Лекция 2. Слабо и сильно односторонние функции

1 Построение сильно односторонней функции из слабой

Напоминание:

Определение 1. Слабо односторонняя функция $f(x)$ — это такая, что $\exists p(x) \geqslant 0 \forall \{C_n\}_{n=1}^{\infty} \exists N \forall n \geqslant N \to P(f(C_n(f(n))) = f(x)) < 1 - \frac{1}{p(n)}$, где C_n — семейство схем полиномиального размера, а $f(x)$ вычислима за полиномиальное время.

Определение 2. Сильно односторонняя функция $f(x)$ — это такая, что $\forall p(x) \geqslant 0 \forall \{C_n\}_{n=1}^{\infty} \exists N \forall n \geqslant N \to P(f(C_n(f(n))) = f(x)) < \frac{1}{p(n)}$.

Теорема 1. *Если существует слабо односторонняя функция, то существует и сильно односторонняя.*

Доказательство. Рассмотрим функцию $F(x_1, \ldots, x_N) = (f(x_1), \ldots, f(x_N))$. Ясно, что такая функция защищена от наивных обратителей, которые пытаются обратить каждую компоненту по отдельности. Однако, неясно, почему не существует более сложного и более эффективного обратителя.

Поэтому мы возьмем гипотетический обратитель R_F для F в обратитель R_f для f. Обратитель $R_f(y) = R_F(y, f(x_2), \ldots, f(x_N))\ |_1$ может не преуспеть, так как при фиксированной первой компоненте вероятность успеха может быть мала. Однако, мы можем запускать обратитель R_F много раз, поэтому сделаем так:

```
for (int i = 0; i < n; ++i) {
    for (int j = 0; j < K; ++j) {
        x_1, ..., x_n = gen_random();           // except i
        X = R_F(f(x_1), ..., y, ... , f(x_n)); // except i
        if (f(X) == y) {
            return X;
        }
    }
}
```

Для всех $i = 1, \ldots, n$ K раз выберем случайные $x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n$, и запустим $R_F(f(x_1), \ldots, f(x_{i-1}), y, f(x_{i+1}), \ldots, f(x_n))$ и выберем i-ю компоненту x. Если $f(x) = y$, вернем x.

Пусть $\rho_i(x) = P\{f(R_F(\ldots)) = f(x)\}$, $\rho_{\max}(x) = \max\limits_{i=1,\ldots,N} \rho_i(x)$. x бывает двух видов: такой, что $\rho_{\max}(x) \geqslant \varepsilon$ и такой, что $\rho_{\max} < \varepsilon$, доля последних равна δ.

Вероятность неудачи в таком случае $R_f \leqslant \delta + (1-\varepsilon)^k$. Если F — не сильно одностороняя, то вероятность успеха $R_F > \frac{1}{q(n)}$.

Для $(x_1, \ldots, x_n)$ вероятность, что все x_i хорошие $\leqslant (1-\delta)^N$, а если хотя бы один x плохой, то условная вероятность обращения $R_F < \varepsilon$.

Тогда вероятность успеха $\frac{1}{q(n)} < R_F < \varepsilon + (1-\delta)^N$. При $\varepsilon = \frac{1}{2q(n)}$ получается, что $(1-\delta)^N > \frac{1}{2q(n)}$. При $N = np(n) \Rightarrow \delta < \frac{1}{2p(n)}$. За счёт выбора K можем сделать $K = nq(n)$ и тогда $(1-\varepsilon)^K < \frac{1}{2p(n)}$. В итоге $\delta + (1-\varepsilon)^K < \frac{1}{p(n)}$, что означает, что f не слабо односторонняя. □

2 Примеры «односторонних» функций

Функция Рабина: $(x, y) \mapsto (x^2 \mod y, y)$. $y = p \cdot q, 0 \leqslant x < y$, притом p, q — простые числа вида $4k + 3$.

Функция RSA: $(x, y, z) \mapsto (x^z \mod y, y, z)$.

$P\{f(R_n(f(x))) = f(x)\}$ определяется по всем $x \in D_n$, притом требование к области D_n таково, что нужно уметь порождать случайные элементы D_n, то есть существует полиномиальный вероятностный алгоритм, порождающий случайную величину, статистически близкую к равномерной на D_n (расстояние между любыми двумя событиями меньше любого обратного полинома).

Можно отметить, что у функции Рабина, например, есть так называемый «секрет» (разложение $y = p \cdot q$), благодаря которому можно расшифровать сообщение. Более формально определим

Определение 3. Семейство односторонних функций с секретом $\{f_\alpha\}_{\alpha \in A}$: $f_\alpha : D_\alpha \to R_\alpha$ это такие функции, что существуют 4 алгоритма:

- Генератор: $1^n \to (\alpha, \tau)$, генерирует ключ и секрет.
- Сэмплер: $\alpha \mapsto$ случайный элемент D_α (с точностью до статистической близости).
- Вычислитель: $(\alpha, x) \mapsto f_\alpha(x)$.
- Обратитель: $(\alpha, \tau, y) \mapsto f_\alpha^{-1}(y)$.

Притом $(\alpha, y) \mapsto f_\alpha^{-1}(y)$ труднообратимо в обычном смысле.

Улучшенная односторонняя перестановка с секретом: y выбирается как случайный элемент D_α, а обратитель помимо α и y получает случайные биты, использованные при порождении y (при этом они все равно ему не помогают).

3 Генераторы псевдослучайных чисел

Определение 4. G — генератор псевдослучайных чисел, если

- $G : \{0,1\}^n \to \{0,1\}^{p(n)}$.
- G вычислима за полином.
- $\forall\{D_n\}_{n=1}^{\infty}\forall q(\cdot) \to \exists N : \forall n > N \to \left|P_{x\sim U_n}(D_n(G(x)) = 1) - P_{y\sim U_{p(n)}}(D_n(y) = 1)\right| < \frac{1}{q(n)}$.

Ясно, что генератор должен быть односторонней функцией, так как иначе обратитель мог бы отличить вывод генератора от случайного вывода.

Теорема 2. *Если существует односторонняя функция, то существует и генератор.*

Мы докажем ослабленную версию этой теоремы:

Теорема 3. *Если существует односторонняя перестановка, то существует и генератор.*

Определение 5. Трудный бит. Схематически: $x \mapsto f(x), x \mapsto b(x)$ вычисляются легко, $|b(x)| = 1$. При этом по $f(x)$ сложно вычислить $b(x)$:
$\forall q(\cdot)\forall\{P_n\}_{n=1}\infty\exists N\forall n > N|P(P_n(f(x)) = b(x)) - \frac{1}{2}| < \frac{1}{q(n)}$.

Схема доказательства теоремы такая:

- Односторонняя перестановка $f \mapsto$ односторонняя перестановка с трудным битом: $g(x,y) = (f(x),y), b(x,y) = x \odot y = \bigoplus_{i=1}^{n} x_i y_i$.
- Генератор $n \to n+1$: $G(x) = g(x)b(x)$.
- Генератор $n \to p(n)$: $g(g(x))b(g(x))b(x), g(g(g(x)))b(g(g(x)))b(g(x))b(x), \ldots$.