системы защиты периметра достаточно сложна, так как шлюз должен пропускать некоторые типы трафика, например, электронную почту. Другие сетевые услуги, такие как NTP (Network Time Protocol) и FTP могут также оказаться желательными. Более того, шлюзовая система периметра должна быть способна пропускать весь трафик всего домена, заключенного в данный периметр.

3.3. Защита от активных атак является крайне желательной

В обозримом будущем потребуются достаточно мощные системы, способные противостоять активным атакам. Многие корпоративные сети, базирующиеся на широковещательной технологии, такой как Ethernet, вероятно нуждаются в такой методике.

Чтобы защититься от активных атак, или обеспечить конфиденциальность, необходимо использовать *протокол с шифрованием сессии*, например, Kerberos, возможно использование аутентификационного механизма, который защищает от атаки воспроизведения. В системе Kerberos, пользователи получают коды доступа от сервера Kerberos и используют их для аутентификации, чтобы осуществить доступ к услугам других ЭВМ сети. Вычислительная мощность локальной рабочей станции может быть использована для дешифрования кодов доступа (используя ключ, извлеченный из пароля, введенного пользователем) и запоминания на время пока это требуется. Если протокол безопасности базируется на синхронизации часов, тогда может быть полезен протокол **NTPv3**, так как он распространяет временные метки для большого числа ЭВМ и является одним из немногих протоколов Интернет, которые содержат механизмы аутентификации [Bishop, Mills92].

Другим подходом для доступа к сетевым ЭВМ является введение для всех внешних машин общего секретного кода Kerberos KDC. Это делает эти машины "серверами", а не рабочими станциями. Этот общий секретный код может быть затем использован для шифрования всех обменов между машинами, обеспечивая безопасную передачу аутентификационной информации KDC.

Наконец, рабочие станции, которые удаленно доступны, могут использовать асимметричную криптографическую технологию для шифрования телекоммуникаций. Общедоступный ключ рабочей станции будет предоставлен всем клиентам. Пользователь может применить общедоступный ключ для шифрования пароля, а удаленная система дешифрует его и аутентифицирует пользователя без угрозы раскрытия пароля при транспортировке. Ограничением этой системы безопасности, ориентированной на рабочую станцию заключается в том, что она не аутентифицирует индивидуальных пользователей, а только индивидуальные рабочие станции. В некоторых средах, например, в многоуровневых правительственных системах безопасности необходима аутентификация пользователь-пользователь.

4. Раздача ключей и управление

Управление доступом для ключей является самой тяжелой проблемой, с которой приходится сталкиваться при обеспечении аутентификации в больших сетях Интернет. Протокол **Нидхема-Шрёдера** [NS78, NS87], который используется в системе Kerberos, базируется на централизованном сервере ключей. В больших корпоративных сетях требуется значительное число таких ключевых серверов, по крайней мере, один ключевой сервер на каждый административный домен. Существует также нужда в механизмах для отдельных ключевых серверов, необходимых для координирования генерации ключей сессий участников в различных административных доменах.

Большинство алгоритмов шифрования с использованием общедоступных ключей требуют достаточно больших вычислительных мощностей и по этой причине они неидеальны для шифрования пакетов в сети. Однако асимметричное свойство делает их очень полезными в начале сессии для получения симметричных ключей сессии.

На практике, коммерческий сектор, вероятно, использует асимметричный алгоритм для цифровых подписей и пересылки ключей, но не для массового шифрования данных. Для целей пересылки ключей можно использовать алгоритмы **RSA** и **Диффи-Хелмана** [DH76].

Преимуществом асимметричной методики является отсутствие необходимости иметь центральный сервер для хранения и рассылки ключей. Система PEM использует цифровые подписи для аутентификации общедоступных ключей пользователей [Kent93]. Результатом этой операции является сертификат, который содержит общедоступный ключ партнера. Сертификаты ключей могут рассылаться различными способами. В одном из вариантов рассылка ключей встраивается в существующие службы каталогов. Это может быть сделано, например, путем расширения возможностей DNS и включения ключа ЭВМ в ресурсную запись нового типа.

Для мультикастных сессий, управление рассылкой ключей сложнее, так как число обменов, необходимых для используемых методик пропорционально числу участников.

5. Аутентификация сетевых услуг

Кроме необходимости аутентификации пользователей и ЭВМ друг другу, многие сетевые услуги сами нуждаются в аутентификации.

Наиболее общий случай в настоящее время - это отсутствие поддержки в протоколе какой-либо аутентификации. Bellovin и другие документировали многие случаи, когда *существующие протоколы могут использоваться для атаки удаленной ЭВМ, так как там не существует встроенной процедуры аутентификации* [Bellovin89].

Некоторые протоколы предоставляют возможность передачи незащищенных паролей совместно с протокольной информацией. Исходные протоколы SNMP использовали этот метод, многие маршрутные протоколы продолжают его использовать и сейчас [Moy91, LR91, CFSD88]. Этот метод полезен, так как несколько повышает безопасность передачи.

Дополнительные материалы для изучения

Аутентификация в Internet

Существует много протоколов, которые нуждаются в поддержке более строгих аутентификационных механизмов. Например, известно, что протокол SNMP нуждается в существенном усилении аутентификации. Это вызвало разработку протоколов Secure SNMP, которые поддерживают опционную аутентификацию, используя цифровую подпись и опционное шифрование с привлечением алгоритма DES. Цифровые подписи, используемые в Secure SNMP, базируются на добавлении криптографической контрольной суммы к SNMP-информации. Криптографическая контрольная сумма вычисляется с использованием алгоритма MD5 и секретного кода, используемого совместно обоими партнерами обмена.

Технология цифровой подписи должна рассматриваться как необходимое средство при разработке новых технологий аутентификации (но не конфиденциальности). Цифровые подписи могут использовать ключи и методы как симметричной, так и асимметричной криптографии. Если доступна централизованная система распределения ключей, опционная поддержка цифровой подписи может быть обеспечена для большинства протоколов с минимальными издержками. Каждый протокол может столкнуться проблемой пересылки ключей и установки параметров обмена, и это приведет к усложнению использования техники цифровой подписи.

Для случаев, когда требуется аутентификация и конфиденциальность для схемы коммуникации ЭВМ-ЭВМ, может быть применено шифрование, базирующееся на симметричной или асимметричной схеме, или даже на их комбинации. Использование асимметричной криптографии упрощает управление раздачей ключей. Каждая ЭВМ шифрует свою информацию перед отправкой, безопасность же внутри машины обеспечивается средствами операционной системы ЭВМ.

В некоторых случаях, возможно включающих e-mail, может оказаться желательным обеспечить безопасность в пределах приложения на уровне пользователь-пользователь, а не ЭВМ-ЭВМ. Безопасная почта PEM использует именно этот подход.

6. Будущие направления

Просматривается тенденция внедрения все более строгих механизмов аутентификации. Следует ожидать введения не раскрывающей пароли аутентификации и более широкого использования механизмов с общедоступным ключом. Растет важность аутентификации сессий и процессов, проблема целостности и конфиденциальности сообщений при передаче по сетевым каналам. Так как коммуникации ЭВМ-ЭВМ становятся все более важными, протоколы связи человек-машина становятся менее существенными.

Использование криптосистем с общедоступным ключом для аутентификации пользователь-ЭВМ упрощает многие аспекты, но хранение простых паролей, а также общедоступных и секретных ключей остается актуальной проблемой. Следует учитывать, что размер общедоступного ключа, используемого в настоящее время, по меньшей мере, составляет 500 бит. В будущем, вероятно, будут применяться еще более длинные ключи. Таким образом, пользователи могут хранить свои ключи в виде пригодном для электронного считывания.

