35. Безопасность компьютерных систем. Системы электронно-цифровой подписи. Хэш-функции. Алгоритм ЭЦП. Схема слепой подписи. Схемы неоспоримой подписи.

Безопасность компьютерных систем.

Информационная безопасность - состояние защищенности информации и информационной среды от случайных или преднамеренных воздействий естественного или искусственного характера, которые могут нанести неприемлемый ущерб субъектам информационных отношений.

Защита информации. **Целостность** информации; **Доступность** информации.
Аутентичность - возможность достоверно установить автора сообщения. **Апеллируемость** - возможность доказать, что автором является именно данный человек
Угроза информационной безопасности
Атака Нарушение Нарушитель Злоумышленник
Безопасность АИС: 1. **Внешняя** 2. **Внутренняя**
Меры противодействия угрозам безопасности: законодательные (правовые), **административные** (организационные), **процедурные** и **программно-технические.**
Криптографическая система — это система обеспечения информационной безопасности сети или АИС, использующая криптографические средства.
. **Шифрование** — это процесс преобразования исходного сообщения M (называемого открытым текстом) в форму M' (зашифрованный текст или шифртекст). При этом провести обратное преобразование M' в M возможно только обладая некоторой дополнительной информацией, называемой ключом.
Симметричное шифрование.
В симметричных алгоритмах шифрования один и тот же ключ K используется для того, чтобы зашифровать сообщение и для его последующей расшифровки. Таким образом, и отправитель и получатель сообщения должны располагать одним и тем же клю чом. Схематично это можно записать в виде:
M' = E(M, K) ; M = D(M', K), где E — функция шифрования (encrypt), а D — функция дешифрования (decrypt), обе используют ключ K в качестве одного из параметров.
Подстановочные алгоритмы шифрования работают по следующему принципу: каждый символ (или последовательность символов) исходного сообщения заменяются другим символом (или другой последовательностью символов). *Например: 1. Шифр Цезаря. 2. Моноалфавитный шифр (шифр простой замены)*
Перестановочные алгоритмы. (в лесу родилась елочка)
Шифрование с открытым ключом (Ассиметричные криптосистемы).
Принципиальное отличие асимметричной криптосистемы от криптосистемы симметричного шифрования состоит в том, что для шифрования информации и ее последующего расшифровывания используются различные ключи:
• *открытый ключ K* используется для шифрования информации, вычисляется из секретного ключа *к*;
• *секретный ключ к* используется для расшифровывания информации, зашифрованной с помощью парного ему открытого ключа *K*.
Эти ключи различаются таким образом, что с помощью вычислений нельзя вывести секретный ключ *к* из открытого ключа *K*. Поэтому открытый ключ *K* может свободно передаваться по каналам связи.

Системы электронно-цифровой подписи.

При обмене электронными документами по сети связи существенно снижаются затраты на обработку и хранение документов, убыстряется их

поиск. Но при этом возникает проблема аутентификации автора документа и самого документа, т.е. установления подлинности автора и отсутствия изменений в полученном документе. В обычной (бумажной) информатике эти проблемы решаются за счет того, что информация в документе и рукописная подпись автора жестко связаны с физическим носителем (бумагой). В электронных документах на машинных носителях такой связи нет.

6.1. Проблема аутентификации данных

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

1) Активный перехват – нарушитель, подключившийся к сети, перехватывает документы (файлы) и изменяет их.

2) Маскарад – абонент С посылает документ абоненту В от имени абонента А.

3) Ренегатство – абонент А заявляет, что не посылал сообщения абоненту В, хотя на самом деле посылал.

4) Подмена – абонент В изменяет или формирует новый документ и заявляет, что получил его от абонента А.

5) Повтор – абонент С повторяет ранее переданный документ, который абонент А посылал абоненту В.

При обработке документов в электронной форме совершенно непригодны традиционные способы установления подлинности по рукописной подписи и оттиску печати на бумажном документе. Принципиально новым решением является электронная цифровая подпись (ЭЦП).

Электронная цифровая подпись используется для аутентификации текстов, передаваемых по телекоммуникационным каналам. Функционально она аналогична обычной рукописной подписи и обладает ее основными достоинствами:

1) Удостоверяет, что подписанный текст исходит от лица, поставившего подпись.

2) Не дает самому этому лицу возможности отказаться от обязательств, связанных с подписанным текстом.

3) Гарантирует целостность подписанного текста.

Цифровая подпись представляет собой относительно небольшое количество дополнительной цифровой информации, передаваемой вместе с подписываемым текстом.

Система ЭЦП включает две процедуры:

1) Процедуру постановки подписи.

2) Процедуру проверки подписи.

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

При формировании ЭЦП отправитель прежде всего вычисляет хеш-функцию $h(M)$ подписываемого текста M. Вычисленное значение хеш-функции $h(M)$ представляет собой один короткий блок информации m, характеризующий весь текст M в целом. Затем число m шифруется секретным ключом отправителя. Получаемая при этом пара чисел представляет собой ЭЦП для данного текста M.

