ЛЕКЦИЯ 13.
Электронная цифровая подпись

13.1. Требования к цифровым подписям и их классификация.

13.2. Основные алгоритмы цифровых подписей.

Постановка задачи

Участники обмена сообщениями нуждаются в защите от следующих действий:

- ***отказ (ренегатство)*** – отправитель впоследствии отказывается от переданного сообщения;
- ***фальсификация*** – получатель подделывает сообщение;
- ***изменение*** – получатель вносит изменения в сообщение;
- ***маскировка*** – нарушитель маскируется под другого пользователя.

Цифровая подпись должна обеспечивать следующие возможности:

- Возможность установить **автора**, а также **дату и время** подписи.
- Возможность установить **достоверность** содержимого сообщения на время подписи.
- Возможность **проверки подписи третьей стороной** на случай возникновения спора.

Требования к цифровой подписи:

- Подпись должна быть ***двоичным кодом***, который **зависит** от *подписываемого сообщения*.
- Подпись должна использовать некоторую информацию, **уникальную** для отправителя, чтобы предотвратить возможность как ***фальсификации***, так и ***отрицания авторства***.
- Цифровую подпись должно быть относительно **просто произвести**.
- Цифровую подпись должно быть относительно **просто распознать и проверить**.
- С точки зрения вычислений должно быть **нереально фальсифицировать** цифровую подпись

ни с помощью ***создания нового сообщения***,
ни с помощью ***расшифровки*** созданной подписи.

Непосредственная цифровая подпись

может быть сформирована с помощью
1. шифрования *всего сообщения* ***личным*** ключом отправителя,
2. шифрования *хэш-кода* сообщения ***личным*** ключом отправителя.

Конфиденциальность может быть обеспечена шифрованием *всего сообщения* вместе с **подписью:**

- с помощью **открытого** ключа получателя (шифрование с **открытым** ключом),
- с помощью общего *секретного* ключа (традиционное шифрование).

Важно *сначала* выполнить **функцию подписи** и только *потом* — ***внешнюю функцию***, обеспечивающую конфиденциальность.

Слабое место **непосредственного применения цифровой подписи**:
пригодность всей схемы зависит от **защищенности *личного*** ключа отправителя.

Арбитражная цифровая подпись

Все схемы применения **арбитражных** цифровых подписей строятся следующим образом:

1. Каждое подписанное сообщение отправителя **X** адресату **Y** сначала попадает к арбитру **A**, который подвергает *сообщение и подпись* к нему ***тестированию*** по ряду *критериев*, чтобы проверить достоверность источника и содержимого сообщения.

2. После этого сообщение датируется и посылается **Y** с указанием того, что это сообщение было проверено и удовлетворило критериям **арбитра**.

Варианты схем арбитражных цифровых подписей

а) Традиционное шифрование, арбитр *может видеть* сообщение
(1) X→A: M || E_{Kxa} [ID_x || H(M)] **(2) A→Y: E_{Kay} [ID_x || M || E_{Kxa} [ID_x || H(M)] || T].**
б) Традиционное шифрование, арбитр *не видит* сообщения
(1) X→A: ID_x || E_{Kxy} [M] || E_{Kxa} [ID_x || H(E_{Kxy} [M])] **(2) A→Y: E_{Kay}[ID_x || E_{Kxy} [M] || E_{Kxa} [ID_x || H(E_{Kxy} [M])] || T].**
***в)* Шифрование с открытым ключом**, арбитр *не видит* сообщения
(1) X→A: ID_x || E_{KRx} [ID_x || E_{KUy}(E_{KRx}[M])] **(2) A→Y: E_{KRa} [ID_x || E_{KUy} (E_{KRx}[M]) || T].**

В таблице использованы обозначения:

X — отправитель,

Y — получатель,

A — арбитр,

M — сообщение.

Основные алгоритмы цифровых подписей

Электронная цифровая подпись (ЭЦП) Эль-Гамаля

1. Выбирается большое простое число **p** и целое число **g.** Эти числа публикуются.
2. Затем выбирается секретное число ***x***
3. и вычисляется **открытый** ключ ***для проверки подписи***

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

4. Далее для подписи сообщения **M** вычисляется его хэш-функция **h = H(M).**
5. Выбирается случайное целое **k: 1 < k < (p – 1),** *взаимно простое* с **p – 1**, и вычисляется

$$r = g^k \pmod p.$$

6. После этого с помощью расширенного алгоритма ***Евклида*** решается относительно **s** уравнение

$$h = xr + ks \pmod{(p-1)}.$$

Подпись образует пара чисел **(r, s).**

После выработки подписи значение **k** *уничтожается*.

