11. Лекция: Протоколирование и аудит, шифрование, контроль целостности

Протоколирование и аудит

Основные понятия

Под ***протоколированием*** понимается сбор и накопление информации о *событиях*, происходящих в информационной системе. У каждого *сервиса* свой набор возможных событий, но в любом случае их можно разделить на внешние (вызванные действиями других сервисов), внутренние (вызванные действиями самого сервиса) и клиентские (вызванные действиями пользователей и администраторов).

Аудит – это анализ накопленной информации, проводимый оперативно, в реальном времени или периодически (например, раз в день). Оперативный аудит с автоматическим реагированием на выявленные нештатные ситуации называется активным.

Реализация протоколирования и аудита решает следующие задачи:

- обеспечение *подотчетности* пользователей и администраторов;
- обеспечение возможности реконструкции последовательности событий;
- обнаружение попыток нарушений информационной безопасности;
- предоставление информации для выявления и анализа проблем.

Протоколирование требует для своей реализации здравого смысла. Какие события регистрировать? С какой степенью *детализации*? На подобные вопросы невозможно дать универсальные ответы. Необходимо следить за тем, чтобы, с одной стороны, достигались перечисленные выше цели, а, с другой, расход ресурсов оставался в пределах допустимого. Слишком обширное или подробное протоколирование не только снижает производительность сервисов (что отрицательно сказывается на *доступности*), но и затрудняет аудит, то есть не увеличивает, а уменьшает информационную безопасность.

Разумный подход к упомянутым вопросам применительно к операционным системам предлагается в "Оранжевой книге", где выделены следующие события:

- вход в систему (успешный или нет);
- выход из системы;
- обращение к удаленной системе;
- операции с файлами (открыть, закрыть, переименовать, удалить);
- смена привилегий или иных атрибутов безопасности (режима доступа, уровня благонадежности пользователя и т.п.).

При протоколировании события рекомендуется записывать, по крайней мере, следующую информацию:

- дата и время события;
- уникальный идентификатор пользователя – инициатора действия;
- тип события;
- результат действия (успех или неудача);
- источник запроса (например, имя терминала);
- имена затронутых объектов (например, открываемых или удаляемых файлов);
- описание изменений, внесенных в базы данных защиты (например, новая метка безопасности объекта).

Еще одно важное понятие, фигурирующее в "Оранжевой книге", – *выборочное протоколирование*, как в отношении пользователей (внимательно следить только за подозрительными), так и в отношении событий.

Характерная особенность протоколирования и аудита – зависимость от других средств безопасности. Идентификация и аутентификация служат отправной точкой подотчетности пользователей, логическое управление доступом защищает

конфиденциальность и целостность *регистрационной информации*. Возможно, для защиты привлекаются и криптографические методы.

Возвращаясь к целям протоколирования и аудита, отметим, что обеспечение подотчетности важно в первую очередь как сдерживающее средство. Если пользователи и администраторы знают, что все их действия фиксируются, они, возможно, воздержатся от незаконных операций. Очевидно, если есть основания подозревать какого-либо пользователя в нечестности, можно регистрировать все его действия, вплоть до каждого нажатия клавиши. При этом обеспечивается не только возможность расследования случаев нарушения режима безопасности, но и откат некорректных изменений (если в протоколе присутствуют данные до и после модификации). Тем самым защищается целостность информации.

Реконструкция последовательности событий позволяет выявить слабости в защите сервисов, найти виновника вторжения, оценить масштабы причиненного ущерба и вернуться к нормальной работе.

Обнаружение попыток нарушений информационной безопасности – функция активного аудита, о котором пойдет речь в следующем разделе. Обычный аудит позволяет выявить подобные попытки с опозданием, но и это оказывается полезным. В свое время поимка немецких хакеров, действовавших по заказу КГБ, началась с выявления подозрительного расхождения в несколько центов в ежедневном отчете крупного вычислительного центра.

Выявление и анализ проблем могут помочь улучшить такой параметр безопасности, как доступность. Обнаружив узкие места, можно попытаться переконфигурировать или перенастроить систему, снова измерить производительность и т.д.

Непросто осуществить организацию согласованного протоколирования и аудита в распределенной разнородной системе. Во-первых, некоторые компоненты, важные для безопасности (например, маршрутизаторы), могут не обладать своими ресурсами протоколирования; в таком случае их нужно экранировать другими сервисами, которые возьмут протоколирование на себя. Во-вторых, необходимо увязывать между собой события в разных сервисах.