При проверке ЭЦП получатель сообщения снова вычисляет хеш-функцию $m = h(M)$ принятого по каналу текста M, после чего при помощи открытого ключа отправителя проверяет, соответствует ли полученная подпись вычисленному значению m хеш-функции.

Каждая подпись содержит следующую информацию:

1) Дату подписи.

2) Срок окончания действия ключа данной подписи.

3) Информацию о лице, подписавшем файл (ФИО, должность, краткое наименование фирмы).

4) Идентификатор подписавшего (имя открытого ключа).

5) Собственно цифровую подпись.

Хэш-функции. 5. Хеш-функция5.1. Однонаправленные хеш-функцииХеш-функция предназначена для сжатия подписываемого документа M до нескольких десятков или сотен бит. Хеш-функция $h(\cdot)$ принимает в качестве аргумента сообщение (документ) M произвольной длины и возвращает хеш-значение $h(M)=H$ фиксированной длины. Обычно хешированная информация является сжатым двоичным представлением основного сообщения произвольной длины. Следует отметить, что значение хеш-функции $h(M)$ сложным образом зависит от документа M и не позволяет восстановить сам документ M.

Хеш-функция должна удовлетворять целому ряду условий:

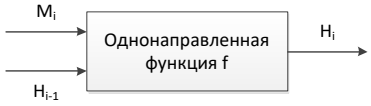
1) Хеш-функция должна быть чувствительна к всевозможным изменениям в тексте M, таким как вставки, выбросы, перестановки и т.п.

2) Хеш-функция должна обладать свойством необратимости, т.е. задача подбора документа M, который обладал бы требуемым значением хеш-функции, должна быть вычислительно неразрешима.

3) Вероятность того, что значения хеш-функций двух различных документов (вне зависимости от их длин) совпадут, должна быть ничтожно мала.

Большинство хэш-функций строится на основе однонаправленной функции $f(\cdot)$, которая образует выходное значение длиной n при задании двух входных значений длиной n. Этими входами являются блок исходного текста M_i и хеш-значение H_{i-1} предыдущего блока текста:

$$H_i = f(M_i, H_{i-1}).$$



Хеш-значение, вычисляемое при вводе последнего блока текста, становится хеш-значением всего сообщения M.

В результате однонаправленная хеш-функция всегда формирует выход фиксированной длины n (независимо от длины входного текста).

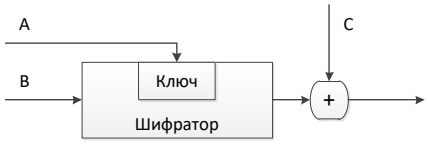
5.2. Однонаправленные хеш-функции на основе симметричных блочных алгоритмов

Однонаправленную хеш-функцию можно построить, используя симметричный блочный алгоритм. Наиболее очевидный подход состоит в том, чтобы шифровать сообщение M посредством блочного алгоритма в режиме CBC или CFB с помощью фиксированного ключа и некоторого вектора инициализации IV. Последний блок шифротекста можно рассматривать в качестве хеш-значения сообщения M. При таком подходе не всегда можно построить безопасную однонаправленную хеш-функцию, но всегда можно получить код аутентификации сообщения MAC (Message Authentication Code).

Более безопасный вариант хеш-функции можно получить, используя блок сообщения в качестве ключа, предыдущее значение – в качестве входа, а текущее хеш-значение – в качестве выхода. Реальные хеш-функции проектируются еще более сложными. Длина блока обычно определяется длиной ключа, а длина хеш-значения совпадает с длиной блока.

Поскольку большинство блочных алгоритмов являются 64-битовыми, некоторые схемы хеширования проектируют так, чтобы хеш-значение имело длину, равную двойной длине блока.

Если принять, что получаемая хэш-функция корректна, безопасность схемы хеширования базируется на безопасности лежащего в ее основе блочного алгоритма. Покажем схему хеширования, у которой длина хеш-значения равна длине блока.



Ее работа описывается выражениями:

$$H_0 = I_H,$$
$$H_i = E_A(B) \oplus C,$$

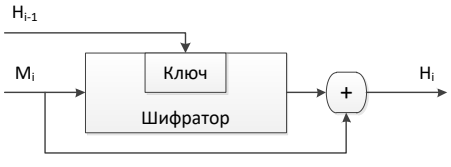
где I_H - некоторое случайное начальное значение; A, B и C могут принимать значения M_i , H_i , $(M_i \oplus H_{i-1})$ или быть константами.

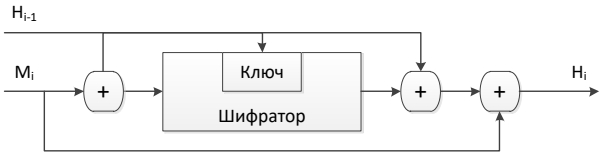
Сообщение M разбивается на блоки M_i принятой длины, которые обрабатываются поочередно.

