1 Необратимые и односторонние функции

1.1 Определения

Пусть имеется семейство функций $f_n\colon \mathbb{B}^{k(n)} \to \mathbb{B}^{l(n)}$, где k, l некоторые функции натурального аргумента с натуральными значениями, ограниченные сверху некоторым полиномом от n.

Мы хотим определить понятие "плохо обратимой" функции. Обратить $f(x)$ означает найти найти какое-то x' такое, что $f(x') = f(x)$. Возможны разные варианты определения: слабый и сильный. Неформально говоря, обращение слабо необратимой функции оказывается неверным с заметной (не пренебрежимо малой) вероятностью; для сильно необратимой функции оно неверно почти всегда (вероятность удачного обращения пренебрежимо мала). В качестве процедур обращения мы будем рассматривать последовательности схем C_n из функциональных элементов, размер которых ограничен некоторым полиномом от n. Схема C_n должна иметь $l(n)$ входов и $k(n)$ выходов.

Дадим точные определения.

Семейство функций f_n описанного вида называется *слабо необратимым*, если существует некоторый многочлен p с таким свойством: для любой последовательности схем C_n из функциональных элементов, размер которых ограничен некоторым полиномом от n вероятность события

$$f_n(C_n(f_n(x))) = f_n(x),$$

(где все слова x длины $k(n)$ считаются равновероятными) не превосходит

$$1 - \frac{1}{p(n)}$$

для всех достаточно больших n. Если $f_n(C_n(f_n(x))) = f_n(x)$, то мы говорим, что схема C_n успешно обращает $f_n(x)$.

В этом определении на вход схемы C_n (гипотетического алгоритма обращения) дается слово длины $l(n)$, и она должна найти слово длины $k(n)$, попадающее в f_n-прообраз входного слова. Такой прообраз существует, так как обращению подлежит слово вида $f_n(x)$ для некоторого (неизвестного алгоритму) слова x.

Важно, что мы не требуем точного обращения (равенства $C_n(f_n(x)) = x$). Если бы требовали, то любая функция, склеивающая все слова в одно,

оказалась бы необратимой.) Ещё отметим, что равномерное распределение берётся на словах x, а не на их образах (иначе трудность обращения была бы связана с тем, что многие слова не имеют прообраза).

Наконец, заметим, что многочлен p, задающий долю неудачных попыток обращения, не должен зависеть от последовательности схем C_n. Если оказывается, что с ростом сложности схем (в пределах полиномиальной) обращение становится возможным всё лучше и лучше, и ошибка может быть сделана меньше любого наперёд заданного полинома, то это не годится.

Семейство функций f_n называется *сильно необратимым*, если для любой последовательности схем C_n из функциональных элементов, размер которых ограничен некоторым полиномом от n, вероятность события

$$f_n(R(n, f_n(x))) = f_n(x),$$

(где все слова x длины $k(n)$ считаются равновероятными) стремится к нулю при $n \to \infty$ быстрее любого обратного полинома. Это означает, что для любого многочлена q она не превосходит

$$\frac{1}{q(n)}$$

для всех достаточно больших n. (Разумеется, та граница, с которой начинаются "достаточно большие" n, может зависеть от q.)

Семейство f_n называется полиномиально вычислимым, если имеется алгоритм, который по n вычисляет $k(n)$ и $l(n)$, а также имеется полиномиальный алгоритм, получающий на вход n и слово x длины $k(n)$ и вычисляющий (за полиномиальное от n время) слово $f_n(x)$. Необратимые полиномиально вычислимые семейства функций называются односторонними. Аналогично, слабо необратимые семейства, вычислимые за полиномиальное время называются слабо односторонними.

В дальнейшем мы будем позволять себе следующую терминологическую вольность: мы будем говорить "[слабо] необратимая функция" вместо "[слабо] необратимое семейство функций" и "[слабо] односторонняя функция" вместо "[слабо] одностороннее семейство функций".

Задача 1. (1) Докажите, что определение слабо необратимой функции можно переформулировать следующим образом: для некоторого полинома p для любого полинома q лишь для конечного числа n существует схема размера $q(n)$, успешно обращающая $f_n(x)$ с вероятностью не менее

$1 - 1/p(n)$. (2) Докажите, что определение сильно необратимой функции можно переформулировать следующим образом: для любого полинома q существует последовательность чисел ε_n, стремящаяся к нулю быстрее любого обратного полинома от n и такая, что лишь для конечного числа n существует схема размера не более $q(n)$, успешно обращающая $f_n(x)$ с вероятностью более ε_n.

Задача 2. Пусть для бесконечно многих n для всех слов x длины $k(n)$ выполнено $f_n(x) = 00\ldots0$. Докажите, что тогда f_n не является слабо обратимой. Докажите, что функция не является слабо необратимой даже, если мощность множества значений f_n ограничена некоторым полиномом от n (для бесконечно многих n).

Задача 3. Пусть даны сильно [слабо] необратимая функция $f_n : \{0,1\}^{k(n)} \to \{0,1\}^{l(n)}$ и числовая функция $m(n)$, ограниченная многочленом и вычислимая за полиномиальное от n время. Докажите, что тогда функция $xy \mapsto f_n(x)y$ определенная на словах длины $k(n) + m(n)$ сильно [слабо] необратима.

Задача 4. Докажите, что необратимые функции существуют.

Задача 5. Пусть даны две биективные полиномиально вычислимые функции $f_n, g_n : \{0,1\}^{k(n)} \to \{0,1\}^{k(n)}$. Докажите, что если хотя бы одна из них сильно [слабо] необратима, то и их композиция сильно [слабо] необратима.

Задача 6. Докажите, что существуют сильно необратимые функции $f_n, g_n : \{0,1\}^{k(n)} \to \{0,1\}^{k(n)}$, композиция которых не является даже слабо необратимой (полиномиальная вычислимость функций не требуется).

Задача 7. Докажите, что существуют слабо необратимые функции, не являющиеся сильно необратимыми.

Задача 8. Пусть $f_n : \{0,1\}^n \to \{0,1\}^n$ выбирается случайно среди всех функций (необязательно биективных). Все функции считаются равновероятными и f_n независимы при разных n. Докажите, что с вероятностью 1 семейство f_n необратимо.

Задача 9. Докажите, что если функция f_n необратима, то и функция $g_n(x) = f_n(x)0$ необратима.

Задача 10. Докажите, что существует необратимая функция f_n такая, что функция $x \mapsto (f_n(x)$ без последнего бита) обратима.

Задача 11. Докажите, что существуют функции f_n, g_n, не являющиеся слабо необратимыми и такие, что функция $x \mapsto f_n(x)g_n(x)$ сильно необратима.

Задача 12. Докажите, что если функция f_n сильно [слабо] необратима, то и функция $x \mapsto f_n(x)f_n(x)$ сильно [слабо] необратима.

Задача 13. Докажите, что существуют сильно необратимые функции f_n, g_n такие, что функция $x \mapsto f_n(x)g_n(x)$ не является даже слабо необратимой.

Назовем вероятностной схемой любую схему C, входы которой разделены на две группы, $y_1, \ldots, y_k$ и $r_1, \ldots, r_s$, называемых основными и случайными входами.

Задача 14. Докажите, что если функция f_n [слабо] необратима, то для любой последовательности вероятностных схем C_n, размер которых ограничен полиномом от n, с $l(n)$ основными входами и $k(n)$ выходами, вероятность события $f_n(C_n(f_n(x))) = f_n(x)$ стремится к нулю быстрее любого обратного полинома от n [не больше $1-1/s(n)$ для некоторого многочлена r для всех достаточно больших n]. Здесь мы считаем, что все слова x длины $k(n)$ считаются равновероятными, а на случайные входы схемы подаются независимые от x и друг от друга исходы бросаний симметричной монетки.

Неизвестно, существуют ли односторонние или хотя бы слабо односторонние функции. Доказать их существование сложно, поскольку из него следует, что P=NP.

Теорема 1. *Если P=NP, то любое полиномиально вычислимое семейство функций f_n, не является слабо обратимым. Более того, существует алгоритм, который для* всех *x по n и $f_n(x)$ за полиномиальное от n время находит некоторый прообраз $f(x)$ длины $k(n)$.*

Доказательство. Задача нахождения прообраза $f_n(x)$ есть частный случай полиномиальной задачи поиска, поскольку отношение "$f_n(z) = y$ и $|z| = k(n)$" разрешимо за полиномиальное время. Как любая задача поиска, она может быть решена за полиномиальное время, если P=NP. □

Приведем два семейства функций, которые возможно являются односторонними.

Произведение натуральных чисел: $f_n(x)$ равно произведению первой и второй половинок x, рассматриваемых как двоичные записи натуральных чисел. Здесь $k(n) = l(n) = 2n$ (произведение двух n-битовых чисел содержит не более $2n$ битов).

Функция SUBSET-SUM. Определение этой функции навеяно NP-полной задачей о сумме подмножеств (SUBSET-SUM). Здесь $k(n) = n^2 + n$, $l(n) = n^2 + 2n + \lceil \log n \rceil$. Значение f_n на словах x длины $n^2 + n$ определяется так. Разрежем x на $n+1$ блоков длины n и будем понимать первые n блоков как n натуральных чисел $x_1, \ldots, x_n$ в двоичной записи. Последний блок будем понимать как подмножество I множества $\{1, 2, \ldots, n\}$. По определению $f_n(x)$ равно конкатенации $x_1, \ldots, x_n$ и $\sum_{i \in I} x_i$.

Задача 15. Пусть $f_n : \{0,1\}^{k(n)} \to \{0,1\}^{l(n)}$ одностороннее семейство такое, что $k(n)$ строго возрастает. Докажите, что существует одностороннее семейство $g_m : \{0,1\}^m \to \{0,1\}^*$ такое, что $f_n = g_{k(n)}$ для всех n. (Семейство g_m можно было бы назвать всюду определенным односторонним продолжением f_n.)

Задача 16. Рассмотрим следующий способ продолжения функции f_n до всюду определенной функции. Положим $g_m(x) = f_n(x00\ldots0)$, где n выбрано как наименьшее такое число, что $m \leq k(n)$. Докажите, что существует необратимое семейство f_n, для которой так определенное семейство g_m не является даже слабо необратимым.

Как говорилось, существование сильно односторонних семейств является лишь гипотезой. (Эта гипотеза лежит в основе почти всех криптографических протоколов.) Но оказывается, что из слабо одностороннего семейства можно получить сильно одностороннее.

1.2 Усиление

Теорема 2. *Если существуют слабо односторонние функции, то существуют и сильно односторонние функции.*

(Заметим, что в этом случае существуют и слабо односторонние функции, не являющиеся сильно односторонними, см. задачу 7.)

Доказательство. Пусть f_n — слабо одностороннее семейство и $p(n)$ такой полином, что вероятность успешного обращения $f_n(x)$ схемами размера $\mathrm{poly}(n)$ при достаточно больших n не превосходит $1 - 1/p(n)$. Для краткости мы будем опускать индекс n и говорить об одной функции.

(Имеется в виду, что это построение и все рассуждения выполняются параллельно для всех n.) Рассмотрим новую функцию f^N, для которой

$$f^N(x_1x_2\ldots x_N) = f(x_1)f(x_2)\ldots f(x_N).$$

Она определена на словах длины $Nk(n)$, которые отождествляются с кортежами из N слов длины $k(n)$, и принимает значения длины $Nl(n)$. В качестве N мы возьмем некоторый достаточно большой многочлен от n (какой — скажем дальше).

Начнём с простого (но, увы, неправильного) объяснения, почему при достаточно большом N функция f^N является сильно односторонней. В самом деле, обращение $f^N(x_1\ldots x_N)$ для случайного $x_1\ldots x_N$ требует одновременного обращения $f(x_1),\ldots,f(x_N)$ при независимых случайных $x_1,\ldots,x_N$. В каждом из N случаев вероятность ошибки не меньше $1/p(n)$. Поэтому общая вероятность успеха не больше

$$\left(1-\frac{1}{p(n)}\right)^N.$$

Если положить $N = np(n)$, то выражение это равно $(1-1/p(n))^{p(n)}$ в степени n, что экспоненциально убывает (примерно равно e^{-n}). Поэтому функция f^N сильно односторонняя. Забегая вперед, отметим, что хотя это рассуждение и ошибочное, выбраное значение N окажется правильным.

Исправление

Что неверно в этом рассуждении? Дело в том, что мы рассматриваем вероятность успешного обращения f^N с помощью алгоритма специального вида, состоящего в том, что некий алгоритм обращения функции f применяется параллельно для всех n. А что делать, если алгоритм обращения такого вида не имеет? Кажется странным, что какие-то алгоритмы обращения функции f^N могут делать что-то более сложное, чем обращать в отдельности каждое $f(x_i)$, но это не доказательство (и непонятно, как это можно было бы формализовать буквально).

Правильное доказательство должно идти в другом направлении: вместо того, чтобы начинать с алгоритма обращения для f и затем смотреть, что получится при его раздельном применении к $f(x_i)$, мы должны рассмотреть произвольный алгоритм R, претендующий на обращение функции f^N, и показать, что если он имеет непренебрежимые шансы на успех,

то для функции f есть алгоритм R', у которого вероятность ошибки обращения меньше $1/p(n)$. Заметим, что алгоритм R' и реализующая его схема из функцинальных элементов по задаче 14 могут быть вероятностными.

Попробуем следующий (ещё не окончательный) вероятностный алгоритм обращения $f(x)$. Пусть нам дано некоторое слово y, прообраз которого мы ищем. Мысленно представим себе, что y есть $f(x_1)$ для неизвестного x_1 и дополним слово y словами $f(x_2), \ldots, f(x_N)$ для случайных слов $x_2, \ldots, x_N$. Применим к полученному слову

$$yf(x_2)\ldots f(x_N)$$

алгоритм R и посмотрим, что он даст на первом месте (каковы первые $k(n)$ битов). Это слово и будет результатом работы алгоритма.

Нетрудно понять, что вероятность успеха алгоритма R' (для $f(x_1)$ при случайном x_1) не меньше вероятности успешного обращения $f(x_1)\ldots f(x_N)$ алгоритмом R для случайных $x_1\ldots x_n$. Этого нам мало, поскольку надо перейти от не очень малой вероятности к вероятности, близкой к единице.

Как можно усовершенствовать алгоритм R'? Сразу возникают две идеи. Во-первых, можно производить несколько попыток обращения подряд (и выбирать из них удачную, если таковая случилась). Ведь сама функция f полиномиально вычислима, поэтому мы можем за полиномиальное время проверить, удалась ли попытка обращения. Другая идея: мы может ставить обращаемое слово y не только на первое место, но и на любое другое место от 1 до N.

Алгоритм обращения

Этих двух идей окажется достаточно для доказательства теоремы. А именно, для данной схемы R, претендующей на обращение f^N, мы рассмотрим вероятностную схему для обращения $f(x)$, действующую следующим образом.

Для каждого i от 1 до N поставим обращаемое слово y на i-е место и окружим его $N-1$ словами, получив набор

$$f(x_1), \ldots, f(x_{i-1}), y, f(x_{i+1}), \ldots, f(x_N).$$

Слова $x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_N$ выбираются независимо и равномерно распределены в $\mathbb{B}^{k(n)}$. К полученному набору применяется схема R. Если i-я

компонента результата лежит в прообразе y, констатируем удачу (и сворачиваем всё дальнейшее).

Эти действия (N попыток, по одной для каждого i) составляют один этап алгоритма R'. Полностью R' состоит в повторении этого действия несколько раз. Обозначим число повторений через M (и выберем его дальше).

[Заметим сразу, что ставить y на разные позиции необходимо. В самом деле, если R почему-то не работает на образах слов, в которых первый бит (первого слова) равен нулю, то он вполне может иметь вероятность успеха $1/2$. Но тогда и алгоритм R' (если y ставится лишь на первое место) не будет работать на образах таких слов, и вероятность успеха не подымется выше $1/2$, сколько повторений ни делай.]

Теперь нам нужно подобрать значение M. (Значение M может зависеть только от n, функции f_n и схемы R.) Нам нужно, чтобы из того, что вероятность успеха R' меньше $1 - 1/p(n)$, следовало бы, что вероятность успеха R меньше $1/q(n)$. Для того чтобы подобрать M с этим свойством, необходимо разобраться, как связаны вероятности успеха для алгоритмов R и R'. Обозначим через $s_1(x_1)$ вероятность того, алгоритм R правильно обращает слово

$$f(x_1)f(x_2)\ldots f(x_N)$$

для случайно выбранных $x_2, \ldots, x_N$. Вероятность успешного обращения для первой попытки (когда мы ставим y на первое место) для слова $f(x_1)$ не меньше $s_1(x_1)$. Аналогичные вероятности можно рассмотреть и для других попыток (всего в каждой фазе N попыток). По аналогии обозначим их $s_2(x_2), \ldots, s_N(x_N)$.

Какова вероятность успеха на одной фазе? Для входа $f(x)$ она заведомо не меньше

$$\max(s_1(x), \ldots, s_N(x)),$$

поскольку успех одной попытки означает успех всей фазы. (Более точно сказать трудно, эта вероятность, вообще говоря, не определяется значениями $s_i(x)$, поскольку успехи в попытках зависимы: может оказаться, скажем, что некоторые слова трудны для обращения во всех координатах.) Обозначим этот максимум через $\hat{s}(x)$.

Дальнейший ход доказательства можно описать так. Посмотрим, сколько имеется слов x, для которых величина $\hat{s}(x)$ очень мала (меньше некоторой границы ε, которую мы выберем впоследствии). Такие слова, нефор-

мально говоря, трудны для обращения. Нетрудные слова хорошо обращаются алгоритмом R', поскольку на каждой фазе вероятность обращения не меньше ε, фазы независимы и их много. Пользуясь тем, что вероятность неудачи R' больше $1/p(n)$ мы докажем, что трудных слов много. Отсюда будет следовать R обращает f^N с вероятностью меньше $1/q(n)$. Эти неформальные разговоры следует, конечно, уточнить.

Оценки

Пусть ε — некоторые положительное число, порядка $1/\mathrm{poly}(n)$. Будем называть слово x "трудным", если $\hat{p}(x) < \varepsilon$. Обозначим через δ долю трудных слов среди всех слов длины n. Тогда вероятность вероятность ошибки при обращении слова $f(x)$ (для случайного x) меньше

$$\delta + (1-\varepsilon)^M.$$

Положим $M = n/\varepsilon$, второе слагаемое тогда примерно равно e^{-n}.

Алгоритм R' реализуется вероятностной схемой, которая по существу состоит из MN схем R и такого же количества схем для вычисления f_n, а значит реализуется схемой размера, ограниченного многочленом от n. По условию вероятность ошибки R' не меньше $1/p(n)$ (если n достаточно велико). Поэтому при всех достаточно n выполнено $\delta \geq 1/2p(n)$.

