4.4. ЭЛЕКТРОННАЯ ЦИФРОВАЯ ПОДПИСЬ

Электронная цифровая подпись используется для аутентификации текстов, передаваемых по телекоммуникационным каналам. При таком обмене электронными документами существенно снижаются затраты на обработку и хранение документов, ускоряется их поиск. Но возникает проблема аутентификации автора электронного документа и самого документа, то есть установления подлинности автора и отсутствия изменений в полученном электронном документе.

Целью аутентификации электронных документов является их защита от возможных видов злоумышленных действий, к которым относятся:

- **активный перехват** – нарушитель, подключившийся к сети, перехватывает документы (файлы) и изменяет их;
- **маскарад** – абонент С посылает документ абоненту В от имени абонента А;
- **ренегатство** – абонент А заявляет, что не посылал сообщения абоненту В, хотя на самом деле послал;
- **подмена** – абонент В изменяет или формирует новый документ и заявляет, что получил его от абонента А;
- **повтор** – абонент С повторяет ранее переданный документ, который абонент А посылал абоненту В.

Эти виды злоумышленных действий могут нанести существенный ущерб банковским и коммерческим структурам, государственным предприятиям и организациям, частным лицам, применяющим в своей деятельности компьютерные информационные технологии.

Проблему проверки целостности сообщения и подлинности автора сообщения позволяет эффективно решить методология электронной цифровой подписи.

4.4.1. ОСНОВНЫЕ ПРОЦЕДУРЫ ЦИФРОВОЙ ПДПИСИ

Функционально цифровая подпись аналогична обычной рукописной подписи и обладает ее основными достоинствами:

- удостоверяет, что подписанный текст исходит от лица, поставившего подпись;
- не дает самому этому лицу возможности отказаться от обязательств, связанных с подписанным текстом;
- гарантирует целостность подписанного текста.

Электронная цифровая подпись (ЭЦП) представляет собой относительно небольшое количество дополнительной цифровой информации, передаваемой вместе с подписываемым текстом. ЭЦП основана на обратимости асимметричных шифров, а также на взаимосвязанности содержимого сообщения, самой подписи и пары ключей. Изменение хотя бы одного из этих элементов сделает невозможным подтверждение подлинности цифровой подписи. ЭЦП реализуется при помощи асимметричных алгоритмов шифрования и хэш-функций.

Технология применения системы ЭЦП предполагает наличие сети абонентов, посылающих друг другу подписанные электронные документы. Для каждого абонента генерируется пара ключей: секретный и открытый. Секретный ключ хранится абонентом в тайне и используется им для формирования ЭЦП. Открытый ключ известен всем другим пользователям и предназначен для проверки ЭЦП получателем подписанного электронного документа.

Система ЭЦП включает две основные процедуры:

- процедуру формирования цифровой подписи;
- процедуру проверки цифровой подписи.

В процедуре формирования подписи используется секретный ключ отправителя сообщения, в процедуре проверки подписи – открытый ключ отправителя.

Процедура формирования цифровой подписи

На подготовительном этапе этой процедуры абонент A – отправитель сообщения генерирует пару ключей: секретный ключ k_A и открытый ключ K_A. Открытый ключ K_A вычисляется из парного ему секретного ключа k_A. Открытый ключ K_A рассылается остальным абонентам сети (или делается доступным, например, на разделяемом ресурсе) для использования при проверке подписи.

Для формирования цифровой подписи отправитель A прежде всего вычисляет значение хэш-функции $h(M)$ подписываемого текста M (рис. 1.20). Хэш-функция служит для сжатия исходного подписываемого текста M в дайджест m – относительно короткое число, состоящее из фиксированного небольшого числа битов и характеризующее весь текст M в целом . Далее отправитель A шифрует дайджест своим секретным ключом k_A. Получаемая при этом пара чисел представляет собой цифровую подпись для данного текста M. Сообщение M вместе с цифровой подписью отправляется в адрес получателя.