Использование ROM, такой как флоппи-диск или магнитная карточка может решить эту проблему, но тогда пользователь неизбежно доверяет свои ключи считывающему устройству. Применение смарт-карты, совмещающей память и программу, более предпочтительно. Такие приборы могут обеспечить аутентификацию без риска разглашения секретных кодов, которые они хранят. Они могут также взаимодействовать с пользователем, осуществляя простую аутентификацию при разблокировании карты. Применение криптосистем с общедоступным ключом при аутентификации ЭВМ-ЭВМ лишено проблем запоминания ключей, которые характерны для интерфейсов человек-машина. Многопользовательская ЭВМ может запоминать свои ключи в области памяти, защищенной от доступа пользователей.

Если рассматривать существующие симметричные алгоритмы как одноключевые, а асимметричные, такие как RSA в качестве двухключевых систем, можно предположить появление в будущем N-ключевых методик (где N больше 2). Если бы такая N-ключевая технология существовала, она могла бы использоваться для реализации масштабируемых протоколов для рассылки ключей в случае мультикастинга. В настоящее время ведется разработка технологии CBT (Core Based Tree), предназначенной для решения подобной задачи [BFC93].

WEB-сервисы с контролем идентификации

Данный подраздел подготовлен на основе Identity-Enabled Web Services. Standards-based Identity for WEB 2.0 (Ping Identity Corporation). Появление **SAML** (Security Assertion Markup Language) сделало возможным обеспечение безопасности приложений WEB 2.0 для сервис провайдеров, которые используют протокол **SOAP** (Simple Object Access Protocol). SAML предоставляет механизм, который позволяет транспортировать параметры идентификации на всех уровнях. Сервис **STS** (WS-Trust Security Token Service) позволяет преобразовывать параметры безопасности (token) из формата SAML в формат, специфический для домена, и наоборот. STS упрощает предоставление WEB-сервисов с контролем идентификации.

WEB-сервисы являются шлюзом, который обеспечивает совместимость самых разных программных технологий. В последнее время стала популярной технология **SOA** (Service Oriented Architecture), которая часто совмещается с SOAP и **REST** (Representational State Transfer).

Дополнительные материалы для изучения

WEB-сервисы с контролем идентификации

Базовым принципом такой архитектуры является обеспечение совместимости базовых стандартов (XML, SOAP, WS-Security и HTTP). Когда источник запроса WEB-сервиса взаимодействует с WEB-сервис провайдером, они не должны зависеть от деталей, скрытых внутри конкретных программных реализаций. Взаимодействие между ними происходит посредством стандартных сообщений.

До последнего времени большинство WEB-сервисов базировалось на аутентификации пользователя системного или прикладного уровня. В результате WEB-сервис провайдер проверяет идентичность источника WEB-запроса (приложения, выдавшего SOAP-запрос), анализируя, содержатся ли параметры идентификации в теле сообщения. Эта модель не вполне безопасна.

Некоторое улучшение безопасности дает использование встроенного механизма аутентификации, имеющегося в транспортных протоколах, таких как HTTP и TLS. такое решение заметно улучшает безопасность канала. В результате сервис провайдер может быть уверен, что сообщение послано узлом, заслуживающим доверия. Однако содержимое сообщения остается неподконтрольным и допускает возможность для атаки инсайдера.

Существует три открытых стандарта - **WSS** (Web Services Security), **SAML** (Security Assertion Markup Language) и **WS-Trust** (Web Services Trust), которые способны гарантировать идентификацию пользователя путем включения нужной информации в SOAP-запрос.

Создатель системы может располагать следующими сервисами безопасности:

- Конфиденциальность
- Целостность
- Аутентификация
- Авторизация

WEB-сервисы с контролем идентификации

Однако, так как архитектура WEB-сервисов предполагает возможность нахождения источника запроса и сервера в разных доменах, совмещение аутентификации и авторизации становится невозможным.

WS-Security является стандартом безопасности WEB-сервисов, широко используемым практически во всех SOAP-системах, таких как IBM WebSphere, Microsofy .Net и Apache Axis. WS-Security не определяет нового механизма безопасности, он описывает, как использовать существующие стандарты безопасности для обеспечения конфиденциальности, целостности сообщений, аутентификации в рамках SOAP-сообщений.

Конфиденциальность и целостность сообщений реализуются за счет криптования XML и XML-подписи (стандарты W3C). WS-Security определяет технику использования электронной подписи и криптозащиты в SOAP.

Аутентификация и авторизация используют разные профайлы для передачи маркеров (token) безопасности в заголовках сообщений WS-Security. Такие маркеры могут содержать идентификационные данные пользователя. Решение об авторизации принимаются на основании этой информации. Повторная аутентификация для предоставления определенного сервиса уже не требуется - работает принцип **SSO** (Single Sign-On). Появляются специальные токены в виде USB-модулей.

WS-Security определяет профайлы для различных типов маркеров безопасности. Сюда входят профайлы для Kerberos ticket'ов, сертификатов X.509, пар имя/пароль, утверждений SAML и лицензий XRML.

Дополнительные материалы для изучения
WEB-сервисы с контролем идентификации

WS-Trust является стандартом **OASIS** (Organization for the Advancement of Structured Information Standards), который определяет протокол сообщений для получения или проверки маркеров безопасности, генерируемых службой STS.

Маркеры безопасности могут быть затем переслан в заголовках WS-Security запросов SOAP с использованием профайла маркеров WS-Security. WS-Trust обеспечивает возможность передавать сообщения безопасности сервисам через неоднородную среду.

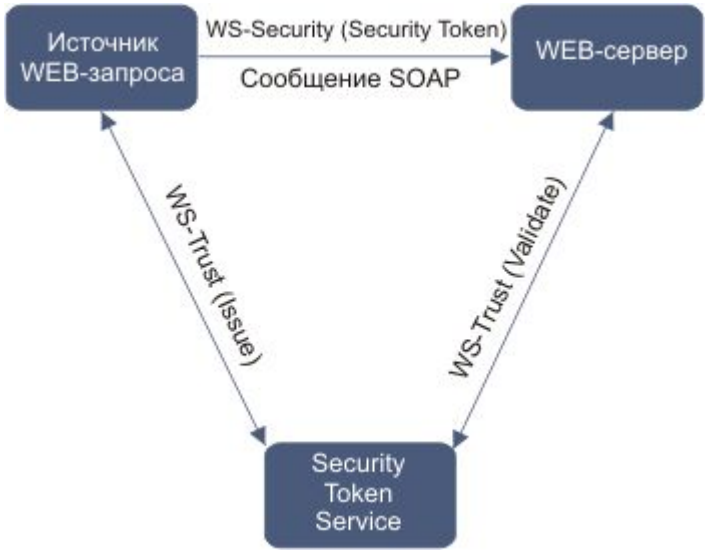

Рис. 1. Безопасность WS передает идентификационные данные с помощью Security Token, созданного STS

Стандарт SAML также является продуктом OASIS. Этот стандарт определяет протоколы и профайлы для объединения различных прикладных требований. Наиболее известным примером этих приложений является реализация безопасности браузера на основе технологии SSO.

Спецификация SAML определяет также маркер безопасности, вызванный утверждением SAML. Утверждение SAML является безопасным предложением идентичности, которое может использоваться совместно с WS-Security для реализации аутентификации, SSO и авторизации между источником запроса сервиса и WEB-сервис провайдером.

SAML хорошо приспособлен для реализации SSO, базирующейся на браузере. WS-Security и WS-Trust образуют опорную магистраль для передачи маркеров безопасности внутри системы.