Выделим из трудных слов часть, образующую долю ровно $1/2p(n)$. Оценим сверху вероятность успешного обращения $f^N(x_1, \ldots, x_N)$ алгоритмом R на случайном входе. Оценивая её, разделим пространство на две части: когда одно из слов x_i является выделенным и когда все они не выделены. Для второй части нам не важны значения $\hat{s}(x)$, мы пользуемся тем, что эта часть сама по себе имеет вероятность не больше $(1-1/2p(n))^N$ (каждое из x_i выделено с вероятностью $1/2p(n)$, а всего их N и они независимы).

Первая часть: мы оценим вероятность в случае, когда выделенным (а значит трудным) оказалось слово x_i, а затем умножим оценку на N (вероятность объединения событий не больше суммы вероятностей). Вероятность того, что слово x_i окажется выделенным, равна $1/2p(n)$, а для каждого из выделенных слов x_i условная вероятность успешной работы алгоритма R (при данном значении i-й координаты) меньше ε. В итоге получаем для первой части оценку $N\varepsilon/2p(n)$, а для суммы (вероятности успешной работы R на случайном входе) оценку

$$N\varepsilon/2p(n) + (1-1/2p(n))^N.$$

Нам надо выбрать ε, N так, чтобы эта величина была строго меньшей $1/q(n)$. Как было уже обещано, положим $N = np(n)$. Тогда второе слагаемое в этой сумме равно

$$\left(1 - \frac{1}{2p(n)}\right)^{np(n)} = \left[\left(1 - \frac{1}{2p(n)}\right)^{2p(n)}\right]^{n/2} \approx e^{-n/2}$$

а значит при больших n намного меньше $1/q(n)$. Поэтому ε можно определить так, чтобы первое слагаемое было равным $1/2q(n)$, то есть положить

$$\varepsilon = \frac{1}{nq(n)}.$$

Теорема доказана.

1.3 Частично необратимые функции

В определении необратимости требуется, чтобы для любой последовательности схем C_n полиномиального размера вероятность успеха C_n была пренебрежимо мала по *равномерному распределению на словах длины* n. Теперь мы разрешим вместо равномерного распределения использовать любое распределение вероятностей, которое можно порождать за полиномиальное от n время. Необратимые в этом смысле функци будем называть частично необратимыми.

Перейдем к формальному определению. Последовательность распределений вероятностей μ_n на множестве двоичных слов называется генерируемой за полиномиальное время, если существует полиномиальный вероятностный алгоритм K такой, что для всех $x \in \{0,1\}^*$ выполнено

$$\Pr[K \text{ выдает } x \text{ на входе } n] = \mu_n(x).$$

Алгоритм K получает на вход число n в унарной записи и при любых исходах своих бросаний должен остановится за полиномиальное от n время и выдать некоторую двоичную строку. Случайная величина называется генерируемой за полиномиальное время, если ее распределение генерируется за полиномиальное время.

Пусть f_n семейство функций вида $\{0,1\}^{k(n)} \to \{0,1\}^{l(n)}$. Последовательность случайных величин $\{\alpha_n\}$, все возможные значения которой

имеют длину $k(n)$ называется трудной для функции f, если для любой последовательности схем C_n размера $\mathrm{poly}(n)$ вероятность события

$$f_n(C_n(f_n(\alpha_n))) = f(\alpha_n)$$

стремится к нулю быстрее любого обратного полинома от n. Последовательность $\{\alpha_n\}$ называется нелегкой для функции f, если для некоторого многочлена p для любой последовательности схем C_n размера $\mathrm{poly}(n)$ вероятность события

$$f_n(C_n(f_n(\alpha_n))) = f(\alpha_n)$$

меньше $1 - 1/p(n)$ для всех достаточно больших n.

Семейство f_n называется частично сильно необратимым, если существует доступная и трудная для f_n последовательность случайных величин. Семейство f_n называется слабо частично необратимой, если существует доступная и нелегкая для f_n последовательность случайных величин. Семейство f называется частично сильно [слабо] односторонним, если оно полиномиально вычислимо и частично сильно [слабо] необратимо.

Когда необходимо отличить частично необратимые функции от необратимых в обычном смысле, будем называть последние полностью необратимыми. Любая полностью сильно необратимая функция частично необратима, трудным распределением для нее является равномерное распределение μ_n на множестве слов длины n. То же самое относится и к слабо необратимым функциям.

Задача 17. Докажите, что существует частично сильно необратимая функция g, не являющаяся полностью сильно необратимой.

Эта задача показывает, что определения частично и полностью сильно односторонней функции неравносильны (при условии, что существуют полностью односторонние функции). Однако из *существования* частично сильно односторонней функции следует существование полностью сильно односторонней функции.

Задача 18. Предположим, существует частично сильно одностороняя функция. Докажите, что тогда существует и полностью сильно одностороняя функция.

Задача 19. Предположим, существует частично слабо одностороняя функция. Докажите, что тогда существует и слабо одностороняя функция.

Задача 20. Предположим, существует частично слабо одностороняя функция. Докажите, что тогда существует и (полностью) сильно одностороняя функция.

Задача 21. Докажите, что существует частично сильно необратимая функция g, не являющаяся полностью слабо необратимой.

Задача 22. Докажите, что существует слабо необратимая функции, не являющаяся частично сильно необратимой.

Примеры предположительно частично сильно односторонных функций

Во всех трех примерах предположительно трудное распределение будет в некоторм смысле близко к равномерному распределению на некотором множестве.

Статистическим расстоянием между распределениями вероятностей μ и ν на множестве двоичных слов называется максимум $|\mu(A) - \nu(A)|$ по всем множествам слов A. Нетрудно понять, что этот максимум достигается на множестве слов $A = \{x \mid \mu(x) > \nu(x)\}$ (а также на его дополнении) и равен $\sum_{x\in\{0,1\}^*} |\mu(x) - \nu(x)|/2$.

Последовательности распределений вероятностей μ_n и ν_n, называются статистически неотличимыми, если статистическое расстояние между μ_n и ν_n стремится к нулю быстрее любого обратного многочлена от n. Аналогично определяется статистическая неотличимость случайных величин: случайные величины α_n и β_n, зависящие от натурального параметра n, называются статистически неотличимыми, если их распределения $\mu_n(x) = \Pr[\alpha_n = x]$ и $\nu_n(x) = \Pr[\beta_n = x]$ статистически неотличимы.

Если случайные величины α_n и β_n статистически неотличимы, и одно из них является трудным для частично необратимой функции f_n, то и другое будет трудным.

Функция Рабина. Функция Рабина f_n определена на словах длины $4n$; на слове вида xy (имеется в виду конкатенация слов), где $|x| = |y| = 2n$ ее значение равно конкатенации $x^2 \bmod y$ и y (слова x, y понимаются как двоичные записи натуральных чисел).

Предположительно трудная случайная величина α_n равномерно распределена на множестве, состоящем из всех слов xy длины $4n$ следующего вида: y есть двоичная запись числа $p \cdot q$, где p, q простые числа из n битов, а x произвольное $2n$-битовое число. Алгоритм генерации случайной величины, статистически неотличимой от α_n работает так. Выбираем

случайно число из n битов и проверяем, просто ли оно, с помощью полиномиального алгоритма проверки простоты. Если оно просто, положим p равным этому числу. Иначе повторяем попытку и т.д. Если после n^2 попыток простое число не найдется, то положим $p = 0$. Из закона распределения простых чисел следует, что вероятность этого события будет меньше $(1 - \varepsilon/n)^{n^2} < e^{-\varepsilon n}$ для некоторого положительного ε. Затем так же генерируем q. Число x выбираем равномерно среди всех n-битовых чисел. (Можно вместо алгоритма трех индийцев использовать и вероятностный тест на простоту с вероятностью ошибки, скажем 2^{-n}. Тогда статистическое расстояние будет больше на величину, равную вероятность того, что p или q составные, то есть 2^{-n+1}.)

Функция RSA. Функция RSA есть обобщение функции Рабина. Она определена на словах длины $6n$. Ее значение на на словах вида xyz, где x, y и z имеют длину $2n$ и понимаются как двоичные записи натуральных чисел, равно конкатенации $(x^z \bmod y)$, y и z.

Предположительно трудная случайная величина равномерно распределена на всех строках вида xyz, где $y = p \cdot q$, p, q — простые n-битовые числа, а x, z произвольные $2n$-битовые числа. Статистически неотличимая от нее случайная величина генерируется за полиномиальное время так же, как для функции Рабина. Точнее, y генерируется так же алгоритмом, как для функции Рабина, а x, z выбираются случайно.

Экпонента в кольце вычетов (функция Блюма и Микэли). Функция f_n словах длины $3n$. Ее значение на слове xyz, где x, y, z слова длины n, понимаемые как двоичные записи натуральных чисел, равно конкатенации $x^z \bmod y$, x и z. Отличие от функции RSA здесь в том, что при обращении нужно найти показатель степени, а не ее основание. То есть, обращение является логарифмированием, а не извлечением корня. Предположительно трудная случайная величина равномерно распределена на всех строках вида xyz, где y есть n-битовое простое число, а x и z произвольные n-битовые числа. Задача обращения функции Блюма и Микэли называется дискретным логарифмированием.

2 Генераторы псевдослучайных чисел

2.1 Определение

Случайные величины α_n и β_n, зависящие от натурального параметра n, со значениями в множестве слов некоторой длины $l(n)$ называются вычислительно неотличимыми, если для любой последовательности схем C_n размера $\mathrm{poly}(n)$ (с $l(n)$ входами и одним выходом) вероятность событий $C_n(\alpha_n) = 1$ и $C_n(\beta_n) = 1$ отличаются на пренебрежимо малую величину. Схема C_n в этом контексте называется тестом и мы говорим, что случайная величина α_n проходит тест C_n, если $C_n(\alpha_n) = 1$. Таким образом, мы требуем, чтобы α_n и β_n проходили любые тесты полиномиального размера с приблизительно равной вероятностью.

Если α_n и β_n статистически неотличимы, то они и вычислительно неотличимыми, поскольку разность вероятностей попадания α_n и β_n в множество $\{x \mid C_n(x) = 1\}$, задаваемое тестом C_n, не превосходит статистического расстоянимя между α_n и β_n. Последовательность распределений (случайных величин) назовем доступной, если она статистически неотличима от некоторой последовательности распределений (случайных величин), генерируемой за полиномиальное время.

Нам понадобятся некоторые простые свойства понятия вычислительной неотличимости.

Лемма 3. *1) Отношение вычислительной неотличимости рефлексивно, симметрично и транзитивно.*
2) Пусть α_n и β_n вычислительно неотличимы, а $l(n)$ некоторая функция натурального аргумента, не превосходящая длины значений α_n. Тогда случайная величины, получаемые из α_n и β_n отрезанием последних $l(n)$ битов, вычислительно неотличимы.
3) Если α_n и β_n вычислительно неотличимы, p некоторый полином, а γ_n случайная величина, независимая от α_n и β_n и равномерно распределенная среди всех строк некоторой длины $k(n)$, то случайные величины $\alpha_n\gamma_n$ и $\beta_n\gamma_n$ вычислительно неотличимы.
4) Если α_n и β_n вычислительно неотличимы, а C_n последовательность схем полиномиального от n размера с $l(n)$ входами, то случайные величины $C_n(\alpha_n)$ и $C_n(\beta_n)$ вычислительно неотличимы.

Доказательство. Первое и второе утверждения очевидны. Докажем третье. Допустим, что существует последовательность схем-тестов T_n поли-

номиального от n размера с $k(n)+l(n)$ входами такая, что вероятности событий $T_n(\alpha_n\gamma_n)=1$ и $T_n(\beta_n\gamma_n)=1$ отличаются на $\varepsilon=1/\mathrm{poly}(n)$. Фиксируем одно из этих n. Вероятность события $T_n(\alpha_n\gamma_n)=1$ есть среднее арифметическое вероятностей событий $T_n(\alpha_n r)=1$ для различных $r\in\{0,1\}^{k(n)}$. Аналогичное верно для вероятности события $T_n(\beta_n\gamma_n)=1$. Поскольку средние арифметические отличаются не менее, чем на ε, найдется такое $r=r_n\in\{0,1\}^{k(n)}$, что вероятности событий $T_n(\alpha_n r)=1$ и $T_n(\beta_n r)=1$ отличаются не менее, чем на ε. Преобразование $x\mapsto T_n(xr_n)$ вычисляется некоторой схемой C_n, размер которой не больше размера T_n ("запаиваем" r_n в схему T_n). Получили противоречие с вычислительной неотличимостью α_n и β_n.

Докажем четвертое свойство. Пусть дана последовательность тестов T_n полиномиального от n размера с $s(n)$ входами (где $s(n)$ — количество выходов C_n) на отличимость $C_n(\alpha_n)$ и $C_n(\beta_n)$. Тогда последовательность схем $D_n(x)=T_n(C_n(x))$ можно рассматривать, как тест на отличимость α_n и β_n. Размер схемы D_n есть сумма размеров T_n и C_n, а значит ограничен полиномом от n. Следовательно вероятность того, что α_n проходит тест D_n пренебрежимо мало отличается от вероятности того, что β_n проходит тест D_n. Осталось заметить, что первая вероятность равна вероятности того, что случайная величина $C_n(\alpha_n)$ пройдет тест T_n, и то же самое верно для $C_n(\beta_n)$. □

Задача 23. Докажите, что определение частично односторонней функции не изменится, если потребовать, что трудная случайная вычислительно неотличима от некоторой случайной величины, генерируемой за полиномиальное время (а не сама генерируется за полиномиальное время, как требовалось).

Пусть даны полиномиально вычислимые функции натурального аргумента $k(n), l(n)$, ограниченные сверху полиномом от n такие, что $l(n)>k(n)$ для всех n. Генератором ПСЧ типа $k(n)\to l(n)$ будем называть семейство функций G_n, где для каждого n функция G_n отображает двоичные слова длины $k(n)$ в слова длины $l(n)$, удовлетворяющее следующим двум условиям:

(1) Семейство G_n вычислимо за полиномиальное время (по данным n и s за полиномиальное от n количество шагов можно найти $G_n(s)$).

(2) Случайная величина $G_n(s)$ (где x выбирается случайно среди всех строк длины $k(n)$) вычислительно неотличима от равномерно распределенной среди слов длины $l(n)$ случайной величины (при стремлении n к

бесконечности). Это свойство генератора называют надежностью.

Задача 24. Докажите, что второе требование нельзя усилить, потребовав статистической неотличимости $G_n(s)$ и равномерно распределенной случайной величины. [Пусть A множество всех исходов случайная величиной величины $G_n(s)$. Вероятность того, что случайно выбранное слово попадет в A значительно меньше вероятности того, что $G_n(s)$ попадет в A.]

Определим также генераторы ПСЧ типа $n \to \infty$, как семейства G_n, где G_n отображает двоичные слова длины n в бесконечные двоичные последовательности, удовлетворяющие следующим требованиям.

(1) Существует алгоритм, который по слову s и натуральному l за полиномиальное от $|s|+l$ время вычисляет l-ый бит последовательности $G_n(s)$.

(2) Случайная величина $G_n(s)$ вычислительно неотличима от равномерно распределенной бесконечной последовательности нулей и единиц.

В этом определении вычислительная неотличимость двух случайных величин ω_n, ξ_n со значениями в множестве всех бесконечных двоичных последовательностей понимается в следующем смысле: для любой последовательности схем C_n размера $\mathrm{poly}(n)$ вероятность событий $C_n(\omega_n) = 1$ и $C_n(\xi_n) = 1$ отличаются на пренебрежимо малую величину. Здесь $C_n(\omega_n)$ обозначает результат применения схемы к k первым битам ω_n, где k — количество входов C_n. Аналогично понимается $C_n(\xi_n)$. Другими словами, ω_n и ξ_n вычислительно неотличимы, если для любого полинома $p(n)$ вычислительно неотличимы случайные величины, равные и первым $p(n)$ битам случайных величин ω_n и ξ_n.

Имеется следующее соотношение между генераторами типа $k(n) \to l(n)$ и генераторами типа $n \to \infty$. Если $G_n : \{0,1\}^* \to \{0,1\}^\infty$ — генератор типа $n \to \infty$, то для любых l, k последовательность функций $H_n(s) = (G_n(s))_{l(n)}$, где $s \in \{0,1\}^{k(n)}$, будет генератором типа $k(n) \to l(n)$. Обратно, мы докажем, что по любому генератору типа $k(n) \to l(n)$ (какие бы ни были k, l) можно построить генератор типа $n \to \infty$.

Задача 25. Докажите, что существует функция G_n (не обязательно вычислимая за полиномиальное время), типа $n \to n+1$ такая, что случайная величина $G_n(s)$ вычислительно неотличима от равномерно распределенной случайной величины.

Задача 26. Докажите, что если функция G_n является генератором типа $k(n) \to l(n)$, а x_n последовательность слов длины $l(n)$, вычислимая за

время $\mathrm{poly}(n)$, то функция $s \mapsto G_n(s) \oplus x_n$ является генератором.

2.2 Генераторы ПСЧ и односторонние функции

Любой генератор G_n типа $k(n) \to l(n)$ является слабо необратимой функцией. Действительно, никакая последовательность схем C_n полиномиального размера не может обратить $G_n(s)$ в вероятностью успеха, большей $3/4$ (для бесконечно многих n). Допустим, такая последовательность схем существует. Тогда рассмотрим следующий тест на последовательностях длины $l(n)$: применяем к последовательности y схему C_n, затем проверяем, с помощью алгоритма, вычисляющего G_n, правильно ли схема C_n обратила y (то есть, $G_n(C_n(y)) = y$). Если обращение произошло удачно, то выдаем 1, а иначе 0. Вероятность того, что тест будет пройден случайной величиной $G_n(s)$, не менее $3/4$ (для бесконечно многих n), тест выдает единицу на y, если C_n правильно обращает y. Вероятность того, что равномерно распределенная случайная величина пройдет тест, не больше $1/2$, поскольку множество значений G_n составляет не более половины всего множества последовательностей длины $l(n)$. Поскольку C_n и G_n можно вычислить схемой полиномиального от n размера, описанный тест имеет полиномиальный размер.

Итак, если для каких-то $l(n) > k(n)$ существует генератор типа $k(n) \to l(n)$, то существуют и слабо однносторонние, а значит и сильно односторонние функции. Оказывается, верно и обратное.

Теорема 4. *[1] Если существует сильно односторонняя функция, то существует и генератор ПСЧ типа $n \to \infty$.*

Мы докажем более слабое утверждение: если существует односторонняя перестановка, то существует и генератор ПСЧ типа $n \to \infty$. Сначала дадим определение односторонней перестановки.