Активный аудит

Основные понятия

Под ***подозрительной активностью*** понимается поведение пользователя или компонента информационной системы, являющееся ***злоумышленным*** (в соответствии с заранее определенной политикой безопасности) или нетипичным (согласно принятым критериям).

Задача активного аудита – оперативно выявлять подозрительную активность и предоставлять средства для *автоматического реагирования* на нее.

Активность, не соответствующую политике безопасности, целесообразно разделить на *атаки*, направленные на незаконное получение полномочий, и на действия, выполняемые в рамках имеющихся полномочий, но нарушающие политику безопасности.

Атаки нарушают любую осмысленную политику безопасности. Иными словами, активность атакующего является разрушительной независимо от политики. Следовательно, для описания и выявления атак можно применять универсальные методы, инвариантные относительно политики безопасности, такие как сигнатуры и их обнаружение во входном потоке событий с помощью аппарата *экспертных систем*.

Сигнатура атаки – это совокупность условий, при выполнении которых атака считается имеющей место, что вызывает заранее определенную реакцию. Простейший пример сигнатуры – "зафиксированы три последовательные неудачные попытки входа в систему с одного терминала", пример ассоциированной реакции – блокирование терминала до прояснения ситуации.

Действия, выполняемые в рамках имеющихся полномочий, но нарушающие политику безопасности, мы будем называть злоупотреблением полномочиями. *Злоупотребления полномочиями* возможны из-за неадекватности средств разграничения доступа выбранной политике безопасности. Простейшим примером злоупотреблений является неэтичное поведение суперпользователя, просматривающего личные файлы других пользователей. Анализируя регистрационную информацию, можно обнаружить подобные события и сообщить о них *администратору безопасности*, хотя для этого необходимы соответствующие средства выражения политики безопасности.

Выделение злоупотреблений полномочиями в отдельную группу неправомерных действий, выявляемых средствами активного аудита, не является общепринятым, однако, на наш взгляд, подобный подход имеет право на существование и мы будем его придерживаться, хотя наиболее радикальным решением было бы развитие средств разграничения доступа (см. "Возможный подход к управлению доступом в распределенной объектной среде").

Нетипичное *поведение* выявляется статистическими методами. В простейшем случае применяют систему *порогов*, превышение которых является подозрительным. (Впрочем, "пороговый" метод можно трактовать и как вырожденный случай сигнатуры атаки, и как тривиальный способ выражения политики безопасности.) В более развитых системах производится сопоставление долговременных характеристик работы (называемых *долгосрочным профилем*) с *краткосрочными профилями*. (Здесь можно усмотреть аналогию биометрической аутентификации по поведенческим характеристикам.)

Применительно к средствам активного аудита различают *ошибки первого* и *второго рода*: *пропуск атак* и *ложные тревоги*, соответственно. Нежелательность ошибок первого рода очевидна; ошибки второго рода не менее неприятны, поскольку отвлекают администратора безопасности от действительно важных дел, косвенно способствуя пропуску атак.

Достоинства сигнатурного метода – высокая производительность, малое число ошибок второго рода, обоснованность решений. Основной недостаток – неумение обнаруживать неизвестные атаки и вариации известных атак.

Основные достоинства статистического подхода – универсальность и обоснованность решений, потенциальная способность обнаруживать неизвестные атаки, то есть минимизация числа ошибок первого рода. Минусы заключаются в относительно высокой доле ошибок второго рода, плохой работе в случае, когда неправомерное поведение является типичным, когда типичное поведение плавно меняется от легального к неправомерному, а также в случаях, когда типичного поведения нет (как показывает статистика, таких пользователей примерно 5-10%).

Средства активного аудита могут располагаться на всех линиях обороны информационной системы. На границе контролируемой зоны они могут обнаруживать подозрительную активность в точках подключения к внешним сетям (не только попытки нелегального проникновения, но и действия по "прощупыванию" сервисов безопасности). В корпоративной сети, в рамках информационных сервисов и сервисов безопасности, активный аудит в состоянии обнаружить и пресечь подозрительную активность внешних и внутренних пользователей, выявить проблемы в работе сервисов, вызванные как нарушениями безопасности, так и аппаратно-программными ошибками. Важно отметить, что активный аудит, в принципе, способен обеспечить защиту от атак на доступность.