Получатель *подписанного* сообщения

1. вычисляет хэш-функцию сообщения **h = H(M)**
2. и проверяет выполнение равенства

$$y^r r^s \pmod p = g^h.$$

Корректность этого уравнения очевидна:

$$y^r r^s = g^{xr} g^{ks} = g^{xr+ks} = g^h \pmod p.$$

ЭЦП Шнорра

1-й вариант

1. Выбирается **p** – большое простое число; **q** – простой делитель **(p – 1); g** – элемент порядка **q** в *Zp*; **k** – случайное число, ***x*** – секретный ключ.
2. Вычисляется

$$y = g^x \pmod p$$ – **открытый** ключ.

3. ***Уравнения выработки подписи*** имеют вид:

$$r = g^k \pmod p;$$

$$h = H(M, r);$$

$$s = k + x\,h \pmod q.$$

Подписью является пара **(r, s).**

На приемной стороне

1. вычисляется значение хэш-функции **h = H (M, r)** ,
2. проверяется выполнение равенства

$$r = g^s\, y^{-h} \pmod p,$$

при этом действия с показателями степени производятся по модулю ***q***.

2-й вариант

Для подписи сообщения **M**

1. выбирается случайное ***k*** ,
2. вычисляется

$$g^k \pmod p,$$

$$h = H(g^k, M),$$

$$z = k + x\,h \pmod q.$$

Подписью является тройка **(M, h, z).**

Проверка подписи заключается в проверке равенства

$$H(g^z\, y^{-h}, M) = h.$$

В самом деле,

$$g^z\, y^{-h} = g^{k+xh}\, g^{-xh} = g^k.$$

Стандарт ЭЦП DSS

Федеральный стандарт обработки информации **FIPS PUB 186**, известный как **DSS (*Digital Signature Standard*** — стандарт цифровой подписи) основан на *алгоритме хэширования* **SHA (*Secure Hash Algorithm*** — защищенный алгоритм хэширования).

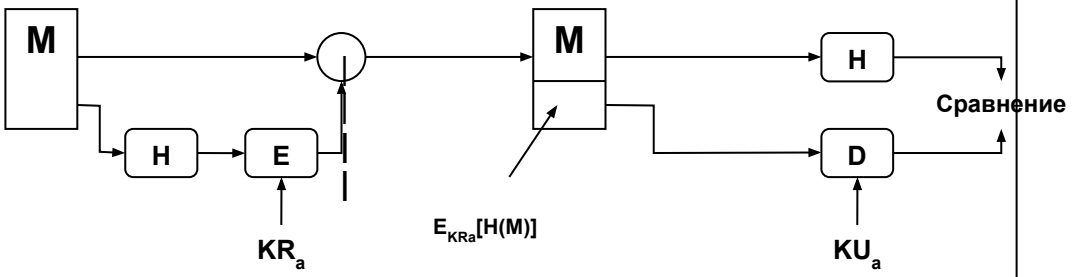
Согласно этому стандарту, электронная цифровая подпись может вырабатываться по одному из трех алгоритмов:

- **DSA (Digital Signature Algorithm) –** алгоритму, основанному на **проблеме логарифма в конечном поле,**
- **ANSI X9.31 (RSA DSA),**
- **ANSI X9.63 (EC DSA)** – алгоритму выработки подписи, основанному на **проблеме логарифма в группе точек эллиптической кривой над конечным полем.**

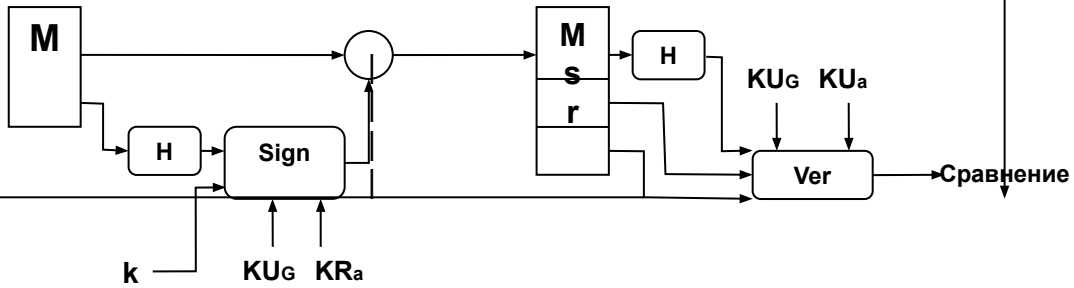
В **России**
вычисление дайджеста и реализацию электронной подписи
регламентируют два стандарта:
– "*Процедуры выработки и проверки электронной цифровой подписи на базе асимметричного криптографического алгоритма*" и
- "*Функция хэширования*",
объединенные общим заголовком
"Информационная технология. Криптографическая защита информации".

В сентябре 2001 г. утвержден, а с 1 июля 2002 г. вступил в силу, новый стандарт электронной цифровой подписи –
ГОСТ Р 34.10–2001.

