Лекция 10

Электронные подписи (продолжение)

(Конспект: В. Николаенко)

10.1 Протокол подписи сообщений произвольной длины на основе односторонней функции

В прошлой лекции мы построили одноразовый протокол подписи сообщений произвольной длины на основе CTOK (семейства хэш-функций с трудно обнаруживаемыми коллизиями). CTOK является самостоятельным криптографическим примитивом, неизвестно, следует ли его существование из существования односторонних функций. В этот раз мы рассмотрим другой примитив — универсальное семейство односторонних хэш-функций.

Рассмотрим полиномиально вычислимое отображение: $h : \{0,1\}^q(n) \times \{0,1\}^p(n) \to \{0,1\}^n$. При каждом фиксированном s из $\{0,1\}^q$ получаем функцию $x \mapsto h(s,x)$, которую будем обозначать h_s. Пусть теперь $q = q(n)$ и $p = p(n)$ — некоторые полиномы от n. Тогда h называется *универсальным семейством односторонних хэш-функций*, если выполнены следующие условия: для любой последовательности строк $\{x_n\}$ такой, что $|x_n| = p(n)$, для любого семейства вероятностных полиномиальных схем C_n вероятность того, что $h_s(x_n) = h_s(C_n(s))$ — пренебрежимо мала. То есть любой схемный противник будет находить коллизии с маленькой вероятностью. Важно, что первый элемент коллизии выбирается заранее, в случае CTOK было сложно найти сразу два эллемента, на которых случается коллизия.

Теорема 10.1. *Если существуют односторонние функции, то для любого полинома $p(n)$ существует полином $q(n)$ такой что существует универсальное семейство односторонних хэш функций $h : \{0,1\}^q(n) \times \{0,1\}^p(n) \to \{0,1\}^n$.*

Доказательство. Доказательство этой теоремы достаточно трудное, его можно найти в статье J. Rompel «One-way functions are necessary and sufficient for secure signatures». □

Теорема 10.2. *Если существуют односторонние функции и существует протокол подписи сообщения фиксированной длины, то существует протокол подписи одного сообщения произвольной полиномиальной длины.*

Доказательство. Напомним, что противнику разрешается дать на подпись одно сообщение, посмотреть на эту подпись, а затем подделать подпись под другим сообщением. (Так бывает и в обычной жизни, для того, чтобы подделывать подпись, надо хотя бы раз ее увидеть).

В дальнейших конструкциях будет удобно считать, что ни для каких двух сообщений, которые подписываются, ни одно не является префиксом другого. Так можно считать, поскольку можно подписывать не само сообщение, а его префиксный код: $\widehat{x} = x_1x_1x_2x_2...x_mx_m01$.

Разобъем сообщение на биты: $x = \sigma_1\sigma_2...\sigma_m$

Согласно условию у нас имеется алгоритм подписи сообщений фиксированной длины, будем считать, что это длина — $n+1$.

Воспользуемся генератором ключей этого алгоритма и сгенерируем m пар открытых и закрытых ключей: $<e_2, d_2> \ldots <e_{m+1}, d_{m+1}>$. Считаем, что первая пара $<e_1, d_1>$ - дана (e_1 - это опубликованный открытый ключ, а d_1 - соответствующий ему закрытый ключ).

Пусть $p(n)$ — это длина ключей e_i, пусть h $h : \{0,1\}^q(n) \times \{0,1\}^p(n) \to \{0,1\}^n$ — универсальное семейство односторонних хэш-функций.

Имеющимся алгоритмом S подписи сообщений длины $n+1$ подпишем следующие сообщения:

$$\begin{aligned} S(h_s(e_2)\sigma_1, d_1) &= s_1 \\ S(h_s(e_3)\sigma_2, d_2) &= s_2 \\ &\ldots \\ S(h_s(e_{m+1})\sigma_m, d_m) &= s_m \end{aligned}$$

Секретным ключом будет пара d_1 и s, открытым ключом — e_1, а подпись сообщения $x = \sigma_1\sigma_2...\sigma_m$ — это $s_1, s_2, ..., s_m, e_2, e_3, ..., e_{m+1}$

Для проверки подписи достаточно проверить, что при всех i от 1 до m выполняется $V(h_s(e_{i+1}\sigma_i), s_i, e_i) = 1$.

Если подпись была сгенерирована описанным выше алгоритмом, V ее примет.