К сожалению, формулировка "в принципе, способен обеспечить защиту" не случайна. Активный аудит развивается более десяти лет, и первые результаты казались весьма многообещающими. Довольно быстро удалось реализовать распознавание простых типовых атак, однако затем было выявлено множество проблем, связанных с обнаружением заранее неизвестных атак, атак распределенных, растянутых во времени и т.п. Было бы наивно ожидать полного решения подобных проблем в ближайшее время.

(Оперативное пополнение базы сигнатур атак таким решением, конечно, не является.) Тем не менее, и на нынешней стадии развития активный аудит полезен как один из рубежей (вернее, как набор прослоек) *эшелонированной обороны*.

Функциональные компоненты и архитектура

В составе средств активного аудита можно выделить следующие функциональные компоненты:

- компоненты генерации регистрационной информации. Они находятся на стыке между средствами активного аудита и контролируемыми объектами;
- компоненты хранения сгенерированной регистрационной информации;
- компоненты извлечения регистрационной информации (*сенсоры*). Обычно различают сетевые и хостовые сенсоры, имея в виду под первыми выделенные компьютеры, сетевые карты которых установлены в режим прослушивания, а под вторыми – программы, читающие регистрационные журналы операционной системы. На наш взгляд, с развитием коммутационных технологий это различие постепенно стирается, так как сетевые сенсоры приходится устанавливать в активном сетевом оборудовании и, по сути, они становятся частью сетевой ОС;
- компоненты просмотра регистрационной информации. Могут помочь при принятии решения о реагировании на подозрительную активность;
- компоненты *анализа* информации, поступившей от сенсоров. В соответствии с данным выше определением средств активного аудита, выделяют пороговый анализатор, анализатор нарушений политики безопасности, экспертную систему, выявляющую сигнатуры атак, а также статистический анализатор, обнаруживающий нетипичное поведение;
- компоненты хранения информации, участвующей в анализе. Такое хранение необходимо, например, для выявления атак, протяженных во времени;
- компоненты принятия решений и реагирования ("*решатели*"). "Решатель" может получать информацию не только от локальных, но и от внешних анализаторов, проводя так называемый корреляционный анализ распределенных событий;
- компоненты хранения информации о контролируемых объектах. Здесь могут храниться как пассивные данные, так и методы, необходимые, например, для извлечения из объекта регистрационной информации или для реагирования;
- компоненты, играющие роль организующей оболочки для менеджеров активного аудита, называемые мониторами и объединяющие анализаторы, "решатели", хранилище описаний объектов и *интерфейсные* компоненты. В число последних входят компоненты интерфейса с другими мониторами, как равноправными, так и входящими в иерархию. Такие интерфейсы необходимы, например, для выявления распределенных, широкомасштабных атак;
- компоненты интерфейса с администратором безопасности.

Средства активного аудита строятся в архитектуре *менеджер/агент*. Основными агентскими компонентами являются сенсоры. Анализ, принятие решений – функции менеджеров. Очевидно, между менеджерами и агентами должны быть сформированы доверенные каналы.

Подчеркнем важность интерфейсных компонентов. Они полезны как с внутренней для средств активного аудита точки зрения (обеспечивают расширяемость, подключение компонентов различных производителей), так и с внешней точки зрения. Между менеджерами (между компонентами анализа и "решателями") могут существовать горизонтальные связи, необходимые для анализа распределенной активности. Возможно

также формирование иерархий средств активного аудита с вынесением на верхние уровни информации о наиболее масштабной и опасной активности.

Обратим также внимание на архитектурную общность средств активного аудита и управления, являющуюся следствием общности выполняемых функций. Продуманные интерфейсные компоненты могут существенно облегчить совместную работу этих средств.

Шифрование

Мы приступаем к рассмотрению криптографических сервисов безопасности, точнее, к изложению элементарных сведений, помогающих составить общее представление о компьютерной криптографии и ее месте в общей архитектуре информационных систем.

Криптография необходима для реализации, по крайней мере, трех сервисов безопасности:

- шифрование;
- контроль целостности;
- аутентификация (этот сервис был рассмотрен нами ранее).