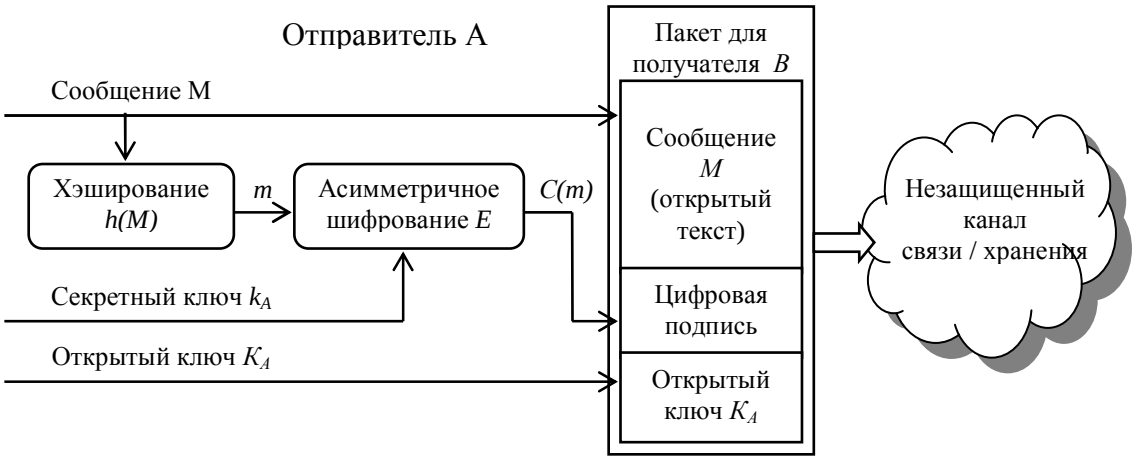

Рис. 1.20. Схема формирования электронной цифровой подписи

Процедура проверки цифровой подписи

Абоненты сети могут проверить цифровую подпись полученного сообщения M с помощью открытого ключа отправителя K_A этого сообщения (рис.1.21). При проверке ЭЦП абонент B – получатель сообщения M – расшифровывает принятый дайджест т открытым ключом K_A отправителя A. Кроме того, получатель сам вычисляет с помощью хэш-функции $h(M)$ дайджест m' принятого сообщения M и сравнивает его с расшифрованным. Если эти два дайджеста m и m' – совпадают, то цифровая подпись является подлинной. В противном случае либо подпись подделана, либо изменено содержание сообщения.

Принципиальным моментом в системе ЭЦП является невозможность подделки ЭЦП пользователя без знания его секретного ключа подписывания. Поэтому необходимо защитить секретный ключ подписывания от несанкционированного доступа.

Секретный ключ ЭЦП, аналогично ключу симметричного шифрования, рекомендуется хранить на персональном ключевом носителе в защищенном виде. Электронная цифровая подпись представляет собой уникальное число, зависящее от подписываемого документа и секретного ключа абонента. В качестве подписываемого документа может быть использован любой файл. Подписанный файл создается из неподписанного путем добавления в него одной или более электронных подписей.

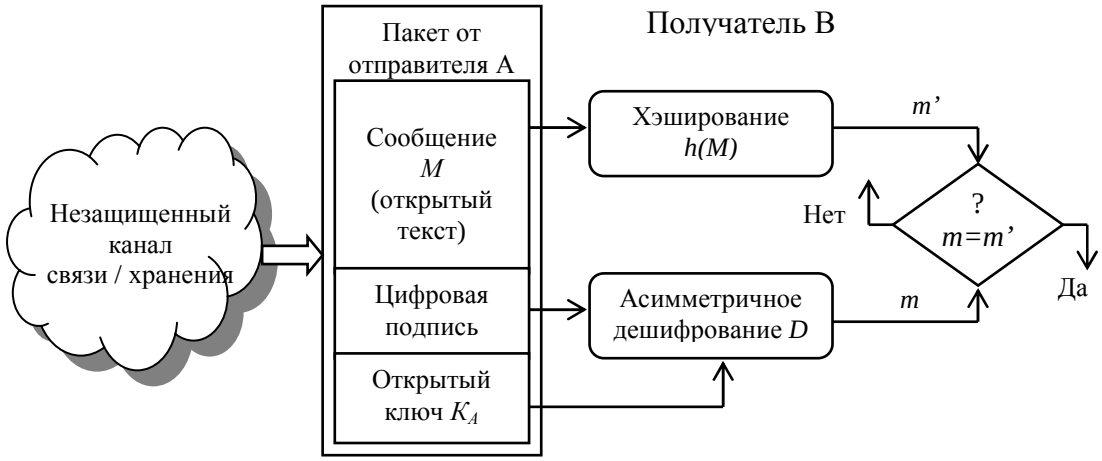

Рис. 1.21. Схема проверки электронной цифровой подписи

Помещаемая в подписываемый файл (или в отдельный файл электронной подписи) структура ЭЦП обычно содержит дополнительную информацию, однозначно идентифицирующую автора подписанного документа. Эта информация добавляется к документу до вычисления ЭЦП, что обеспечивает и ее целостность. Каждая подпись содержит следующую информацию:

- дату подписи;
- срок окончания действия ключа данной подписи;
- информацию о лице, подписавшем файл (Ф.И.О., должность, краткое наименование фирмы);
- идентификатор подписавшего (имя открытого ключа);
- собственно цифровую подпись.

Важно отметить, что, с точки зрения конечного пользователя, процесс формирования и проверки цифровой подписи отличается от процесса криптографического закрытия передаваемых данных следующими особенностями.

При формировании цифровой подписи используется закрытый ключ отправителя, тогда как при шифровании применяется открытый ключ получателя. При проверке цифровой подписи используется открытый ключ отправителя, а при дешифровании – закрытый ключ получателя.

Проверить сформированную подпись может любое лицо, так как ключ проверки подписи является открытым. При положительном результате проверки подписи делается заключение о подлинности и целостности полученного сообщения, то есть о том, что это сообщение действительно отправлено тем или иным отправителем и не было модифицировано при передаче по сети. Однако, если пользователя интересует, не является ли полученное сообщение повторением ранее отправленного или не было ли оно задержано на пути следования, то он должен проверить дату и время его отправки, а при наличии – порядковый номер.

Аналогично асимметричному шифрованию, необходимо обеспечить невозможность подмены открытого ключа, используемого для проверки ЭЦП. Если предположить, что злоумышленник n имеет доступ к открытым ключам, которые хранит на своем компьютере абонент B, в том числе к открытому ключу K_A абонента A, то он может выполнить следующие действия:

- прочитать из файла, в котором содержится открытый ключ K_A, идентификационную информацию об абоненте A;
- сгенерировать собственную пару ключей k_n и K_n, записав в них идентификационную информацию абонента A;
- подменить хранящийся у абонента B открытый ключ K_A своим открытым ключом K_n, но содержащим идентификационную информацию абонента A.

После этого злоумышленник n может посылать документы абоненту B, подписанные своим секретным ключом k_n. При проверке подписи этих документов абонент B будет считать, что документы подписаны абонентом A и их ЭЦП верна, то есть они не были модифицированы кем-либо. До выяснения отношений непосредственно с абонентом A у абонента B может не появиться сомнений в полученных документах.

Открытые ключи ЭЦП можно защитить от подмены с помощью соответствующих цифровых сертификатов.

Сегодня существует большое количество алгоритмов ЭЦП.

4.4.2. АЛГОРИТМ ЦИФРОВОЙ ПОДПИСИ DSA

Алгоритм цифровой подписи DSA (Digital Signature Algorithm) был предложен в 1991 году Национальным институтом стандартов и технологии США (National Institute of Standards and Technology – NIST) и стал стандартом США в 1993 году. Алгоритм DSA является развитием алгоритмов цифровой подписи Эль Гамаля и К. Шнорра. Ниже приводятся процедуры генерации ключей, генерации подписи и проверки подписи в алгоритме DSA.

Генерация ключей DSA

Отправитель и получатель электронного документа используют при вычислениях большие целые числа: g и p – простые числа, длиной L битов каждое ($512 \le L \le 1024$);

q – простое число длиной 160 бит (делитель числа $(p - 1)$. Числа g, p, q являются открытыми и могут быть общими для всех пользователей сети.

Отправитель выбирает случайное целое число x, $1 < x < q$. Число x является секретным ключом отправителя для формирования электронной цифровой подписи.

Затем отправитель вычисляет значение

$$y = g^x \bmod p.$$

Число y является открытым ключом для проверки подписи отправителя. Число y передается всем получателям документов.

Генерация подписи DSA

Этот алгоритм предусматривает использование односторонней функции хэширования $h(\cdot)$. В стандарте определен алгоритм безопасного хэширования SHA-1. Для того чтобы подписать сообщение M, участник A выполняет следующие шаги:

Шаг 1 – выбирает случайное целое k в интервале $[1, q - 1]$.

