5. АСИММЕТРИЧНОЕ ШИФРОВАНИЕ И ЭЛЕКТРОННАЯ ЦИФРОВАЯ ПОДПИСЬ

5.1. Теоретические сведения

Электронная цифровая подпись ЭЦП – реквизит электронного документа, полученный в результате криптографического преобразования информации с использованием закрытого ключа подписи и позволяющий проверить отсутствие искажения информации в электронном документе с момента формирования подписи (целостность), принадлежность подписи владельцу сертификата ключа подписи (авторство), а в случае успешной проверки подтвердить факт подписания электронного документа (неотказуемость).

Существуют две основные схемы построения цифровой подписи:

1) Схема арбитражной подписи (рис. 5.1). В этой схеме необходимо наличие в системе арбитра – третьего лица, пользующегося доверием сторон, генерирующих и проверяющих подпись. Схема строится с использованием симметричных алгоритмов шифрования.

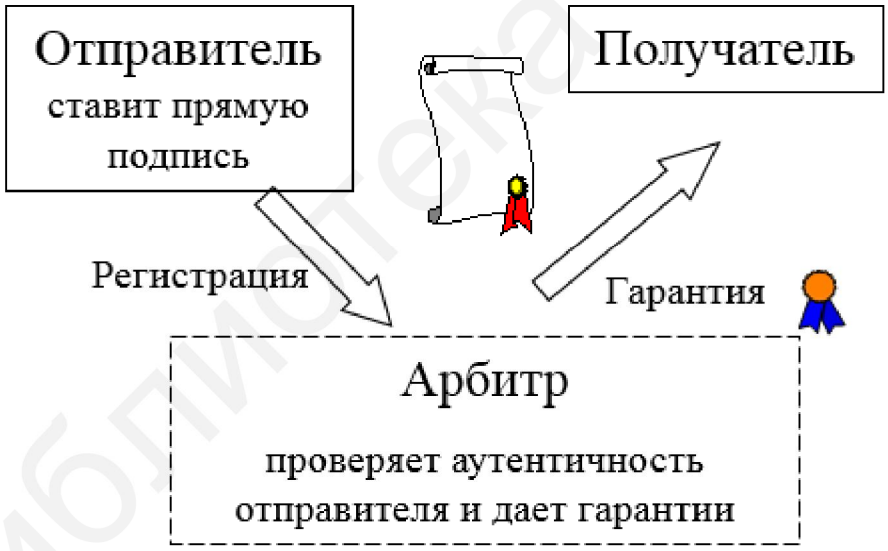

Рис. 5.1. Схема арбитражной подписи

2) Схема прямой подписи (рис. 5.2). Схема строится на основе алгоритмов асимметричного шифрования. Для формирования и проверки подписи третья сторона не нужна. Однако наличие арбитра требуется в случае возникновения конфликтной ситуации между сторонами.

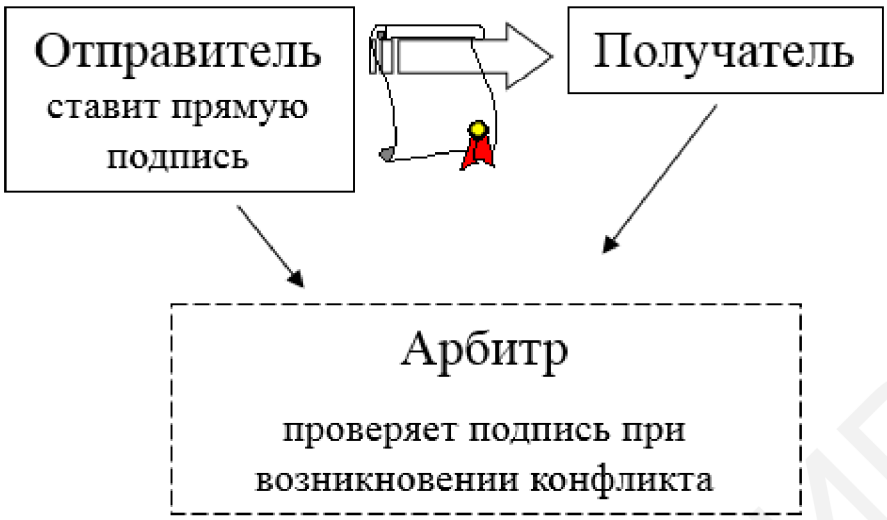

Рис. 5.2. Схема прямой подписи

Кроме этого, существуют другие виды цифровых подписей (групповая подпись, неоспоримая подпись, доверенная подпись), которые являются вариациями двух основных схем. Их появление обусловлено разнообразием задач, решаемых с помощью ЭЦП.