Теперь докажем, что такой протокол будет сложно взломать вероятностными полиномиальными схемами. Как было замечено, у взломщика должно быть право подписать какое-нибудь сообщение, значит у него есть полиномиальная схема $C(e_1) = x = \sigma_1\sigma_2...\sigma_m$, которая генерирует сообщение на подпись. Пусть подпись x, это $s(x) = s_1, s_2, ..., s_m, e_2, e_3, ..., e_{m+1}$. Взломщик имеет и другую полиномиальную схему, $E(x, e_1, s(x)) = x' \neq x$, пусть $|x| = l$. Эта схема генерирует поддельное сообщение. И, наконец, у взломщика есть схема, которая подделывает подпись под x': $D(e_1, s(x), x') = s'_1, s'_2, ..., s'_m, e'_2, e'_3, ..., e'_{m+1}$.

Атаку считаем успешной, если подпись принята с вероятностью по крайней мере $\varepsilon = 1/poly(n)$.

Утверждение 10.1. *Если атака успешна, то существует такое* $j \leq \min\{m, l\}$ *такое что либо (1)* $e_j \neq e'_j$ *и* $h_s(e_j) = h_s(e'_j)$*, либо (2)* $e_j = e'_j$*, но* $\sigma_j h_s(e_{j+1}) \neq \sigma'_j h_s(e'_{j+1})$

Доказательство. Иными словами утверждается, что у противника есть только две возможности подделать подпись в новой системе: отыскать такой e'_j отличное от e_j с тем же хэш значением, что и у e_j. Вторая возможность — правильно подделать подпись в старой системе под сообщением $\sigma'_j h_s(e'_{j+1})$, отличным от сообщения $\sigma_j h_s(e_{j+1})$, подписанного владельцем секретного ключа.

Пусть это не так.

Так как $e_1 = e'_1$ (открытый ключ одинаков) и поскольку первое предположение не верно, то $\sigma_1 h_s(e_2) = \sigma'_1 h_s(e'_2)$, поскольку не верно второе, то $e_2 = e'_2 \Rightarrow \sigma_2 h_s(e_3) = \sigma'_2 h_s(e'_3) \Rightarrow e_3 = e'_3$ и т.д. Но такая цепочка обязана оборваться, иначе одно из сообщений $\sigma_1\sigma_2...\sigma_m$, $\sigma'_1\sigma'_2...\sigma'_l$ будет префиксом другого. А мы работаем в предположении обратного. □

Таким образом, если взломщик работает успешно, то есть взламывает с вероятностью ε, то с вероятностью $\frac{\varepsilon}{2}$ случится либо первое, либо второе событие из утверждения.

Пусть с вероятностью не меньше $\frac{\varepsilon}{2}$ случилось (1), но тогда мы успешно атаковали семейство односторонних хэш-функций. Действительно, сгенерируем случайно e_1, d_1 и s, дадим противнику открытый ключ — e_1. Он попросит нас подписать какое-то сообщение x, мы подписываем его, генерируя нужное количество пар ключей. Когда же он выдаст фальшивую подпись s', возьмем в ней j-ый ключ e'_j, с вероятностью $\frac{\varepsilon}{2 \cdot poly(n)}$ $h_s(e_j) = h_s(e'_j)$, то есть $<e_j, e'_j>$ будет коллизией. Как нетрудно заметить, описанная атака вычисляется вероятностными схемами полиномиального размера.

Пусть теперь с вероятностью не меньше $\frac{\varepsilon}{2}$ случилось (2), то есть нашлось такое $j \leq min(m, l)$, что $e_j = e'_j$ и $\sigma_j h_s(e_{j+1}) \neq \sigma'_j h_s(e'_{j+1})$. Опять-таки для какого-то конкретного j вероятность этого события не меньше $\frac{\varepsilon}{2 \cdot poly(n)}$. Будем атаковать систему подписи одного сообщения длины $n+1$. Сгенерируем e, d, s. Противник сгенерирует сообщение, которое мы подпишем, при этом запомним $\sigma_j h_s(e_{j+1})$. Противник подпишет фальшивое сообщение, которое будет принято с вероятностью больше ε. Это значит, что $V(h_s(e_{j+1}\sigma_j), s_j, e_j) = V(h_s(e'_{j+1}\sigma'_j), s'_j, e'_j = e_j) = 1$. То есть, сообщение $e'_{j+1}\sigma'_j$ является отличным от того, которое нас просили подписать и подпись s'_j под которым принимается с тем же ключом e_j. Таким образом мы успешно атаковали алгоритм G, S, V подписи сообщения длины $n+1$. □