Шаг 2 – вычисляет $r = (g^k \bmod p) \bmod q$.

Шаг 3 – вычисляет $k^{-1} \bmod q$.

Шаг 4 – вычисляет $s = k^{-1}\{h(M) + xr\} \bmod q$, где h есть алгоритм хэширования SHA-1.

Шаг 5 – если s = 0 тогда перейти к шагу 1. (Если s = 0, тогда $s^{-1} \bmod q$ не существует; s требуется на шаге 2 процедуры проверки подписи.)

Шаг 6 – подпись для сообщения *M* есть пара целых чисел (*r*, *s*).

Проверка подписи DSA

Для того чтобы проверить подпись (*r*, *s*) сообщения *M* от участника *A*, участник *B* делает следующие шаги:

Шаг 1 – Получает подлинную копию открытого ключа *y* участника *A*.

Шаг 2 – Вычисляет $w = s^{-1} \bmod q$ и хэш-значение *h(M)*.

Шаг 3 – Вычисляет значения $u_1 = h(M)w \bmod q$ и $u_2 = (rw) \bmod q$.

Шаг 4 – Используя открытый ключ *y*, вычисляет значение

$v = (g^{u1} y^{u2} \bmod p) \bmod q$.

Шаг 5 – Признает подпись (*r*, *s*) под документом *M* подлинной, если *v* = *r*.

Поскольку *r* и *s* являются целыми числами, причем каждое меньше *q*, подписи DSA имеют длину 320 бит. Безопасность алгоритма цифровой подписи DSA базируется на трудностях задачи дискретного логарифмирования.

4.4.3. СТАНДАРТ ЦИФРОВОЙ ПОДПИСИ ГОСТ Р 34.10-94

Первый российский стандарт цифровой подписи обозначается как ГОСТ Р 34.10-94. Алгоритм цифровой подписи, определяемый этим стандартом, концептуально близок к алгоритму DSA. В нем используются следующие параметры:

p – большое простое число длиной от 509 до 512 бит либо от 1020 до 1024 бит;

q – простой сомножитель числа (*p* – 1), имеющий длину 254–256 бит;

a – любое число, меньшее (*p* – 1), причем такое, что $a^q \bmod p = 1$;

x – некоторое число, меньшее *q*;

$y = a^x \bmod p$.

Кроме того, этот алгоритм использует однонаправленную хэш-функцию *H*(*x*).

Стандарт ГОСТ Р 34.11-94 определяет хэш-функцию, основанную на использовании стандартного симметричного алгоритма ГОСТ 28147-89.

Первые три параметра – *p*, *q* и *a* – являются открытыми и могут быть общими для всех пользователей сети. Число *x* является секретным ключом. Число *y* является открытым ключом.

Чтобы подписать некоторое сообщение *m*, а затем проверить подпись, выполняются следующие шаги:

1) Пользователь *A* генерирует случайное число *k*, причем *k* < *q*.
2) Пользователь *A* вычисляет значения:

   $r = (a^k \bmod p) \bmod q$,

   $s = (x \times r + kH(m))) \bmod q$.

   Если $H(m) \bmod q = 0$, то значение $H(m) \bmod q$ принимают равным единице.

   Если *r* = 0, то выбирают другое значение *k* и начинают снова.

Цифровая подпись представляет собой два числа: r и s.

Пользователь A отправляет эти числа пользователю B.

3) Пользователь B проверяет полученную подпись, вычисляя:

$v = H(m)^{q-2} \bmod q$,

$z_1 = (s \times v) \bmod q$,

$z_2 = ((q - r) \times v) \bmod q$,

$u = ((a^{z_1} \times y^{z_2}) \bmod p) \bmod q$.

Если $u = r$, то подпись считается верной.

Различие между этим алгоритмом и алгоритмом DSA заключается в том, что в DSA

$s = (k(x \times r + (H(m)))) \bmod q$,

что приводит к другому уравнению верификации.

Следует также отметить, что в российском стандарте ЭЦП параметр q имеет длину 256 бит. Современных криптографов вполне устраивает q длиной примерно 160 бит. Различие в значениях параметра q является отражением стремления разработчиков российского стандарта к получению более безопасной подписи. Этот стандарт вступил в действие с начала 1995 года.