Шифрование – наиболее мощное средство обеспечения конфиденциальности. Во многих отношениях оно занимает центральное место среди программно-технических регуляторов безопасности, являясь основой реализации многих из них, и в то же время последним (а подчас и единственным) защитным рубежом. Например, для портативных компьютеров только шифрование позволяет обеспечить конфиденциальность данных даже в случае кражи.

В большинстве случаев и шифрование, и контроль целостности играют глубоко инфраструктурную роль, оставаясь прозрачными и для приложений, и для пользователей. Типичное место этих сервисов безопасности – на сетевом и транспортном уровнях реализации стека сетевых протоколов.

Различают два основных метода *шифрования*: *симметричный* и *асимметричный*. В первом из них один и тот же ключ (хранящийся в секрете) используется и для зашифрования, и для *расшифрования* данных. Разработаны весьма эффективные (быстрые и надежные) методы симметричного шифрования. Существует и национальный стандарт на подобные методы – ГОСТ 28147-89 "Системы обработки информации. Защита криптографическая. Алгоритм криптографического преобразования".

Рис. 11.1 иллюстрирует использование симметричного шифрования. Для определенности мы будем вести речь о защите сообщений, хотя события могут развиваться не только в пространстве, но и во времени, когда зашифровываются и расшифровываются никуда не перемещающиеся файлы.

Рис. 11.1. Использование симметричного метода шифрования.

Основным недостатком симметричного шифрования является то, что *секретный ключ* должен быть известен и отправителю, и получателю. С одной стороны, это создает новую проблему *распространения ключей*. С другой стороны, получатель на основании наличия зашифрованного и расшифрованного сообщения не может доказать, что он получил это сообщение от конкретного отправителя, поскольку такое же сообщение он мог сгенерировать самостоятельно.

В асимметричных методах используются два ключа. Один из них, несекретный (он может публиковаться вместе с другими открытыми сведениями о пользователе), применяется для шифрования, другой (секретный, известный только получателю) – для расшифрования. Самым популярным из асимметричных является метод RSA (Райвест, Шамир, Адлеман), основанный на операциях с большими (скажем, 100-значными) простыми числами и их произведениями.

Проиллюстрируем использование асимметричного шифрования (см. рис. 11.2).

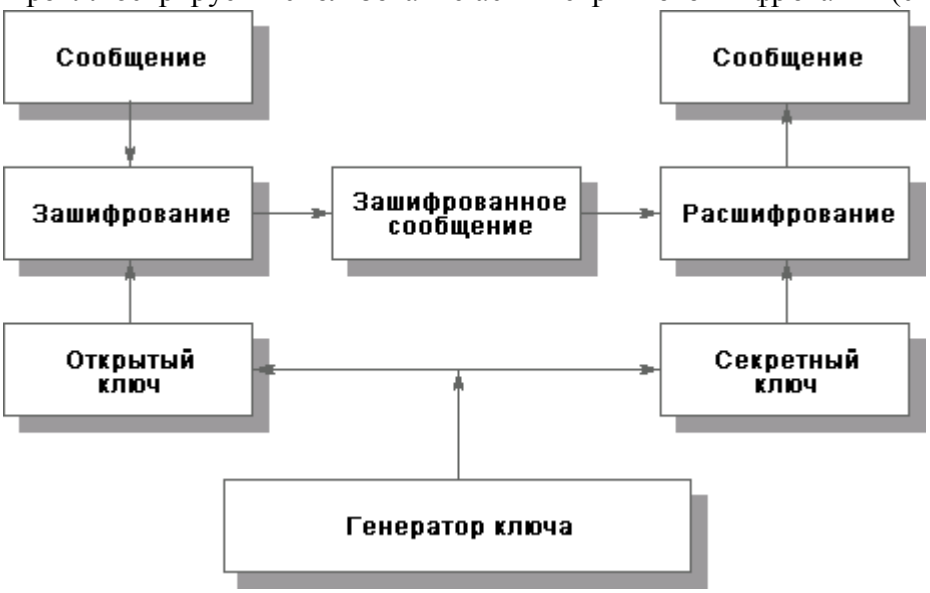

Рис. 11.2. Использование асимметричного метода шифрования.

Существенным недостатком асимметричных методов шифрования является их низкое быстродействие, поэтому данные методы приходится сочетать с симметричными (асимметричные методы на 3 – 4 порядка медленнее). Так, для решения задачи эффективного шифрования с передачей секретного ключа, использованного отправителем, сообщение сначала симметрично зашифровывают случайным ключом, затем этот ключ зашифровывают *открытым* асимметричным *ключом* получателя, после чего сообщение и ключ отправляются по сети.