10.1.1 Протокол подписи произвольного количества сообщений произвольной длины

Протоколом подписи произвольного количества сообщений произвольной длины называется совокупность алгоритмов: генератор ключей $G(1^n) = <e_n, d_n>$, вероятностного подписывающего алгоритма $S(x, d_n, 1^n) = s_n$ и проверяющего детерминированного $V(x, s, e_n) = \{0, 1\}$, причем

1) $V(x, S(x, d_n, 1^n), e_n) = 1$ — проверяющий алгоритм всегда принимает правильную подпись

2) Протокол должен быть устойчив к противнику, который может попросить подпись для полиномиального числа сообщений, после чего подделать подпись под фальшивым сообщением. Говоря формально, для любой полиномиальной вероятностной схемы C,

после подписи $q(n)$ сообщений:

$$\begin{aligned} C(e) = x_1, S(x_1, d) = s_1 \\ C(e, s_1) = x_2, S(x_2, d) = s_2 \\ \dots \\ C(e, s_1, s_2, ..., s_{q(n)-1}) = x_{q(n)} \\ S(x_{q(n)}, d) = s_{q_n} \end{aligned}$$

для любой вероятностной схемы D, которая порождает сообщение, отличное от предыдущих, и фальшивую подпись: $D(e, x_1, s_1, x_2, s_2, ..., x_{q(n)}, s_{q(n)}) = (x_{q(n)+1}, s_{q(n)+1})$, вероятность, что проверяющий алгоритм примет эту подпись пренебрежимо мала.

Протокол, удовлетворяющий этим требованиям, существует в предположениях, о которых гласит следующая теорема:

Теорема. *Если существует одноразовый алгоритм подписи сообщения произвольной длины и существует семейство псевдослучайных функций, то существует и многоразовый алгоритм подписи сообщения произвольной длины.*

Доказательство. Пусть S, V, G — одноразовый протокол подписи.

Пусть G использует $l(n)$ случайных битов. ($l(n)$ — это некоторый полином).

Рассмотрим семейство псевдослучайных функций $f_s : \{0,1\}^* \to \{0,1\}^{l(n)}$.

Пусть $x = \alpha_1\alpha_2...\alpha_m$ — сообщение, которое мы хотим подписать.

Пусть (e, d) — открытый и закрытый ключи одноразового (старого) протокола.

Выберем случайно s — номер псевдослучайной фунции из семейства.

Рассмотрим строку бит: β, длина которой не больше $l(n)$. Под $<e_\beta, d_\beta>$ будем понимать ключи, сгенерированные алгоритмом G, которому вместо случайных битов, подается строка $f_s(\beta)$.

Для пустой строки - Λ, положим $d_\Lambda = d$, $e_\Lambda = e$.

Теперь новая подпись сообщения:

$$\begin{aligned} \tilde{S}(\tilde{d}, s, x) = \\ & (S(d_\Lambda, (e_0, e_1)), e_0, e_1, \\ & S(d_{\alpha_1}, (e_{\alpha_1,0}, e_{\alpha_1,1})), e_{\alpha_1,0}, e_{\alpha_1,1}, \\ & \dots, \\ & S(d_{\alpha_1\alpha_2...\alpha_i}, (e_{\alpha_1\alpha_2...\alpha_i 0}, e_{\alpha_1\alpha_2...\alpha_i 1})), e_{\alpha_1\alpha_2...\alpha_i 0}, e_{\alpha_1\alpha_2...\alpha_i 1}, \\ & \dots, \\ & S(d_{\alpha_1\alpha_2...\alpha_m}, x)) \end{aligned}$$

Открытый ключ — e, закрытый ключ — пара (d, s).

Проверка подписи: $\widetilde{V}(s_1, e_0^1 e_1^1, s_2, e_0^2 e_1^2, ..., s_m, e_0^m e_1^m, s_{m+1}, x, e)$

Выполняем проверку, последовательно проверяя все тройки, начиная с начала:

$$\begin{aligned} V(s_1, e_0^1 e_1^1, e) = 1 \\ V(s_2, e_0^2 e_1^2, e_{\alpha_1}^1) = 1 \end{aligned}$$

мы знаем, чему равно α_1, поскольку знаем сообщение и поверили, что открытые ключи на предыдущем шаге - правильные, значит, мы знаем $e^1_{\alpha_1}$ и т.д..

Покажем, что этот протокол сложно взломать. Пусть есть взломщик - это два семейства вероятностных схем: C и D. Схемы C генерируют на подпись полиномиальное число сообщений (пусть их будет k), а схемы D генерируют новое сообщение — x_{k+1} и фальшивую подпись, причем эти сообщение с подписью принимаются проверяющим алгоритмом с вероятностью по крайней мере $\varepsilon = \frac{1}{poly}$ для бесконечного числа длин входов.