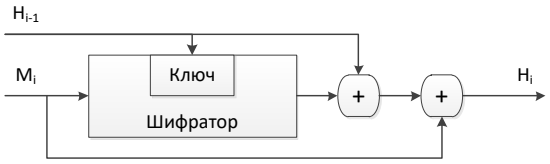
Три различные переменные A, B и C могут принимать одно из четырех возможных значений, поэтому в принципе можно получить 64 варианта общей схемы этого типа. Из них 52 варианта являются либо тривиально слабыми, либо небезопасными. Остальные 12 безопасных схем хеширования перечислены в таблице.

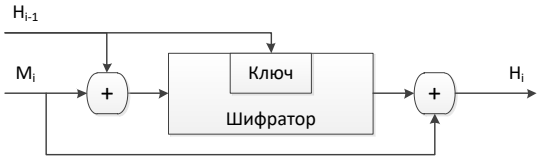
Номер схемы	Функция хеширования
1	$H_i = E_{H_{i-1}}(M_i) \oplus M_i$
2	$H_i = E_{H_{i-1}}(M_i \oplus H_{i-1}) \oplus M_i \oplus H_{i-1}$
3	$H_i = E_{H_{i-1}}(M_i) \oplus H_{i-1} \oplus M_i$
4	$H_i = E_{H_{i-1}}(M_i \oplus H_{i-1}) \oplus M_i$
5	$H_i = E_{M_i}(H_{i-1}) \oplus H_{i-1}$
6	$H_i = E_{M_i}(M_i \oplus H_{i-1}) \oplus M_i \oplus H_{i-1}$
7	$H_i = E_{M_i}(H_{i-1}) M_i \oplus H_{i-1}$
8	$H_i = E_{M_i}(M_i \oplus H_{i-1}) \oplus H_{i-1}$
9	$H_i = E_{M_i \oplus H_{i-1}}(M_i) \oplus M_i$
10	$H_i = E_{M_i \oplus H_{i-1}}(H_{i-1}) \oplus H_{i-1}$
11	$H_i = E_{M_i \oplus H_{i-1}}(M_i) \oplus H_{i-1}$
12	$H_i = E_{M_i \oplus H_{i-1}}(H_{i-1}) \oplus M_i$

Приведем первые четыре схемы хеширования, являющиеся безопасными при всех атаках.









5.3. Отечественный стандарт хеш-функции

Российский стандарт ГОСТ Р 34.11-94 определяет алгоритм и процедуру вычисления хеш-функции для любых последовательностей двоичных символов, применяемых в криптографических методах обработки и защиты информации. Этот стандарт базируется на блочном алгоритме шифрования ГОСТ 29147-89, хотя в принципе можно было бы использовать и другой блочный алгоритм шифрования с 64-битовым блоком и 256-битовым ключом.

Данная хеш-функция формирует 256-битовое хеш-значение.

Функция сжатия $H_i = f(M_i, H_{i-1})$ (оба операнда M_i и H_{i-1} являются 256-битовыми величинами) определяется следующим образом:

1) Генерируются 4 ключа шифрования K_j, $j = 1 \ldots 4$, путем линейного смешивания M_i, H_{i-1} и некоторой константы C_j.

2) Каждый ключ K_j используют для шифрования 64-битовых подслов h_i слова H_{i-1} в режиме простой замены: $S_j = E_{K_j}(h_j)$. Результирующая последовательность S_4, S_3, S_2, S_1 длиной 256 бит запоминается во временной переменной S.

3) Значение H_i является сложной, хотя и линейной функцией смешивания S, M_i и H_{i-1}.

При вычислении окончательного хеш-значения сообщения M учитываются значения трех связанных между собой переменных:

H_n - хеш-значение последнего блока сообщения;

Z – значение контрольной суммы, получаемой при сложении по модулю 2 всех блоков сообщения;

L – длина сообщения.

Эти три переменные и дополнительный последний блок M' сообщения объединяются в окончательное хеш-значение следующим образом:

$$H = f(Z \oplus M', f(L, f(M', H_n))).$$

Данная хеш-функция определена стандартом ГОСТ Р 34.11-94 для использования совместно с российским стандартом электронной цифровой подписи.

Алгоритм ЭЦП.

6.2. Алгоритм цифровой подписи RSA

Первой и наиболее известной во всем мире конкретной системой ЭЦП стала система RSA, математическая схема которой была разработана в 1977 г. в Массачусетском технологическом институте США.