Источник запроса WEB-услуги взаимодействует с STS (служба маркеров), которая формирует *утверждение SAML*, которое характеризует идентичность пользователя. Источник запроса помещает *утверждение SAML* в WSS-заголовок каждого SAML-запроса. STS становится агрегатором и арбитром безопасности.

Провайдер WEB-услуг должен доверять STS, которая сформировала *утверждение SAML*. В противном случает провайдер может использовать STS, чтобы проверить утверждение SAML самостоятельно. Утверждение SAML содержит данные, которые позволяют сервис провайдеру принять правильные решения в отношении аутентификации и авторизации.

WEB-сервисы с контролем идентификации

Существующие провайдеры WEB-сервисов нуждаются либо в идентификаторе пользователя, либо в достойном доверия указании относительно роли пользователя, прежде чем принять запрос.

Рис. 2. Web-сервис провайдер должен неявно доверять приложению

Такое установление доверия может быть реализовано разными путями. Web-сервис провайдер может решить доверять приложению, которое прислало сообщение SOAP. В этом случае Web-сервис провайдер верит, что приложение источника запроса имеет право послать запрос и идентичность пользователя как-то проверена. По существу, провайдер доверяет всем сообщениям, пришедшим от источника запросов.

Рис. 3. Идеальная ситуация. Web-сервис провайдер доверяет пользователю, обеспечивая безопасность end-to-end

WEB-сервисы с контролем идентификации

Альтернативой является ситуация, когда Web-сервис провайдер аутентифицирует и доверяет объекту, который отправляет запрос. В этом случае Web-сервис провайдер ищет подтверждение того, что запрос отправил действительный пользователь.

Утверждение SAML может предоставить данные, идентифицирующие приложение (или пользователя) источник запроса, пославшее сообщение SAML. В любом случае отправитель запроса должен позаботиться о маркере безопасности, чтобы получить доступ к WEB-сервисам из различных доменов безопасности. Существует несколько способов решить эту проблему:

- Web-сервис провайдер может верить или проверять маркеры безопасности различных источников запросов из разных доменов безопасности. Это становится достаточно дорогостоящим процессом, когда число источников запросов становится значительным. Например, Web-сервис провайдер доверяет Kerberos-тикитам пользователей, посылающим запросы из под Windows, и cookie сессии от пользователей, входящих через определенный портал.
- Источник запроса может получить различные маркеры безопасности для доступа к Web-сервис провайдерам из разных доменов безопасности. Это становится дорогим решением, когда сильно увеличивается число доменов безопасности.
- Источник запроса и Web-сервис провайдер могут использовать общий маркер безопасности и выполнять преобразование локальных маркеров безопасности в общий формат. Это решение проще и лучше масштабируется.

Практика последнего времени склоняется к третьему, более дешевому и гибкому варианту.

SAML является стандартом OASIS, который определяет XML-протоколы и профайлы для целей идентификации. Утверждение SAML является безопасным предложением, характеризующим идентичность, которое может быть использовано WS-Security для выполнения аутентификации, SSO и авторизации между источником запроса и Web-сервис провайдером. WS-Security определяет специфический профайл, который описывает то, как использовать утверждение SAML с WS-Security. Этот профайл называется SAML Token profile.

Как маркер безопасности утверждение SAML обладает многими уникальными свойствами, которые делают крайне полезным для WEB-сервисов, управляемых идентификаторами. К особенностям *утверждения SAML* следует отнести:

- это открытый стандарт
- расширяет возможности синтаксиса XML в XML-документах
- поддерживает неоднородную среду
- является гибким и расширяемым
- является опционно самопроверяемым

Cookie сессии. Все маркеры SSO и сессии, используемые системой управление WEB-доступа, являются частными по своей природе. Все они передают сходную информацию, но имеют разные синтаксические особенности и обеспечивают безопасность по-разному.

Kerberos Tickets. Kerberos является открытым стандартом, но он не предназначался для междоменного использования. Достаточно трудно сделать так, чтобы тикит, сформированный в одном домене безопасности, мог быть применен для доступа к ресурсам в другом домене.

Дополнительные материалы для изучения

WEB-сервисы с контролем идентификации

Сертификаты X.509. Модель инфраструктуры с открытым ключом **PKI** и сопряженные с ней сертификаты X.509, становится проблематичной при попытке использовать ее для идентификации людей. Генерация большого числа сертификатов X.509 для идентификации пользователей в пределах домена безопасности достаточно сложное дело.

Технология *утверждений SAML* базируется на сертификатах X.509 и на модели PKI. Для организации междоменной схемы SSO требуется лишь ограниченного числа сертификатов и сопряженных с ними ключей. Сертификат X.509, сформированный сертификационным центром Y и заявление "I am Security Domain A" используется для подписи утверждений SAML, которые транспортируют идентичность пользователя, которая гласит "I am User X from Security Domain A".

Утверждение SAML станет универсальным языком для WEB-сервисов с контролем идентичности. Его можно рассматривать как универсальный формат маркеров безопасности.

STS является функцией, определенной спецификацией WS-Trust. Отправитель запроса взаимодействует с STS для получения маркера безопасности для работы с сообщениями SOAP. WEB-сервис провайдер взаимодействует с STS, чтобы проверить маркеры безопасности, которые получены в сообщении SOAP. STS является **арбитром** между различными форматами маркеров, такими что сообщение SOAP может быть исполнено с полным учетом контекста безопасности запроса.

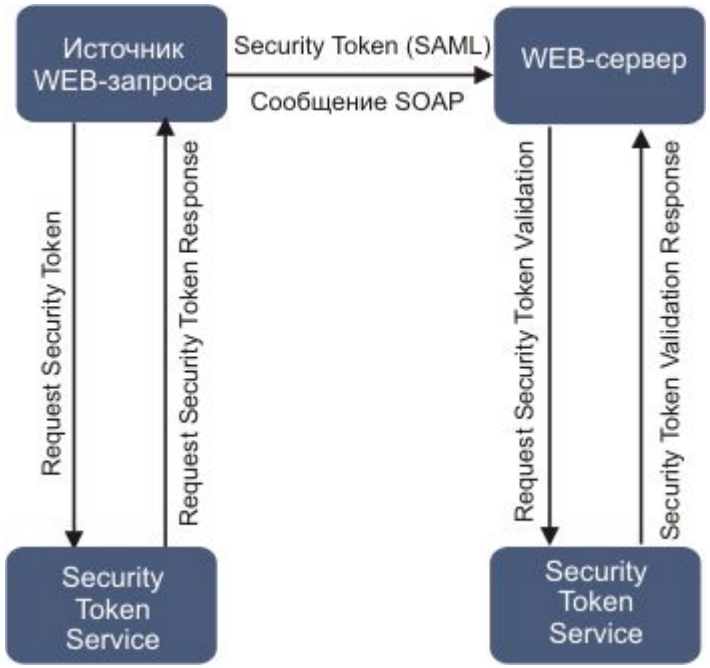

Рис. 4. Арбитраж безопасности нескольких доменах в рамках SAML

Обработка ансамбля идентификации является многоступенчатым и повторяемым процессом, который реализуется вне зависимости от протоколов и профайлов, которые используются для перемещения маркеров безопасности через сеть.