4.4.4. АЛГОРИТМ ЦИФРОВОЙ ПОДПИСИ ECDSA

В алгоритме ЭЦП ECDSA (Elliptic Curve Digital Signature Algorithm) определение параметров системы и генерация ключей аналогичны алгоритму асимметричного шифрования ECES.

Генерация ЭЦП (пользователь A подписывает сообщение M):

- вычисляется хэш-собщения *H(M);*
- выбирается случайное целое число k, взаимно простое с n (то есть не имеющее других общих с n делителей, кроме 1; поскольку n является простым числом по определению, данное условие выполняется автоматически), $1 < k < n - 1$;
- вычисляется точка $(x, y) = kP$ и $r = x \bmod n$. В случае если $r = 0$, повторяется выбор k;
- вычисляется $s = k^{-1}(H(M) + rd) \bmod n$;
- цифровой подписью сообщения M является пара чисел (r, s).

Проверка ЭЦП (пользователь B проверяет ЭЦП пользователя A под сообщением M):

- если $r = 0$, то полученная ЭЦП неверна;
- вычисляется хэш-собщения H(M);
- вычисляются $u = s^{-1}H(M) \bmod n$ и $v = s^{-1}r \bmod n$;
- вычисляется точка $(x_1, y_1) = uP + vQ$;
- вычисляется $r' = x_1 \bmod n$;
- ЭЦП считается верной, если $r' = r$.

4.4.5. СТАНДАРТ ЦИФРОВОЙ ПОДПИСИ ГОСТ Р 34.10-2001

Российский стандарт цифровой подписи ГОСТ Р 34.10-2001 был принят в 2001 году. Этот стандарт разработан взамен первого стандарта цифровой подписи

ГОСТ Р 34.10-94. Необходимость разработки стандарта ГОСТ Р 34.10-2001 вызвана потребностью в повышении стойкости электронной цифровой подписи к несанкционированным изменениям. Стойкость ЭЦП основывается на сложности вычисления дискретного логарифма в группе точек эллиптической кривой, а также на стойкости используемой хэш-функции по ГОСТ Р 34.11.

Принципиальное отличие нового стандарта от предыдущего ГОСТ Р 34.10-94 состоит в том, что все вычисления при генерации и проверке ЭЦП в новом алгоритме производятся в группе точек эллиптической кривой, определенной над конечным полем F_p.

Принадлежность точки (пары чисел x и y) к данной группе определяется следующим соотношением:

$$y^2 = x^3 + ax + b (\mathrm{mod}\ p),$$

где модуль системы p является простым числом, большим 3, а a и b - константы, удовлетворяющие следующим соотношениям: $a, b \in F_p$ и $4a^3 + 27b^2$ не сравнимо с нулем по модулю p.

При этом следует отметить, что принципы вычислений по данному алгоритму схожи с предшествующим российским стандартом ЭЦП. Математические подробности реализации этого алгоритма приводятся ниже.

Обозначения

В данном стандарте использованы следующие обозначения:

V_{256} – множество всех двоичных векторов длиной 256 бит;

V_∞ – множество всех двоичных векторов произвольной конечной длины;

Z – множество всех целых чисел;

p – простое число, $p > 3$;

F_p – конечное простое поле, представляемое как множество из p целых чисел $\{0, 1, ..., p-1\}$;

$b (\mathrm{mod}\ p)$ – минимальное неотрицательное число, сравнимое с b по модулю p;

M – сообщение пользователя, $M \in V_\infty$;

$(\overline{h_1} \| \overline{h_2})$ – конкатенация (объединение) двух двоичных векторов;

a, b – коэффициенты эллиптической кривой;

m – порядок группы точек эллиптической кривой;

q – порядок подгруппы группы точек эллиптической кривой;

O – нулевая точка эллиптической кривой;

P – точка эллиптической кривой порядка q;

d – целое число – ключ подписи;

Q – точка эллиптической кривой – ключ проверки;

w – цифровая подпись под сообщением M.

Общие положения

Механизм цифровой подписи реализуется посредством двух основных процессов:

- формирования цифровой подписи;
- проверки цифровой подписи.

В процессе формирования цифровой подписи в качестве исходных данных используются сообщение *M*, ключ подписи *d* и параметры схемы ЭЦП, а в результате формируется цифровая подпись *w*.

Ключ подписи *d* является элементом секретных данных, специфичным для субъекта и используемым только данным субъектом в процессе формирования цифровой подписи.

