Лекция 4.

1 Псевдолучайные функции с неадаптивным отличителем

Определение 1. Семейство функций $\{f_s^n\} : f_s^n : \{0,1\}^n \to \{0,1\}^n, s \in \{0,1\}^{p(n)}$ называется псевдослучайным, если:

- Существует полиномиальный алгоритм, который по s и x вычисляет $f_s^n(x)$.
- Надёжность против неадаптивных отличителей:
  $\forall q(\cdot)\ \forall\{D_n\}\forall\{x_1,\ldots,x_{q(n)}\}\,\forall w(\cdot)\ \exists N : \forall n > N \to$
  $|P_s(D_n(x_1,\ldots,x_{q(n)}, f_s(x_1),\ldots,f_s(x_{q(n)})) = 1) -$
  $P_g(D_n(x_1,\ldots,x_{q(n)}, g(x_1),\ldots,g(x_{q(n)})) = 1)| < \frac{1}{w(n)}$

Теорема 1. *Если существует генератор псевдослучайных чисел из* $\{0,1\}^n$ *в* $\{0,1\}^{2n}$*, то существует и семейство псевдослучайных функций.*

Доказательство. Конструкция такова: для некотрого x длины n делаем следующее:

- Считаем $G(s) = s_0 s_1$, если 1й бит x равен 1, то берем s_1, иначе s_0.
- Считаем $G(s_{x_1}) = s_{x_1 0} s_{x_1 1}$, выбираем одну из половин в зависимости от x_2.
- Продолжаем аналогично.

Доказательство индукцией по дереву: нарисуем бинарное дерево, которое является частью полного бинарного, содержащей $x_1,\ldots,x_{q(n)}$. На каждом следующем уровне мы имеем $s_{a_1},\ldots,s_{a_r}$, однако в силу псевдослучайности мы можем вычислительно неотличимо заменить их на действительно случайные значения. Поскольку размер дерева полиномиален, то мы использовали вычислительную неотличимость полиномиально много раз, что делать можно.

Более формально: если $G(y) = G_0(y)G_1(y)$, то можно записать $f_s(x) = G_{x_n}(G_{x_{n-1}}(\ldots Gx1(s)\ldots))$.
$h_{i,t}(x) = G_{x_n}(G_{x_{n-1}}(\ldots Gx_{i+1}(t_{x_1\ldots x_i})\ldots))$, $|t| = 2^{n+i}$.
$h_{0,t}(x) = f_t(x)$, $h_{n,t}(x)$ — случайная функция. Цепочка эквивалентностей приводит к тому, что они вычислительно неотличимы. □

2 Адаптивные отличители

Пример 1. Пример, когда адапитивный отличитель сильнее неадаптивного: пусть есть f, такая что:
$f(0\ldots0) = v$ — случайное, $f(v) = 0\ldots0$, все остальные слова случайны.

Адаптивный отличитель легко справится с такой задачей, а для неадаптивного отличителя вероятность найти нужное значение v очень мала.

Мы воспользуемся тем, что адаптивные алгоритмы — это то же самое, что алгоритмы с подсказкой и доступом к оракулу-функции. Алгоритм получает на вход 1^n и подсказку a_n длины $poly(n)$.

$A^g(1^n, a_n)$ — это результат работы такого алгоритма с функцией g в качестве оракула.

Определение 2. Систему псевдослучайных функций будем называть устойчивой относительно адаптивного отличителя, если $\forall A$ — отличителя $\forall q(\cdot)\, \exists N : \forall n > N \to |P_s(A^{f_s}(1^n, a_n) = 1) - P_g(A^g(1^n, a_n) = 1)| < \frac{1}{q(n)}$.

Утверждение 1. Построенная система функций устойчива относительно адаптивного отличителя.

Доказательство. Доказательство в целом такое же, только одного общего дерева нет, оно стоится по ходу алгоритма. Однако в ходе рассуждений ничего особо не меняется. $\square$

Вариации с параметрами могут быть следующие:

- Уменьшение длины - легко, если уменьшить длину выхода, ничего не нарушится.
- $f_s : \{0,1\}^* \to \{0,1\}^{r(n)}, s \in \{0,1\}^{p(n)}$. Используется генератор $G : \{0,1\}^{p(n)} \to \{0,1\}^{2p(n)+r(n)}$, $G(s) = \underbrace{G_0(s)}_{p(n)}\underbrace{G_1(s)}_{p(n)}\underbrace{G_2(s)}_{r(n)}$, а функции вычисляются так: $f_s(x) = G_2(G_{x_k}(G_{x_{k-1}}(\ldots)))$.