Сначала необходимо вычислить пару ключей (секретный ключ и открытый ключ). Для этого отправитель (автор) электронных документов вычисляет два больших простых числа P и Q, затем находит их произведение

$$N = P \cdot Q$$

и значение функции

$$\varphi(N) = (P-1)(Q-1).$$

Далее отправитель вычисляет число E из условий:

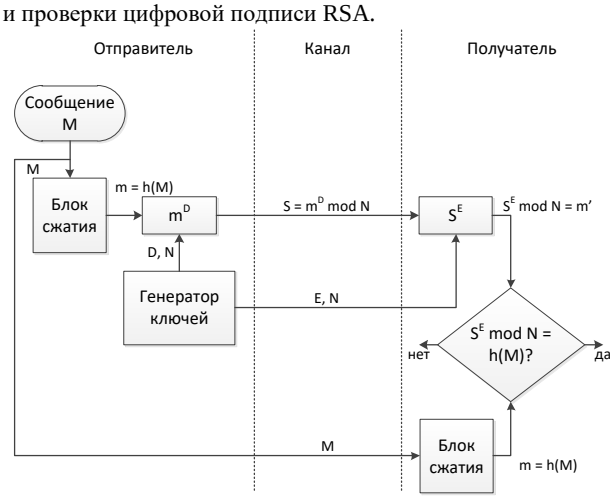
$$E \le \varphi(N), \quad НОД(E, \varphi(N)) = 1$$

и число D из условий:

$$D < N, \quad E \cdot D \equiv 1 \bmod \varphi(N).$$

Пара чисел (E, N) является открытым ключом. Эту пару чисел автор передает партнерам по переписке для проверки его цифровых подписей. Число D сохраняется автором как секретный ключ для подписывания.

Покажем обобщенную схему формирования и проверки цифровой подписи RSA.



Допустим, что отправитель хочет подписать сообщение M перед его отправкой. Сначала сообщение M (блок информации, файл, таблица) сжимают с помощью хеш-функции $h(\cdot)$ в целое число m:

$$m = h(M).$$

Затем вычисляют цифровую подпись S под электронным документом M, используя хеш-значение m и секретный ключ D:

$$S = m^D \bmod N.$$

Пара (M, S) передается партнеру-получателю как электронный документ M, подписанный цифровой подписью S, причем подпись S сформирована обладателем секретного ключа D.

После приема пары (M, S) получатель вычисляет хеш-значение сообщения M двумя разными способами. Прежде всего он восстанавливает хеш-значение m', применяя криптографическое преобразование подписи S с использованием открытого ключа E:

$$m' = S^E \bmod N.$$

Кроме того, он находит результат хеширования принятого сообщения M с помощью такой же хеш-функции $h(\cdot)$:

$$m = h(M).$$

Если соблюдается равенство вычисленных значений, т.е.

$$S^E \bmod N = h(M),$$

то получатель признает пару (M, S) подлинной. Доказано, что только обладатель секретного ключа D может сформировать цифровую подпись S по документу M, а определить секретное число D по открытому числу E не легче, чем разложить модуль N на множители.

Кроме того, можно строго математически доказать, что результат проверки цифровой подписи S будет положительным только в том случае, если при вычислении S был использован секретный ключ D, соответствующий открытому ключу E. Поэтому открытый ключ E иногда называют "идентификатором" подписавшего.

Недостатки алгоритма цифровой подписи RSA:

1. При вычислении модуля N, ключей E и D для системы цифровой подписи RSA необходимо проверять большое количество дополнительных условий, что сделать практически трудно. Невыполнение любого из этих условий делает возможным фальсификацию цифровой подписи со стороны того, кто обнаружит такое невыполнение. При подписании важных документов нельзя допускать такую возможность даже теоретически.

2. Для обеспечения криптостойкости цифровой подписи RSA по отношению к попыткам фальсификации на уровне, например, национального стандарта США на шифрование информации (алгоритм DES), т.е. 10^{18}, необходимо использовать при вычислениях N, D и E целые числа не менее 2^{15} (или около 10^{154}) каждое, что требует больших вычислительных затрат, превышающих на 20…30% вычислительные затраты других алгоритмов цифровой подписи при сохранении того же уровня криптостойкости.

3. Цифровая подпись RSA уязвима к так называемой мультипликативной атаке. Иначе говоря, алгоритм цифровой подписи RSA позволяет злоумышленнику без знания секретного ключа D сформировать подписи под теми документами, у которых результат хеширования можно вычислить как произведение результатов хеширования уже подписанных документов.

Пример. Допустим, что злоумышленник может сконструировать три сообщения M_1, M_2 и M_3, у которых хеш-значения

$$m_1 = h(M_1), \quad m_2 = h(M_2), \quad m_3 = h(M_3),$$

причем

$$m_3 = (m_1 \cdot m_2) \bmod N.$$

Допустим также, что для двух сообщений M_1 и M_2 получены законные подписи

$$S_1 = m_1^{\,D} \bmod N \text{ и } S_2 = m_2^{\,D} \bmod N.$$

Тогда злоумышленник может легко вычислить подпись S_3 для документа M_3, даже не зная секретного ключа D:

$$S_3 = (S_1 \cdot S_2) \bmod N.$$

Действительно,

$$(S_1 \cdot S_2) \bmod N = (m_1^D \cdot m_2^D) \bmod N = (m_1 \cdot m_2)^D \bmod N = m_3^D \bmod N = S_3.$$

Более надежный и удобный для реализации на персональных компьютерах алгоритм цифровой подписи был разработан в 1984 г. американцем арабского происхождения Тахером Эль Гамалем. В 1991 г. НИСТ США обосновал перед комиссией Конгресса США выбор алгоритма цифровой подписи

Эль Гамаля в качестве основы для национального стандарта.