Два подхода к использованию цифровых подписей



(а) Подход
RSA



(б) Подход
DSS

Алгоритм цифровой подписи DSA

Глобальные компоненты открытого ключа

p — простое число, $2^{L-1} < p < 2^{L}$, где $512 < L < 1024$ и L является ***кратным 64***, т.е. длиной *между 512 и 1024 битами с шагом 64 бита;*

q — простой делитель $(p - 1)$, где $2^{159} < q < 2^{160}$, т.е. *длиной 160 битов;*

$g = h^{(p-1)/q} \pmod p$,

где **h** является любым целым числом таким, что $1 < h < (p - 1)$ и $h^{(p-1)/q} \pmod p > 1$.

Личный ключ пользователя

x – случайное или псевдослучайное число, $0 < x < q$

Открытый ключ пользователя

$y = (g^{x}) \bmod p$

Секретный номер сообщения пользователя

k – случайное или псевдослучайное число, $0 < k < q$

Создание подписи

$r = (g^{k} \bmod p) \bmod q$

$s = [k^{-1} (H(M) + xr)] \bmod q$

Подпись = (r, s)

Верификация

$w = (s')^{-1} \bmod q$

$u1 = H(M') w \bmod q$

$u2 = (r')w \bmod q$

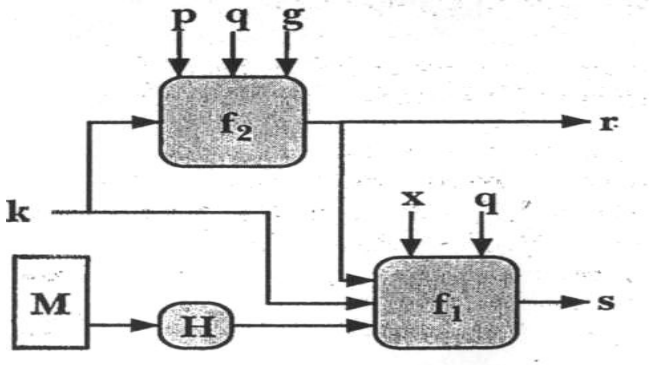
$v = [(g^{u1}) (y^{u2}) \bmod p] \bmod q$

ПРОВЕРКА: $v = r'$

M – подписываемое сообщение,

H(M) — хэш-код **M** по методу ***SHA-1,***

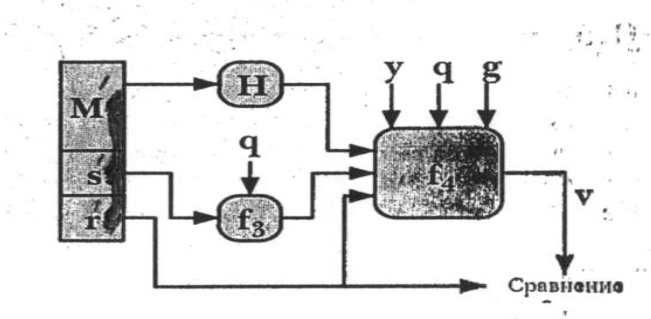
M', r', *s'* — полученные версии **M, r** и **s.**



$s = f_1(H(M), k, x, r, q) = (k^{-1}(H(M) + xr)) \bmod q$

$r = f_2(k, p, q, g) = (g^k \bmod p) \bmod q$

(а) Создание подписи



$w = f_3(s9, q) = (s')^{-1} \bmod q$

$v = f_4(y, q, g, H(M'), w, r') =$

$= ((g(H(M')w) \bmod q\, y^{r'} w \bmod q \bmod p) \bmod q$

(б) Верификация

Подпись и верификация
DSS

Проверка осуществляется со значением ***r***, которое не зависит от сообщения вообще:
r является функцией ***k*** и *трех* компонентов **глобального открытого ключа**.

Уравнение проверки для сообщения **m** имеет вид:

$$r \equiv (g^{H(m)*s^{-1}} \times y^{r*s^{-1}} \bmod p) \pmod q.$$

Действительно,

$$(g^{H(m)*s^{-1}} \times y^{r*s^{-1}} \bmod p) \pmod q = (g^{H(m)*s^{-1}} \times g^{x*r*s^{-1}} \bmod p) \pmod q =$$
$$= (g^{(H(m)+x*r)*s^{-1}} \bmod p) \pmod q =$$
$$= (g^{k^{-1}*(H(m)+x*r)^{-1}*(H(m)+x*r)} \bmod p) \pmod q =$$
$$= (g^{(k^{-1})^{-1}*(H(m)+x*r)^{-1}*(H(m)+x*r)} \bmod p) \pmod q =$$
$$= (g^{k} \bmod p) \pmod q \equiv r.$$

Интенсивные вычисления потребуются только при вычислении
$(g^k) \bmod p$.
Поскольку это значение *не зависит от подписываемого сообщения*, оказывается возможным вычислить значение заранее.