Функция $f_n : D_n \to D_n$ называется односторонней перестановкой, если (1) $D_n \subset \{0,1\}^{k(n)}$ и отображение $n, x \mapsto f_n(x)$ вычислимо за время $\mathrm{poly}(n)$, (2) функция f_n является перестановкой множества D_n и (3) случайная величина, равномерно распределенная на D_n является доступной и трудной для f_n.

Задача 27. Докажите, что любая односторонняя перестановка может быть продолжена до сильно частично односторонней функцией.

Приведем примеры (предположительно) односторонних перестановок. Известны три примера и все они являются односторонними перестановками. А именно, в областях определения функции Рабина, функции RSA и функции Блюма-Микэли мы выделим некоторое подмножество D_n с такими свойствами: (1) равномерное распределение на D_n доступно и (2) представляется правдоподобным, что равномерное распределение на D_n является трудным (говоря точнее, до сих пор не удалось опровергнуть это предположение).

Функция Рабина. Рассмотрим следующее множестве D_n. Оно состоит из всех строк вида xy, где $y = p \cdot q$ и p, q взаимно простые n-битовые числа вида $4k+3$, причем x и y взаимно просты, а $x \in [0, y)$ является полным квадратом по модулю y. Докажем, что функция Рабина является перестановкой D_n. Ясно, что f отображает D_n в себя. Поэтому достаточно доказать, что f инъективна. Пусть $x = z^2$ и z взаимно просто с p. Тогда по x^2 можно найти x по модулю p, возведя x в степень $(p+1)/4$. Действительно, $(x^2)^{(p+1)/4} = z^{p+1} = z^2 = x \pmod{p}$. То же самое верно и по модулю q, поэтому по x^2 и $y = pq$ можно восстановить x. (Из этого, конечно, не следует обратимость функции Рабина, поскольку для ее обращения нужно разложить y на простые множители.)

Статистически неотличимую от равномерно распределенной на D_n величину можно генерировать следующим образом так же, как раньше. При этом нам важно, что простых чисел вида $4k + 3$ достаточно много: среди всех n-битовых чисел они составляют долю не менее $1/\mathrm{poly}(n)$.

Функция RSA. Для функции RSA рассмотрим следующее подмножество D_n ее области определения: в него включаются только те xyz, для которых y есть произведение двух простых n-битовых чисел p и q, число x находится в интервале $[0, pq)$, а z взаимно просто со значением функции Эйлера на pq, $\phi(pq) = (p-1)(q-1)$. Очевидно, что функция RSA отображает D_n в себя. Инъективность следует из того, что по модулю $\phi(pq)$ число z имеет обратный элемент u. Поэтому по x^z можно найти x, возведя x^z в степень u. Действительно, $\phi(pq)$ есть мощность мультипликативной группы вычетов по модулю pq и по теореме Лагранжа мы имеем $x^{zu} = x^{1+k\phi(pq} = x \pmod{pq}$. Из этого, конечно не следует, обратимность функции RSA, поскольку для обращения нужно знать u (или числа p и q, по которым u можно вычислить).

Доступность равномерного распределения на D_n не очевидна из-за того, что надо уметь порождать числа, взаимно простые с $(p-1)(q-1)$. Для этого нам нужно знать разложение $(p-1)(q-1)$ на простые мно-

жители. Оказывается, можно порождать равномерное распределение на простых n-битовых числах вместе с разложением числа $p-1$ на простые множители (см. [2]).

Функция Блюма и Микэли. Для функции Блюма и Микэли рассмотрим следующее подмножество D_n ее области определения: в него включаются только те xyz, y в для которых y есть n-битовое простое число, x является генератором мультипликативной группы вычетов по модулю y, а $z \in [0, y-1)$. По определению генератора, функция Блюма и Микэли является перестановкой D_n. Доступность равномерного распределения на D_n неочевидна, это доказано в [2].

Теорема 5. *Если существует односторонняя обобщенная перестановка, то существует генератор типа $n \to \infty$.*

2.3 Трудный бит

Сначала мы научимся добавлять один случайный бит к уже имеющимся. Для этого нам понадобится понятие трудного бита для данной функции. Пусть β_n, γ_n последовательность пар совместно распределенных случайных величин, причем случайная величина β_n имеет значения 0,1. Мы говорим, что β_n является трудно вычислимой по для γ_n если для любой последовательности схем C_n размера $\mathrm{poly}(n)$ вероятность события $C_n(\gamma_n)) = \beta_n$ приблизительно равна $1/2$ (то есть, стремится к $1/2$ быстрее любого обратного полинома от n). То есть, случайную величину β_n нельзя вычислить по γ_n с вероятностью существенно лучшей, чем при простом угадывании.[1] Если β_n является трудно вычислимой по γ_n, то оба своих значения она принимает с примерно равной вероятностью (иначе в качестве C_n можно взять схему, которая выдает 0 независимо от входа).

Полиномиально вычислимая функция $h_n : D_n \to \{0,1\}$ является трудным битом для функции $g_n : D_n \to D_n$, если случайная величина $h_n(\alpha_n)$ трудно вычислима по случайной величине $g_n(\alpha_n)$, где α_n — случайная величина, равномерно распределенная в D_n.

Задача 28. Докажите, что если имеется трудный бит для перестановки $g_n : D_n \to D_n$, то перестановка сильно необратима.

[1] В общем случае вероятность должна быть приблизительно равной $1/|M_n|$, где M_n — множество возможных значений β_n.

Трудным битом для функции Рабина является последний бит (четность), в предположении необратимости функции Рабина). Трудным битом для функции Блюма–Микэли (в предположении необратимости этой функции) является функция $h_n(xyz) = 0$, если $z < (y-1)/2$, и $h_n(xyz) = 1$ иначе.

Лемма 6. *Случайная величина β_n трудно вычислима по γ_n тогда и только тогда, когда случайные величины $\beta_n\gamma_n$ и $r\gamma_n$ вычислительно неотличимы. (Здесь r есть равномерно распределенный бит, независимый от γ_n.)*

Доказательство. В одну сторону утверждение леммы очевидно: если $\beta\gamma$ и $r\gamma$ вычислительно неотличимы, то β является трудным битом для γ. (Мы опускаем индекс n.) Действительно, пусть имеется последовательнось схем C_n полиномиального от n размера такая, что для бесконечно многих n вероятность события $C(\gamma) = \beta$ отличается от $1/2$ не менее, чем на $\varepsilon = 1/\mathrm{poly}(n)$. Рассмотрим следующий тест D: он применяет ко входной последовательности без первого бита C и выдает 1, если результат равен первому биту входной последовательности. Случайная величина $r\gamma$ проходит этот тест с вероятностью ровно $1/2$, поскольку ее первый бит не зависит от последующих битов и принимает значения 0,1 с равными вероятностями. Вероятность того, что последовательность $\beta\gamma$ пройдет этот тест, в точности равна вероятности того, что $C(\gamma) = \beta$. Ясно, что тест D задается схемой полиномиального от n размера.

Теперь докажем обратное. Допустим, что существует последовательность схем D размера $\mathrm{poly}(n)$ такая, что для бесконечно многих n выполнено

$$\Pr[D(\beta\gamma) = 1] - \Pr[D(r\gamma) = 1] = \varepsilon,$$

где $|\varepsilon| \geq 1/\mathrm{poly}(n)$. Фиксируем любое из этих бесконечно многих n. Пусть для определенности $\varepsilon > 0$. Это значит, что схема D "больше любит" β, чем случайный бит. Рассмотрим следующий алгоритм C вычисления β по γ.

Если $D(0\gamma) = D(1\gamma)$, то выдаем 0 и 1 с равными вероятностями. Если $D(0\gamma) = 0$, $D(1\gamma) = 1$, выдаем 1 (неформальное объяснение: β скорее равен 1, чем 0, поскольку схема больше любит 1, чем 0). Если $D(0\gamma) = 1$, $D(1\gamma) = 0$, выдаем 0.

Вероятность того, что этот алгоритм выдаст β равна $1/2 + \varepsilon$. Чтобы доказать это, достаточно доказать это при любом фиксированном зна-

чении $\bar{\gamma}$ случайной величины γ выполнено равенство

$$\Pr[C(\bar{\gamma}) = \beta] = 1/2 + \Pr[D(\beta\bar{\gamma}) = 1] - \Pr[D(r\bar{\gamma}) = 1]. \qquad (1)$$

(В этом уровнении все вероятности вычисляются при условии $\gamma = \bar{\gamma}$.) Пусть $\bar{\gamma}$ фиксировано. Если $D(0\bar{\gamma}) = D(1\bar{\gamma})$ то схема не чувствует замены 0 на 1. Поэтому правая часть (1) равна $1/2$. Левая часть также равна $1/2$ по определению C.

Если $D(0\bar{\gamma}) = 0$, $D(1\bar{\gamma}) = 1$, то левая часть (1) равна $\Pr[1 = \beta]$, а в правая равна $1/2 + \Pr[\beta = 1] - 1/2$, и равенство выполнено. Если $D(0\bar{\gamma}) = 1$, $D(1\bar{\gamma}) = 0$, то левая часть (1) равна $\Pr[0 = \beta]$, а правая равна $1/2 + \Pr[\beta = 0] - 1/2$.

Алгоритм C задается вероятностной схемой примерно вдвое большей, чем D, поэтому имеющей также полиномиальный размер. Можно обойтись и схемой размера, примерно равного размеру схемы D, если заметить, что $C(\gamma) = D(\gamma r) \oplus r \oplus 1$, где r случайный бит, независимый от γ. □

Пусть h_n — трудный бит для односторонней перестановки $g_n : D_n \to D_n$, а случайная величина α_n равномерно распределена в D_n. По лемме случайные величины $h_n(\alpha_n)g_n(\alpha_n)$ и $rg_n(\alpha_n)$ вычислительно неотличимы. Вторая имеет то же распределение, что и $rg_n(\alpha_n)$. Таким образом, случайные величины $h_n(\alpha_n)g_n(\alpha_n)$ и $r\alpha_n$ вычислительно неотличимы. Говоря неформально, случайная величина $h_n(\alpha_n)g_n(\alpha_n)$ имеет на один бит случайности больше, чем α_n. Повторяя этот прием, можно добавить второй, третий случайные биты, и так далее. На этом приеме и основано построение генераторов ПСЧ.

Пусть дан произвольный полином $p(n)$, задающий количество случайных битов, которые нам нужно получить. Рассмотрим последовательность

$$\alpha, g(\alpha), g(g(\alpha)), \ldots, g^{p(n)-1}(\alpha)$$

(для наглядности мы опускаем индекс n). Применим ко всем членам этой последовательности функцию h. Утверждается, что полученная последовательность

$$h(\alpha), h(g(\alpha)), h(g(g(\alpha)), \ldots, h(g^{p(n)-1}(\alpha))$$

будет вычислительно неотличима от случайной равномерно распределенной последовательности (при $n \to \infty$). На самом деле, мы докажем

чуть больше: если добавить в хвост этой последовательности, $g^{p(n)}(\alpha)$, то полученная последовательность будет вычислительно неотличима от равномерно распределенной последовательности длины $p(n)$ с приписанным в конец α_n. Это усиление нам понадобится в дальнейшем.

Лемма 7. *Пусть h_n является трудным битом для перестановки $g_n : D_n \to D_n$. Тогда для любого полинома $p(n)$ случайные величины*

$$h(\alpha), h(g(\alpha)), \ldots, h(g^{p(n)-1}(\alpha)), g^{p(n)}(\alpha) \quad \textit{и} \quad r_1, r_2, \ldots, r_{p(n)}, \alpha$$

вычислительно неотличимы (для наглядности мы опускаем индекс n). Здесь α равномерно распределена в D_n, а $r_1 r_2 \ldots r_{p(n)}$ случайная равномерно распределенная строка длины $p(n)$, независимая от α.

Доказательство. Сначала докажем утверждение для случаев $p(n) \equiv 1$ и $p(n) \equiv 2$. В первом случае вычислительная неотличимость случайных величин $h(\alpha)g(\alpha)$ и $r_1\alpha$ следует из Леммы 6: $h(\alpha)g(\alpha) \equiv r_1 g(\alpha) \equiv r_1\alpha$. Вторая из эквивалентностей имеет место, поскольку g перестановка.

Теперь докажем утверждение для $p(n) \equiv 2$. Надо установить неотличимость случайных величин $h(\alpha)h(g(\alpha))g^2(\alpha)$ и $r_1 r_2 \alpha$. Рассмотрим "промежуточную" случайную величину $r_1 h(\alpha)g(\alpha)$.

Докажем ее неотличимость от обеих случайных величин. Неотличимость от $r_1 r_2 \alpha$ следует из неотличимости $h(\alpha)g(\alpha)$ и $r_2\alpha$, данной нам в условии (обе случайные величины получаются приписыванием к этим случайным величинам одного бита).

Неотличимость случайных величин $h(\alpha)h(g(\alpha))g^2(\alpha)$ и $r_1 h(\alpha)g(\alpha)$ устанавливается немного сложнее. Рассмотрим преобразование T, которое переводит строку $r_1\alpha$ в строку $r_1 h(\alpha)g(\alpha)$. Это преобразование вычислимо схемами полиномиального размера. С другой стороны, это же преобразование переводит случайную величину $h(\alpha)g(\alpha)$ в случайную величину $h(\alpha)h(g(\alpha))g^2(\alpha)$. Поскольку случайные величины $r_1\alpha$ и $h(\alpha)g(\alpha)$ неотличнимы, будет неотличимы и результаты применения к ним преобразования T.

Теперь докажем утверждение в общем случае. Допустим последовательность схем C_n полиномиального размера отличает случайные величины

$$h(\alpha), h(g(\alpha)), \ldots, h(g^{p(n)-1}(\alpha)), g^{p(n)}(\alpha) \quad \text{и} \quad r_1, r_2, \ldots, r_{p(n)}, \alpha.$$

То есть, для бесконечно многих n вероятность того, что первая случайная величина проходит тест C_n, отличается не менее, чем на $\varepsilon_n = 1/\mathrm{poly}(n)$,

от вероятности того, что вторая случайная величина проходит тест C_n. Фиксируем любое такое n и для каждого $i = 0, \ldots, p(n)$ рассмотрим промежуточные величины

$$H_i = r_1, r_2, \ldots, r_i, h(\alpha), h(g(\alpha)), \ldots, h(g^{p(n)-1-i}(\alpha)), g^{p(n)-i}(\alpha).$$

Случайные величины H_0 и $H_{p(n)}$ совпадают со случайными величинами из условия леммы. Поэтому для некоторого i будет выполнено

$$|\Pr[C_n(H_i) = 1] - \Pr[C_n(H_{i+1}) = 1]| \geq \varepsilon_n / p(n).$$

Случайная величина H_{i+1} получена некоторым преобразованием T строки $r_1 r_2 \ldots r_i r_{i+1} \alpha$, вычисляемым схемой размера $\mathrm{poly}(n)$. Если это преобразование применить к строке $r_1 r_2 \ldots r_i h(\alpha) g(\alpha)$ то получится случайная величина H_i. Поэтому схема $C_n(T(r_1 r_2 \ldots r_i r_{i+1} \alpha))$ различает величины $r_1 r_2 \ldots r_i r_{i+1} \alpha$ и $r_1 r_2 \ldots r_i h(\alpha) g(\alpha)$, которые неотличимы, поскольку таковы $r_{i+1}\alpha$ и $h(\alpha)g(\alpha)$. □

Итак, мы почти доказали следующее утверждение.

Теорема 8. *Если существет одностороняя перестановка (или хотя бы обобщенная перестановка) и трудный бит для нее, то для некоторого полинома $q(n)$ существует генератор ПСЧ типа $q(n) \to \infty$.*

Доказательство. Сначала предположим, что D_n есть множество всех слов длины $k(n)$. Тогда в качестве генератора $q(n)$ можно взять $k(n)$ и положить

$$G_n(s) = h(s)h(g(s))h(g^2(s)) \ldots$$

По Лемме 7 это отображение является генератором ПСЧ. Действительно, лемма утверждает, что любое начало этой последовательности полиномиальной длины вычислительно неотличимо от равномерно распределенной последовательности той же длины.

В общем случае воспользуемся доступностью равномерного распределения на D_n. Рафиксируем вероятностнй алгоритм K, который по n за полиномиальное от n время генерирует случайную величину, которая статистически неотличима от случайной величины α_n, равномерно распределенной в D_n. Пусть K использует случайную строку s длины $q(n)$ и $K_n(s)$ обозначает выдаваемую им строку. Определим генератор G_n. Его значение на строке s длины $q(n)$ как

$$G_k(s) = h(K_n(s))h(g(K_n(s)))h(g^2(K_n(s))) \ldots$$

В этой последовательности l-ый бит равен $h(g^l(K_n(s)))$ и может быть вычислен за полиномиальное от $l+n$ время.

Докажем, что построенный генератор G_n надежен. Пусть $p(n)$ — произвольный полином. Случайная величина $K_n(s)$ неотличима от случайной величины α_n тестами размера $\mathrm{poly}(n)$. Поэтому и первые $p(n)$ битов $G_n(s)$ неотличимы от

$$h(\alpha_n)h(g(\alpha_n))h(g^2(\alpha_n))\ldots h(g^{p(n)-1}(\alpha_n))$$

тестами того же размера. По Лемме 7 последняя неотличима от равномерно распределенной последовательности тестами размера $\mathrm{poly}(n)$. □

Генератор, построенный в доказательстве теоремы 8 обладает двумя практически важными свойствами, не отраженным в определении. (1) По данному n и s биты последовательности $G_n(s)$ можно вычислять по очереди так, что на вычисление очередного бита уходит время, ограниченное некоторым полиномом от n, не зависящим от номера этого бита. Действительно, вычислив l первых битов $G_n(s)$, мы сохраняем слово $g^l(s)$. Для вычисления очередного бита надо применить к сначала функцию g, а потом к полученному слову применить функцию h. (2) Для любого полинома p любые $p(n)$ подряд идущих битов (а не только первые) сгенерированной последовательности вычислительно неотличимы от равномерно распределенной последовательности длины $p(n)$.

Замечание. В каком месте доказательства Леммы 7 мы использовали то, что g_n является перестановкой? Ровно в одном месте, а именно, когда использовали вычислительную неотличимость случайных величин α_n и $g_n(\alpha_n)$. Поэтому в условии леммы требование инъективности g_n можно заменить на более слабое требование: случайные величины α_n и $g_n(\alpha_n)$ вычислительно неотличимы (где α_n) равномерно распределена в D_n). Будем функции g_n удовлетворяющие этому ослабленному требованию называть *обобщенными перестановками.*

Итак, мы получаем следующее утверждение.