Основными функциями цифровой подписи являются:

– контроль целостности электронного документа;

– защита от изменений (подделки) документа;

– обеспечение невозможности отказа от авторства документа;

– формирование доказательств подтверждения авторства документа.

В настоящее время наибольшее распространение получили системы, использующие схему прямой подписи. Формирование и проверка прямой цифровой подписи основана на использовании идеи ***односторонних функций с потайным ходом***.

В статье, опубликованной Диффи и Хеллманом в 1976 году, односторонняя функция была определена как семейство обратимых функций f_z с параметром z, таких, что для данного z можно найти алгоритмы E_z и D_z, позволяющие легко вычислить значение $f_z(x)$ для всех x из области определения, а также $f_z^{-1}(y)$ для всех y из области значений, однако практически для всех значений параметра z и практически для всех значений y из области значений f_z нахождение $f_z^{-1}(y)$ вычислительно неосуществимо даже при известном E_z.

Независимо от выбранного вида односторонней функции с потайным ходом схемы прямой подписи относятся к асимметричным криптографическим системам с открытым ключом. В рамках такой системы каждый пользователь имеет пару ключей – секретный (private key) и открытый (public key).

Использовать их можно как для цифровой подписи, так и для шифрования сообщений (чаще всего для обмена секретными ключами по открытым каналам связи).

Для формирования цифровой подписи подписант использует свой секретный ключ, а для проверки проверяющий использует открытый ключ подписанта.

Для зашифрования сообщения отправитель использует открытый ключ получателя, а для расшифрования получатель использует свой секретный ключ.

Схема формирования цифровой подписи включает в себя три процесса:

1) Генерация ключевой пары. Секретный ключ равновероятным образом выбирают из множества возможных секретных ключей, затем вычисляют соответствующий ему открытый ключ.

2) Формирование ЭЦП. Для блока данных с помощью секретного ключа вычисляют значение ЭЦП.

3) Проверка (верификация) ЭЦП. Для блока данных и значения ЭЦП с помощью открытого ключа проверяют действительность ЭЦП.

Для эффективного использования цифровой подписи необходимо выполнение следующих условий:

– верификация ЭЦП должна выполняться с помощью открытого ключа, соответствующего секретному ключу, использовавшемуся для формирования ЭЦП;

– без обладания секретным ключом должно быть вычислительно сложно сформировать легитимную цифровую подпись.

Для обеспечения выполнения этих условий, как правило, используют алгоритмы, основанные на следующих задачах:

– задача дискретного логарифмирования;

– задача факторизации, то есть разложение числа на простые множители.

В августе 1977 года трое ученых из Массачусетского технологического института Рональд Ривест, Ади Шамир и Леонард Адлеман предложили **криптосистему RSA**, основанную на задаче факторизации.

Алгоритм создания ключа:

1) Выбирают два случайных простых числа p и q заданного размера, $p \neq q$.

2) Вычисляют произведение $n = p \cdot q$.

3) Вычисляют значение функции Эйлера от числа n:

$$\varphi(n) = (p-1)(q-1).$$

4) Выбирают целое число e, $1 < e < \varphi(n)$, взаимно простое со значением функции $\varphi(n)$. Обычно в качестве e берут простые числа, содержащие небольшое количество единичных битов в двоичной записи, например простые

числа Ферма 17, 257 или 65537 (что обеспечивает высокую скорость шифрования, так как время шифрования пропорционально количеству единичных битов в числе e). Число e называют открытой экспонентой.

5) Вычисляют число d, мультипликативно обратное к числу e по модулю n, удовлетворяющее сравнению:

$$d \cdot e \equiv 1 \bmod \varphi(n).$$

Число d называют секретной экспонентой. Для вычисления d используют расширенный алгоритм Евклида.

Пара чисел $\{e, n\}$ публикуется в качестве открытого ключа RSA.

Пара $\{d, n\}$ играет роль закрытого ключа RSA и держится в секрете.

Шифрование. В качестве исходного сообщения m выступают целые числа в интервале от 0 до $n - 1$.