Параметры схемы ЭЦП – элементы данных, общие для всех субъектов схемы цифровой подписи, известные или доступные всем этим субъектам. Электронная цифровая подпись *w* представляет собой строку битов, полученную в результате процесса формирования подписи. Данная строка имеет внутреннюю структуру, зависящую от конкретного механизма формирования подписи.

В процессе проверки цифровой подписи в качестве исходных данных используются подписанное сообщение, ключ проверки *Q* и параметры схемы ЭЦП, а результатом проверки является заключение о правильности или ошибочности цифровой подписи.

Ключ проверки *Q* является элементом данных, математически связанным с ключом подписи *d* и используемым проверяющей стороной в процессе проверки цифровой подписи.

Схематическое представление подписанного сообщения показано на рис. 1.22. Поле «Текст», дополняющее поле «Цифровая подпись», может содержать идентификаторы субъекта, подписавшего сообщение, и/или метку времени.

Рис. 1.22. Схема подписанного сообщения

Установленная в данном стандарте схема цифровой подписи должна быть реализована с использованием операций группы точек эллиптической кривой, определенной над конечным простым полем, а также хэш-функции.

Криптографическая стойкость данной схемы цифровой подписи основывается на сложности решения задачи дискретного логарифмирования в группе точек эллиптической кривой, а также на стойкости используемой хэш-функции. Алгоритм вычисления хэш-функции установлен в ГОСТ Р 34.11.

Цифровая подпись представляет собой двоичный вектор длиной 512 бит, вычисляется и проверяется с помощью определенного набора правил изложенных ниже.

Параметры схемы цифровой подписи для ее формирования и проверки:

- простое число p – модуль эллиптической кривой. Это число должно удовлетворять неравенству $p > 2^{255}$. Верхняя граница данного числа должна определяться при конкретной реализации схемы цифровой подписи;
- эллиптическая кривая *E*, задаваемая своим инвариантом $J(E)$ или коэффициентами $a, b \in F_p$;
- целое число *m* - порядок группы точек эллиптической кривой *E*;
- простое число *q* – порядок циклической подгруппы группы точек эллиптической кривой *E*, для которого выполнены следующие условия:

$$\begin{cases} m = nq, n \in Z, n \geq 1 \\ 2^{254} < q < 2^{256} \end{cases};$$

- точка $P \neq 0$ эллиптической кривой E с координатами (x_p, y_p), удовлетворяющая равенству $qP = 0$;
- хэш-функция $h(\cdot)$: $V_\infty \rightarrow V_{256}$, отображающая сообщения, представленные в виде двоичных векторов произвольной конечной длины, в двоичные век торы длины 256 бит. Хэш-функция определена в ГОСТ Р 34.11.

Каждый пользователь схемы цифровой подписи должен обладать личными ключами:

- ключом подписи – целым числом d, удовлетворяющим неравенству $0 < d < q$;
- ключом проверки – точкой эллиптической кривой Q с координатами (x_q, y_q), удовлетворяющей равенству $dP = Q$.

На приведенные выше параметры схемы цифровой подписи накладываются следующие требования:

- должно быть выполнено условие $p^t \neq 1 (\mathrm{mod}\ p)$ для всех целых $t = 1, 2, ..., B$, где B удовлетворяет неравенству $B \geq 31$;
- должно быть выполнено неравенство $m \neq p$;
- инвариант кривой должен удовлетворять условию $j(E) \neq 0$ или 1728.

Двоичные векторы

Для определения процессов формирования и проверки цифровой подписи необходимо установить соответствие между целыми числами и двоичными векторами длиной 256 бит.

Рассмотрим следующий двоичный вектор длиной 256 бит, в котором младшие биты расположены справа, а старшие - слева:

$$\overline{h} = (\alpha_{255}, ..., \alpha_0), \overline{h} \in V_{255},$$

где α_i, i = 0..256, равно либо 1, либо 0. Будем считать, что число $\alpha \in Z$ соответствует двоичному вектору h, если выполнено равенство

$$\alpha = \sum_{i=0}^{255} \alpha_i 2^i.$$

Для двух двоичных векторов $\overline{h}_1$ и $\overline{h}_2$, соответствующих целым числам α и β, определим операцию конкатенации (объединения) следующим образом. Пусть

$$\overline{h}_1 = (\alpha_{255}, ..., \alpha_0),$$

$$\overline{h}_2 = (\beta_{255}, ..., \beta_0),$$

тогда их объединение имеет вид

$$\overline{h}_1 \| \overline{h}_2 = (\alpha_{255}, ..., \alpha_0, \beta_{255}, ..., \beta_0)$$

и представляет собой двоичный вектор длиной 512 бит, составленный из коэффициентов векторов $\overline{h}_1$ и $\overline{h}_2$.