Дополнительные материалы для изучения
WEB-сервисы с контролем идентификации

Процесс обработки ансамбля идентификации включает в себя следующие три фундаментальные процедуры:

- **Аутентификация и установление доверия**. Аутентификация необходима, чтобы проверить идентичность пользователя (представляемую маркером безопасности) в одном домене безопасности, перед генерацией другого маркера безопасности, который обеспечит доверие к сообщениям в домене партнера. Доверие устанавливается в результате обмена маркерами безопасности.
- **Идентичность пользователя**. Пользователь может идентифицироваться разными идентификаторами в разных доменах и в разных его ролях. Предполагается, что маркер безопасности содержит в себе информацию, однозначно определяющую личность и роль пользователя. Идентификационные данные могут быть получены из маркера или из внешних информационных источников, таких как LDAP-каталог. Идентификационная информация может включать в себя значения атрибутов, таких как e-mail, роль, адрес, любимый цвет и т.д.
- **Авторизация, аудит и предоставление данных**. Аудит гарантирует то, что соответствующие данные пользователя и его партера имеются, остаются корректными для SLA и соответствуют существующим требованиям. Интегрированная авторизация может быть базовой, гарантирующей, что в маркере безопасности имеется корректный идентификатор роли пользователя. *Предоставление данных* предполагает динамическое обновление идентификационных данных пользователя, хранящейся в различных доменах.

Дополнительные материалы для изучения
WEB-сервисы с контролем идентификации

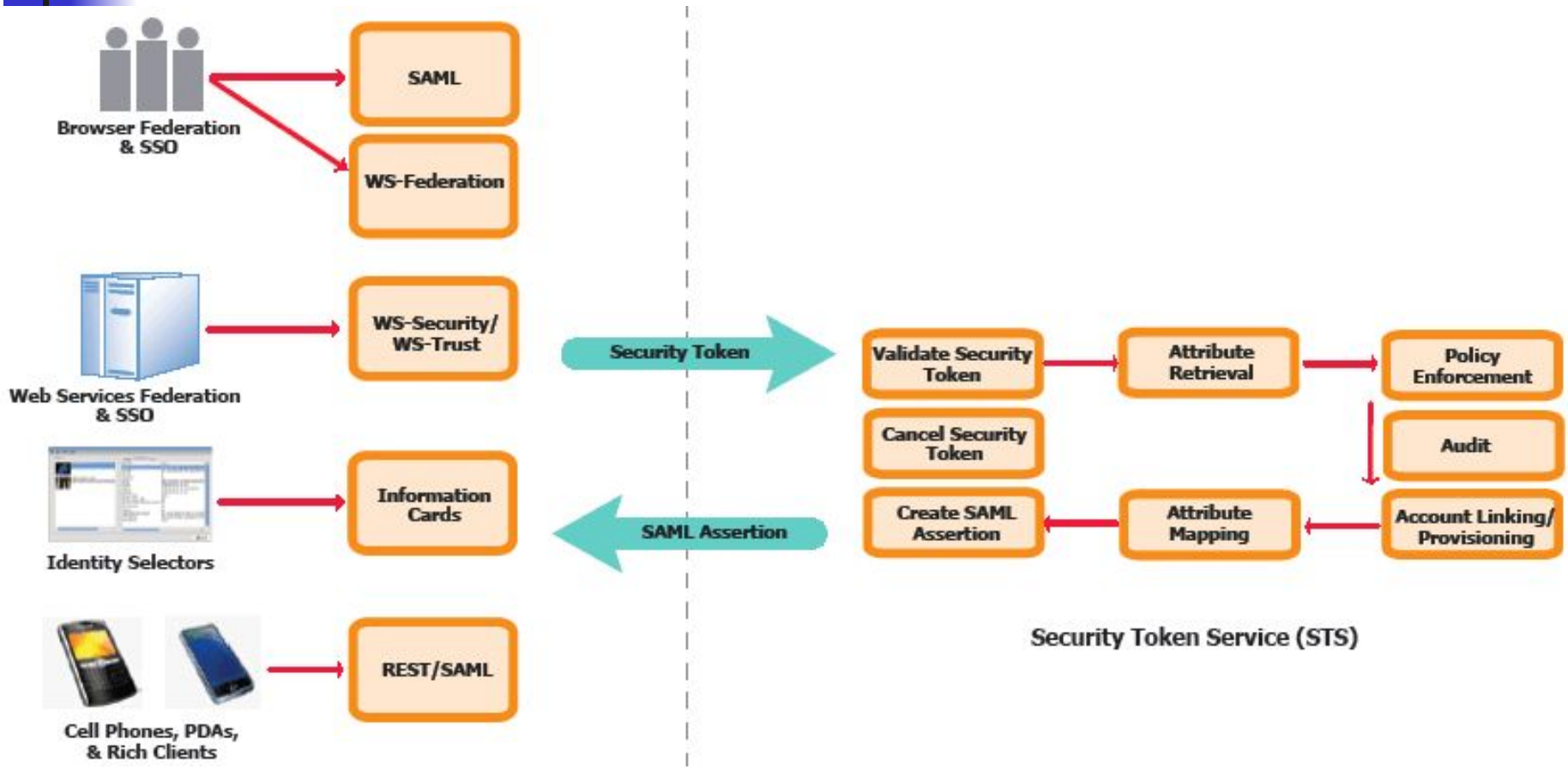

Рис. 5. Сервисы маркеров безопасности транслируют их в *утверждения SAML*

Дополнительные материалы для изучения
WEB-сервисы с контролем идентификации

В качестве примера источник запроса WEB-услуги может попросить STS генерировать утверждение SAML, которое характеризует идентичность пользователя или приложения, которое ссылается на тело сообщения SOAP. Отправитель запроса должен доказать, что он действительно авторизован посылать запрос утверждения SAML для данного пользователя. В случае более сложного приложения это может быть тикит Kerberos, сформированный активным каталогом Microsoft. В случае WEB-портала это может быть Cookie сессии, сформированное программой управления WEB-доступом. WEB-сервис провайдер может обрабатывать запросы SOAP от любых источников запроса, которые предоставляют идентификационные данные о пользователе в соответствии со стандартом.

