Лекция 4

Криптосистемы с открытым ключом, кодирующие слова произвольной длины

(Конспект: К. Ушаков)

4.1 Семантическая надежность и неразличимость

В прошлой лекции мы научились кодировать ровно один бит. Теперь нам хочется закодировать более одного бита, ибо мы породили большой ключ, а с его помощью закодировали всего один бит, что обидно. К сожалению, когда мы давали определение криптосистемы, мы определяли e и d как схемы, которые построены по параметру надежности, а про длину кодового слова ничего не знают. Так что если мы хотим получить криптосистему, которая кодирует слова произвольной длины, то мы должны добавить к ней два полиномиальных по времени детерминированных (т.е. случайная строка будет для них одним из параметров) алгоритма E и D, которые будут заниматься тем, что применять e и d к сообщениям произвольной длины:

$$E(m, e, r_e),$$

$$D(d, r_d).$$

(Старое определение для однобитовых сообщений получится, если положить $E(m, e, r) = e(m, r)$ и $D(c, d, r) = d(c, r)$.)

Замечание 4.1. Можно было бы поступить иначе: можно было бы генератор заставить получать на вход security parameter и длину сообщения, тогда генератор выдавал бы схемы не для кодирования одного бита, а для кодирования сообщения определенной длины. Это было бы красиво, но неудобно.

Для криптосистемы, кодирующей один бит, было понятное определение того, что значит, что ее взламывают (отличают 0 от 1 с хорошей вероятностью). Для криптосистем, которые кодируют слова произвольной длины, имеется два определения: semantic security (на вид — более сильное) и indistinguishability (с ним проще работать).

Определение 4.1 (Semantic Security). Криптосистема называется *семантически надежной*, если $\forall h\ \forall f\ \forall C_S\ \forall p\ \exists \widetilde{C_S}\ \forall M_S$

$$\Pr\{C_S(E(m,e,r_e),e,f(m))=h(m)\} \le \Pr\{\widetilde{C_S}(e,f(m))=h(m)\}+\frac{1}{p(n)},$$

где f и h — полиномиально вычислимые функции, M_S, C_S и $\widetilde{C_S}$ — некоторые «противники» (заданные схемами полиномиального размера), а p — многочлен. Здесь все желающие (включая f и h) получают на вход также 1^n и $1^{|m|}$, но мы этого не пишем, чтобы не загромождать обозначения. Сообщения генерируются как $M_S(1^n)$. Вероятность берется по r_g, r_e и M_S.

Определение 4.2 (Indistinguishability). Криптосистема называется *неразличимой*, если для любого генератора пар сообщений M_I и любого «противника» C_I (заданных схемами полиномиального размера), а также для любого многочлена p

$$\Pr\{C_I(E(e,m_i,r_e),e,1^n,m_0,m_1)=i\} < \frac{1}{p(n)}+\frac{1}{2},$$

где (m_0,m_1) генерируется как $M_I(1^n)$. Вероятность берется по r_g, r_e, $i \in \{0,1\}$ и случайным битам M_I.

Теорема 4.1. *Определения семантической надежности и неразличимости равносильны.*

Доказательство. Сложная сторона $\Uparrow$.

Идея: пусть есть функция, которую мы умеем угадывать (от противного), тогда построим два такие сообщения m_0 и m_1, что $h(m_0) \ne h(m_1)$ и C_I, использующий наш вычислитель функции h, с хорошей вероятностью различает эти сообщения (M_I ровно эту пару и будет порождать).

Пусть у нас есть C_S, такой что $C_S(...,m) = h(m)$ с хорошей вероятностью. Сначала определим M_I: он всегда генерирует пару сообщений (m_0, m_1) (существование подходящих нам m_0 и m_1 мы докажем позже). Различающий же C_I будет работать следующим образом: выдавать 0, если $C_S(\ldots, f(m_0), m) = h(m_0)$ (обратите внимание, что здесь C_S опять получает на вход $f(m_0)$, а не $f(m)$, это позволяет зашить $f(m_0)$ в схему C_I раз и навсегда для данного n); и равновероятно 0 или 1, если $C_S(m)$ выдал что-то другое. Обозначим $p_k(x) = \Pr\{C_S(\ldots, f(x), x) = k\}$, $q_k(x) = \Pr\{C_S(\ldots, f(m_0), x) = k\}$ и $h_i = h(m_i)$.

$$\Pr\{\text{успеха } C_I\} =$$
$$\Pr\{\text{дали } m_0\}\cdot\left(p_{h_0}(m_0) + \frac{1}{2}(1 - p_{h_0}(m_0))\right) + \Pr\{\text{дали } m_1\}\cdot\left(\frac{1}{2}(1 - q_{h_0}(m_1))\right) =$$
$$\frac{1}{2}(1 + p_{h_0}(m_0) - q_{h_0}(m_1))$$

Докажем, что действительно существуют m_0 и m_1, для которых вероятность успеха будет больше, чем $\frac{1}{\mathrm{poly}(n)} + \frac{1}{2}$.

