7.2. Прямая и арбитражная цифровые подписи

При использовании *прямой цифровой подписи* взаимодействуют только сами участники, т.е. отправитель и получатель. Предполагается, что получатель знает открытый ключ отправителя. *Цифровая подпись* может быть создана шифрованием всего сообщения или его хэш-кода закрытым ключом отправителя.

Конфиденциальность может быть обеспечена дальнейшим шифрованием всего сообщения вместе с подписью открытым ключом получателя (асимметричное шифрование) или разделяемым секретным ключом (симметричное шифрование). Заметим, что обычно функция подписи выполняется первой, и только после этого выполняется функция конфиденциальности. В случае возникновения спора некая третья сторона должна просмотреть сообщение и его подпись. Если функция подписи выполняется над зашифрованным сообщением, то для разрешения споров придется хранить сообщение как в незашифрованном виде (для практического использования), так и в зашифрованном (для проверки подписи). Либо в этом случае необходимо хранить ключ симметричного шифрования, для того чтобы можно было проверить подпись исходного сообщения. Если *цифровая подпись* выполняется над незашифрованным сообщением, получатель может хранить только сообщение в незашифрованном виде и соответствующую подпись к нему.

Все прямые схемы, рассматриваемые далее, имеют общее слабое место. Действенность схемы зависит от безопасности закрытого ключа отправителя. Если отправитель впоследствии не захочет признать факт отправки сообщения, он может утверждать, что закрытый ключ был потерян или украден, и в результате кто-то подделал его подпись. Можно применить административное управление, обеспечивающее безопасность закрытых ключей, для того чтобы, по крайней мере, хоть в какой-то степени ослабить эти угрозы. Один из возможных способов состоит в требовании в каждую подпись сообщения включать отметку времени (дату и время) и сообщать о скомпрометированных ключах в специальный центр.

Другая угроза состоит в том, что закрытый ключ может быть действительно украден у X в момент времени T. Нарушитель может затем послать сообщение, подписанное подписью X и помеченное временной меткой, которая меньше или равна T.

Проблемы, связанные с *прямой цифровой подписью*, могут быть частично решены с помощью арбитра. Существуют различные схемы с применением *арбитражной подписи*. В общем виде *арбитражная подпись* выполняется следующим образом. Каждое подписанное сообщение от отправителя X к получателю Y первым делом поступает к арбитру A, который проверяет подпись для данного сообщения. После этого сообщение датируется и посылается к Y с указанием того, что оно было проверено

арбитром. Присутствие А решает проблему схем *прямой цифровой подписи*, при которых Х может отказаться от сообщения.

Рассмотрим некоторые возможные технологии *арбитражной цифровой подписи*.

Симметричное шифрование, арбитр видит сообщение:

$X \rightarrow A$: $M \parallel E_{Kxa} [ID_X \parallel H (M)]$

Предполагается, что отправитель Х и арбитр А разделяют секретный ключ K_{XA} и что А и Y разделяют секретный ключ K_{AY}. Х создает сообщение М и вычисляет его хэш-значение Н (М). Затем Х передает сообщение и подпись А. Подпись состоит из идентификатора Х и хэш-значения, все зашифровано с использованием ключа K_{XA}. А дешифрует подпись и проверяет хэш-значение.

$A \rightarrow Y$: $E_{Kay} [ID_X \parallel M \parallel E_{Kxa} [ID_X \parallel H (M)], T]$

Затем А передает сообщение к Y, шифруя его K_{AY}. Сообщение включает ID_X, первоначальное сообщение от Х, подпись и отметку времени. Y может дешифровать его для получения сообщения и подписи. Отметка времени информирует Y о том, что данное сообщение не устарело и не является повтором. Y может сохранить М и подпись к нему. В случае спора Y, который утверждает, что получил сообщение М от Х, посылает следующее сообщение к А:

$E_{Kay} [ID_X \parallel M \parallel E_{Kxa} [ID_X \parallel H (M)]]$

Арбитр использует K_{AY} для получения ID_X, М и подписи, а затем, используя K_{XA}, может дешифровать подпись и проверить хэш-код. По этой схеме Y не может прямо проверить подпись Х; подпись используется исключительно для разрешения споров. Y считает сообщение от Х аутентифицированным, потому что оно прошло через А. В данном сценарии обе стороны должны иметь высокую степень доверия к А:

1. Х должен доверять А в том, что тот не будет раскрывать K_{XA} и создавать фальшивые подписи в форме $E_{Kка} [ID_X \parallel H (M)]$.
2. Y должен доверять А в том, что он будет посылать $E_{Kay} [ID_X \parallel M \parallel E_{Kxa} [ID_X \parallel H (M)]]$ только в том случае, если хэш-значение является корректным и подпись была создана Х.
3. Обе стороны должны доверять А в решении спорных вопросов.

Симметричное шифрование, арбитр не видит сообщение:

Если арбитр не является такой доверенной стороной, то Х должен добиться того, чтобы никто не мог подделать его подпись, а Y должен добиться того, чтобы Х не мог отвергнуть свою подпись.

Предыдущий сценарий также предполагает, что А имеет возможность читать сообщения от Х к Y и что возможно любое подсматривание. Рассмотрим сценарий, который, как и прежде, использует арбитраж, но при этом еще обеспечивает конфиденциальность. В таком случае также предполагается, что Х и Y разделяют секретный ключ K_{XY}.

$X \rightarrow A$: $ID_X \parallel E_{Kxy} [M] \parallel E_{Kxa} [ID_X \parallel H (E_{Kxy}M)]]$

Х передает А свой идентификатор, сообщение, зашифрованное K_{XY}, и подпись. Подпись состоит из идентификатора и хэш-значения

зашифрованного сообщения, которые зашифрованы с использованием ключа K_{XA}. А дешифрует подпись и проверяет хэш-значение. В данном случае А работает только с зашифрованной версией сообщения, что предотвращает его чтение.

$A \rightarrow Y: E_{Kay} [ID_X || E_{Kxy}] || E_{Kxa} [ID_X || H (E_{Kxy}M])], T]$

А передает Y все, что он получил от X плюс отметку времени, все шифруя с использованием ключа K_{AY}.

Хотя арбитр и не может прочитать сообщение, он в состоянии предотвратить подделку любого из участников, X или Y. Остается проблема, как и в первом сценарии, что арбитр может сговориться с отправителем, отрицающим подписанное сообщение, или с получателем, для подделки подписи отправителя.

Шифрование открытым ключом, арбитр не видит сообщение:

Все обсуждаемые проблемы могут быть решены с помощью схемы открытого ключа.

$X \rightarrow A: ID_X || E_{KRx} [ID_X || E_{KUy} [E_{KRx} [M]]]$

В этом случае X осуществляет двойное шифрование сообщения M, сначала своим закрытым ключом KR_X, а затем открытым ключом Y KU_Y. Получается подписанная секретная версия сообщения. Теперь это подписанное сообщение вместе с идентификатором X шифруется KR_X и вместе с ID_X посылается A. Внутреннее, дважды зашифрованное, сообщение недоступно арбитру (и всем, исключая Y). Однако А может дешифровать внешнюю шифрацию, чтобы убедиться, что сообщение пришло от X (так как только X имеет KR_X). Проверка дает гарантию, что пара закрытый/открытый ключ законна, и тем самым верифицирует сообщение.

$A \rightarrow Y: E_{KRa} [ID_X || E_{KUy} [E_{KRx} [M]] || T]$

Затем А передает сообщение Y, шифруя его KR_A. Сообщение включает ID_X, дважды зашифрованное сообщение и отметку времени.

Эта схема имеет ряд преимуществ по сравнению с предыдущими двумя схемами. Во-первых, никакая информация не разделяется участниками до начала соединения, предотвращая договор об обмане. Во-вторых, некорректные данные не могут быть посланы, даже если KR_X скомпрометирован, при условии, что не скомпрометирован KR_A. В заключение, содержимое сообщения от X к Y неизвестно ни А, ни кому бы то ни было еще.