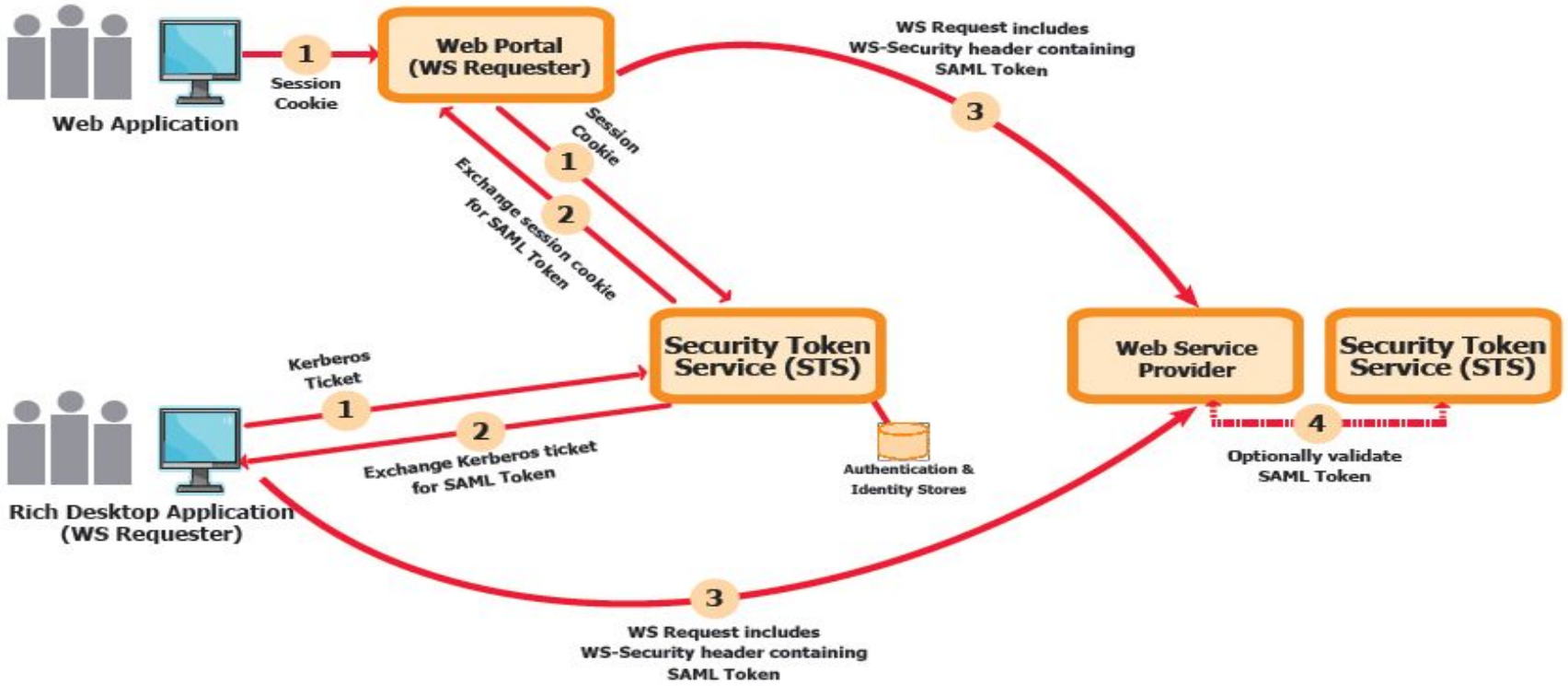

Рис. 6. Сервисы маркеров безопасности транслируют их в *утверждения SAML*

Дополнительные материалы для изучения

WEB-сервисы с контролем идентификации

В принципе должен быть реализован вариант автономной системы контроля идентичности сервер/STS. Такая архитектура делает возможным:

- Быстрота адаптации к изменению числа клиентов, приложений и партнеров
- Централизация контроля безопасности, уменьшающая степень риска, и простота согласования
- Интеграция как внутренних так и внешних доменов безопасности
- Гибкость хостинг-опций для WEB-сервисов

Автономная система идентификации сервер/STS позволяет сконцентрировать обработку и администрирование в одном месте.

Сопоставление разных видов аутентификации

Разные виды аутентификации обеспечивают различную мобильность пользователя и надежность. Мобильность пользователей сама по себе существенно усложняет их аутентификацию (переменный IP-адрес, дополнительные возможности атак типа MITM). Смотри **рис. 7.**

Соотношение мобильности и эффективности аутентификации для разных технологий

OTP - одноразовые пароли; **OOB** - Out of Band Authentication; **TPM** -Trusted Platform Module

Аутентификация OOB (внеканальная - Out of Band) базируется на использовании двухфакторной модели с программной аутентификацией и является еще одной разновидностью идентификации, которая предполагает наличие отдельных каналов для аутентификации и передачи данных. В одной из версий OOB предполагается применение SMS, посылаемых на мобильный телефон с последующим голосовым вызовом для аутентификации. Этот вид идентификации недостаточно безопасен, но в сочетании с другими может оказаться достаточно надежным. В последнее время появились новые виды USB-аутентификаторов.

Аутентификация с помощью SMS

Актуальность обеспечения безопасности мобильных средств коммуникации, например ip-phone, стимулирует новые разработки в этой области. Среди них можно назвать аутентификация с помощью SMS-сообщений. Процедура такой аутентификации включает в себя следующие шаги:

1. Ввод имени пользователя и пароля
2. Сразу после этого PhoneFactor (служба безопасности) присылает одноразовый аутентификационный ключ в виде текстового SMS-сообщения.
3. Полученный ключ используется для аутентификации

Привлекательность данного метода заключается в том, что ключ получается не по тому каналу, по которому производится аутентификация (out-of-band), что практически исключает атаку типа "человек-по-середине". Дополнительный уровень безопасности может дать требование ввода PIN-кода мобильного средства.

Авторизация и аутентификация

Очень часто пользователи не различают авторизацию и аутентификацию, хотя эти две процедуры имеют разное назначение. Авторизация (процедура Logon/Logoff) требует ввода имени и пароля. Пароль и имя могут передаваться открытым текстом или зашифрованным. Аутентификация предполагает проверку идентификацию клиента. Аутентификация может производиться не на том компьютере, где производится авторизация. Аутентификация может включать процедуру сертификации или проверку IP-адреса, откуда производится попытка аутентификации. Организовать проверку IP-адреса можно, поместив список разрешенных адресов в файл /etc/hosts_allow (ОС Linux).

Тип Logon	Описание
2	Интерактивный (с консоли)
3	Сетевой (т.е.соединение с разделяемым ресурсом откудо-то через сеть)
4	Batch (запуск задания по времени)
5	Сервис (запуск сервиса)
10	Удаленный интерактивный (терминальный сервис, использование удаленного компьютера в качестве услуги или удаленная помощь)

Дополнительные материалы для изучения

WEB-сервисы с контролем идентификации

Современные системы управления доступом к ресурсам используют комплексную модель **AAA** (Authentication, Authorization, and Accounting). В эту схему часто входит и контроль принадлежности того или иного ресурса. Смотри The technology of trust. A short course in authentication, authorization, and accounting — the security mechanisms that keeps modern Internet-connected computing humming.

Одним из источников угрозы является также ненадлежащий уровень доступа (привилегий) при выполнении тех или иных приложений. 88% из почти 12000 инцидентов в 2013 году были связаны неправильно выбранным уровнем доступа инсайдеров ("Privileged Identity Management", GEMALTO.COM. Security to be free). Иллюстрацией этой проблемы может быть случай с Э.Сноуденом. Дифференциация уровней доступа инсайдеров является одной из основ безопасности. Важным средством обеспечения безопасности является многофакторная аутентификация. Оптимальным путем решения проблемы можно считать использование смарткарт.

Схема реализации применения смарткарты и сертификатов для аутентификации показана на рис. 8. **RDP** - Remote Desktop Protocol.