Теорема 9. *Если существет одностороняя обобщенная перестановка и трудный бит для нее, то для некоторого полинома $q(n)$ существует генератор ПСЧ типа $q(n) \to \infty$.*

Из этой теоремы мы получаем следующее следствие:

Теорема 10. *Если для некоторой полиномиально вычислимой функции $k(n)$, ограниченной сверху полиномом, существует генератор G_n типа $k(n) \to k(n)+1$, то для некоторого полинома $q(n)$ существует генератор ПСЧ типа $q(n) \to \infty$.*

Доказательство. Обозначим через $h_n(s)$ первый бит $G_n(s)$, а через $g_n(s)$ остальные $k(n)$ битов. Мы получили полиномиально вычислимую обобщенную перестановку $g_n : \{0,1\}^{k(n)} \to \{0,1\}^{k(n)}$ вместе с трудным битом для нее. Осталось воспользоваться Теоремой 9. □

Задача 29. Докажите, что если для некоторого полинома $q(n)$ существует генератор ПСЧ типа $q(n) \to \infty$, то существует и генератор ПСЧ типа $n \to \infty$.

2.4 Отступление: надежность генераторов по Яо

В некоторых приложениях генераторов псевдослучайных чисел нам всего лишь нужно, чтобы невозможно было предсказывать никакой бит сгенерированной последовательности по ее предыдущим битам с вероятностью, существенно отличающейся от $1/2$ (например, в казино). Генераторы, удовлетворяющие этому требованию называются надежными по Яо (Yao).

Определение. Пусть имеется последовательность случайных величин $\alpha^1, \alpha^2, \alpha^3, \ldots$ и каждая α^n принимает значения в в множестве бесконечных последовательностей нулей и единиц. Мы будем говорить, что α^n неотличима по Яо от равномерно распределенной последовательности, если для любой последовательности индексов $i_n \leq \mathrm{poly}(n)$ случайная величина $\alpha^n_{i_n+1}$ (i_n+1-ый бит α^n) является трудно вычислима по случайной величине $(\alpha^n)_{i_n}$ (начало длины i_n последовательности α^n). [2] Генератор типа $n \to \infty$ называется *надежным по Яо*, если генерируемая им случайная величина неотличима по Яо от равномерно распределенной последовательности.[3]

Если случайная величина вычислительно неотличима от равномерно распределенной случайной величины, то она неотличима от нее и по Яо. Действительно, по лемме 6 случайная величина $(\alpha^n)_{i_n+1}$ трудно вычислима по случайной величине $(\alpha^n)_{i_n}$ тогда и только тогда, когда случайные

[2] Можно дать определение неотличимости по Яо любых двух случайных величин в множестве слов одной длины, но оно нам не понадобится.

[3] Аналогично дается определение надежности по Яо генераторов вида $k(n) \to l(n)$.

величины $(\alpha^n)_{i_n+1}$ и $(\alpha^n)_{i_n} r$ вычислительно неотличимы. А последнее верно, поскольку обе случайные величины неотличимы от случайной величины, от равномерно распределенной среди слов длины $i_n + 1$. Верно и обратное.

Теорема 11. *Если случайная величина α^n в множестве бесконечных последовательностей нулей и единиц неотличима по Яо от равномерно распределенной случайной величины, то она вычислительно неотличима от нее. Следовательно, любой надежный по Яо генератор надежен в обычном смысле.*

Доказательство. Пусть C_n произвольная последовательность схем размера $\mathrm{poly}(n)$. Нам надо доказать, что C_n не отличает α^n от равномерно распределенной случайной величины. Обозначим через $l(n)$ количество входов C_n. Фиксируем произвольное n и для каждого $0 \le i \le l(n)$ рассмотрим случайную величину H_i, которая получится, если в последовательности α^n заменить все биты, начиная с $i+1$-ого, на случайные биты, независимые от α^n. Ясно, что начало $H_{l(n)}$ длины $l(n)$ распределено так же, как начало α^n длины $l(n)$, а H_0 равномерно распределена. Обозначим через i_n тот индекс, для которого $|\Pr[C_n(H_i) = 1] - \Pr[C_n(H_{i+1}) = 1]|$ максимально. Сравним случайные величины H_i и H_{i+1}. У них первые i битов равны i первым битам α^n. Последние $n - i - 1$ битов выбираются случайно. Разница только в $i+1$-ом бите, который в H_i выбирается случайным образом, а в H_{i+1} берется из α^n. Поскольку α^n неотличима по Яо от равномерно распределенной последовательности, α_{i_n+1} (верхний индекс n опускаем) невозможно вычислить по $(\alpha)_{i_n}$ с вероятностью, существенно отличающейся от $1/2$. По лемме 6 случайные величины $(\alpha)_{i_n+1}$ и $(\alpha)_{i_n} r$ вычислительно неотличимы (здесь r случайный равномерно распределенный бит, независимый от α). Значит и случайные величины H_{i_n} и H_{i_n+1} вычислительно неотличимы (они получаются приписыванием к $(\alpha)_{i_n+1}$ и $(\alpha)_{i_n} r$, соответственно, случайной независимой последовательности). Следовательно, величина $|\Pr[C_n(H_{i_n}) = 1] - \Pr[C_n(H_{i_n+1}) = 1]|$ стремится к нулю быстрее любого обратного полинома от n. Поскольку величина $|\Pr[H_0) = 1] - \Pr[C_n(H_{l(n)}) = 1]|$ не более, чем в $l(n)$ раз превосходит эту, то и она стремится к нулю быстрее любого обратного полинома от n. □

2.5 Построение функции, являющейся трудным битом

Пусть $f_n : D_n \to D_n$ необратимая обобщенная перестановка и $D_n \subset \{0,1\}^{l(n)}$. Рассмотрим функции g_n, h_n определенные на множестве $D_n \times \{0,1\}^{l(n)}$ равенствами $g_n(xy) = f_n(x)y$ и $h_n(xy) = x \odot y$. Здесь выражение xy обозначает конкатенацию слов x, y длины $l(n)$, а $x \odot y = \sum_{i=1}^{l(n)} x_i y_i \bmod 2$. Ясно, что функции g_n и h_n полиномиально вычислимы и g_n является обобщенной перестановкой. Теорема Гольдрайха—Левина утверждает, что h_n является трудным битом для нее.

Теорема 12 (Гольдрайха—Левина)**.** *Если f_n сильно необратима, то функция $x \odot y$ является трудным битом для функции $g_n(x,y) = f_n(x)y$.*

Доказательство основано на возможности вероятностного декодирования так называемых кодов Адамара. *Кодом Адамара* слова x длины m называется последовательность длины 2^m, состоящая из значений $x \odot y$ для всех слов y длины m. Ясно, что по коду Адамара можно восстановить само слово — i-ый бит x есть скалярное произведение x с i-ым единичным вектором e_i. Нам понадобится способность кода Адамара "исправлять ошибки". Это означает, что если в коде Адамара слова x изменить не более чем $(1/2 - \varepsilon)2^m$ битов, то по полученной строке можно отыскать список из $\mathrm{poly}(m/\varepsilon)$ строк, среди которых есть x.

Лемма 13. *Существует вероятностный алгоритм, который, для любого слова x длины m и любого ε, получив на вход m и ε, и имея в качестве оракула любую функцию $s(y)$ такую, что $Pr[s(y) = x \odot y] \geq 1/2 + \varepsilon$, за полиномиальное от $m, 1/\varepsilon$ время с вероятностью не менее $1/2$ находит некоторый список слов (размера $poly(m, 1/\varepsilon)$), содержащий x.*

Эту лемму мы докажем позже. А сейчас завершим доказательство теоремы, используя ее.

Допустим противное — для бесконечно многих n существует схема C_n размера $\mathrm{poly}(n)$ такая, что вероятность события $C_n(f(x)y) = x \odot y$ не меньше $1/2 + \varepsilon_n$, где $\varepsilon_n = 1/\mathrm{poly}(n)$ (слово x выбирается равномерно из D_n, слово y выбирается равномерно из $\{0,1\}^{l(n)}$ независимо от x). Зафиксируем некоторое n из этого бесконечного множества и построим вероятностную схему полиномиального от n размера, которая будет обращать $f(x)$ с вероятностью не меньше $\varepsilon_n/4$. Эта схема работает так:

она запускает алгоритм из леммы 13, используя в качестве внешней процедуры $s(y) = C_n(f(x)y)$, и $\varepsilon_n/2$ в качестве ε. Затем для каждого из $\tilde{x}$, выданных алгоритмом, схема вычисляет $f(\tilde{x})$, сравнивая полученное значение с $f(x)$ и выдает первое $\tilde{x}$, для которого $f(\tilde{x}) = f(x)$. Если такого $\tilde{x}$ в списке не нашлось, то схема выдает, что угодно.

Оценим вероятность с которой эта схема правильно обращает $f(x)$. Обозначим через $p(x)$ вероятность $\Pr[C_n(f(x)y) = x \odot y]$. По лемме 13 схема обратит $f(x)$ с вероятностью не меньше $1/2$ для любого такого x, что $p(x) \geq 1/2 + \varepsilon_n/2$. Поэтому достаточно доказать, что с вероятностью не меньше $\varepsilon_n/2$ выполнено $p(\alpha) \geq 1/2 + \varepsilon_n/2$. Это легко доказать от противного. Действительно, по условию среднее значение величины $p(x)$ не меньше $1/2 + \varepsilon_n$. Разделим все x на те, для которых $p(x) \geq 1/2 + \varepsilon_n/2$, и остальные. Если бы x попадало в первый класс с вероятностью меньше $\varepsilon_n/2$, то среднее значение величины $p(x)$ было бы меньше $(\varepsilon_n/2) \cdot 1 + 1 \cdot (1/2 + \varepsilon_n/2) = 1/2 + \varepsilon_n$ (первое слагаемое в этой сумме ограничивает сверху вклад в среднее всех x из первой группы, второе слагаемое — вклад всех остальных).

Теперь нам нужно доказать Лемму 13.

2.6 Доказательство Леммы 13

Декодирование кода Адамара, испорченного менее чем в четверти битов

Сначала объясним, как можно за экспоненциальное от n (длины слова x) исправлять код Адамара слова x, испорченный менее чем в четверти битов. Пусть z неиспорченный код Адамара слова x, то есть слово длины 2^n, у которого бит номер y равен $x \odot y$. Пусть $\tilde{z}$ — испорченный код Адамара, то есть некоторое слово, которое отличается от z менее, чем в чевтерти позиций. Нам дано $\tilde{z}$ и надо найти x.

Будем находить биты слова x по очереди. Для того, чтобы найти i-ый бит x_i, перенумерум мысленно биты слова z следующим образом. Бит номер y теперь будет равен $x \odot (y \oplus e_i)$, а не $x \odot y$, как раньше. (Напомним, что e_i обозначает слово, имеющее в i-ой позиции, и нули в остальных.) Обозначим получившееся слово через z^i. Нетрудно понять, что

$$z^i(y) = x \odot (y \oplus e_i) = (x \odot y) \oplus (x \odot e_i) = (x \odot y) \oplus x_i.$$

Поэтому, если i-ый бит слова x равен нулю, то $z^i = z$, а иначе $z^i = \bar{z}$ (так

мы обозначили слово z, в котором все биты инвертированы). Обозначим через $\tilde{z}^i$ результат аналогичной перенумерации битов слова $\tilde{z}$. Поскольку слово $\tilde{z}$ отличается от слова z менее чем в четверти позиций, для него верно следующее. Если i-ый бит слова x равен нулю, то $\tilde{z}^i$ отличается от $\tilde{z}$ менее чем в половине позиций, а иначе они отличаются более чем в половине позиций. Поэтому, сравнив $\tilde{z}^i$ и $\tilde{z}$ мы сможем понять, чему равен i-ый бит x.

Декодирование списком кода Адамара, испорченного менее чем в половине битов, за экспоненциальное время

Пусть теперь нам известно, что слово $\tilde{z}$ отличается от кода Адамара z слова x не более чем в $(1/2-\varepsilon)2^n$ позиций. Объясним, как за экспоненциальное от n время найти список размера $\mathrm{poly}(1/\varepsilon)$, содержащий x. Ясно, что надо найти список всех слов x' коды Адамара которых отличаются от $\tilde{z}$ не более чем в $(1/2-\varepsilon)2^n$ позиций (слово x содержится в этом списке, и обратно, любое слово из этого списка могло бы быть исходным x). Поэтому нам необходимо и достаточно убедиться, что этот содержит $\mathrm{poly}(1/\varepsilon)$ слов. Мы докажем, что в нем не более $(1/2\varepsilon)^2$ слов.

Для этого рассмотрим на линейном пространстве всех функций из $\{0,1\}^n$ в $\mathbb{R}$ скалярное произведение (r,t) определяемое как среднее значение случайной величины $r(y)t(y)$:

$$(r,t) = 2^{-n}\sum_y r(y)t(y).$$

(Нетрудно проверить, все аксиомы скалярного произведения выполнены.) Сделаем в коде Адамара слова x замены $1 \to -1$ и $0 \to 1$. То есть отныне будем считать, что y-ый бит кода Адамара слова x равен $(-1)^{x\odot y}$. Что можно сказать о скалярном произведении z и $\tilde{z}$ (биты $\tilde{z}$ заменяются аналогичным образом)? В побитовом произведении z и $\tilde{z}$ будет не более чем $(1/2-\varepsilon)2^n$ минус единиц, а остальные члены побитового произведения будут равны 1. Поэтому

$$(z,\tilde{z}) \geq (1/2+\varepsilon) - (1/2-\varepsilon) = 2\varepsilon.$$

Алгоритм из предыдущего параграфа (восстановления кода Адамара, испорченного менее чем в четверти битов) можно изложить в этих терминах так. Инвертирование менее четверти битов в одном из векторов

u, v меняет скалярное произведение (u, v) менее чем на $1/2$. Поэтому скалярное произведение $(\tilde{z}, \tilde{z}^i)$ отличается от (z, z^i) менее чем на 1. Последнее же равно $+1$ или -1 в зависимости от x_i. Поэтому по знаку $(\tilde{z}, \tilde{z}^i)$ можно найти x_i.

Нетрудно проверить, что коды Адамара всевозможных слов длины n образуют ортонормированный базис в пространстве $\mathbb{R}^{2^n}$: они попарно ортогональны, скалярный квадрат каждого из них равен 1, а их количество равно размерности пространства. Поэтому для любой функции r сумма квадратов скалярных произведений r с этими функциями равна скалярному квадрату r. В частности, это верно и для $\tilde{z}$:

$$\sum_z (\tilde{z}, z)^2 = (\tilde{z}, \tilde{z})^2 = 1.$$

(Здесь z пробегает коды Адамара всех строк длины n.) Следовательно, количество z, для которых $(z, \tilde{z}) \geq 2\varepsilon$, не превосходит $1/(2\varepsilon)^2$. Все такие z можно найти перебором за время $2^{O(n)}$ и среди них присутствует код Адамара исходной строки.

Полиномиальное декодирование списком кода Адамара, испорченного менее чем в половине битов

В доказательстве нам понадобится способ "дешево" производить много попарно независимых равномерно распределенных векторов длины n. Для этого рассмотрим булеву матрица A размера $n \times l$. Матрица A задает линейное отображение $y \mapsto Ay$ из l-мерного в n-мерное линейного пространства над полем вычетов по модулю 2 (все операции выполняются в поле вычетов по модулю 2).

Лемма 14. *Пусть $y_1 \neq y_2$ два различных ненулевых вектора. Тогда случайные величины Ay_1 и Ay_2 независимы и равномерно распределены среди строк длины k (если A выбирается случайно по равномерному распределению).*

Доказательство. Равномерная распределенность Ay очевидна. Докажем независимость. Отображение $A \mapsto (Ay_1, Ay_2)$ является линейным отображением из пространства матриц размера $k \times l$ в пространств матриц размера $k \times 2$. Поэтому все элементы в образе этого отображения имеют одинаковую кратность (равную мощности его ядра). Значит, нам достаточно доказать его сюръективность. Нетрудно видеть, что это отображение состоит в умножении справа на матрицу M, состоящую из столбцов

y_1, y_2. Нам нужно доказать, что ранг этой матрицы равен 2. Это следует из того, что столбцы этой матрицы различны и не равны нулю. □

Алгоритм декодирования списком основан на той же идее, что и алгоритм исправления менее чем четверти ошибок. Каждый вектор в списке мы будем искать бит за битом. При изложении алгоритма мы опять будем использовать скалярное произведение, введеное в предыдущем параграфе.

Пусть z — код Адамара слова x (составленный из плюс-минус единиц). Пусть $\tilde{z}$ отличается от z инвертированием не более, чем $(1/2-\varepsilon)2^n$ битов, то есть, $(z, \tilde{z}) \geq 2\varepsilon$. Нетрудно проверить, что $(z, \tilde{z}^i) = (-1)^{x_i}(z, \tilde{z})$. (Действительно, $(z, \tilde{z}^i) = (z^i, \tilde{z}) = ((-1)^{x_i} z, \tilde{z}) = (-1)^{x_i}(z, \tilde{z})$.) Поэтому, для нахождения x_i достаточно вычислить $(z, \tilde{z}^i)$ с точностью 2ε: если полученное число отрицательно, то $x_i = 1$, а если положительно, то $x_i = 0$.

Как нам найти $(z, \tilde{z}^i)$ с нужной точностью? Проблемы здесь две: во-первых, мы не знаем z, во-вторых, даже при известном z вычисление скалярного произведения требует экспоненциального времени. Вторую проблему решить несложно: ведь нам не нужно точно вычислять $(z, \tilde{z}^i)$. Можно оценить $(z, \tilde{z}^i)$ вероятностным алгоритмом с помощью закона больших чисел. Для этого рассмотрим величину

$$S_i = \sum_{j=1}^{N} z(y_j)\tilde{z}^i(y_j),$$

где $y_1, \ldots, y_N$ выбираются случайно среди строк длины n. Насколько большим надо взять N, чтобы с большой вероятностью это приближение было 2ε-близко к $(z, \tilde{z}^i)$? Для ответа на этот вопрос нам достаточно следующей формы закона больших чисел.

Лемма 15. *Если случайные величины $\xi_1, \ldots, \xi_N$ попарно независимы и имеют одинаковое распределение, то для всех положительных δ вероятность события*

$$\left| \frac{\xi_1 + \cdots + \xi_N}{N} - \mathbf{E}\,\xi_i \right| \geq \delta$$

не превосходит $\mathbf{D}\,\xi_i/(N\delta^2)$. Здесь $\mathbf{E}\,\xi_i$ и $\mathbf{D}\,\xi_i$ обозначают математическое ожидание и дисперсию случайной величины ξ_i (по условию они не зависят от i).