Заметим, что если мы при подписывании всегда будем использовать не псевдослучайную функцию, а случайную, но общую для всех сообщений, то противник все равно будет взламывать нашу подпись, так как иначе с помощью противника можно было бы построить взломщика для семейства псевдослучайных функций. Поясним, что имеется в виду под тем, что мы будем использовать случайную функцию: вместо запросов к псевдослучайной функции можно генерировать нужную строчку случайным образом, но при этом все запросы нужно сохранять, чтобы в следующий раз на тот же запрос выдать тот же самый ответ.

Посмотрим на поддельную подпись, она должна разбиваться на те же блоки — иначе мы могли бы отличить ее от настоящей чисто синтаксически. Поддельная подпись: $\widehat{s}_1\widehat{e}^1_0\widehat{e}^1_1...\widehat{s}_m\widehat{e}^m_0\widehat{e}^m_1\widehat{s}_{m+1}$.

Будем сравнивать настоящую и поддельную подписи фальшивого сообщения. Пусть j первое место, в котором открытые ключи различаются. Если такого места нет, положим $j = m + 1$.

Поскольку подпись полиномиальной длины, различных j тоже полиномиальное число, значит с вероятностью $\frac{1}{poly}$ мы угадаем нужное j.

Если атака успешна, то $\widehat{s}_{j-1}$ является правильной подписью под фальшивым сообщением. То есть $V(e_{\alpha_1...\alpha_{j-1}}, \widehat{s}_{j-1}, \widehat{e}^i_0\widehat{e}^i_1) = 1$, если $1 \le j \le m$, и $V(e_{\alpha_1...\alpha_{i-1}}, \widehat{s}_m, x_{k+1}) = 1$, если $j = m + 1$.

Будем атаковать старую схему подписи на паре ключей (e, d) (e мы знаем, а d — нет). Рассмотрим сначала простой случай. Пусть схема C выдает в начале кроме сообщения x_1, последовательность фальшивого сообщения: $\alpha_1...\alpha_{j-1}$. (j угадываем, как выше).

Если последовательность $\alpha_1...\alpha_{j-1}$ не является префиксом x_1, мы генерируем ключи e_β, d_β (используя для этого настоящие случайные биты, а не псевдослучайные) для всех префиксов x_1 и запоминаем их (таким образом ключи для одинаковых последовательностей генерируем по одному разу). Если же последовательность $\alpha_1...\alpha_{j-1}$ является префиксом x_1, то делаем то же самое, только вместо ключей $e_{\alpha_1...\alpha_{j-1}}, d_{\alpha_1...\alpha_{j-1}}$ берем ключи e, d. Причем, подпись под $e_{\alpha_1...\alpha_{j-1}0}e_{\alpha_1...\alpha_{j-1}1}$ спрашиваем у старой схемы одноразовой подписи (у владелицы закрытого ключа — d). (Мы можем один раз попросить подписать сообщение). Затем проделываем то же самое для остальных сообщений, которые генерирует схема — $x_2, ..., x_k$.

Если последовательность $\alpha_1...\alpha_{j-1}$ оказалась префиксом двух различных сообщений x_{i_1}, x_{i_2}, то мы не просим вторично подпись у владелицы ключа d, а используем первую (в обоих случаях нам нужно подписать одно и тоже сообщение).

Если атака успешна, вероятность того, что $V(e, \widehat{s}_{j-1}, \widehat{e}^j_0\widehat{e}^j_1) = 1$, если $1 \le j \le m$, и $V(e, \widehat{s}_m, x_{k+1}) = 1$, если $j = m+1$ не меньше ε. Таким образом мы сломали алгоритм од-

норазовой подписи, правильно подписав фальшивое сообщение $(\widehat{e}_0^j \widehat{e}_1^j)$, после получения подписи под другим $(e_{\alpha_1 \ldots \alpha_{j-1} 0} e_{\alpha_1 \ldots \alpha_{j-1} 1})$.

Теперь, пусть схема не сообщает $\alpha_1 ... \alpha_{j-1}$, тогда мы можем вместе с j угадывать первый номер $1 \leq i \leq k+1$ такой, что $\alpha_1 ... \alpha_{j-1}$ является префиксом x_i. И соответственно, вместо $\alpha_1 ... \alpha_{j-1}$ будем использовать префикс x_i длины $j-1$. Так как i и j ограничены полиномом, вероятность успешной атаки останется $\frac{1}{poly}$.

□