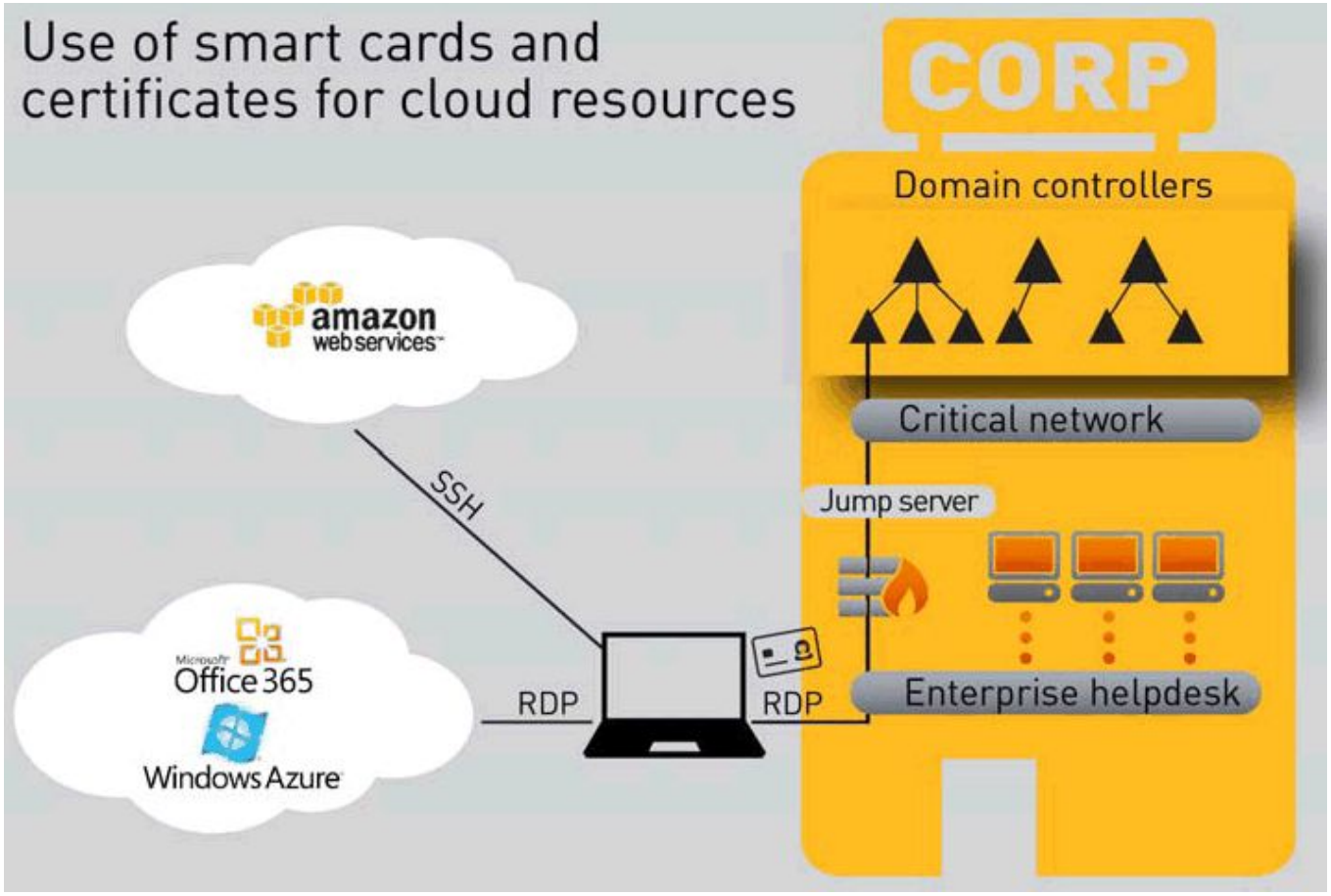

Рис. 8. Использование смарткарт и сертификатов для доступа к облачным ресурсам

Протокол аутентификации Нидхэма-Шредера в случаях симметричной и асимметричной системы шифрования

Протокол Нидхэма-Шрёдера предназначен для решения проблемы аутентификации (см., например, http://www.cs.sunysb.edu/~zhaoming/np.html).

Протоколу уже более 20 лет. Алгоритм предназначен для организации аутентифицированного канала между разными ЭВМ в сети по схеме точка-точка. Задача решается с помощью одного или двух серверов аутентификации с использованием общедоступных или общих секретных ключей. Данный протокол предоставляет децентрализованную услугу аутентификации.

Операция аутентификации может охватывать несколько процессов.

1. Установление виртуального канала двунаправленного обмена сообщениями между двумя субъектами, работающими на разных ЭВМ.
2. Установление однонаправленного обмена, который, например, имеет место при отправке почты. Здесь ситуация осложняется тем, что субъекты могут не быть одновременно доступны через сеть и не могут непосредственно обмениваться сообщениями.
3. Коммуникация, при которой источник информации и ее целостность может гарантироваться третей стороной.

Безопасная передача данных по сети, которая сама не является безопасной, предполагает шифрование передаваемой информации. Будем предполагать, что каждая из сторон, участвующих в обмене, способна шифровать и дешифровать данные. Протокол Нидхэма-Шрёдера может работать как для симметричной, так и несимметричной схем шифрования (с общим секретным ключом и с двумя парами ключей, соответственно). Будем также считать, что злоумышленник может подключить свою ЭВМ в любую точку пути, по которому происходит обмен и, таким образом, способен перехватить, воспроизвести или исказить любое сообщение. ЭВМ же субъектов обмена и сервер аутентификации предполагаются защищенными от вторжения.

Протокол аутентификации Нидхэма-Шредера в случаях симметричной и асимметричной системы шифрования

Сервер аутентификации может предоставить идентификационную информацию, вычисляемую на основе секретного ключа субъекта аутентификации. Сначала рассмотрим вариант с использованием симметричного шифрования/дешифрования (один ключ).

При схеме шифрования с одним ключом, предполагается, что секретный ключ известен обоим субъектам обмена (A и B) и серверу аутентификации. Инициатором обмена будем считать субъекта A. Сообщения, посылаемые от A к B, могут быть дешифрованы только B и субъект B должен быть уверен, что сообщение пришло именно от A. В начале предположим, что оба субъекта находятся в области действия общего сервера аутентификации (AS). AS знает секретные ключи субъектов A и B (KA и KB, соответственно).

Обмен начинается с того, что субъект A генерирует свой идентификатор I_{A1}, который будет использоваться только один раз. Первое сообщение, посылаемое от A к AS, содержит:

$$A \rightarrow AS: \quad (A, B, I_{A1}) \qquad (1)$$

Здесь предполагается, что сообщение послано открытым текстом, но в принципе оно может быть и зашифровано с использованием ключа KA

$$A \rightarrow AS: \quad (A, B, I_{A1})^{KA} \qquad (1.1)$$

Получив это сообщение AS, извлекает из базы данных секретные ключи KA и KB, а также вычисляет новый ключ CK (ключ сессии), который будет использован для осуществления процедуры аутентификации. Этот новый ключ должен быть непредсказуемым, он используется только для одной операции аутентификации.

Далее AS посылает субъекту **A** следующее сообщение:

$$AS \rightarrow A: \quad (I_{A1}, B, CK, \{CK, A\}^{KB})^{KA} \qquad (1.2).$$

Верхний индекс в данном выражении означает, что содержимое в скобках зашифровано с использованием ключа-индекса. KA и KB - секретные ключи субъектов **A** и **B**, соответственно.

Протокол аутентификации Нидхэма-Шредера в случаях симметричной и асимметричной системы шифрования

Так как выражение (I_{A1}, B, CK, {CK, A}^{KB}) зашифровано ключом КА, то только субъект **A** может его дешифровать и прочесть. Субъект **A** проверяет наличие идентификатора I_{A1} (это подтверждает то, что данное сообщение является откликом на сообщение **A**), и имени субъекта, с которым **A** намерен обмениваться данными (**B**). В результате дешифровки сообщения от AS **A** получает во владение рабочий ключ СК. Наличие **B** в сообщение является обязательным. В противном случае злоумышленник может заменить **B** на, например, **X** в сообщении (1), и в дальнейшем **A** будет взаимодействовать с **X** а не с **B**, сам того не подозревая. Заметим, что часть текста {CK, A}^{KB} субъект А прочесть не может.

Если все прошло нормально, субъект **A** посылает **B** следующее сообщение:

A B: $\rightarrow$ {CK, A}^{KB} (1.3)

Нетрудно видеть, что содержимое {CK, A}^{KB} является частью сообщения, полученного от AS. Дешифровать это послание может только субъект **B**, так как оно зашифровано его секретным ключом. После дешифровки **B** также становится владельцем ключа сессии СК. Наличие **A** в сообщении подтверждает факт, что код получен именно от данного субъекта. Все обмены между **A** и **B** далее будут выполняться с использованием ключа шифрования СК. Чтобы сделать схему симметричной и уменьшить вероятность атаки воспроизведения, **B** следует послать **A** свой идентификатор:

B A: $\rightarrow$ {I_B}^{CK} (1.4)

зашифрованный ключом СК. При этом ожидается отклик:

A B: $\rightarrow$ {I_B-1}^{CK} (1.5)

Таким образом, в данной версии протокола используется 5 сообщений.