Доказательство. Из попарной независимости следует, что дисперсия случайной величины $\xi_1 + \cdots + \xi_N$ равна сумме дисперсий $\xi_1, \ldots, \xi_N$, то есть, $N\,\mathbf{D}\,\xi_i$. Поскольку случайные величины $\xi_1, \ldots, \xi_N$ одинаково распределены, среднее значение суммы $\xi_1 + \cdots + \xi_N$ равно $N\,\mathbf{E}\,\xi_i$. Поэтому, если указанное в лемме событие имеет место, то величина $(\xi_1 + \cdots + \xi_N - N\,\mathbf{E}\,\xi_i)^2$ не меньше $\delta^2 N^2$, а значит превосходит свое среднее значение в $N\delta^2/\,\mathbf{D}\,\xi_i$ раз. Вероятность этого не превосходит $\mathbf{D}\,\xi_i/(N\delta^2)$. □

Мы будем выбирать $y_1, \ldots, y_N$ некоторым хитрым образом. При этом они будут попарно независимы и каждое y_j будет равномерно распределено среди строк длины n. Поэтому среднее значение случайной величины $\xi_j = z(y_j)\tilde{z}^i(y_j)$ равно (z, z^i). Дисперсия этой величины не превосходит 1 (потому что она принимает значения ± 1). Поэтому вероятность события $|S_i - (z, \tilde{z}^i)| \geq 2\varepsilon$ не превосходит $1/(4N\varepsilon^2)$. Положим $N = \lceil n/2\varepsilon^2 \rceil$. Тогда вероятность этого события будет не больше $1/(2n)$. Значит, с вероятностью не менее $1/2$ для всех $i = 1, \ldots, n$ будет выполнено $|S_i - (z, \tilde{z}^i)| < 2\varepsilon$.

Теперь мы укажем способ выбора $y_1, \ldots, y_N$. Положим $l = \lceil \log_2(N + 1) \rceil$ и пусть $u_1, \ldots, u_N$ первые N ненулевых векторов l-мерного пространства, занумерованные в любом порядке. Возьмем случайную матрицу A размера $n \times l$ и положим $y_j = Au_j$. По Лемме 14 вектора $y_1, \ldots, y_N$ попарно независимы и равномерно распределены среди строк длины n. А следовательно, по Лемме 15 с вероятностью не менее $1/2$ для всех i выполнено $|S_i - (z, \tilde{z}^i)| < 2\varepsilon$.

Осталось только понять, как можно для данных $y_1, \ldots, y_N$ вычислить S_i, не зная z. Здесь нам приходит на помощь особый способ вычисления y_j. Вспомним, что $y_j = Au_j$, то есть k-ый бит y_j получается умножением k-ой строки A_k матрицы A на вектор u_j, что можно записать, пользуясь нашими обозначениями, как $A_k \odot u_j$. Поэтому

$$z(y_j) = (-1)^{x \odot y_j} = (-1)^{\sum_{k=1}^{n} x_k (A_k \odot u_j)} = (-1)^{(\sum_{k=1}^{n} x_k A_k) \odot u_j}.$$

В этом выражении $\sum_{k=1}^{n} x_k A_k$ есть неизвестный нам вектор длины l. Всего имеется 2^l различных векторов длины l, что меньше $2N$. Поэтому нам не нужно знать z — мы можем перебрать все возможные значения вектора $\sum_{k=1}^{n} x_k A_k$.

Точнее, пусть w — произвольный вектор длины l. Обозначим через

$S_i(A,w)$ результат подстановки $z(y_j) \to (-1)^{w \odot u_j}$ и $y_j \to Au_j$ в сумму

$$S_i = \sum_{j=1}^{N} z(y_j)\tilde{z}^i(y_j).$$

То есть,

$$S_i(A,w) = \sum_{j=1}^{N} (-1)^{w \odot u_j} \tilde{z}^i(Au_j).$$

С вероятностью не менее $1/2$ для некоторого w (а именно, для $w = \sum_{k=1}^{n} x_k A_k$) для всех $i = 1, \ldots, n$ будет выполнено $|S_i(A,w) - (z, \tilde{z}^i)| < 2\varepsilon$. Поэтому можно действовать так: выбираем случайно матрицу A; затем для каждого l-мерного вектора w делаем следующее. Вычисляем суммы $S_i(A,w)$ для всех $i = 1, \ldots, n$ и по знаку каждой суммы формируем вектор $x = x(w)$. Затем выдаем список, состоящий из векторов $x(w)$.

3 Шифрование с закрытым ключом

Схема шифрования с закрытым ключом состоит из доступной последовательности e_n случайных величин (на строках полиномиальной от n длины) вероятностного полиномиального алгоритма E, называемого алгоритмом *шифровки* и детерминированного полиномиального алгоритма D, называемого алгоритмом *расшифровки*. Случайная величина e_n называется *ключом*,

Алгоритмы E и D получают на вход натуральное число n и пару двоичных строк и выдают на выход одну двоичную строку. Точнее, алгоритм E получает на вход параметр безопасности n, ключ e_n и *сообщение* $x \in \{0,1\}^*$ и выдает некоторую строку $y = E_n(e_n, x)$, называемую *шифрограммой*, длина которой зависит только от n и длины x. Алгоритм D получает на вход n, y и e_n и выдает строку $D_n(e_n, y)$, которая должна совпадать с x. Оба алгоритма E, D должны работать полиномальное от n и длины x время (какие бы случайные биты ни получил E).

При этом, при любом фиксированном x и случайно выбранном ключе e_n шифрограмма $E_n(e_n, x)$ не должна нести никакой информации об x. Эти требования уточняются следующим образом.

(1) Правильность расшифровки: для всех n, x с вероятностью 1 выполнено $D(e, E(e,x)) = x$. (Для упрощения обозначений мы опускаем

индекс n.) Источником случайности здесь является не только случайная величина e, но и случайные биты алгоритма E. Мы предполагаем, что случайные биты алгоритма E независимы от e.

(2) Неразглашение информации: для любого полинома $p(n)$ и для любых двух последовательностей сообщений a_n, b_n таких, что длины a_n, b_n равны $p(n)$, случайные величины $E(e, a_n)$ и $E(e, b_n)$ вычислительно неотличимы. Источник случайности в этом определении имеется в e и в случайных битах алгоритма E, которые предполагаются независимыми от e.

Условие (2) очевидно эквивалентно своему частному случаю: неотличимости случайных величин $E(e, a_n)$ и $E(e, 00\ldots0)$ (подряд n нулей). Вторую из этих случайных величин можно генерировать, зная только n. Поэтому, какое бы сообщение мы не зашифровывали случайно выбранным ключом, шифрограмма не будет нести никакой информации — некоторую неотличимую от нее случайную величину можно сгенерировать и не зная исходного сообщения. Заметим, что в наших определениях важно, что ключ выбирается случайно, а не фиксирован. Мы не можем требовать неотличимости шифрограмм $E(e, a_n)$ и $E(e, b_n)$ при любом фиксированном ключе e, поскольку любой фиксированный ключ e можно "запаять" в схему-отличитель, которая будет применять алгоритм D и, скажем, выдавать первый бит расшифрованного сообщения.

Из условия (2) следует, что, не зная ключа, никакой бит исходного сообщения нельзя найти по шифрограмме с вероятностью существенно больше $1/2$, если сообщение выбирается случайно с равномерным распределением среди всех строк какой-то фиксированной полиномиальной от n длины k. Действительно, пусть противник C (схема полиномиального от n размера) пытается по шифрограмме восстановить, скажем, первый бит сообщения. Из условия (2) следует, что вероятности событий $C(E(e, 0a)) = 1$ и $C(E(e, 1a)) = 1$ (где a случайная строка длины $k-1$) приблизительно равны. Другими словами сумма вероятностей событий $C(E(e, 0a)) = 0$ и $C(E(e, 1a)) = 1$ приблизительна равна 1, следовательно, вероятность события $C(E(e, b)) = b[1]$ (где b случайная строка длины k) примерно равна $1/2$.

Задача 30. Пусть противник интересуется некоторой информацией $f(x)$ о передаваемом сообщении x (где $f : \{0,1\}^* \to \{0,1\}^*$ — некоторая функция). Допустим противник применяет к шифрограмме $E(e, x)$ некоторую схему D_n полиномиального от n размера для извлечения этой информа-

ции. Докажите, что существует доступная случайная величина α_n такая, что для любой последовательности сообщений x_n полиномиальной длины, вероятности событий $D_n(E(e, x_n)) = f(x_n)$ (вероятность успеха атаки) и $\alpha_n = f(x_n)$ приблизительно равны. То есть, информация $f(x_n)$ может быть вычислена и без подслушивания с примерно той же вероятностью успеха.

Пара алгоритмов E, D и последовательность случайных величин e_n называется *схемой шифрования с закрытым ключом*, если выполнены все перечисленные условия. Схему шифрования с закрытым ключом можно построить на основе любого генератора ПСЧ.

Теорема 16. *Если существует генератора G_n, то существует и схема шифрования с закрытым ключом.*

Доказательство. Сначала построим схему шифрования с закрытым ключом, для которой свойство (2) выполнено только для какого-нибудь одного фиксированного полинома $p(n)$ (а не для всех полиномов, как требуется). Эта схема называется *гаммированием* (one-time pad). В качестве ключа возьмем случайную строчку γ_n длины $p(n)$ и положим $E(\gamma_n, x) = \gamma_n \oplus x$ (побитовое сложение по модулю 2) и $D(\gamma_n, y) = \gamma_n \oplus y$. (Если длина сообщения не равна $p(n)$, то положим $E(\gamma_n, x) = x$, $D(\gamma_n, y) = y$; второе условие в этом случае, разумеется, не выполнено, но это и не требуется.) Условие (1) очевидно выполнено. В условии (2) вероятность события $C_n(\gamma_n \oplus a_n) = 1$ равна вероятности события $C_n(\gamma_n) = 1$, а потому не зависит от a_n. При этом нам даже не важно, что схема C_n имеет полиномиальный размер — случайные величины $\gamma_n \oplus a_n$ и γ_n имеют одинаковое распределение.

Недостатком гаммирования является то, что длина ключа равна длине сообщения, поэтому на практике его можно применять только для шифрования коротких сообщений.

Теперь модифицируем схемму гаммирования, заменив случайную величину γ_n на псевдослучайную величину, порожденную генератором. Ключ s равномерно распределен среди строк длины n. Алгоритм $E(s, x)$ является детерминированным и выдает побитовую сумму x и первых $|x|$ битов последовательности $G_n(s)$. Будем обозначать выданную им последовательность через $x \oplus G(s)$. Алгоритм $D(s, y)$ выдает $y \oplus G(s)$, побитовую сумму y и первых $|y|$ битов последовательности $G(s)$. Ясно, что условие (1) выполнено. Для проверки условия (2) фиксируем произволь-

ный полином $p(n)$. Нам надо доказать неотличимость случайных величин $a_n \oplus G(s)$ и $b_n \oplus G(s)$, где a_n, b_n любые последовательности слов длины $p(n)$. Поскольку функция G является генератором, случайная величина $G(s)[1 : p(n)]$ вычислительно неотличима от случайной величины $\gamma_{p(n)}$, равномерно распределенной среди слов длины $p(n)$. Значит, случайная величина $a_n \oplus G(s)$ неотличима от случайной величины $a_n \oplus \gamma_{p(n)}$. В самом деле, если бы они были отличимы тестом $T_n(y)$ размера $\mathrm{poly}(n)$ то тест $T_n(a_n \oplus y)$, размер которого лишь немного превышает размер T_n, отличал бы $G(s)$ и γ). Случайная величина $a_n \oplus \gamma_{p(n)}$ имеет равномерное распределение, значит случайная величина $a_n \oplus G(s)$ неотличима от случайной величины $\gamma_{p(n)}$. То же самое справедливо для случайной величины $b_n \oplus G(s)$, значит она неотличима от $a_n \oplus G(s)$. □

4 Шифрование с открытым ключом

В схеме шифрования с открытым ключом имеется два ключа — открытый и закрытый. Открытый ключ используется для шифрования, а закрытый для расшифровки. От схемы шифрования требуется, чтобы противник из открытого ключа и шифрограммы любого сообщения не смог получить никакой информации о сообщении. Точнее для любых двух сообщений x_1, x_2 одной длины случайная величина, состоящая из открытого ключа и шифрограммы сообщения x_1, полученной с помощью этого ключа, была вычислительно неотличима от случайная величины, состоящей из открытого ключа и шифрограммы сообщения x_2, полученной с помощью этого ключа. Открытый ключ можно опубликовать, тогда любой может посылать зашифрованные сообщения, расшифровывать которые может только владелец закрытого ключа.

Схема шифрования с открытым ключом состоит из полиномиального вероятностного алгоритма E и полиномиального детерминированного алгоритма D, называемых, соответственно, алгоритмами *шифровки* и *расшифровки* и доступной последовательности случайных величин $\langle e_n, d_n \rangle$ (то есть, случайная величина $e_n d_n$ доступна; без ограничения общности можно предполагать, что длины e_n и d_n одинаковы, так что по $e_n d_n$ можно найти e_n и d_n). Строки e_n и d_n называются, соответственно, *открытым и закрытым ключами.*

Алгоритм E получает на вход параметр безопасности n, ключ e_n и *сообщение* $x \in \{0,1\}^*$ и выдает некоторую строку $y = E_n(e_n, x)$, назы-

ваемую *шифрограммой*, длина которой зависит только от длин n и x. Алгоритм D получает на вход n, y и d и выдает строку $D_n(d_n, y)$, которая должна совпадать с x. При этом, при любом фиксированном x и случайно выбранном ключе e_n шифрограмма $E_n(e_n, x)$ вместе с ключом e_n не должна нести никакой информации об x. Эти требования уточняются следующим образом.

(1) Правильность расшифровки: для всех n, x с вероятностью приблизительно равной 1 выполнено $D(e, E(d, x)) = x$ (то есть, вероятность отличичается от 1 на пренебрежимо малую величину). (Для упрощения обозначений мы опускаем индекс n.) Источником случайности здесь является не только случайная величина $\langle e, d\rangle$, но и случайные биты алгоритма E. Мы предполагаем, что случайные биты алгоритма E независимы от $\langle e, d\rangle$.

(2) Для любого полинома $p(n)$ и для любых двух последовательностей сообщений a_n, b_n таких, что длины a_n, b_n равны $p(n)$, случайные величины $\langle e, E(e, a_n)\rangle$ и $\langle e, E(e, b_n)\rangle$ вычислительно неотличимы. Источником случайности здесь является не только случайная величина $\langle e, d\rangle$, но и случайные биты алгоритма E. Мы предполагаем, что случайные биты алгоритма E независимы от $\langle e, d\rangle$.

Пара алгоритмов E, D и последовательность случайных величин $\langle e_n, d_n\rangle$ называется *схемой шифрования с открытым ключом*, если выполнены все перечисленные условия. Для существования схемы шифрования с открытым ключом требуется более сильная гипотеза, чем существование генератора. Например, удается построить такую схему при условии существования так называемой односторонней функции с секретом.

5 Односторонние функции с секретом

На самом деле речь идет не об одной функции, а о семействе функций. Грубо говоря, необратимой функцией с секретом (trapdoor function) называется доступная последовательность случайных величин $\langle e_n, d_n\rangle$ и семейство перестановок f_e такие, что функция $\langle e, x\rangle \mapsto \langle e, f_e(x)\rangle$ односторонняя, но при этом существует полиномиальный алгоритм, который по $f_e(x)$ и d находит x (в частности, все функции f_e инъективны).

Теперь формальное определение. Пусть задана доступная последовательность случайных величин $\langle e_n, d_n\rangle$ со значениями в множестве $A_n \times B_n$ (A_n и B_n некоторое множество слов, длина которых равна некоторому

полиному от n; то же самое верно и для B_n). Пусть для каждого $e \in A_n$ имеется функция f_e из некоторого множества слов D_e длины $p(n)$, в множество слов длины $p(n)$, где $p(n)$ некоторый полином. Пусть задана семейство случайных величин α_e, $e \in A_n$, $n \in \mathbb{N}$, со значениями в D_e. Последовательноть троек $\langle\langle e_n, d_n\rangle, \{f_e\}, \alpha_n\rangle$ называется односторонней функцией с секретом, если выполнены следующие условия.

(1) Функция $\langle e, x\rangle \mapsto f_e(x)$ вычислима за полиномиальное от n время (существует полиномиальный алгоритм, который по любой паре $\langle e, x\rangle$, где $e \in A_n$, $x \in D_e$, вычисляет $f_e(x)$).

(2) Для любой последовательности схем C_n размера $\mathrm{poly}(n)$ вероятность того, что C_n по $\langle e, f_e(\alpha_e)\rangle$ найдет α_e стремится к нулю быстрее любого обратного полинома от n.

(3) Семейство α_e доступно равномерно по e: существует вероятностный полиномиальный алгоритм, который по любому $e \in A_n$ за полиномиальное от n время генерирует случайную величину X'_e, статистически неотличимую от α_n. Это означает, что статистическое растояние между α_e и X'_e меньше некоторой пренебрежимо малой последовательности α_n для всех $e \in A_n$ (при стремлении n к бесконечности). Из условий (1), (2) и (3) следует, что функция $\langle e, x\rangle \mapsto \langle e, f_e(x)\rangle$ односторонняя.

(4) Случайные величины $\langle e, \alpha_e\rangle$ и $\langle e, f_e(\alpha_e)\rangle$ вычислительно неотличимы.

(5) Существует алгоритм, который за полиномиальное от n время по паре $\langle d, f_e(\alpha_e)\rangle$ с вероятностью приблизительно равной 1 вычисляет α_e.

Поскольку неизвестно, существуют ли односторонние функции, тем более неизвестно, существуют ли односторонние функции с секретом. Гипотеза об их существовании почти столь же правдоподобна, как и гипотеза о существовании необратимых перестановок. Действительно, если преположительно трудная случайная величина для функции Рабина в самом деле трудна для нее, или преположительно трудная случайная величина для функции RSA, в самом деле трудна для нее, то существуют односторонние функции с секретом. Убедимся в этом.

Функция Рабина. Закрытый ключ d_n равномерно распределен среди пар n-битовых простых чисел p, q вида $4k + 3$. Открытый ключ e_n равен произведению pq этих чисел. Множество D_e есть множество всех квадратичных вычетов по модулю pq, взаимно простых с pq, при этом $f_e(x) = x^2 \bmod pq$. Случайная величина α_e равномерна распределена в D_e. Условия (1), (3), (4), (5) очевидно выполнены. Условие (2) будет выполнено выполнено если преположительно трудная случайная величина

в определении функции Рабина в самом деле трудна для нее.