Для зашифрования используют открытый ключ $\{e, n\}$ получателя сообщения и вычисляют криптограмму с следующим образом:

$$c = E(m) = m^e \bmod n.$$

Для расшифрования используют секретный ключ $\{d, n\}$ получателя сообщения и вычисляют исходное сообщение следующим образом:

$$m = D(c) = c^d \bmod n.$$

На практике RSA не применяют для шифрования сообщений, а используют смешанный алгоритм, состоящий из RSA и блочного симметричного шифра, например AES.

С помощью RSA зашифровывается секретный ключ блочного алгоритма шифрования, а затем с помощью этого ключа шифруют сообщения блочным симметричным шифром. Как правило, такой ключ используют в течение одного сеанса связи, а затем уничтожают. Поэтому ключ называют сеансовым.

Если сеансовый ключ больше n, то перед зашифрованием его разделяют на блоки и зашифровывают поблочно.

Цифровая подпись. Алгоритм цифровой подписи отличается от алгоритма шифрования использованием ключей.

При создании цифровой подписи входом является исходный текст m и секретный ключ подписанта $\{d, n\}$.

Для создания цифровой подписи s с помощью секретного ключа $\{d, n\}$ вычисляют

$$s = m^d \bmod n.$$

Затем формируют пару $\{m, s\}$ и отправляют получателю.

Для проверки цифровой подписи входом является пара $\{m, s\}$ и открытый ключ подписанта $\{e, n\}$.

Проверяющий вычисляет прообраз сообщения из подписи

$$m^* = s^e \bmod n$$

и сравнивает m и m^*. Если $m = m^*$, значит подпись верна, в противном случае – ложна.

Важным является то, что создать подпись может только автор – владелец секретного ключа $\{e, n\}$, а проверить любой, имеющий доступ к открытому ключу подписанта.

Цифровая подпись не обеспечивает конфиденциальность подписанного сообщения, так как не зашифровывает его.

Для обеспечения конфиденциальности автор должен подписать исходное сообщение, затем зашифровать пару исходное сообщение – подпись $\{m, s\}$ с помощью открытого ключа получателя. Получатель расшифровывает полученное сообщение с помощью своего секретного ключа, восстанавливая пару $\{m, s\}$, затем проверяет цифровую подпись.

Стойкость RSA. Стойкость алгоритма RSA базируется на предположении о сложности вычисления функции, обратной по отношению к функции шифрования

$$c = E(m) = m^e \bmod n.$$

Поскольку атакующему известна тройка $\{c, e, n\}$, то для вычисления m необходимо найти d, удовлетворяющее условию

$$e \cdot d \equiv 1 \bmod \varphi(n).$$

Для решения этой задачи атакующему необходимо знать значение функции Эйлера $\varphi(n)$. Поскольку

$$\varphi(n) = (p-1)(q-1),$$
$$n = p \cdot q,$$

то для нахождения $\varphi(n)$ надо разложить число n на простые множители p и q, т. е. решить задачу факторизации.

В настоящее время самым быстрым из известных методов факторизации является общий метод решета числового поля. Его скорость для целого числа длиной k бит оценивается как

$$\exp\left(\left(c + o(1)\right)k^{\frac{1}{3}}\log^{\frac{2}{3}}k\right) \text{ для некоторого } c < 2.$$

В 2010 году были успешно взломаны данные, зашифрованные RSA с длиной ключа 768 бит, и был сделан вывод о том, что надежной может считаться система RSA с длиной ключа не менее 1024 бит. Браузер Mozilla с 2013 года не поддерживает сертификаты удостоверяющего центра с длиной ключа менее 2048 бит, с 2014 года протокол TLS для сертификата сервера также рекомендует использовать RSA с ключом 2056 бит и более.

5.2. Задание для самостоятельного выполнения

Разработать программное обеспечение, реализующее функции генерации секретного и открытого ключей, шифрования и цифровой подписи для алгоритма RSA. Обмен входными и выходными данными должен осуществляться через файлы:

– открытого ключа;

– секретного ключа;

– исходного сообщения;

– зашифрованного сообщения.

Для повышения скорости шифрования использовать метод последовательного возведения в квадрат и умножения.

Выполнить тестирование разработанного программного обеспечения на 10 наборах тестовых данных.

Длина чисел p и q должна быть не менее 1024 бит.