Рис. 11.3 иллюстрирует эффективное шифрование, реализованное путем сочетания симметричного и асимметричного методов.

На рис. 11.4 показано расшифрование эффективно зашифрованного сообщения.

Отметим, что асимметричные методы позволили решить важную задачу совместной выработки секретных ключей (это существенно, если стороны не доверяют друг другу), обслуживающих сеанс взаимодействия, при изначальном отсутствии общих секретов. Для этого используется алгоритм Диффи-Хелмана.

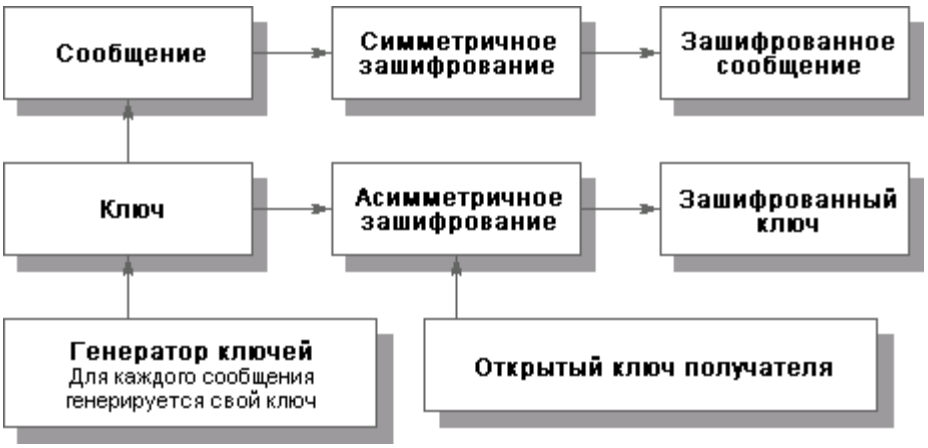

Рис. 11.3. Эффективное шифрование сообщения.

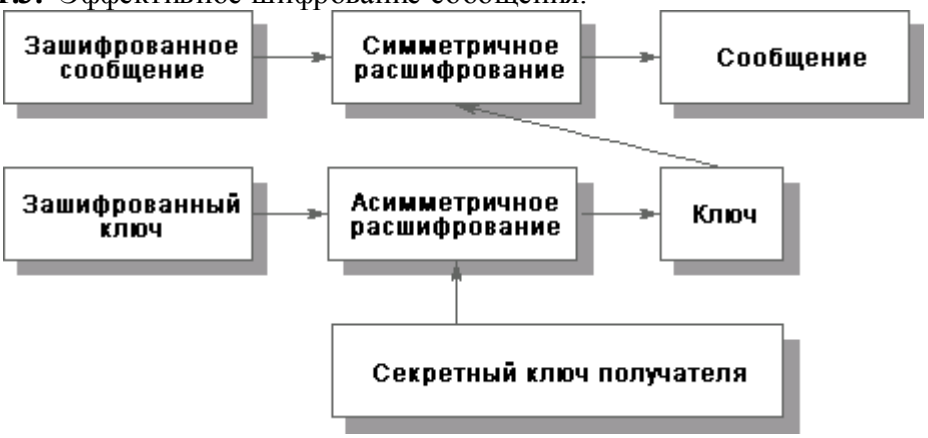

Рис. 11.4. Расшифрование эффективно зашифрованного сообщения.

Определенное распространение получила разновидность симметричного шифрования, основанная на использовании *составных ключей*. Идея состоит в том, что секретный ключ делится на две части, хранящиеся отдельно. Каждая часть сама по себе не позволяет выполнить расшифрование. Если у правоохранительных органов появляются подозрения относительно лица, использующего некоторый ключ, они могут в установленном порядке получить половинки ключа и дальше действовать обычным для симметричного расшифрования образом.

Порядок работы с составными ключами – хороший пример следования принципу разделения обязанностей. Он позволяет сочетать права на разного рода тайны (персональную, коммерческую) с возможностью эффективно следить за нарушителями закона, хотя, конечно, здесь очень много тонкостей и технического, и юридического плана.