Функция RSA. Закрытый ключ равномерно распределен на множестве троек $\langle p, q, i\rangle$, где p, q n-битовые простые числа, а $i = 1, \ldots, pq - 1$ взаимно просто со значением функции Эйлера $\phi(pq) = (p-1)(q-1)$. Открытый ключ e равен паре $\langle pq, i\rangle$. Множество D_e есть множество всех вычетов по модулю pq, взаимно простых с pq, при этом $f_e(x) = x^i \bmod pq$. Случайная величина α_e равномерна распределена в D_e. Условия (1), (3), (4), (5) очевидно выполнены. Условие (2) будет выполнено выполнено если преположительно трудная случайная величина в определении функции RSA в самом деле трудна для нее.

Теорема 17. *Если существует необратимая функция с секретом, то существует и схема шифрования с открытым ключом.*

Доказательство. Сначала научимся шифровать один бит. Гаммирование нам уже не поможет, потому что в схеме гаммирования закрытый ключ совпадал с открытым. Нам опять понадобится понятие трудного бита. Пусть имеется необратимая функция с секретом. Назовем семейство функций $h_e : D_e \to \{0, 1\}$, $e \in A_n$, трудным битом к f_e, если случайная величина $h_e(\alpha_e)$ является трудным битом для $f_e(\alpha_e)$ (распределения случайных величин e и α_e заданы в определении необратимой функции).

Лемма 18. *Если существует необратимая фукнция с секретом, то существует другая необратимая функция с секретом и трудный бит для нее.*

Доказательство. Это следует из теоремы Левина–Голдрайха. Пусть f_e исходная необратимая функция с секретом. В новой необратимой функции случайная величина $\langle e_n, d_n\rangle$ та же самая, а функция g_e определяется как $g_e(xy) = f_e(x)y$, где y строка из нулей и единиц той же длины, что и x. Таким образом область определения g_e состоит из всех слов вида xy, где $x \in D_e$, $|y| = |x|$. Случайная величина α'_n получается приписыванием к α_n независимой равномерно распределенной строки γ той же длины. Трудный бит h_e определяется как скалярное произведение: $h_e(xy) = x \odot y$.

Условия (1)–(5) очевидно выполнены для g_e. Докажем, что h_e трудный бит к g_e. Предположим, что схема C_n полиномиального от n размера с вероятностью $1/2 + \varepsilon_n$ по $e, f_e(\alpha), \gamma$ вычисляет $\alpha \odot \gamma$. Тогда ее можно переделать в схему полиномиального размера, которая по $e, f_e(\alpha), \gamma, Z$

(где случайная величина Z равномерна распределена среди слов той же длины, что e, независимая от α, γ) вычисляет с вероятностью $1/2 + \varepsilon_n$ скалярное произведение конкатенации $e\alpha$ и $Z\gamma$, поскольку последнее равно $e \odot Z \oplus \alpha \odot \gamma$. По теореме Левина–Голдрайха отсюда следует, что ε_n пренебрежимо мало. □

Теперь, имея необратимую функцию g_e с секретом и трудный бит h_e для нее, построим схему шифрования с открытым ключом одного бита b. Случайную величину $\langle e_n, d_n\rangle$ заимствуем из необратимой функции. Алгоритм шифрования выдает пару $y = \langle b \oplus h_e(\alpha), g_e(\alpha)\rangle$ (точнее, надо вместо α взять полиномиально генерируемую и статистически неотличимую от нее случайную величину). Зная d расшифровать y просто: надо найти s, обратив $g_e(\alpha)$, и сложить первую компоненту y с $h_e(\alpha)$. Шифрограмма любого бита вместе с открытым ключом e будет вычислительно неотличима от случайной величины $\langle r, g_e(\alpha), e\rangle$, где r случайный равномерно распределенный бит, независимый от α, e. Действительно, по лемме 6 случайная величина $\langle r, g_e(\alpha), e\rangle$ вычислительно неотличима от случайной величины $\langle h_e(\alpha), g_e(\alpha), e\rangle$. Поэтому для любого бита b случайные величины $\langle b \oplus r, g_e(\alpha), e\rangle$ и $\langle b \oplus h_e(\alpha), g_e(\alpha), e\rangle$ вычислительно неотличимы. Первая из них имеет то же распределение, что и $\langle r, g_e(\alpha), e\rangle$. При замене α на статистически неотличимую от нее случайную величину, получим опять неотличимые случайные величины.

Теперь построим схему шифрования произвольного числа битов. Опять заимствуем алгоритм генерации ключей из необратимой функции. Шифрограмму сообщения $b_1 b_2 \ldots b_m$ из m битов определим как

$$b_1 \oplus h_e(\alpha), b_2 \oplus h_e(g_e(\alpha)), \ldots, b_m \oplus h_e(g_e^{m-1}(\alpha)), g_e^m(\alpha).$$

Все условия, кроме условия (2), очевидны. Условие (2) следует из леммы 7. Пусть $p(n)$ произвольный полином. Применяя лемму 7 к функциям $\langle e, x\rangle \mapsto \langle e, g_e(x)\rangle$, $\langle e, x\rangle \mapsto h_e(x)$ и случайной величине $\alpha_n = \langle e, \alpha_e\rangle$, мы можем заключить, что случайные величины

$$e, h_e(\alpha_e), h_e(g_e(\alpha_e)), \ldots, h_e(g_e^{p(n)-1}(\alpha_e)), g_e^{p(n)}(\alpha_e)$$

и $e, r_1, r_2, \ldots, r_{p(n)}, \alpha_e$ вычислительно неотличимы.

Отсюда следует вычислительная неотличимость случайных величина

$$e, b_1 \oplus h_e(\alpha_e), b_2 \oplus h_e(g_e(\alpha_e)), \ldots, b_{p(n)} \oplus h_e(g_e^{p(n)-1}(\alpha_e)), g_e^{p(n)}(\alpha_e)$$

и $e, b_1 \oplus r_1, b_2 \oplus r_2, \ldots, b_{p(n)} \oplus r_{p(n)}, \alpha_e$ вычислительно неотличимы. Последняя из них имеет то же распределение, что и случайная величина

$$e, r_1, r_2, \ldots, r_{p(n)}, \alpha_e,$$

которая не зависит от $b_1 \ldots, b_{p(n)}$. Поэтому при разных сообщениях $b_1 \ldots, b_{p(n)}$ случайные величины, состоящие из открытого ключа и шифофрграммы, вычислительно неотличимы друг от друга. □

Задача 31. Докажите, что можно из любой схемы шифрования с открытым ключом одного бита можно изготовить схему шифрования с открытым ключом произвольного количества битов с помощью шифрования каждого бита по отдельности.

6 Протоколы привязки к биту (bit commitment)

Протоколы привязки к биту имитируют сундучок с замком, в который первый игрок (отправитель) может положить один бит, закрыть его на ключ и передать второму игроку (получателю). Получатель не может открыть сундучок до тех пор, пока отправитель не передаст ему ключ. Отправитель не может подменить бит, положенный в сундучок. Различают два вида таких протоколов — интерактивные и неинтерактивные. Начнем со вторых, как более простых.

Неинтерактивный протокол привязки к биту состоит из двух алгоритмов: вероятностного полиномиального алгоритма S и детерминированного полиномиального алгоритма R. Алгоритм S получает на вход параметр безопасности n и один бит σ и выдает на выход ключ $k = k_n(\sigma, r)$ и привязку $c = c_n(\sigma, r)$ (r — использованные алгоритмом S случайные биты); привязка имитирует закрытый сундучок. Алгоритм R получает на вход ключ k и привязку c и выдает на выход один бит $R_n(c, k)$ или специальный символ $\perp$ (последнее означает, что ключ k не подошел к замку).

Требования к этим алгоритмам такие.

(1) Для любого n и любого бита σ с вероятностью, приблизительно равной 1, выполнено $R_n(c_n(\sigma, r), k_n(\sigma, r)) = \sigma$ (настоящий ключ открывает сундучок).

(2) Для любого n не существует таких c, k', k'', что $R_n(c, k') = 0$ и $R_n(c, k'') = 1$ (любая строка c может быть привязкой только к одному биту).

(3) Последовательности случайных величин $c_n(0,r)$ и $c_n(1,r)$ вычислительно неотличимы (не зная ключа, получатель не имеет никакой информации о содержимом сундучка).

Если выполнены все три условия, то пара алгоритмов R, S называется протоколом привязки к биту.

Теорема 19. *Если существует инъективная односторонняя функция $f_n : D_n \to \{0,1\}^*$, область определения которой разрешима за время poly(n) (то есть, существует алгоритм, который по любой паре $\langle n, x\rangle$ за время poly(n) определяет, принадлежит ли x к D_n), то существует и протокол привязки к биту.*

Доказательство. По теореме Левина-Голдрайха из условия теоремы следует существование функции g, удовлетворяющей условию теоремы и трудного бита h к ней. Пусть α_n доступная трудная случайная величина для пары $\langle g_n, h_n\rangle$. Алгоритм S выбирает случайную строку x из области определения g_n по полиномиально генерируемому распределению, статистически неотличимому от распределения α_n. Ключ k равен x, а привязка c равна $\langle \sigma \oplus h(x), g(x)\rangle$ (мы опускаем индекс n). Алгоритм R, получив на вход x и $\langle \delta, y\rangle$, проверяет равенство $g(x) = y$ и то, что $x \in D_n$. Если равенство неверно или $x \notin D_n$, он выдает $\perp$. Иначе он выдает $\delta \oplus h(x)$.

Условие (1) очевидно выполнено. Условие (2) следует из инъективности g. Действительно, если $g(x') = y = g(x'')$, и $x', x'' \in D_n$, то $x' = x''$, следовательно, $\delta \oplus h(x'') = \delta \oplus h(x'')$.

Условие (3) означает неотличимость случайных величин $\langle h(x), g(x)\rangle$ и $\langle 1 \oplus h(x), g(x)\rangle$. Это следует из того, что они обе неотличимы от случайной величины $\langle \gamma, g(x)\rangle$, где γ случайный бит, независимый от x. □

Замечание. Без ограничения общности можно считать, что в неинтерактивных протоколах привязки к биту ключ всегда равен σr. Действительно, любой протокол R, S привязки к биту можно преобразовать в протокол R', S', удовлетворяющий этому ограничению. Для этого положим $c'_n(\sigma, r) = c_n(\sigma, r)$ и $R'_n(c, \sigma r) = R_n(c, k_n(\sigma, r))$.

Заметим, что область определения всех кандидатов в односторонние инъективные функции не является полиномиально разрешимой (точнее, неизвестен полиномиальный алгоритм, ее разрешающий). Поэтому, неясно, к какой функции можно было бы применить теорему 19. Перейдем

поэтому к интерактивным протоколам. В интерактивном протоколе получаетель также участвует в положении бита в сундучок. Отправитель и получатель обмениваются сообщениями, которые записываются на магнитофонную ленту. Эта запись и играет роль привязки к биту. Определение интерактивный протокола привязки к биту сложнее, чем неинтерактивного. Но зато, его можно построить, исходя из любого генератора псевдослучайных чисел.

6.1 Интерактивные алгоритмы

Интерактивными алгоритмы — это алгоритмы, которые, кроме обычного входа, имеют еще входной и выходной каналы обмена информацией. Чтобы дать определение интерактивными алгоритма, начнем с понятия стратегии. Стратегией называется функция, отображающей конечные последовательности слов в бинарном алфавите в слова в бинарном алфавите. Пусть имеются стратегии F, G. Последовательность сообщений $c_1, c_2, \ldots$ между стратегиями определяется по правилу

$$\begin{aligned} c_1 &= F(), \\ c_2 &= G(c_1), \\ c_3 &= F(c_1, c_2), \\ c_4 &= G(c_1, c_2, c_3), \\ &\ldots \end{aligned}$$

Общение длится до тех пор, пока некоторое c_i не станет пустым; посылка пустого слова означает желание прекратить общение. Если пустое слово не послано, то последовательность сообщений бесконечна. Последовательность сообщений $c_1, c_2, c_3, \ldots$ будем обозначать через $c(F, G)$ и называть диалогом между F и G.

Стратегией с параметром из некоторого множества M называется отображение, сопоставляющее каждому элементу $x \in M$ некоторую стратегию F_x. Интерактивным алгоритмом со входами из $\{0,1\}^*$ называется вычислимая стратегия с параметром из $\{0,1\}^*$. Чтобы запустить интерактивный алгоритм A, задаваемый стратегией с параметром F_x, нужен вход и собеседник, то есть, некоторая стратегия. Пусть x бинарное слово, а G — некоторая стратегия. Результат $A^G(x)$ работы алгоритма A на входе x с собеседником G определяется следующим образом. Запустим

стратегии F_x и G. Пусть $c(F_x, G)$ полученный диалог. Если диалог бесконечен, то результат не определен. Иначе применим к $c(F_x, G)$ стратегию F_x, полученное слово и будет результатом. Пусть A, B два интерактивных алгоритма, а x, y двоичные слова. Запустим алгоритм A на входе x, а B на входе y так, чтобы они общались друг с другом. Результат, выданный алгоритмом A будем обозначать через $A^{B(y)}(x)$.

Обозначение $A^G(x)$ напоминает обозначения, связанные с машинами с оракулом. Подчеркнем важное отличие машин с оракулом от интерактивных алгоритмов. Ответ оракула на очередной вопрос зависит только от этого вопроса. Очередное сообщение стратегии зависит от всех предыдущих сообщений полученных от алгоритма.

Теперь определим понятие полиномиального интерактивного алгоритма. Полиномиальной стратегией F_x с параметром из некоторого множества двоичных слов называется стратегия, вычислимая за полиномиальное от $n = |x|$ время. Такая стратегия может сделать только полиномиальное от n количество ходов, полиномиальной от n длины (точнее, сделанные ей ходы могут зависеть только от полинома первых символов последовательности сделанных ходов). Интерактивный алгоритм A называется полиномиальным, если стратегия с параметром, его задающая, полиномиальна.

Полиномиальные вероятностные интерактивные алгоритмы (и стратегии с параметром) можно определить через полиномиальные детерминированные интерактивные алгоритмы. Пусть A полиномиальный детерминированный интерактивный алгоритм, а p некоторый полином. Будем понимать результат работы A на входе вида xr с собеседником G, где r двоичное слово длины $p(|x|)$, выбираемое случайно по равномерному распределению, как результат работы вероятностного интерактивного алгоритма на входе вида x с собеседником G.

Последовательность стратегий $\{F_n\}$ называется неравномерно полиномиальной, если существуют схемы C_{in}, где $i, n = 0, 1, 2, \ldots$ такие, что размер схемы C_{in} ограничен полиномом от n, и функция $F_n(y)$ на словах y длины i вычисляется схемой C_{in}. (Мы считаем, что последовательности битовых строк закодированы словами в бинарном алфавите. Например, 0 кодируется как 00, 1 кодируется как 11, а запятая как 01.) Стратегия с параметром $x \mapsto F_x$ называется неравномерно полиномиальной, если существуют схемы C_{in}, где $i, n = 0, 1, 2, \ldots$ такие, что размер схемы C_{in} ограничен полиномом от n, и функция $x, y \mapsto F_x(y)$ на словах y длины i вычисляется схемой $C_{i|x|}$. Аналогично определяются неравномерно

полиномиальные вероятностные стратегии с параметром и алгоритмы.

Интерактивный протокол привязки к биту состоит из трех алгоритмов: вероятностных полиномиальных алгоритмов S, T и детерминированного полиномиального алгоритма R. Алгоритм S получает на вход параметр безопасности n и один бит σ, а алгоритм T получает на вход только n. После этого эти алгоритмы общаются друг с другом, то есть по очереди посылают друг другу сообщения. Через $c(S_n(\sigma, s), T_n(r))$ мы будем обозначать последовательность всех переданных сообщений между S и T, если S получил на вход n, σ и случайные биты s, а T получил n и случайные биты r. Роль привязки к биту в неинтерактивном протоколе играет последовательность сообщений $c(S_n(\sigma, s), T_n(r))$. Кроме того, алгоритм S выдает на выход строку $k_n(\sigma, s)$, называемую ключом (для простоты обозначений мы предполагаем, что выданный ключ зависит только от σ, s). Алгоритм R получает на вход n, строку c и строку k и выдает на выход один бит или сообщение о том, что один из двух игроков блокировал протокол. Будем эти сообщения обозначать $\perp_S, \perp_T$ ($\perp_S$ означает, что протокол блокирован посылающим, а $\perp_T$ — получателем).

Требования к этим алгоритмам такие.

Первое требование заключается в следующем. Для любого бита σ с вероятностью, приблизительно равной 1, выполнено

$$R_n(c(S_n(\sigma, s), T_n(r)), k(\sigma, s)) = \sigma.$$

Более того, это требование должно быть выполнено, даже если, получатель жульничает. Уточним, что это означает.

Пусть получатель на стадии закрытия сундучка выполняет не алгоритм T, а какой-то другой алгоритм T^*. При этом он хочет добиться того, чтобы при его открытии появился противоположный запечатанному бит. Требуется, чтобы это было невозможно, если T^* вычисляется схемами полиномиального от n размера. Итак, уточненное требование выглядит так.

(1) Для любой неравномерно полиномиально последовательности стратегий $\{T_n^*\}$ с вероятностью, приблизительно равной 1, для любого бита σ выполнено

$$R_n(c(S_n(\sigma, s), T_n^*), k(\sigma, s)) \in \{\sigma, \perp_T\}.$$

Второе требование: представим, что отправитель хочет обмануть получателя: на стадии закрытия сундучка он выполняет не алгоритм S,

а неравномерно полиномиальную стратегию S^*. При этом он хочет добиться появления такой последовательности сообщений c, которую можно раскрыть двумя способами, то есть, существуют такие k_0, k_1, что $R_n(c, k_0) = 0$ и $R_n(c, k_1) = 1$. Требуется, чтобы это было невозможно, если алгоритм S^* полиномиален по времени.

(2) Для неравномерно полиномиальной последовательности стратегий $\{S_n^*\}$ для всех n с вероятностью, приблизительно равной 1, выполнено следующее. Во-первых, не существует ключа k, для которого

$$R_n(c(S_n^*, T_n(r)), k) = \perp_T.$$

Во-вторых, не существует таких k_0, k_1, что

$$R_n(c(S_n^*, T_n(r)), k_0) = 0 \quad \text{и} \quad R_n(c(S_n^*, T_n(r)), k_1) = 1.$$

Вместе с условием (1) это гарантирует, что для всех σ с вероятностью приблизительно равной 1, $R_n(c(S(\sigma, s), T(r)), k_1) = \sigma$.