6.3. Алгоритм цифровой подписи Эль Гамаля (EGSA)

Название EGSA происходит от слов El Gamal Signature Algorithm (алгоритм цифровой подписи Эль Гамаля). Идея EGSA основана на том, что для обоснования практической невозможности фальсификации цифровой подписи может быть использована более сложная вычислительная задача, чем разложение на множители большого целого числа, - задача дискретного логарифмирования. Кроме того, Эль Гамалю удалось избежать явной слабости алгоритма цифровой подписи RSA, связанной с возможностью подделки цифровой подписи под некоторыми сообщениями без определения секретного ключа.

Рассмотрим подробнее алгоритм цифровой подписи Эль Гамаля. Для того чтобы генерировать пару ключей (открытый ключ – секретный ключ), сначала выбирают некоторое большое простое целое число P и большое целое число G, причем $G < P$. Отправитель и получатель подписанного документа используют при вычислениях одинаковые большие целые числа P ($\sim 10^{154}$ или $\sim 2^{512}$), которые не являются секретными.

Отправитель выбирает случайное целое число X, $1 < X \le (P-1)$, и вычисляет

$$Y = G^X \bmod P.$$

Число Y является открытым ключом, используемым для проверки подписи отправителя. Число Y открыто передается всем потенциальным получателям документов.

Число X является секретным ключом отправителя для подписывания документов и должно храниться в секрете.

Для того чтобы подписать сообщение M, сначала отправитель хеширует его с помощью хеш-функции $h(\cdot)$ в целое число m:

$$m = h(M), \quad 1 < m < (P-1),$$

и генерирует случайное целое число K, $1 < K \le (P-1)$, такое, что K и $(P-1)$ являются взаимно простыми. Затем отправитель вычисляет целое число a:

$$a = G^K \bmod P$$

и, применяя расширенный алгоритм Евклида, вычисляет с помощью секретного ключа X целое число b из уравнения

$$m = (X \cdot a + K \cdot b) \bmod (P-1).$$

Пара чисел (a,b) образует цифровую подпись S:

$$S = (a,b),$$

проставляемую под документом M.

Тройка чисел (M,a,b) передается получателю, в то время как пара чисел (X,K) держится в секрете.

После приема подписанного сообщения (M,a,b) получатель должен проверить, соответствует ли подпись $S = (a,b)$ сообщению M. Для этого получатель сначала вычисляет по принятому сообщению M число

$$m = h(M),$$

т.е. хеширует принятое сообщение M.

Затем получатель вычисляет значение

$$A = (Y^a a^b) \bmod P$$

и признает сообщение M подлинным, если, и только если

$$A = G^m \bmod P.$$

Иначе говоря, получатель проверяет справедливость соотношения

$$\left(Y^a a^b\right) \bmod P = G^m \bmod P.$$

Пример. Выберем: числа $P = 11$, $G = 2$ и секретный ключ $X = 8$. Вычисляем значение открытого ключа:

$$Y = G^X \bmod P = 2^8 \bmod 11 = 256 \bmod 11 = 3.$$

Предположим, что исходное сообщение M характеризуется хеш-значением $m = 5$.

Для того чтобы вычислить цифровую подпись для сообщения M, имеющего хеш-значение $m = 5$, сначала выберем случайное целое число $K = 9$. Убедимся, что числа K и $(P-1)$ являются взаимно простыми. Действительно,

$$НОД(K, P-1) = НОД(9,10) = 1.$$

Далее вычисляем элементы a и b подписи:

$$a = G^K \bmod P = 2^9 \bmod 11 = 512 \bmod 11 = 6,$$

элемент b определяем, используя расширенный алгоритм Евклида:

$$m = \left(X \cdot a + K \cdot b\right) \bmod (P-1).$$

При $m = 5$, $a = 6$, $X = 8$, $K = 9$, $P = 11$ получаем

$$5 = \left(8 \cdot 6 + 9 \cdot b\right) \bmod (11-1)$$

$$5 = \left(48 + 9 \cdot b\right) \bmod 10$$

или

$$9 \cdot b = -43 \bmod 10.$$

$$9 \cdot b = 7 \bmod 10.$$

Сначала решаем сравнение

$$9 \cdot y \equiv 1 \bmod 10.$$

$$y = 9^{-1} \bmod 10.$$

Используя расширенный алгоритм Евклида, выполним вычисления, записывая результаты отдельных шагов в таблицу.

q	u_1	u_2	u_3	v_1	v_2	v_3
-	0	1	n=10	1	0	a=9
1	1	0	9	-1	1	1
9	-1	1	1	10	-9	0

Итерация 1:

1) $(u_1, u_2, u_3) := (0,1,10), (v_1, v_2, v_3) := (1,0,9)$

2) $u_3 \neq 1$

3) $q = [u_3 / v_3] = [10/9] = 1$

$$(t_1, t_2, t_3) = (u_1, u_2, u_3) - (v_1, v_2, v_3) \cdot q =$$

$$(0,1,10) - (1,0,9) \cdot 1 = (0,1,10) - (1,0,9) = (-1,1,1),$$

$$(u_1, u_2, u_3) = (v_1, v_2, v_3) = (1,0,9),$$

$$(v_1, v_2, v_3) = (t_1, t_2, t_3) = (-1,1,1).$$

Итерация 2:

2) $u_3 \neq 1$

3) $q = [u_3 / v_3] = [9/1] = 9$

$$(t_1, t_2, t_3) = (u_1, u_2, u_3) - (v_1, v_2, v_3) \cdot q =$$

$$(1,0,9) - (-1,1,1) \cdot 9 = (1,0,9) - (-9,9,9) = (10,-9,0),$$

$$(u_1, u_2, u_3) = (v_1, v_2, v_3) = (-1,1,1),$$

$$(v_1, v_2, v_3) = (t_1, t_2, t_3) = (10,-9,0).$$

Итерация 3:

2) $u_3 = 1$

При $u_1 = -1, u_2 = 1, u_3 = 1$

$$(a \cdot u_1 + n \cdot u_2) \bmod n = (9 \cdot (-1) + 10 \cdot 1) \bmod 10 = (-9 + 10) \bmod 10 = 1 \equiv$$
$$\equiv (a \cdot u_1) \bmod n = (9 \cdot (-1)) \bmod 10 = -9 \bmod 10 = 1 \equiv 1,$$
$$a^{-1} \bmod n = 9^{-1} \bmod 10 \equiv u_1 \bmod n = -1 \bmod 10 = 9.$$

Получаем $y = 9^{-1} \bmod 10 = 9.$ Затем находим

$$b = (9^{-1} \cdot 7) \bmod 10 = (9 \cdot 7) \bmod 10 = 63 \bmod 10 = 3.$$

Решение: $b = 3$. Цифровая подпись представляет собой пару: $a = 6$, $b = 3$.

Далее отправитель передает подписанное сообщение. Приняв подписанное сообщение и открытый ключ $Y = 3$, получатель вычисляет хеш-значение для сообщения $M : m = 5$, а затем вычисляет два числа:

1)

$$(Y^a a^b) \bmod P = (3^6 \cdot 6^3) \bmod 11 = 157464 \bmod 11 = 10 \bmod 11 = 10$$

;

2)

$$G^m \bmod P = 2^5 \bmod 11 = 32 \bmod 11 = 10 \bmod 11 = 10.$$

Так как эти два целых числа равны, принятое получателем сообщение признается подлинным.

Следует отметить, что схема Эль Гамаля является характерным примером подхода, который допускает пересылку сообщения M в открытой форме вместе с присоединенным аутентификатором (a,b). В таких случаях процедура установления подлинности принятого сообщения состоит в проверке соответствия аутентификатора сообщению.

1. Схема слепой подписи.

В отличие от обычных схем цифровой подписи, схемы слепой подписи (иногда называемые схемами подписи вслепую) являются двусторонними протоколами между отправителем A и стороной B, подписывающей документ.

Основная идея этих схем заключается в следующем. Отправитель A посылает порцию информации стороне B, которую B подписывает и возвращает A. Используя полученную подпись, сторона A может вычислить подпись стороны B на более важном для себя сообщении m. По завершении этого протокола сторона B ничего не знает ни о сообщении m, ни о подписи под этим сообщением.

Цель слепой подписи состоит в том, чтобы воспрепятствовать подписывающему лицу B ознакомиться с сообщением стороны A, которое он подписывает, и с соответствующей подписью под этим сообщением. Поэтому в дальнейшем подписанное сообщение невозможно связать со стороной A.

Приведем пример применения слепой подписи. Схема слепой подписи может найти применение в тех случаях, когда отправитель A (клиент банка) не хочет, чтобы подписывающая сторона B (банк) имела возможность в дальнейшем связать сообщение m и подпись $s_B(m)$ с определенным шагом выполненного ранее протокола.

В частности, это может быть важно при организации анонимных безналичных расчетов, когда сообщение m могло бы представлять денежную сумму, которую A хочет потратить. Когда сообщение m с подписью $s_B(m)$ предъявляется банку B для оплаты, банк B не может проследить, кто именно из клиентов предъявляет подписанный документ. Это позволяет пользователю A остаться анонимным.

Для построения протокола слепой подписи необходимы следующие компоненты:

1. Механизм обычной цифровой подписи для подписывающей стороны B. Пусть $s_B(X)$ обозначает подпись стороны B на документе X.