Многие криптографические алгоритмы в качестве одного из параметров требуют псевдослучайное значение, в случае предсказуемости которого в алгоритме появляется уязвимость (подобное уязвимое место было обнаружено в некоторых вариантах Web-навигаторов). Генерация *псевдослучайных последовательностей* – важный аспект криптографии, на котором мы, однако, останавливаться не будем.

Более подробную информацию о компьютерной криптографии можно почерпнуть из статьи Г. Семенова "Не только шифрование, или Обзор криптотехнологий" (Jet Info, 2001, 3).

Контроль целостности

Криптографические методы позволяют надежно контролировать целостность как отдельных порций данных, так и их наборов (таких как поток сообщений); определять подлинность источника данных; гарантировать невозможность отказаться от совершенных действий ("неотказуемость").

В основе криптографического контроля целостности лежат два понятия:

- хэш-функция;
- *электронная цифровая подпись (ЭЦП)*.

Хэш-функция – это труднообратимое преобразование данных (*односторонняя функция*), реализуемое, как правило, средствами симметричного шифрования со связыванием блоков. Результат шифрования последнего блока (зависящий от всех предыдущих) и служит результатом хэш-функции.

Пусть имеются данные, целостность которых нужно проверить, хэш-функция и ранее вычисленный результат ее применения к исходным данным (так называемый *дайджест*). Обозначим хэш-функцию через h, исходные данные – через T, проверяемые данные – через T'. Контроль целостности данных сводится к проверке равенства h(T') = h(T). Если оно выполнено, считается, что T' = T. Совпадение дайджестов для различных данных называется коллизией. В принципе, коллизии, конечно, возможны, поскольку мощность множества дайджестов меньше, чем мощность множества хэшируемых данных, однако то, что h есть функция односторонняя, означает, что за приемлемое время специально организовать коллизию невозможно.

Рассмотрим теперь применение асимметричного шифрования для *выработки* и *проверки электронной цифровой подписи*. Пусть E(T) обозначает результат зашифрования текста T с помощью открытого ключа, а D(T) – результат расшифрования текста T (как правило, шифрованного) с помощью секретного ключа. Чтобы асимметричный метод мог применяться для реализации ЭЦП, необходимо выполнение тождества

E(D(T)) = D(E(T)) = T

На рис. 11.5 показана процедура выработки электронной цифровой подписи, состоящая в шифровании преобразованием D дайджеста h(T).

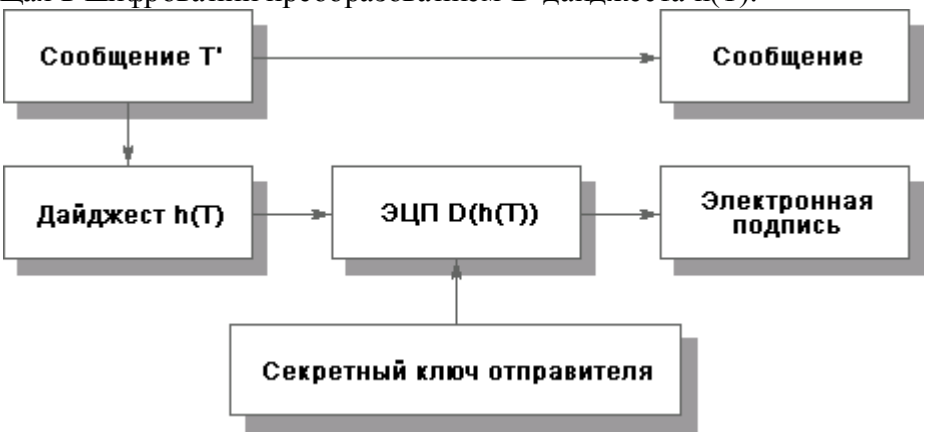

Рис. 11.5. Выработка электронной цифровой подписи.

Проверка ЭЦП может быть реализована так, как показано на рис. 11.6.