В третьем требовании мы опять заботимся об интересах отправителя. Допустим, жульничает получатель, и вместо алгоритма T использует неравномерно полиномиальную последовательность стратегий T_n^*. Требуется, чтобы возникающая привязка не несла никакой информации о сообщении.

(3) Для любой неравномерно полиномиальной последовательности стратегий $\{T_n^*\}$ случайные величины $c(S_n(0, s), T_n^*)$ и $c(S_n(1, s), T_n^*)$ вычислительно неотличимы.

Если выполнены все три условия, то тройка алгоритмов R, S, T называется интерактивным протоколом привязки к биту.

Теорема 20. *Если существует генератор ПСЧ, то существует интерактивный протокол привязки к биту.*

Доказательство. Пусть G генератор ПСЧ $\{0,1\}^n \to \{0,1\}^{3n}$. Алгоритм T посылает случайную строчку r длины $3n$, после чего останавливается. Алгоритм S ждет от T строчку r длины $3n$. Если присланная строка имеет другую длину, то он заканчивает диалог (посылая пустое слово). Иначе, он выбирает случайную строчку s длины n и посылает $G(s)$, если $\sigma = 0$, и посылает $G(s) \oplus r$ иначе (то есть, он посылает $G(s) \oplus \sigma \cdot r$). Ключ, выдаваемый алгоритмом S есть σs.

Входом алгоритма R, кроме n, является пара строк r, x и строка k. Если все строки имеют правильную длину, причем $x = G(s) \oplus \sigma \cdot r$, где σ

первый бит строки k, а s — остальные ее биты, то алгоритм R выдает σ. Иначе, алгоритм R разбирается, кто заблокировал протокол: если длина r не равна $3n$, то заблокировал получатель. Если длина k не равна $n+1$ или $x \neq G(s) \oplus \sigma \cdot r$, то заблокировал отправитель.

Ясно, что условие (1) выполнено. Проверим условие (2). Пусть дана последовательность схем S_n^* полиномиального от n размера (поскольку алгоритм S выдает только одно сообщение, нам нужна для каждого n только одна схема с $3n$ входами). В нашем случае требование заключается в том, что вероятность события

$$(\exists s^0, s^1) \quad S_n^*(r) = G(s^0), \quad S_n^*(r) = G(s^1) \oplus r$$

пренебрежимо мала. Вероятность этого события не больше вероятности того, что найдутся s^0, s^1 такие, что $G(s^0) = G(s^1) \oplus r$. При фиксированных s^0, s^1 вероятность этого события равна 2^{-3n}. Суммируя по всем s^0, s^1, получаем не больше 2^{-n}, что стремится к нулю быстрее любого обратного полинома от n.

Проверим условие (3). Пусть дана последовательность схем T_n^* полиномиального от n размера, которые выдают на выход одну строку $3n$, не имея входа. Другими словами T_n^* просто задает последовательность строк r_n^*. При тех n, для которых длина r_n^* не равна $3n$, диалог имеет вид $\langle r_n^*, \text{пустое слово} \rangle$, и не зависит от σ. При остальных n нам надо доказать вычислительную неотличимость случайных величин $\langle r_n^*, G(s) \rangle$ и $\langle r_n^*, G(s) \oplus r_n^* \rangle$. Они вычислительно неотличимы друг от друга, поскольку обе вычислительно неотличимы от случайной величины $\langle r_n^*, \gamma_{3n} \rangle$, где γ_{3n} равномерно распределенная среди слов длины $3n$ случайная величина. □

Задача 32. Докажите, что без ограничения общности можно считать, что интерактивный протокол в качестве ключа всегда выдает свои случайные биты и σ.

7 Протоколы бросания монетки по телефону

Протоколом бросания монетки по телефону называется тройка алгоритмов A, B, J. Каждый из этих алгоритмов получает на вход натуральное число n в унарной записи и останавливается, проработав полиномиальное от n число шагов. Кроме того, алгоритм J получает на вход еще и

двоичную строку c. Алгоритмы A, B являются вероятностными интерактивными алгоритмами без выхода, а алгоритм J детерминированным неинтерактивным алгоритмом, выдающим на выход `A`или `B`.

Эти три алгоритма следующим образом используются для игры в орлянку по телефону. Чтобы подбросить монетку, игроки, Алиса и Боб, выполняют алгоритмы A, B соответственно. Эти алгоритмы общаются по телефону (без ограничения общности можно считать, что первое сообщение посылается алгоритмом A). Запись их общения $c(A, B)$ дается на вход алгоритму J и он определяет, кто выиграл: если $J(c(A, B)) = \texttt{A}$, то выиграла Алиса, а если если $J(c(A, B)) = \texttt{B}$, то выиграл Боб.

Требуется, чтобы даже если Боб жульничает, и вместо алгоритма B запускает некоторую неравномерно полиномиальную стратегию B_n^*, то вероятность того, что Алиса проиграет превосходить $1/2$ (вероятность проиграть при обычной игре в орлянку) не более, чем на пренебрежимо малую величину (точнее, на величину, стремяющуюся с ростом n к нулю быстрее любого обратного полинома). Аналогичное требование должно быть выполнено, и для вероятности проигрыша Боба, если жульничает Алиса.

Формально, должны быть выполнены следующие два требования.

(1) Для любой неравномерно полиномиальной стратегии B_n^* вероятность события $J(1^n, A(1^n), B_n^*) = \texttt{B}$ не превосходит $1/2+\alpha_n$, где α_n пренебрежимо мало (стремится с ростом n к нулю быстрее любого обратного полинома).

(2) Для любой неравномерно полиномиальной стратегии A_n^* вероятность события $J(1^n, A_n^*, B(1^n) = \texttt{A}$ не превосходит $1/2 + \beta_n$, где β_n пренебрежимо мало.

Замечание 1. Важно, что алгоритм J детерминирован. Иначе, во-первых, задача была бы тривиальной, поскольку алгоритм J сам бы мог подбросить монетку и объявить победителя, исходя из результата бросания. Во-вторых, вероятностные алгоритмы бесполезны, поскольку игроки не могут запустить его из-за отсутствия общего источника случайности.

Замечание 2. Алгоритм J задает некоторую игру G_n (одну для каждого значения параметра безопасности) с конечным числом ходов. А именно, количество ходов в этой игре и длина каждого хода ограничено полиномом $p(n)$, ограничивающим время работов алгоритмов A и B. Игроки ходят по очереди, начиная с Алисы. Последовательность сделанных ходов дается на вход алгоритму J, который и определяет, кто выиграл. По

теореме Кенига, в любой игре с конечным числом ходов один из игроков имеет в этой игре выигрышную стратегию. Если для бесконечно многих n этим игроком является Алиса, то условие (2) не может быть выполнено для любой вообще ее стратегии B_n^*, поскольку если она применяет выигрывающую стратегию (для тех n, для которых такая существует), то вероятность ее выигрыша равна 1 (для этих n). Аналогично, если для бесконечно многих n этим игроком является Боб, то условие (1) не может быть верным для любой стратегии Боба. Поэтому в обоих условиях (1) и (2) важно, что мы ограничиваем вычислительные возможности жульничающего игрока.

Теорема 21. *Если существует интерактивный протокол привязки к биту, то существует и протокол подбрасывания монетки по телефону.*

Доказательство. Пусть нам дан некоторый протокол (S, R, T) привязки к биту.

Алгоритм A (выполняемый Алисой) действует так: подбрасываем монетку один раз и получаем случайный бит, который будем обозначать через σ. Запускаем алгоритм $S(\sigma)$, который будет общаться с Бобом. Когда общение закончится и алгоритм $S(\sigma)$ выдаст некоторый ключ k, мы останавливаемся и ждем, пока нам не пришлют следующее сообщение. После этого посылаем ключ k.

Алгоритм B (выполняемый Бобом) действует так: запускаем алгоритм R, который будет общаться с Алисой. Когда общение закончится, подбрасываем монетку и посылаем результат бросания τ Алисе.

Алгоритм J действует так. Пусть c обозначает протокол общения на первой стадии. (Чтобы можно было отделить в протоколе, где заканчиваются сообщения первой стадии, алгоритмы S и R надо модифицировать так, чтобы они приписывали ко всем ссобщениям 0 слева. На второй стадии все сообщения будут начинаться на 1.) Запускаем алгоритм T на паре c, k, где k последнее сообщение Алисы. Объявляем выигравшим Боба, если T выдает, сообщение, что протокол привязки блокирован Алисой, или на второй стадии Боб послал бит, совпадающий с битом, выданным T.

Проверим, что этот протокол соблюдает инетерсы обоих игроков, то есть, условия (1) и (2).

Интересы Алисы. Пусть Алиса действует честно, а Боб руководствуется некоторой неравномерно полиномиальной стратегией B_n^*. Это означает, что на первой стадии он выполняет некоторую неравномерно по-

линомиальную стратегию R_n^*, а на второй стадии применяет к протоколу беседы c на первой стадии некоторую схему τ_n^* полиномиального от n размера. Оценим сверху вероятность того, что алгоритм J объявит проигравшей Алису. Такое может произойти в двух случаях: алгоритм T объявит Алису блокировавшей протокол и алгоритм T выдаст бит, равный $\tau^*(c(S(\sigma), R^*))$. Поскольку Алиса не жульничает, в силу первого требования к протоколу привязки вероятность первого события пренебрежимо мала. В силу того же требования вероятность второго события приблизительно равна вероятности того, что $\tau^*(c(S(\sigma), R^*))$ равно σ. Так как Алиса выбирает свой бит σ случайно, вероятность этого события равна

$$\frac{\Pr[\tau^*(c(S(0), R^*)) = 0] + \Pr[\tau^*(c(S(1), R^*)) = 1]}{2}.$$

В силу второго требования к протоколу привязки случайные величины $c(S(0), R^*)$ и $c(S(1), R^*)$ вычислительно неотличимы. В частности, они неотличимы и схемой τ^*. Поэтому, если во втором слагаемом заменить вторую случайную величину на первую, вероятность изменится на пренебрежимо малую величину. После такой замены в числителе дроби

$$\frac{\Pr[\tau^*(c(S(0), R^*)) = 0] + \Pr[\tau^*(c(S(0), R^*)) = 1]}{2}$$

мы получим сумму вероятностей взаимно исключающих событий, поэтому числитель не превосходит 1, а значит вся дробь не превосходит $1/2$.

Интересы Боба. Пусть Боб действует честно, а Алиса руководствуется некоторой неравномерно полиномиальной стратегией A_n^*. Это означает, что на первой стадии она выполняет некоторую неравномерно полиномиальную стратегию S_n^*, а на второй стадии применяет к протоколу беседы c, и биту τ, посланному Бобом, некоторую схему k_n^* полиномиального от n размера. Оценим сверху вероятность того, что алгоритм J объявит проигравшей Боба. Такое может произойти в двух случаях: алгоритм T объявит Боба блокировавшим протокол и алгоритм T выдаст бит, отличный от τ, то есть $\bar{\tau}$.

Поскольку Боб действует честно, по третьему требованию вероятностью первого события можно пренебречь. Поскольку свой бит τ Боб выбирает случайно, вероятность второго события равна

$$\frac{\Pr[T(c(S^*, R), k^*(c(S^*, R), 0)) = 1] + \Pr[T(c(S^*, R), k^*(c(S^*, R), 1)) = 0]}{2}.$$

Оценим сверху вероятность этого события, испльзуя оценку

$$\Pr[U] + \Pr[V] = \Pr[U \cup V] + \Pr[U \cap V] \leq 1 + \Pr[U \cap V],$$

верную для любых двух событий U, V. В числителе нашей дроби суммируются вероятности двух событий U, V. Вероятность их пересечения пренебрежимо мала, поскольку третье требование к протоколу привязки гарантирует, что вероятность существования ключей k_0, k_1, для которых

$$T(c(S^*, R), k_0)) = 0, \qquad T(c(S^*, R), k_1)) = 1,$$

пренебрежимо мала. Поэтому $\Pr[U \cap V]$ пренебрежимо мала. □

Задача 33. Пусть в качестве протокола привязки в этой схеме использован неинтерактивный протокол, основанный на однозначной односторонней функции. У кого из двух игроков есть выигрышная стратегия в полученной игре (без ограничения на равномерную полиномиальность)?

Задача 34. Пусть в качестве протокола привязки в этой схеме использован интерактивный протокол из предыдущего раздела, основанный на существовании генератора ПСЧ. У кого из двух игроков есть выигрышная стратегия в полученной игре (без ограничения на равномерную полиномиальность)?

8 Интерактивные доказательства и класс IP

Скажем, что множество слов L принадлежит классу IP, если существует полиномиальный вероятностный интерактивный алгоритм V с таким двумя свойствами.
(1) Для всех $x \in L$ существует стратегия P такая, что $\Pr[V^P(x) = 1] \approx 1$.
(2) Для всех $x \notin L$ для любой стратегии P выполнено $\Pr[V^P(x) = 1] \approx 0$.

Вероятность в обоих случаях берется по случайным битам алгоритма V, так что стоило бы писать $V^P(x, r)$, а не $V^P(x)$. Приблизительное равенство в обоих требованиях понимается при стремлении $n = |x|$ к бесконечности. Обычно алгоритм V называется Проверяющим, а стратегия P — Доказывающим.

Теорема 22. *Язык GI, состоящий из множества всех пар $\langle G_0, G_1 \rangle$ неизоморфных графов, принадлежит IP.*

Мы считаем, что множества вершин графов G_0, G_1 совпадают и равны множеству чисел $\{1, 2, \ldots, n\}$ для некоторого n. Оба графа задаются матрицей смежности, которая, в свою очередь, задается $n(n-1)/2$ битами.

Доказательство. Стратегия Проверяющего заключается в следующем. Выбираем случайно один бит α (оба значения 0,1 с вероятностями $1/2$). Выбираем случайную перестановку π множества вершин $\{1, 2, \ldots, n\}$ (все перестановки равновероятны). Переставляем в графе G_α вершины с помощью перестановки π и посылаем полученный граф πG_α (точнее, его матрицу смежности). Если в ответ Доказывающий прислал что-то, отличное от бита α, то заканчиваем общение досрочно. Иначе по повторяем то же самое, используя новый бит α и новую перестановку π. И так далее, n раз. Если не произошло досрочной остановки, то выдаем 0, а иначе 1.

Ясно, что если графы G_0, G_1 не изоморфны, то у Доказывающего имеется стратегия P такая, что с вероятностью 1 выполнено $V^P(G_0, G_1) = 1$. Пусть графы неизоморны. Докажем, что вероятность того, что в первом раунде Доказывающий прислал α, не больше $1/2$. Действительно, поскольку мы выбираем оба бита $0, 1$ равновероятно, вероятность этого события равна

$$\frac{\Pr[P(\pi G_0) = 0] + \Pr[P(\pi G_1) = 1]}{2}.$$

Поскольку графы G_0, G_1 изоморфны, случайные величины $\pi G_0, \pi G_1$ имеют одинаковое распределение, поэтому в первом слагаемом в этой сумме πG_0 можно заменить на πG_1 (вероятность от этого не изменится). В результате мы получим полусумму вероятностей непересекающихся событий, которая не может превосходить $1/2$.

Рассуждая таким образом, легко доказать по индукции, что после выполнения k раундов вероятность того, что мы не остановимся досрочно не больше 2^{-k}. Чтобы сделать индуктивный переход, достаточно установить, что при любых значениях $\alpha_1, \ldots, \alpha_k$, $\pi_1, \ldots, \pi_k$ вероятность того, что в $k+1$ раунде Доказывающий правильно найдет α_{k+1} при условии, что в предыдущих раундах Проверяющий выбрал $\alpha_1, \ldots, \alpha_k$, $\pi_1, \ldots, \pi_k$, не превосходит $1/2$. Это доказывается так. Ход, делаемый Доказывающем в раунде $k+1$ зависит от наших ходов в предыдущих k раундах и от хода πG_α сделанного нами в раунде $k+1$. Поскольку мы предполагаем $\alpha_1, \ldots, \alpha_k$, $\pi_1, \ldots, \pi_k$, фиксированными, ход Доказывающего зависит

только от $\pi(G_\alpha)$. Поэтому годится то же рассуждение, что и для первого раунда. □

9 Доказательства с нулевым разглашением

Пусть F_x вероятностная стратегия с параметром x, являющимся двоичным словом. Пусть $f(x)$ некоторая функция от x. Мы говорим, что стратегия F_x не разглашает ничего, кроме $f(x)$ (то есть, разглашает только $f(x)$), если существует вероятностный алгоритм S с оракулом, для которого выполнено следующее. Для любой неравномерно полиномиальной последовательности стратегий $\{F_n\}$ для каждого x алгоритм S, получая на вход $f(x)$, и используя в качестве оракула F_n, где $n = |x|$, должен сгенерировать за полиномиальное от n время некоторую случайную величину $\alpha_{f(x)}$, вычислительно неотличимую от случайной величины $c(F_x, G_n)$ равномерно по x (напомним, что так обозначается диалог между стратегиями F_x и G_n). Это означает, что для любой неравномерно полиномиальной последовательности отличителей E_n и любой последовательности слов x_n, где $|x_n| = n$, вероятности событых $E_n(\alpha_{f(x)}) = 1$ и $E_n(c(F(x), G_n)) = 1$ приблизительно равны. В частности, случайные величины $c(F(x), G_n)$ при разных x с одинаковым значением $f(x)$ вычислительно неотличимы друг от друга. Алгоритм S будем называть симулятором. В этом определении важно, что симулятору разрешается не просто общаться со стратегией G_n, а можно запрашивать значение G_n на произвольных последовательностях слов. Например, симулятор может запросить значения $G_n(c_1)$ и $G_n(c_1')$, что невозможно в ходе беседы.

Заметим, что в этом в этом определении стратегия G_n предполагается детерминированной. Однако это несущественное ограничение. Пусть имеется неравномерно полиномиальная последовательность вероятностных стратегий $\{G_n\}$. Стратегию, полученную из G_n фиксацией случайных битов r, будем обозначать через $G_n(r)$. По условию длина r есть полином от n, и слово, выдаваемое стратегией $G_n(r)$ на последовательности сообщений y, можно вычислить по r, y схемой полиномиального от n размера со входами r, y. Тогда с помощью алгоритма S можно сгенерировать случайную, вычислительно неотличимую от случайной величины $\langle r, c(F_x, G_n(r))\rangle$. Для этого выберем r случайным образом, запустим алгоритм S с оракулом $G_n(r)$ и выдадим на выход результат его работы, спаренный с r.

