Лекция 2

Надежность криптосистемы

(Конспект: С. Николенко)

2.1 Семантическая надежность vs. неразличимость

Рассмотрим произвольную криптосистему (G, E, D). Мы хотели бы дать формальное определение того, что эта криптосистема является надежной.

Определение 2.1. Криптосистема (G, E, D) называется *семантически надежной*, если для любой функции h, для всякого (задаваемого полиномиальным по времени вероятностным алгоритмом M) вероятностного распределения на входах этой функции, для всякого работающего полиномиальное время "противника" A и для любого многочлена Q существует полиномиальный алгоритм $\tilde{A}$, такой, что

$$\mathbf{P}\{A(E_e(x), e, k) = h(x)\} < \mathbf{P}\{\tilde{A}(e, k) = h(x)\} + \frac{1}{Q(k)},$$

где e – публичный ключ криптосистемы, k – длина этого ключа, а $E_e(x)$ – результат работы криптосистемы на входе x, т.е. закодированное сообщение.

Замечание 2.1. Здесь и в дальнейшем вероятность берется "по всему": по случайным выборам алгоритмов, по случайным входам x (порожденным алгоритмом M), по случайным ключам (взятым с распределением, соответствующим генератору).

Иными словами, система семантически надежна, если никакой полиномиальный алгоритм не угадает никакую функцию от входа с большей вероятностью, чем это сделал бы алгоритм, являющийся просто случайным распределением на строках (т.е., на вход которого не поступает закодированное сообщение, а только общедоступная информация).

Это определение кажется довольно сильным. Попробуем дать другое определение, послабее, а затем докажем их эквивалентность.

Определение 2.2. Криптосистема (G, E, D) называется *неразличимой*, если для всяких полиномиальных "противников" M и A и для всякого многочлена Q

$$\mathbf{P}\{A(E_e(m_i), m_0, m_1, e, k) = i)\} < \frac{1}{2} + \frac{1}{Q(k)},$$

где $(m_0, m_1) = M(1^k)$.

Иными словами, машина M производит два сообщения m_0 и m_1, криптосистема их кодирует, и после этого никакой полиномиальный алгоритм не сможет их различить, то есть по коду сказать, m_0 это или m_1.

Теорема 2.1. *Семантическая надежность эквивалентна неразличимости. Т.е. всякая семантически надежная криптосистема неразличима, и наоборот.*

Замечание 2.2. Мы докажем эту теорему для схемной сложности (противники могут быть произвольными булевыми схемами, т.е. иметь полиномиальную подсказку, зависящую только от длины входа. Это несколько проще.

Доказательство. Докажем сначала легкую часть теоремы. Пусть есть семантическая надежность, но нет неразличимости, т.е. существуют M и A такие, что

$$\mathbf{P}\{A(\ldots, m_0, m_1, E(m_i), \ldots) = i\} \geq \frac{1}{2} + \varepsilon$$

(здесь и далее под ε понимается величина, обратная к некоторому многочлену). Тогда определим $h(m_i) = i$ на той достаточно хорошей паре (m_0, m_1), на которой выполняется оценка из верхнего неравенства. Возьмем в качестве $\tilde{A}$ алгоритм, моделирующий работу M, то есть производящий то m_0, то m_1 с соответствующей вероятностью (поскольку на вход M подается лишь длина ключа, мы можем смоделировать ее алгоритмом $\tilde{A}$). Получится противоречие с определением семантической надежности. Итак, из семантической надежности следует неразличимость.

Обратно, пусть есть функция h, которую противник умеет достаточно хорошо угадывать. Докажем существование различимых сообщений $\tilde{x}$

и $\tilde{y}$. Наш различающий алгоритм будет работать следующим образом: он будет выдавать $\tilde{x}$, если A выдаст $h(\tilde{x})$, $\tilde{y}$, если A выдаст $h(\tilde{y})$, и с вероятностью $\frac{1}{2}$ что-нибудь из них, если выдано что-то иное либо если $h(\tilde{x}) = h(\tilde{y})$. Машина же M будет равновероятно порождать то $\tilde{x}$, то $\tilde{y}$. Обозначим для краткости $p_k(x) = \mathbf{P}\{A(x) = k\}$, $h(\tilde{x}) = m$, $h(\tilde{y}) = n$. Тогда

$$
\begin{aligned}
&\mathbf{P}\{\text{успеха}\} = \\
&\mathbf{P}\{\text{дали } \tilde{x}\}(p_m(\tilde{x}) + \frac{1}{2}(1 - p_m(\tilde{x}) - p_n(\tilde{x}))) + \\
&\mathbf{P}\{\text{дали } \tilde{y}\}(p_n(\tilde{y}) + \frac{1}{2}(1 - p_m(\tilde{y}) - p_n(\tilde{y}))) = \\
&\qquad\qquad \frac{1}{4}(p_m(\tilde{x}) + p_n(\tilde{y}) - p_m(\tilde{y}) - p_n(\tilde{x}) + 2).
\end{aligned}
$$

Чтобы она была больше $\frac{1}{2} + \varepsilon$, нужно чтобы

$$p_m(\tilde{x}) - p_m(\tilde{y}) + p_n(\tilde{y}) - p_n(\tilde{x}) > \varepsilon$$

(для другого ε). Предположим, что таких $\tilde{x}$ и $\tilde{y}$ не существует:

$$\sum_{x,y} (p_{h(x)}(x) + p_{h(y)}(y) - p_{h(x)}(y) - p_{h(y)}(x)) p(x) p(y) < \varepsilon,$$

где $p(z)$ — вероятность элемента z согласно распределению на входах (из определения семантической надежности). Если теперь раскрыть скобки, в первых двух слагаемых участвует только одна из переменных суммирования, и от суммы по другой (равной 1) можно избавиться. Две другие суммы также равны, и, очередной раз меняя ε, имеем

$$\sum_x p_{h(x)}(x) p(x) - \sum_{x,y} p(x) p(y) p_{h(y)}(x) < \varepsilon.$$

Обозначим событие $H_k = \{x : h(x) = k\}$ и его вероятность $\chi_k = \mathbf{P}(H_k)$; тогда

$$
\begin{aligned}
&\sum_k \sum_{x \in H_k} p(x) p_k(x) - \sum_m \sum_{y \in H_m} \sum_x p(x) p(y) p_m(x) = \\
&\sum_k \sum_{x \in H_k} p(x) p_k(x) - \sum_m \left(\sum_{y \in H_m} p(y) \cdot \sum_x p(x) p_m(x) \right) = \\
&\qquad\qquad \sum_k \sum_{x \in H_k} p(x) p_k(x) - \sum_m \chi_m \sum_x p(x) p_m(x) < \epsilon.
\end{aligned}
$$

Это уже победа, так как первая из полученных сумм – это $\mathbf{P}\{A(E(x)) = h(x)\}$, а вторая – это тот самый алгоритм $\tilde{A}$, который в данном случае будет выбирать m с вероятностью $\sum_x p(x)p_m(x)$ (совершенно независимо от входа, которого у него, впрочем, и нет). Итого получилось противоречие с семантической ненадежностью данной криптосистемы; значит, нужные $\tilde{x}$, $\tilde{y}$ существуют. Отметим напоследок, что это доказательство не дает ни малейшей идеи о том, как именно нужно будет строить $\tilde{x}$, $\tilde{y}$ – это чистое доказательство существования. $\square$