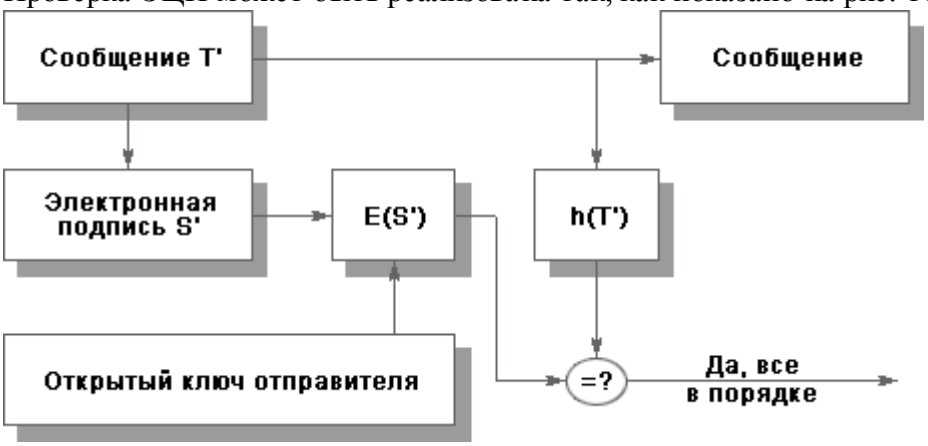

Рис. 11.6. Проверка электронной цифровой подписи.

Из равенства

$E(S') = h(T')$

следует, что $S' = D(h(T'))$ (для доказательства достаточно применить к обеим частям преобразование D и вычеркнуть в левой части тождественное преобразование $D(E())$). Таким образом, электронная цифровая подпись защищает целостность сообщения и удостоверяет личность отправителя, то есть защищает целостность источника данных и служит основой неотказуемости.

Два российских стандарта, ГОСТ Р 34.10-94 "Процедуры выработки и проверки электронной цифровой подписи на базе асимметричного криптографического алгоритма" и ГОСТ Р 34.11-94 "Функция хэширования", объединенные общим заголовком "Информационная технология. Криптографическая защита информации", регламентируют вычисление дайджеста и реализацию ЭЦП. В сентябре 2001 года был утвержден, а 1 июля 2002 года вступил в силу новый стандарт ЭЦП – ГОСТ Р 34.10-2001, разработанный специалистами существовавшего в то время Федерального агентства правительственной связи и информации (ФАПСИ).

Для контроля целостности последовательности сообщений (то есть для защиты от кражи, дублирования и переупорядочения сообщений) применяют временные штампы и нумерацию элементов последовательности, при этом штампы и номера включают в подписываемый текст.

Цифровые сертификаты

При использовании асимметричных методов шифрования (и, в частности, электронной цифровой подписи) необходимо иметь гарантию подлинности пары (имя пользователя, открытый ключ пользователя). Для решения этой задачи в спецификациях X.509 вводятся понятия *цифрового сертификата* и *удостоверяющего центра*.

Удостоверяющий центр – это компонент *глобальной службы каталогов*, отвечающий за управление криптографическими ключами пользователей. Открытые ключи и другая информация о пользователях хранится удостоверяющими центрами в виде цифровых сертификатов, имеющих следующую структуру:

- порядковый номер сертификата;
- идентификатор алгоритма электронной подписи;
- имя удостоверяющего центра;
- срок годности;
- имя владельца сертификата (имя пользователя, которому принадлежит сертификат);
- открытые ключи владельца сертификата (ключей может быть несколько);
- идентификаторы алгоритмов, ассоциированных с открытыми ключами владельца сертификата;
- электронная подпись, сгенерированная с использованием секретного ключа удостоверяющего центра (подписывается результат хэширования всей информации, хранящейся в сертификате).

Цифровые сертификаты обладают следующими свойствами:

- любой пользователь, знающий открытый ключ удостоверяющего центра, может узнать открытые ключи других клиентов центра и проверить целостность сертификата;
- никто, кроме удостоверяющего центра, не может модифицировать информацию о пользователе без нарушения целостности сертификата.

В спецификациях X.509 не описывается конкретная процедура генерации криптографических ключей и управления ими, однако даются некоторые общие рекомендации. В частности, оговаривается, что пары ключей могут порождаться любым из следующих способов:

- ключи может генерировать сам пользователь. В таком случае секретный ключ не попадает в руки третьих лиц, однако нужно решать задачу безопасной связи с удостоверяющим центром;
- ключи генерирует доверенное лицо. В таком случае приходится решать задачи безопасной доставки секретного ключа владельцу и предоставления доверенных данных для создания сертификата;
- ключи генерируются удостоверяющим центром. В таком случае остается только задача безопасной передачи ключей владельцу.

Цифровые сертификаты в формате X.509 версии 3 стали не только формальным, но и фактическим стандартом, поддерживаемым многочисленными удостоверяющими центрами.