Протокол аутентификации Нидхэма-Шредера в случаях симметричной и асимметричной системы шифрования

Злоумышленник не может имитировать такой обмен, так как не владеет ключом СК. Это число для регулярно взаимодействующих партнеров можно сократить до трех, убрав обмен сообщениями 1.1 и 1.2. При этом ключ СК будет использоваться многократно. Здесь желательно заменить обмены 1.3 и 1.4 на:

A B: $\rightarrow \{CK, A\}^{KB}, \{I_{A2}\}^{CK}$ (1.3a)

B A: $\rightarrow \{I_{A2} - 1, I_B\}^{CK}$ (1.4a)

Теперь рассмотрим вариант протокола для случая асимметричного шифрования (двух ключевая схема).

Предполагается, что субъекты **A** и **B** вычислили пары ключей (PKA-SKA) и (PKB-SKB), соответственно. Имена ключей, начинающиеся с буквы P, относятся к общедоступным ключам (public), а имена, начинающиеся с буквы S, - к секретным. Инициатором, как и в предыдущем случае, будем считать субъект **A**. Обмен начинается с посылки AS запроса открытого ключа B (PKB).

A AS: $\rightarrow (A, B)$ (2.1)

AS откликнется сообщением:

AS A: $\rightarrow (PKB, B)^{SKAS}$ (2.2)

Сообщение зашифровано секретным ключом AS (SKAS). Открытый ключ AS (PKAS) предполагается **A** известным, что позволяет **A** успешно дешифровать данное сообщение. Здесь предполагается, что подмена ключей (SKAS-PKAS) злоумышленником невозможна.

Шифрование данных с использованием ключа SKAS не гарантирует конфиденциальности, но исключает модификацию сообщения по дороге (ведь никто посторонний не знает ключ SKAS). Важно, чтобы субъект **A** был уверен, что он получил именно PKB, а не что-то иное. Следующим шагом будет посылка сообщения от A к B:

A B: $\rightarrow \{I_A, A\}^{PKB}$ (2.3).

Протокол аутентификации Нидхэма-Шредера в случаях симметричной и асимметричной системы шифрования

Это сообщение может быть дешифровано только субъектом B. Смысл его заключается в том, что **A** уведомляет **B** о намерении установить с ним связь, и передает ему свой одноразовый идентификатор I_A. Далее **B** запрашивает у AS открытый ключ **A**:

B AS: $\rightarrow$ B, A (2.4)

AS B: $\rightarrow \{PKA, A\}^{SKAS}$ (2.5)

После этого производится взаимная аутентификация субъектов, завершающая сессию, для чего посылаются сообщения:

B A: $\rightarrow \{I_A, I_B\}^{PKA}$ (2.6)

A B: $\rightarrow \{ I_B \}^{PKB}$ (2.7)

Таким образом, в этом варианте аутентификация потребовала семи шагов, но 4 из них (2.1, 2.2, 2.4 и 2.5) могут быть устранены, если партнеры помнят общедоступные ключи друг друга. В этом случае схема становится эквивалентной приведенной выше версии с симметричным шифрованием.

Так как открытые ключи общедоступны, во многих случая для обеспечения большей достоверности следует использовать шифрование типа:

$\{\{сообщение\}^{SKA}\}^{SKB}$.

В реальной жизни субъекты не всегда могут находиться в пределах зоны ответственности одного общего сервера аутентификации. По этой причине в общем случае каждый из субъектов может иметь свой сервер аутентификации (ASA и ASB). Так как и в этом варианте перед субъектом **A** стоит задача сформировать для **B** сообщение типа $\{CK, A\}^{KB}$ (шаг 1.3). В вычисление таких выражений будут вовлечены оба сервера, так как только ASA может шифровать объекты посредством ключа KA, и только ASB может воспользоваться ключом KB. Не исключается необходимость обеспечения безопасного обмена между AS.

Протокол аутентификации Нидхэма-Шредера в случаях симметричной и асимметричной системы шифрования

Примерами такого обмена могут служить операции, завершающие сессию аутентификации:

ASA ASB: $\rightarrow$(CK, A, B, I_{A1}) (1.11)

ASB ASA: $\rightarrow$(CK, A)^{KB,} I_{A1}, A (1.12)

I_{A1} передается для того, чтобы сохранить состояние ASA между сообщениями 1.11 и 1.12.

При работе с открытыми ключами возможно непосредственное обращение **A** к ASB, если субъект **A** владеет общедоступным ключом PKASB. По минимуму аутентификация при асимметричной схеме шифрования требует пересылки трех сообщений.

Протокол Нидхэма-Шрёдера пригоден и для работы с электронными подписями. Электронная подпись, как обычно, формируется на основе дайджеста D (например, MD5) пересылаемого документа. Сначала рассмотрим вариант с традиционной схемой шифрования. Субъект A начинает передачу с посылки AS сообщения:

A AS: $\rightarrow$ (A, {D}^{KA}) (3.1)

AS откликается, послав:

AS A: $\rightarrow$ {A, D}^{KAS} (3.2)

Сообщение 3.2 зашифровано ключом AS и, следовательно, не может дешифровано **A**. Субъект **A** шлет документ субъекту **B** с блоком подписи, следующим за текстом. При получении **B** сначала дешифрует текст и вычисляет дайджест документа CD, затем посылает блок подписи в AS для дешифровки.

B AS: $\rightarrow$ B, {A, D}^{KAS} (3.3)

Сервер дешифрует блок подписи и возвращает результат **B**:

AS B: $\rightarrow${A, D}^{KB} (3.4)

Если возвращенный дайджест D соответствует CD, тогда субъект, упомянутый в 3.4, является отправителем подписанного текста. Если соответствия нет, это означает, что на этапах 3.1 -3.4 произошло искажение блока подписи или самого текста.

Протокол аутентификации Нидхэма-Шредера в случаях симметричной и асимметричной системы шифрования

В случае варианта шифрования с общедоступными ключами схема электронной подписи упрощается. В этом варианте можно даже не формировать дайджест можно послать текст, зашифрованный сначала секретным ключом **A**, а затем с привлечением общедоступного ключа **B**.

A B: $\rightarrow$ { (текст)^{SKA} }^{PKB}

Субъект-получатель **B** дешифрует полученный текст сначала с помощью своего секретного ключа (SKB), а затем с привлечением общедоступного ключа **A** (PKA). При такой схеме **A** не сможет отказаться от того, что именно он послал текст, так как только он владеет секретным ключом SKA. Прочесть же текст может только субъект **B**, так как только он владеет секретным ключом SKB.

В настоящее время разработан улучшенный протокол Нидхэма-Шрёдера (см. http://dimacs.rutgers.edu/Workshops/Security/program2/dedecker/node4.html).

[1] Roger M. Needham and Michael D. Schroeder, Using Encryption for Authentication in Large Networks of Computers. Communication of the ACM, V.21, N12, December 1978.

Электронная подпись

В конце любого письма мы привыкли ставить подпись с тем, чтобы уведомить получателя о том, кто является отправителем данного документа. Кроме того, подпись ответственного лица придает документу юридическую силу. По мере внедрения электронных средств доставки документов (факс и электронная почта) проблема их достоверности обрела крайнюю актуальность. Ведь копирование любой последовательности битов или пикселей не представляет никакой трудности. Современные телекоммуникационные каналы уязвимы для перехвата и искажения пересылаемых документов.

Рассмотрим сначала то, от каких действий злоумышленника должна защищать система идентификации.

- *Отказ от выполненных действий*. Субъект утверждает, что он не посылал некоторый документ, хотя на самом деле он его послал.
- *Модификация документа*. Получатель модифицирует полученный документ и утверждает, что именно такую версию документа он и получил.
- *Подделка*. Субъект фабрикует сообщение и утверждает, что оно ему прислано.
- *Перехват*. Злоумышленник **С** перехватывает сообщение, посланное **А** к **В** с целью модификации.
- *Маскировка*. Посылка сообщения от чужого имени.
- *Повтор*. Злоумышленник **С** посылает повторно сообщение от **А** к **Б**, перехваченное им ранее.