Предположим противное, т.е. пусть для всех пар (m_0, m_1) вероятность успеха мала:

$$p_{h_0}(m_0) - q_{h_0}(m_1) < \frac{1}{\mathrm{poly}(n)}.$$

Просуммируем по всем возможным m_0 и m_1 (с весами, соответствующими вероятностям сообщений согласно M_S), используя сокращение $p(x) = \Pr\{M_S(1^n) = x\}$:

$$\sum_{m_0,m_1} p(m_0)p(m_1)(p_{h_0}(m_0) - q_{h_0}(m_1)) =$$
$$= \left(\sum_x p(x)p_{h(x)}(x) - \sum_{m_0,m_1} p(m_0)p(m_1)q_{h(m_0)}(m_1)\right).$$

Перепишем второе слагаемое: пусть $H_k = \{x | h(x) = k\}$, тогда второе слагаемое равно

$$\sum_k \sum_{m_0 \in H_k} \sum_{m_1} p(m_0)p(m_1)q_k(m_1) = \ldots$$

(из под суммы по m_1 можно вынести $p(m_0)$, а из под суммы по m_0 можно вынести сумму по m_1)

$$\ldots = \sum_k \left(\left(\sum_{m_1} p(m_1)q_k(m_1)\right)\sum_{m_0 \in H_k} p(m_0)\right) = \sum_k \left(\mu_k \left(\sum_{m_1} p(m_1)q_k(m_1)\right)\right),$$

где $\mu_k = \Pr\{M_S(1^n) \in H_k\}$. Таким образом, после преобразования второго слагаемого наша разность принимает вид

$$\sum_x p(x)p_{h(x)}(x) - \sum_k \left(\mu_k \left(\sum_{m_1} p(m_1)q_k(m_1) \right) \right)$$

Первое слагаемое — вероятность того, что старый взломщик правильно угадывает h. А второе слагаемое — вероятность того, что следующий самоуверенный (работающий без зашифрованного сообщения) взломщик угадывает h: он берет случайное сообщение, шифрует его, запускает старого вломщика C_S (используя $f(m_0)$ в аргументе!) и выдает ответ. Разность двух этих вероятностей меньше $\frac{1}{\mathrm{poly}(n)}$, а мы предполагали, что C_S взламывает нашу криптосистему... противоречие!

$\Downarrow$ Легкая сторона.

Пусть есть противник, который умеет различать закодированые слова m_0 и m_1 (вообще говоря, это случайные переменные, порожденные M_I, но мы выберем именно ту пару, на которой вероятность различить максимальна). Мы научимся угадывать функцию h: $h(m_i) = i$, а что в других точках — несущественно, ибо мы будем генерировать только m_i. Есть взломщик C_I, построим по нему C_S. Наш новый взломщик, естественно, будет запускать старого, но ему еще надо дать на вход пару (m_0, m_1). Это не проблема: пару мы зашьем в нового взломщика (благо он — схема). Ломать он будет с вероятностью $\frac{1}{2} + \frac{1}{\mathrm{poly}(n)}$. Но нам надо, чтобы разность его вероятности и вероятности «самоуверенного» взломщика была $\frac{1}{\mathrm{poly}(n)}$. А какова же вероятность взлома для самоуверенного взломщика? На вход он никакой информации об $i \in \{0,1\}$ не получает, т.е. ему надо угадать исход эксперимента подбрасывания симметричной монеты, так что вероятность его успешной работы — $\frac{1}{2}$. $\square$

4.2 Генераторы псевдослучайных чисел

Определение 4.3. $G : \{0,1\}^k \to \{0,1\}^{f(k)}$, где $f(k) > k$, называется *$f(k)$-генератором псевдослучайных чисел* ($f(k)$-*PRG*), если для любого полиномиального по времени вероятностного алгоритма A, для любого многочлена p выполняется

$$|\Pr\{A(G(x)) = 1\} - \Pr\{A(y) = 1\}| < \frac{1}{p(k)},$$

где вероятность берется по случайным числам A и по равномерно распределенным $x \in \{0,1\}^k$ и $y \in \{0,1\}^{f(k)}$.

Существование PRG эквивалентно существованию owf. В следующей лекции мы это (частично) докажем. А воспользуемся уже сейчас следующим вариантом этого утверждения (докажем его на следующей лекции).

Утверждение 4.1. *Если g — односторонняя перестановка (т.е. инъективная owf), сохраняющая длину, B — ее трудный бит, то*

$$G(x) = \left(g^{f(k)-k}(x), B(x), B(g(x)), ..., B(g^{f(k)-k-1}(x))\right)$$

является $f(k)$-генератором.

Построим криптосистему для кодирования более одного бита. Пусть g — кодирующая функция tdpf, B — ее трудный бит. Пусть

$$E(b_1 \dots b_m, g, r) = (g^m(r), B(r) \oplus b_1, B(g(r)) \oplus b_2, \dots),$$

где $b_1 \dots b_m$ — сообщение, r — случайные биты. Заметим, что длина зашифрованного сообщения $O(m+n)$, где n — параметр надежности[1]. Как мы раскодируем? Есть g, значит мы можем узнать r, т.е. мы сможем узнать $B(\dots(r))$, после чего мы возьмем XOR с кодом и получим исходное сообщение.

Почему то, что получилось, — надежная криптосистема? Предположим, что это не так. Вспомним определение неразличимости. Кто-то умеет различать коды двух разных сообщений u и v. Это означает, что одно из них (пусть u) можно отличить от случайного сообщения.

Упражнение 4.1. Вообще-то неразличимость у нас определялась не как разница вероятностей, а как выбор из двух поданных на вход сообщений; как последнюю фразу превратить в строгое доказательство?

Тогда мы сможем отличить выход генератора из утверждения 4.1 от случайных чисел: в первом случае мы, воспользовавшись данным нам выходом генератора, зашифруем u; во втором случае при попытке зашифровать u мы получим код случайного сообщения.

К сожалению, u придётся зашить в эту схему сведения, поэтому данная схема является надёжной (против булевых схем), если исходное tdpf было надёжным *против булевых схем.*

[1] … или $O(m + p(n))$, если tdpf разрешается работать на строках длины, отличной от n.