2. Функции $f(\cdot)$ и $g(\cdot)$ (известные только отправителю) такие, что

$$g(s_B(f(m))) = s_m(m),$$

где $f(\cdot)$ - маскирующая (blinding) функция; $g(\cdot)$ - демаскирующая (unblinding) функция; $f(m)$ - замаскированное (blinded) сообщение m.

Пример

При выборе s_B , f и g существует ряд ограничений.

Выберем в качестве алгоритма подписи s_B для стороны B схему цифровой подписи RSA с открытым ключом (N, E) и секретным ключом D, причем $N = P \cdot Q$ - произведение двух больших случайных простых чисел.

Пусть k – некоторое фиксированное целое число, взаимно простое с N, т.е. $НОД(N, k) = 1$.

Маскирующая функция $f : Z_n \to Z_n$ определяется как $f(m) = (m \cdot k^E) \bmod N$, а демаскирующая $g : Z_n \to Z_n$ - как $g(m) = (k^{-1} \cdot m) \bmod N$. При таком выборе f, g и s получаем

$$g(s_B(f(m))) = g(s_B((m \cdot k^E) \bmod N)) =$$
$$= g((m^D \cdot k) \bmod N) = m^D \bmod N = s_m(m),$$

что соответствует требованию 2.

Согласно протоколу слепой подписи, который предложил Д. Чом, отправитель A сначала получает подпись стороны B на замаскированном сообщении m. Используя эту подпись, сторона A вычисляет подпись B на заранее выбранном сообщении m, где $0 \le m \le N - 1$. При этом стороне B ничего не известно ни о значении m, ни о подписи, связанной с m.

Пусть сторона B имеет для подписи по схеме RSA открытый ключ (N, E) и секретный ключ D. Пусть k – случайное секретное целое число, выбранное стороной A и удовлетворяющее условиям $0 \le k \le N - 1$ и $НОД(N, k)$.

Протокол слепой подписи Д. Чома включает следующие шаги:

1. Отправитель A вычисляет замаскированное сообщение $m = (m \cdot k^E) \bmod N$ и посылает его стороне B.

2. Подписывающая сторона B вычисляет подпись $s = m^D \bmod N$ и отправляет эту подпись стороне A.

3. Сторона A вычисляет подпись $s = (k^{-1} \cdot s) \bmod N$, которая является подписью B на сообщении m.

Нетрудно видеть, что

$$m^D \equiv (m \cdot k^E)^D \equiv (m^D \cdot k) \bmod N,$$

поэтому

$$k^{-1} \cdot s \equiv m^D \cdot k \cdot k^{-1} = m^D \bmod N.$$

Д. Чом разработал несколько алгоритмов слепой подписи для создания системы анонимных безналичных электронных расчетов eCash.

2. Схемы неоспоримой подписи.

Неоспоримая подпись, как и обычная цифровая подпись, зависит от подписанного документа и секретного ключа. Однако в отличие от обычных цифровых подписей неоспоримая подпись не может быть верифицирована без участия лица, поставившего эту подпись. Возможно, более подходящим названием для этих подписей было бы "подписи, не допускающие подлога".

Рассмотрим два возможных сценария применения неоспоримой подписи.

Сценарий 1. Сторона A (клиент) хочет получить доступ в защищенную зону, контролируемую стороной B (банком). Этой защищенной зоной может быть, например, депозитарий (хранилище ценностей клиентов). Сторона B требует от A поставить до предоставления клиенту доступа на заявке о допуске в защищенную зону подпись, время и дату. Если A применит неоспоримую подпись, тогда сторона B не сможет впоследствии доказать кому-либо, что A получил допуск без непосредственного участия A в процессе верификации подписи.

Сценарий 2. Предположим, что известная корпорация A разработала пакет программного обеспечения. Чтобы гарантировать подлинность пакета и отсутствие в нем вирусов, сторона A подписывает этот пакет неоспоримой подписью и продает его стороне B. Сторона B решает сделать копии этого пакета программного обеспечения и перепродать его третьей стороне C. При использовании стороной A неоспоримой подписи сторона C не сможет убедиться в подлинности этого пакета программного обеспечения и отсутствии в нем вирусов без участия стороны A.

Каждая сторона A должна выполнить следующее:

1. Выбрать случайное простое число $p = 2q + 1$, где q – также простое число.

2. Выбрать генераторное число α для подгруппы порядка q в циклической группе Z_p:

2.1. Выбрать случайный элемент $\beta \in Z_p$ и вычислить $\alpha = \beta^{\frac{p-1}{q}} \bmod p$.

2.2. Если $\alpha = 1$, тогда возвратиться к шагу 2.1.

3. Выбрать случайное целое $x \in \{1,2,\ldots,q-1\}$ и вычислить $y = \alpha^x \bmod p$.

4. Для стороны A открытый ключ равен (p,α,y), секретный ключ равен x.

Согласно алгоритму неоспоримой подписи Д. Чома, сторона A подписывает сообщение m, принадлежащее подгруппе порядка q в Z_p. Любая сторона B может проверить эту подпись при участии A.