С другой стороны, приведенные формулы определяют способ разбиения двоичного вектора $\overline{h}$ длиной 512 бит на два двоичных вектора длиной 256 бит, конкатенацией которых он является.

Основные процессы

В данном разделе определены процессы формирования и проверки электронной цифровой подписи под сообщением пользователя.

Для реализации данных процессов необходимо, чтобы всем пользователям были известны параметры схемы цифровой подписи, удовлетворяющие приведенным выше требованиям.

Кроме того, каждому пользователю необходимо иметь ключ подписи d и ключ проверки подписи $Q(x_q,y_q)$, которые также должны удовлетворять приведенным выше требованиям.

Формирование цифровой подписи. Для получения цифровой подписи под сообщением $M \in V\mu$ необходимо выполнить следующие действия (шаги).

Шаг 1 – вычислить хэш-код сообщения M:

$\bar{h} = h(M)$.

Шаг 2 – вычислить целое число α, двоичным представлением которого является вектор $\bar{h}$, и определить значение

$e \equiv \alpha (\mathrm{mod}\, q)$.

Если $e = 0$, то определить $e = 1$.

Шаг 3 – сгенерировать случайное (псевдослучайное) целое число k, удовлетворяющее неравенству

$0 < k < q$.

Шаг 4 – вычислить точку эллиптической кривой $C = kP$ и определить

$r \equiv x_c (\mathrm{mod}\, q)$,

где x_i – x-координата точки C. Если $r = 0$, то вернуться к шагу 3.

Шаг 5 – вычислить значение

$s = (rd + ke)(\mathrm{mod}\, q)$.

Если $s = 0$, то вернуться к шагу 3.

Шаг 6 – вычислить двоичные векторы $\bar{r}$ и $\bar{s}$, соответствующие r и s, и определить цифровую подпись $w = (\bar{r} \| \bar{s})$ как конкатенацию двух двоичных векторов.

Исходными данными этого процесса являются ключ подписи d и подписываемое сообщение M, а выходным результатом – цифровая подпись w.

Проверка цифровой подписи. Для проверки цифровой подписи w, под полученным сообщением M необходимо выполнить следующие действия (шаги).

Шаг 1 – по полученной подписи w вычислить целые числа r и s. Если выполнены неравенства $0 < r < q$, $0 < s < q$, то перейти к следующему шагу. В противном случае подпись неверна.

Шаг 2 – вычислить хэш-код полученного сообщения M:

$\bar{h} = h(M)$.

Шаг 3 – вычислить целое число a, двоичным представлением которого является вектор $\bar{h}$, и определить

$e \equiv \alpha (\mathrm{mod}\, q)$.

Если $e = 0$, то определить $e = 1$.

Шаг 4 – вычислить значение

$v \equiv e^{-1} (\mathrm{mod}\, q)$.

Шаг 5 – вычислить значения

$$z_1 \equiv sv(\mathrm{mod}\, q)\,, \quad z_2 \equiv -rv(\mathrm{mod}\, q)\,.$$

Шаг 6 – вычислить точку эллиптической кривой $C = z_1 P + z_2 Q$ и определить

$$R \equiv x_c\,(\mathrm{mod}\, q),$$

где x_c – x-координата точки C.

Шаг 7 – если выполнено равенство $R = r$, то подпись принимается, в противном случае подпись неверна.

Исходными данными этого процесса являются подписанное сообщение *M*, цифровая подпись *w* и ключ проверки *Q*, а выходным результатом – свидетельство о достоверности или ошибочности данной подписи.

Внедрение цифровой подписи на базе стандарта ГОСТ Р 34.10-2001 повышает, по сравнению с предшествующей схемой цифровой подписи, уровень защищенности передаваемых сообщений от подделок и искажений. Этот стандарт рекомендуется использовать в новых системах обработки информации различного назначения, а также при модернизации действующих систем.