Лекция 8. Электронная подпись I

1 Электронная подпись с закрытым ключом

Схема электронной подписи с закрытым ключом:

- K — генератор закрытого ключа d.
- $S(m, d) = s$ — подпись.
- $V(m, d, s) \in \{0, 1\}$ — верификатор.
- Корректность: $P(V(m, d, S(m, d)) = 1) \approx 1$.
- Атака происходит следующим образом: схема C несколько раз может сгенерировать сообщение x_i и получить подпись для него. После этого схема D пытается сгенерировать новое слово x, не встречавшееся среди запросов и подписать его так, чтобы верификатор принял подпись.

  $m_1 = C(), s_1 = S(m_1, d),$
  $m_2 = C(s_2), s_2 = S(m_2, d),$
  $\dots,$
  $m_k = C(s_1, \dots, s_{k-1}), s_k = S(m_k, d),$
  $m_{k+1} = C(s_1, \dots, s_k), s_{k+1} = D(s_1, \dots, s_k).$

  Условие неподделываемости: $P(m_{k+1} \notin \{m_1, \dots, m_k\}, V(m_{k+1}, d, s_{k+1}) = 1) \approx 0$.

Если существует протокол подписи, то его можно использовать для задачи идентификации: сервер посылает клиенту случайное сообщение m на подпись и проверяет подпись под ним. Несложно увидеть, что атака на такой протокол это и есть атака на электронную подпись.

С закрытым ключом алгоритм довольно простой: пусть d — идентификатор псевдослучайной функции. Положим $S(m, d) = f_d(m), V(m, d, s) = I(s = f_d(m))$. Взломщик в таком случае получает значение псевдослучайной функции на полиномиальном числе входов. Однако семейство псевдослучайных функций вычислительно неотличимо от семейства всех функции, поэтому подписывая значение m_{k+1} он угадает с вероятностью, близкой к вероятности угадать значение случайной функции, которое не зависит от значений в предыдущих точках, то есть с экспоненциально малой вероятностью.

2 Электронная подпись с открытым ключом

- K — генератор закрытого и открытого ключей d, e.
- $S(m, d) = s$ — подпись.
- $V(m, e, s) \in \{0, 1\}$ — верификатор.

- Корректность: $P(V(m, e, S(m, d)) = 1) \approx 1$.
- Атака происходит таким же образом, однако схемы C, D дополнительно получают аргумент e.

  Условие неподделываемости: $P(m_{k+1} \notin \{m_1, \ldots, m_k\}, V(m_{k+1}, e, s_{k+1}) = 1) \approx 0$.

Подпись одного бита можно сделать на базе любой односторонней функции f. $d = (x_0, x_1), e = (f(x_0), f(x_1))$ (нужно выбрать так, чтобы $f(x_0), \neq f(x_1)$).

$m_1 = \sigma = C(y_0, y_1), s_1 = x_\sigma, m_2 = 1 - \sigma, s_2 = D(y_0, y_1, x_\sigma)$. Нужно показать, что $P(f(s_2) = y_{1-\sigma}) \approx 0$.

Пусть существует алгоритм, взламывающий такую подпись, тогда построим алгоритм, обращающий одностороннюю функцию. Обратитель f будет действовать так:

- Получает y.
- Выбирает случайный x и случайный бит τ.
- На всякий случай проверяет $f(x) \neq y$ (если равно, то обратил).
- $y' = f(x'), y_\tau = y', y_{1-\tau} = y$.
- $C(y_0, y_1) = \tau \Rightarrow$ можно вернуть $D(y_0, y_1, x')$.
- В противном случае вернуть что угодно.

Так как вероятность последнего условия $\frac{1}{2}$, то вероятность обращения не меньше $\frac{1}{2}$ вероятности подделки подписи.

Построим теперь подпись одного сообщения фиксированной длины. Для этого переделаем схему подписи одного бита (K, S, V) в $(\hat{K}, \hat{S}, \hat{V})$, так что $\hat{K}$ запускает k раз алгоритм K, генерируя $(d_1, e_1), \ldots, (d_k, e_k)$ то есть $d = (d_1, \ldots, d_k), e = (e_1, \ldots, e_k)$.

$\hat{S}(\sigma_1, \ldots, \sigma_k) = S(\sigma_1, d_1) \ldots S(\sigma_k, d_k)$. Верификатор устроен очевидным образом.

Такая подпись работает только для фиксированной длины (нельзя даже брать слова длины $\leqslant k$, так как после любого сообщения можно корректно подписать любой его префикс), притом подпись одноразовая, поскольку подпись сообщений $0 \ldots 0$ и $1 \ldots 1$ позволяет подделать любую подпись.

Часть проблем решает беспрефиксным кодированием: например удваиваем каждый бит и дописываем в конце 01. Так, вместо длины ровно k мы сможем подписывать сообщения длины $\leqslant k$.

3 Снятие ограничений по длине и одноразовости

Идея: подпись под хеш-значением.

Определение 1. Семейство хеш-функций с трудно обнаружимыми коллизиями — $h : \{0,1\}^{p(n)} \times \{0,1\}^* \to \{0,1\}^{q(n)}$, такая что для схемы, ищущей коллизии $C(s) = \{x_1, x_2\}, x_1 \neq x_2, h_s(x_1) = h_s(x_2)$ верятность успеха близка к 0 при слуайно выбранном s.

Если такое семейство существует, то достаточно взять хеш-значение сообщения и подписать только его. Атакующий должен будет сделать одну из двух вещей — либо найти коллизию хеш-функции, либо использовать соощение с другим хешом, но подделать его подпись. В обеих ситуациях вероятность успеха мала.

Идея для многоразовой подписи: S, V интерактивны и хранят всю историю, взломщик прослушивает канал и может потом попытаться подделать подпись. S вместе с подписью присылает новый открытый ключ для следующего раунда, притом подпись распространяется и на него.

Идею интерактивности можно использовать для подписи слова произвольной длины. $m = \sigma_1 \dots \sigma_k$, (d_0, e_0) ключ для однократной подписи k-битов $e = (e_0, l)$. Генерируются $(d_1, e_1), s_1$ — подпись под $h_l(\sigma_1, e_1)$ при помощи d, $(d_2, e_2), s_2$ — подпись под $h_l(\sigma_2, e_2)$ при помощи d_1 и так далее.

Определение 2. Универсальное семейство односторонних хеш-функций — семейство, такое что трудно обнаружить коллизию для заранее заданного x.

Общая идея для многократной электронной подписи без внутренней памяти: вместо следующего ключа подписываются 2 следующих ключа: какой из них используется зависит от битов сообщения.