В работе алгоритма неоспоримой подписи можно выделить два этапа:

1) Генерация подписи.

2) Верификация подписи.

На этапе генерации подписи сторона A вычисляет $s = m^x \bmod p$, где s – подпись стороны A на сообщении m. Сообщение m с подписью s отсылается стороне B.

Этап верификации подписи выполняется стороной B с участием стороны A и включает следующие шаги:

1. B получает подлинный открытый ключ (p,α,y) стороны A.

2. B выбирает два случайных секретных целых числа $a,b \in \{1,2,\ldots,q-1\}$.

3. B вычисляет $z = s^a y^b \bmod p$ и отправляет значение z стороне A.

4. A вычисляет $w = z^{\frac{1}{x}} \bmod p$, где $x \cdot x^{-1} \equiv 1 \bmod q$, и отправляет значение w стороне B.

5. B вычисляет $w' = m^a \alpha^b \bmod p$ и признает подпись s подлинной, если и только если $w = w'$.

Убедимся, что проверка подписи s работает:

$$w \equiv z^{\frac{1}{x}} \equiv \left(s^a y^b\right)^{\frac{1}{x}} \equiv \left(m^{xa} \alpha^{xb}\right)^{\frac{1}{x}} \equiv m^a \alpha^b \equiv w' \bmod p.$$

Подписавшая сторона A при некоторых обстоятельствах могла бы попытаться отказаться от своей подлинной подписи одним из трех способов:

а) Отказаться от участия в протоколе верификации.

б) Некорректно выполнить протокол верификации.

в) Объявить подпись фальшивой, даже если протокол верификации оказался успешным.

Отречение от подписи способом (а) рассматривалось бы как очевидная попытка неправомерного отказа. Против способов (б) и (в) бороться труднее, здесь требуется специальный протокол дезавуирования. Этот протокол определяет, пытается ли подписавшая сторона A дезавуировать правильную подпись s или эта подпись является фальшивой. В этом протоколе по существу дважды применяется протокол верификации и затем производится проверка с целью убедиться, что сторона A выполняет этот протокол корректно.

Протокол дезавуирования для схемы неоспоримой подписи Д. Чома включает следующие шаги:

1. B принимает от стороны A сообщение m с подписью s и получает подлинный открытый ключ (p, α, y) стороны A.

2. B выбирает случайные секретные целые числа $a, b \in \{1,2,\ldots,q-1\}$, вычисляет $z = s^a y^b \bmod p$ и отправляет значение z стороне A.

3. A вычисляет $w = z^{\frac{1}{x}} \bmod p$, где $x \cdot x^{-1} \equiv 1 \bmod q$, и отправляет значение w стороне B.

4. Если $w = m^a \alpha^b \bmod p$, тогда B признает подпись s подлинной и выполнение протокола прекращается.

5. B выбирает случайные секретные целые числа $a', b' \in \{1,2,\ldots,q-1\}$, вычисляет $z' = s^{a'} y^{b'} \bmod p$ и отправляет значение z' стороне A.

6. A вычисляет $w' = (z')^{\frac{1}{x}} \bmod p$ и отправляет значение w' стороне B.

7. Если $w' = m^{a'} \alpha^{b'} \bmod p$, тогда B принимает подпись s и выполнение протокола останавливается.

8. B вычисляет $c = \left(w \alpha^{-b}\right)^{a'} \bmod p$, $c' = \left(w' \alpha^{-b'}\right)^{a} \bmod p$. Если $c = c'$, тогда B заключает, что подпись s фальшивая; в противном случае B делает вывод, что подпись s подлинная, а сторона A пытается дезавуировать подпись s.

Нетрудно убедиться в том, что этот протокол достигает поставленной цели. Пусть m – сообщение и предположим, что s – подпись стороны A под сообщением m. Если подпись s фальшивая, т.е. $s \neq m^x \bmod p$, и если стороны A и B следуют протоколу должным образом, тогда $w = w'$ (и поэтому справедливо заключение B, что подпись s фальшивая). Пусть s на самом деле является подписью стороны A под сообщением m, т.е. $s = m^x \bmod p$. Предположим, что B точно следует протоколу, а A не следует. Тогда вероятность того, что $w = w'$ (и A преуспевает в дезавуировании подписи), составляет только $\frac{1}{q}$.

Следует отметить, что третья сторона C никогда не должна принимать в качестве доказательства подлинности подписи s запись стороной B протокола верификации, поскольку сторона B может выдумать успешную запись шага 2 и последующих шагов протокола верификации без участия подписывающей стороны A.

Неоспоримая подпись может быть верифицирована только путем непосредственного взаимодействия с подписывающей стороной A.

Разработан также алгоритм для обратимой неоспоримой подписи, которая может быть верифицирована, дезавуирована, а также преобразована в обычную цифровую подпись. Этот алгоритм основан на использовании алгоритма цифровой подписи Эль Гамаля.