Решение практически всех этих проблем может быть реализовано с помощью электронной подписи, базирующейся на алгоритме RSA. Рассмотрим принципы, на которых базируется электронная подпись.

Электронная подпись

Пусть имеются секретные коды **d, p** и **q**, а также открытые **e** и **n=pq**. Пусть также **А** передает сообщение DATA адресату **Б**. Электронная подпись отправителя **А** базируется на его секретном ключе и открытом ключе получателя **Б**. Сначала отправитель с помощью хэш-функции (SHS - Secure Hash Standard; www.nist.gov/itl/div897/pubs/fip180-1.htp) генерирует дайджест своего сообщения длиной 160 бит (5 слов). Затем с помощью своего секретного ключа он формирует электронную подпись. При этом **А** не может отказаться от того, что именно он послал сообщение, так как только он знает свой секретный ключ. Электронную подпись нельзя использовать повторно и подписанный документ нельзя модифицировать, так как любые модификации неизбежно изменят его дайджест, а, следовательно, и электронную подпись. Получатель с помощью открытого ключа дешифрует код электронной подписи, а затем с использованием дайджеста проверяет ее корректность.

Национальный институт стандартов США принял стандарт DSS (Digital Signature Standard; www.itl.nist.gov/div897/pubs/fip198.htm), в основу которого легли алгоритмы Эль-Гамаля и RSA.

Рассмотрим алгоритмы вычисления дайджеста сообщения, электронной подписи и идентификации отправителя. Начнем с алгоритма **SHA** (Secure Hash Algorithm).

Сначала сообщение разбивается на блоки длиной 512 бит. Если длина сообщения не кратна 512, к последнему блоку приписывается справа 1, после чего он дополняется нулями до 512 бит. В конец последнего блока записывается код длины сообщения. В результате сообщение приобретает вид **n** 16-разрядных двоичных слов $M_1,M_2,\ldots,M_n$. M_1 содержит первый символ.

Электронная подпись

Алгоритм SHA использует 80 логических функций $f_0, f_1, \ldots, f_{79}$, которые производят операции над тремя 32-разрядными словами (B,C,D):

$f_t(B,C,D) = (B \text{ AND } C) \text{ OR } ((\text{NOT } B) \text{ AND } D)$ для $0 \le t \le 19$

$f_t(B,C,D) = B \text{ XOR } C \text{ XOR } D$ для $20 \le t \le 39$

$f_t(B,C,D) = (B \text{ AND } C) \text{ OR } (B \text{ AND } D) \text{ OR } (C \text{ AND } D)$ для $40 \le t \le 59$

$f_t(B,C,D) = B \text{ XOR } C \text{ XOR } D$ для $60 \le t \le 79$

В алгоритме используется также 80 констант $K_1, K_2, \ldots, K_{79}$:

K_t = 5A827999 для $0 \le t \le 19$

K_t = 6ED9EBA1 для $20 \le t \le 39$

K_t = 8F1BBCDC для $40 \le t \le 59$

K_t = CA62C1D6 для $60 \le t \le 79$

Вводится 5 переменных H_i инициализируемых как:

H_0 = 67452301 H_3 = 10325476

H_1 = EFCDAB89 H_4 = C3D2E1F0

H_2 = 98BADCFE

Делим массив M на группы из 16 слов $W_0, W_1, \ldots, W_{15}$ (W_0 самое левое слово).

Для t = 16 - 79 $w_t = S^1(W_{t-3} \text{ XOR } W_{t-8} \text{ XOR } W_{t-14} \text{ XOR } W_{t-16})$.

A^k означает операцию циклического сдвига влево на k разрядов.

Пусть теперь $A = H_0$, $B = H_1$, $C = H_2$, $D = H_3$, $E = H_4$.

for t = 0 to 79 do

TEMP = $S^5(A) + f_t(B,C,D) + E + W_t + K_t$. (TEMP - временная переменная).

$E = D$; $D = C$; $C = S^{30}(B)$; $B = A$; A = TEMP;

Пусть $H_0 = H_0 + A$; $H_1 = H_1 + B$; $H_2 = H_2 + C$; $H_3 = H_3 + D$; $H_4 = H_4 + E$.

Электронная подпись

В результате обработки массива M будет получено 5 слов H_0, H_1, H_2, H_3, H_4 с общей длиной 160 бит, которые и образуют дайджест сообщения. Полученная кодовая последовательность с высокой степенью уникальности характеризует сообщение. Любое редактирование сообщения практически неизбежно приведет к изменению дайджеста. Поскольку алгоритм вычисления дайджеста общеизвестен, он не может рассматриваться как гарантия предотвращения модификации сообщения. Смысл вычисления дайджеста заключается в уменьшении объема данных, подлежащих шифрованию. Для того чтобы превратить дайджест в электронную подпись надо воспользоваться секретным ключом. Схема реализации алгоритма DSA (Digital Signature Standard) показана на рис. 9.

Рис. 9. Схема вычисления и верификации электронной подписи (DSA)

Электронная подпись

DSA использует следующие параметры (www.itl.nist.gov/div897/pubs/fip186.htm):

p - простое число, которое при 512≤ L≤1024 удовлетворяет условию $2^{L-1} < p < 2^{L}$, L кратно 64.

q - простой делитель p-1, где $\mathbf{2^{159} < q < 2^{160}}$.

g = $\mathbf{h^{(p-1)/q}\ mod\ p}$, где h любое целое, для которого **1 < h < p-1** и $\mathbf{h^{(p-1)/q}\ mod\ p > 1}$.

x равно случайному или псевдослучайному целому числу, для которого 0 < x < q.

y = $\mathbf{g^{x}\ mod\ p}$.

k равно случайному или псевдослучайному целому числу, для которого 0 < k < q.

Целые **p, q** и **g** могут быть общедоступными и использоваться группой пользователей. Секретным и открытым ключами являются **x** и **y**, соответственно. Параметры **x** и **k** используются только для формирования электронной цифровой подписи и должны храниться в секрете. Параметр **k** генерируется для каждой подписи.

Подпись сообщения M представляет собой два числа **r** и **s**, вычисленные согласно формулам:

r = (gk mod p) mod q

s = ($\mathbf{k^{-1}}$(SHA(M) + xr)) mod q. (здесь k^{-1} величина обратная k).

SHA(M) - представляет собой дайджест сообщения M (160-битовая строка). После вычисления **r** и **s** следует проверить, не равно ли одно из них нулю.

Для верификации электронной подписи проверяющая сторона должна иметь параметры **p, q** и **g,** а также открытый ключ отправителя (подписанта) **y.**

Электронная подпись

Пусть **M`, r`** и **s`** представляют собой полученное сообщение и электронную подпись. Получатель начинает верификацию с проверки условия 0 < r` < q и 0 < s` < q. Если хотя бы одно из условий не выполнено, электронная подпись некорректна. Далее производится вычисление:

w = (s`)$^{-1}$ mod q
u1 = ((SHA(M`)w) mod q
u2 = ((r`)w) mod q
v = (((g)^{u1} (y)^{u2}) mod p) mod q.

Если **v = r`**, верификация подписи завершилась успешно и получатель может с высокой вероятностью быть уверен, что он получил сообщение от партнера, владеющего секретным ключом **x**, соответствующим открытому ключу **y**. Если же **v** не равно **r`**, то сообщение было модифицировано или подписано самозванцем. В ссылке 3 на предыдущей странице (www.itl.nist.gov/div897/pubs/fip186.htm) можно найти описание алгоритма нахождения (проверки) простых чисел и генерации псевдослучайных чисел.