Установим одно важное свойство этого понятия. Для этого определим операцию последовательного выполнения (композиции) стратегий. Пусть F_1, F_2 две стратегии. Выполним сначала стратегию F_1, а затем, забыв все сделанные нами и собеседником сообщения, выполним стратегию F_2. Иными словами, результат применения новой стратегии к последовательности ходов $c_1, c_2, \ldots, c_k$ определяется так: если не существует такого i, что $F_1(c_1, c_2, \ldots, c_{2i})$ есть пустое слово (то есть, стратегия F_1 не остановила диалог), то выдаем $F_1(c_1, c_2, \ldots, c_k)$. Иначе выдаем $F_2(c_{2i+1}, \ldots, c_k)$, где i наименьшее такое, что $F_1(c_1, c_2, \ldots, c_{2i})$ — пустое слово. Полученную стратегию обозначим через F_1F_2.

Аналогично определяется последовательное выполнение двух вероятностных стратегий. При выполнении F_2 мы используем случайные биты, независимые от случайными битами стратегии F_1. Иными словами, если стратегии F_1 нужна последовательность r_1 из n случайных битов, а стратегии F_2 — последовательность r_2 из m случайных битов, то композиции стратегий F_1, F_2 нужно $n+m$ случайных битов. Стратегия F_1F_2 со случайными битами r_1r_2 есть композиция стратегии F_1 со случайными битами r_1 и стратегии F_2 со случайными битами r_2.

Пусть $F(x)$ — вероятностная стратегия с параметром x (чтобы в дальнейшем избежать двойных индексов, мы записываем параметр не как индекс, а как аргумент). Рассмотрим стратегию с параметром $\langle x_1, x_2\rangle$, заключающуюся в последовательном выполнении $F(x_1)$ и $F(x_2)$. Полученную стратегию обозначим через $F(x_1)F(x_2)$.

Лемма 23. *Если стратегия $F(x)$ разглашала только $f(x)$, то стратегия $F(x_1)F(x_2)$ разглашает только $\langle f(x_1), f(x_2)\rangle$.*

Доказательство. Зафиксируем симулятор S для стратегии $F(x)$ и построим симулятор $\tilde{S}$ для стратегии $F(x_1)F(x_2)$. Новый симулятор должен, получив на вход пару $\langle y_1, y_2\rangle = \langle f(x_1), f(x_2)\rangle$ и имея в качестве оракула некоторую неравномерно полиномиальную стратегию G_n, где $n = |x_1x_2|$, сгенерировать случайную величину, вычислительно неотличимую от $c(F(x_1)F(x_2), G)$ (мы будем опускать в дальнейшем индекс n).

После того, как статегия $F(x_1)$ останавливается, в диалоге между $F(x_1)F(x_2)$ и G сделаны какие-то сообщения $c_1, \ldots, c_{2k}$ (последнее сообщение c_{2k+1} сделано стратегией $F(x_1)$ и является пустым словом). Мы можем считать, что со стратегией $F(x_2)$ беседует стратегия $G_{c_1,\ldots,c_{2k}}$, определяемая равенством

$$G_{c_1,\ldots,c_{2k}}(c_{2k+1}, c_{2k+2}, \ldots) = G(c_1, c_2, \ldots, c_{2k}, c_{2k+1}, c_{2k+2}, \ldots).$$

Поэтому диалог стратегий $F(x_1)F(x_2)$ и G равен $u(r_1), v(r_1, r_2)$, где $u(r_1)$ — диалог стратегий $F(x_1, r_1)$ и G, а v — диалог стратегий $F(x_2, r_2)$ и $G_{u(r_1)}$. Здесь r_1 и r_2 независимые равномерно распределенные строки, используемые вероятноснтыми стратегиями $F(x_1)$ и $F(x_2)$, соответственно. Чтобы сгенерировать эту случайную величину, запускаем симулятор S на входе y_1 с оракулом G. Он выдаст некоторую последовательность сообщений $\eta(s_1)$, которая вычислительно неотличима от случайной величины $u(r_1)$. Через s_1 мы использовали последовательность случайных битов, использованную симулятором S. Запустим опять симулятор S с оракулом $G_{\eta(s_1)}$ на входе y_2, используя новые случайные биты s_2. Он нам выдаст последовательность $\theta(s_1, s_2)$, вычислительно неотличимую от случайной величины $c(F(x_2, r_2), F_{\eta(s_1)})$.

Нам надо доказать вычислительную неотличимость случайных величин

$$\xi_0 = u(r_1), v(r_1, r_2).$$

и

$$\xi_2 = \eta(s_1), \theta(s_1, s_2).$$

Достаточно доказать, что они обе они неотличимы от случайной величины

$$\xi_1 = \eta(s_1), w(s_1, r_2),$$

где $w(s_1, r_2) = c(F(x_2, r_2), G_{\eta(s_1)})$ есть диалог стратегий $F(x_2, r_2)$ и $G_{\eta(s_1)}$. Неотличимость ξ_0 и ξ_1 доказывается так. Пусть дан тест E полиномиального размера вероятностей прохождения которого E случайными величинами ξ_0 и ξ_1 не меньше $\varepsilon = 1/\mathrm{poly}(n)$. Вспомним, что $v(r_1, r_2)$ есть диалог стратегий $F(x_2, r_2)$ и $G_{u(r_1)}$. Рассмотрим вероятностную схему $E'(z) = E(z, c(F(x_2, r_2), G_z))$. Если этой схеме подать на вход случайную величину $u(r_1)$ то на выходе будет случайная величина $E(\xi_0)$, а если ей подать на вход $\eta(s_1)$, то на выходе будет случайная величина $E(\xi_1)$. Поэтому разность вероятностей прохождения теста E' случайными величинами $u(r_1)$ от $\eta(s_1)$ равна разности вероятностей прохождения теста E случайными величинами ξ_0 и ξ_1, а значит не меньше ε. Поскольку стратегия G неравномерно полиномиальна, схема E' имеет полиномиальный от n размер.

Неотличимость ξ_2 и ξ_1 доказывается так. Пусть дан тест E полиномиального размера вероятностей прохождения которого E случайными величинами ξ_1 и ξ_2 не меньше $\varepsilon = 1/\mathrm{poly}(n)$. Зафиксировав подходящим образом s_1 получим тест полиномиального размера E', разность

вероятность прохождения которого случайными величинами $\theta(s_1, s_2)$ и $w(s_1, s_2)$ не меньше ε. Вспомним теперь, что $\theta(s_1, s_2)$ есть диалог стратегий $F(x_2, r_2)$ и $G_{\eta(s_1)}$), а $\theta(s_1, s_2)$ есть результат применения симулятора S ко входу $f(x_2)$ с оракулом $G_{\eta(s_1)}$. В этой части рассуждения нам не важно, что G неравномерно полиномиальная стратегия. □

Точно таким же рассуждением можно доказать следующее обобщение доказанного утверждения на полиномиальное количество повторений. Пусть $p(n)$ произвольный полином и пусть стратегия $F(x)$ разглашает только $f(x)$. Пусть $x_1, \ldots, x_m$, где $m = p(n)$, слова длины n. Рассмотрим стратегию $F(x_1) \ldots F(x_m)$ заключающуюся в последовательном независимом выполнении стратегий $F(x_1), \ldots, F(x_m)$.

Лемма 24. *Если стратегия $F(x)$ разглашала только $f(x)$, то стратегия $F(x_1) \ldots F(x_m)$ разглашает только $\langle f(x_1), \ldots, f(x_m)\rangle$.*

Доказательство. Поскольку в рассуждении n и $x_1, \ldots, x_m$ меняться не будут, будем опускать их в обозначениях. Зафиксируем симулятор S для стратегии F и построим симулятор $\tilde{S}$ для стратегии $F(x_1) \ldots F(x_m)$. Новый симулятор должен, получив на вход $\langle y_1, \ldots, y_m\rangle = \langle f(x_1), \ldots, f(x_m)\rangle$ и имея в качестве оракула некоторую неравномерно полиномиальную стратегию G, сгенерировать случайную величину неотличимую от диалога стратегий $F(x_1) \ldots F(x_m$ и G. Для этого запускаем симулятор S на входе y_1 с оракулом G. Он выдаст некоторую последовательность сообщений $\eta(s_1)$. Запустим опять симулятор S с оракулом $G_{\eta_1(s_1)}$ на входе y_2, используя новые случайные биты s_2. Обозначим через $\eta(s_1, s_2)$ результат приписывания выданной им последовательности в конец последовательности$\eta(s_1)$. Повторяя этот процесс m раз получим случайные величины, $\eta(s_1, s_2, \ldots, s_i)$ определяемые индуктивно: $\eta(s_1, s_2, \ldots, s_i)$ есть соединение последовательности $\eta(s_1, \ldots, s_{i-1})$ и результата работы симулятора S на входе y_i со случайными битами s_i с оракулом $G_{\eta(s_1,\ldots,s_{i-1})}$.

Мы утверждаем, что последовательность $\eta(s_1, s_2, \ldots, s_m)$ вычислительно неотличима от диалога стратегий $F(x_1) \ldots F(x_m)$ и G. Чтобы показать это, дадим индуктивное определение диалога стратегий $F(x_1) \ldots F(x_m)$ и G. Определим $\theta(r_1, \ldots, r_i)$ как последовательность, полученную из последовательности $\theta(r_1, \ldots, r_{i-1})$ приписыванием в конец диалога стратегии $F(x_i)$ со случайными битами r_i и и стратегии $G_{\theta(r_1,\ldots,r_{i-1})}$). Тогда $\theta(r_1, \ldots, r_m)$ есть диалог стратегии $F(x_1) \ldots F(x_m)$ со случайными битами $r_1 \ldots r_m$ и G.

Допустим существует неравномерно полиномиальный отличитель E_n случайных величин $\eta(s_1, s_2, \ldots, s_m)$ и $\theta(r_1, \ldots, r_m)$, который для бесконечно многих n имеет успех $\varepsilon_n = 1/\mathrm{poly}(n)$. Фиксируем одно из таких n и найдем i, стратегию $\tilde{G}$ и отличитель диалога стратегий $F(x_i)$ и $\tilde{G}$ от случайной величины, генерируемой симулятором S на входе y_i с оракулом $\tilde{G}$.

Для каждого $i = 0, 1, \ldots, m$ рассмотрим гибридную случайную величину $\xi_{im}(s_1, \ldots, s_i, r_{i+1}, \ldots, r_m)$. В определении этой случайной величины i раз используется первая индуктивная схема, и $m - i$ — вторая. Случайная величина $\xi_0(r_1, \ldots, r_m)$ есть $\theta(r_1, \ldots, r_m)$, а $\xi_m(s_1, \ldots, s_m)$ есть $\eta(s_1, s_2, \ldots, s_m)$. Рассмотрим то i, для которого разность вероятностей $\Pr[E(\xi_i) = 1]$ и $\Pr[E(\xi_{i-1}) = 1]$ больше ε/m по абсолютной величине. Случайная величина ξ_{i-1} является функцией от $s_1, \ldots, s_{i-1}, r_{i+1}, \ldots, r_m$ и диалога c_i стратеги $F(x_i)$ со случайными битами r_i и стратегии $\tilde{G} = G_{\eta_{i-1}(s_1,\ldots,s_{i-1})}$. Обозначим эту функцию через f_i. Эта функция неравномерно полиномиальна, потому что F и G таковы. При замене настоящего диалога c_i на псевдо-диалог (результат работы S с оракулом $\tilde{G}$ на входе y_i со случайными битами s_i) значением этой функции вместо ξ_{i-1} станет случайная величина ξ_i. Поэтому схема $E(f_i(s_1, \ldots, s_{i-1}, r_{i+1}, \ldots, r_m, z))$ при подходящей фиксации значений $s_1, \ldots, s_{i-1}, r_{i+1}, \ldots, r_m$ будет отличать настоящий диалог от псевдо-диалога с успехом не менее ε/m. □

Определим теперь понятие интерактивного доказательства с нулевым разглашением. Пусть имеется интерактивный алгоритм V и вероятностная стратегия P_x с параметром x. Такая пара называется протоколом интерактивного доказательства с нулевым разглашением для множества слов L, если выполнены три условия.

(1) Для всех $x \in L$ выполнено $\Pr[V^{P_x}(x) = 1] \approx 1$.

(2) Для всех $x \notin L$ для любой стратегии F выполнено $\Pr[V^F(x) = 1] \approx 0$.

Приблизительные равенства в обоих условиях при стремлении длины x к бесконечности. Эти условия гарантируют, то что L принадлежит классу IP.

(3) Стратегия P_x разглашает только $\langle x, L(x)\rangle$.

Множество всех языков, для которых существует протокол интерактивного доказательства с нулевым разглашением называется ZKP (Zero Knowledge Proofs).

Теорема 25. *Язык GI, состоящий из множества всех пар $\langle G_0, G_1\rangle$ изоморфных графов, принадлежит ZKP.*

Доказательство. Игру начинает Доказывающий. Если графы G_0, G_1 неизоморфны, то он останавливает игру. Иначе он фиксирует некоторый изоморфизм τ этих графов: $\tau G_1 = G_0$. Затем он посылает Проверяющему граф $H = \pi G_0$, где π случайную перестановка множества $\{1, \ldots, n\}$ вершин графа G_0. Получив граф H, проверяющй посылает Доказывающему случайно выбранный бит α. Затем Доказывающий оставливает игру, если ему прислали не бит (то есть длина присланного сообщения не равна 1), а иначе посылает перестановку $\rho = \pi \circ \tau^\alpha$, Эта перестановка устанавливает изомофризм графов G_α и H, то есть, $\pi \circ \tau^\alpha(G_\alpha) = \pi(G_0) = H$. Проверяющий проверяет, в самом ли деле $\rho(G_\alpha) = H$ и останавливает игру досрочно, если это не так. Если игра не остановлена, то все продолжается сначала, с новыми π, α (изоморфизм τ не меняется). После n повторений Проверяющий останавливает игру, и, если игра не была остановлена досрочно, выдает 1, а иначе 0.

Условие (1) очевидно выполнено. Проверим условие (2). Пусть Доказывающий применяет некоторую стратегию F (возможно отличную от стратегии P). Если графы G_0, G_1 неизоморфны, то посланный им в i-ом раунде граф H не изоморфен хотя бы одному из графов G_0, G_1. С вероятностью не менее $1/2$ Доказывающий укажет в качестве α номер того графа, которому H не изоморфен. Если это произошло, то какую бы перестановку ρ ни прислал Доказывающий, равенство $\rho(G_\alpha) = H$ не выполнено, и Проверяющий досрочно остановит игру. Вероятность этого события не менее одной второй при условию любых ходов в предыдущих раундах, поэтому вероятность досрочной остановки игры хотя бы в одном раунде не менее $1 - 2^{-n}$.

Осталость проверить условие (3). Поскольку Доказывающий применяет одну и ту же стратегию во всех раундах, по лемме 24 достаточно доказать, что в первом раунде разглашается только $\langle x, L(x)\rangle$.

Симулятор действует так. Пусть F некоторая (детерминированноая) стратегия Проверяющего. Будем сначала предполагать, что нам заранее известно, что F на любом графе выдает 0 или 1. Если $L(x) = 0$, то генерация записи общения тривиальна, потому что игра заканчивается, не начавшись. Пусть $L(x) = 1$, то есть графы изоморфны. Тогда нам нужно по G_0, G_1 сгенерировать случайной величину, вычислительно неотличимую от величины $\xi = \langle \pi(G_0), \alpha, \pi \circ \tau^\alpha\rangle$. При этом мы не знаем τ

(иначе генерация была бы тривиальной — мы могли бы просто запустить стратегии Проверяющего и Доказывающего).

Нетрудно понять, что распределение ξ есть равномерное распределение на тройках $\langle H, \alpha, \rho\rangle$ таких, что $\rho(G_\alpha) = H$ и $F(H) = \alpha$ (будем называть такие тройки удачными). Действительно, для каждой перестановки π имеется удачная тройка $\langle H, \alpha, \rho\rangle$ — та, которая получена, если на первом шаге Доказывающий выбрал перестановку π. С другой стороны, каждая удачная тройка H, α, ρ появится в беседе, если на первом шаге Доказывающий выберет перестановку $\pi = \rho \circ \tau^{-\alpha}$. Таким образом, удачных троек ровно $n!$ и все они равновероятны.

Теперь можно запустить хорошо известный алгоритм генерации равномерного распределения в данном множестве. Сначала заметим, что первая компонета любой удачной тройки однозначно определяется второй и третьей компонентами. Поэтому можно взять случайную пару α, ρ, найти $H = \rho(G_\alpha)$ и проверить будет ли тройка $\langle H, \alpha, \rho\rangle$ удачной (то есть, $F(H) = \alpha$). Если нет, то повторить попытку, взяв новую пару α, ρ. Чтобы этот алгоритм за полиномиальное от n время находил с большой вероятностью удачную тройку, нужно чтобы удачных троек было достаточно много. Мы утверждаем, что вероятность того, что тройка $\langle \rho(G_\alpha), \alpha, \rho\rangle$ удачна, равна $1/2$. Действительно вероятность того, что $F(\rho(G_\alpha)) = \alpha$ равна среднему арифметическому вероятностей событий $F(\rho(G_0)) = 0$ и $F(\rho(G_1)) = 1$. Поскольку графы G_0 и G_1 изоморфны, случайные величины $\rho(G_0)$ и $\rho(G_1)$ имеют одинаковое распределение, поэтому вероятность события $F(\rho(G_0)) = 0$ равна вероятности события $F(\rho(G_1)) = 0$. Значит вероятность найти успешную тройку в одной попытке равна среднему арифметическому вероятностей событий $F(\rho(G_1)) = 0$ и $F(\rho(G_1)) = 1$. Поскольку эти события взаимно дополнительны (здесь мы используем, что стратегия F выдает всегда бит), эта вероятность равна $1/2$. Таким образом, после n попыток вероятность неудачи будет равна 2^{-n} и в этом случае мы можем выдать любую тройку.

Как генерировать запись сообщений в общем случае. Обозначим через $\hat{F}$ стратегию, которая в том случае, когда F выдает не бит, выдает, скажем ноль, а в остальных случаях работает так же, как F. Симулятор действует как описано, используя $\hat{F}$ вместо F. После получения тройки $\langle H, \alpha, \rho\rangle$ он применяет F к графу H и, если получится не бит, то заменяет в этой тройке α (которое в этом случае равно 0) на $F(H)$, а ρ на пустое слово. Это преобразование переводит распределение $c(P_x, \hat{F})$ в распределение $c(P_x, F)$ (поскольку при каждом фиксированном π, тройка по-

лученная, в игре стратегий P_x и F получается этим преобразованием из тройки полученной в игре стратегий P_x и $\hat{F}$). Поэтому это преобразование переводит и любое распределение, статистически неотличимое от распределения $c(P_x, \hat{F})$ в распределение, статистически неотличимое от распределения $c(P_x, F)$. □

10 Протоколы аутентификации

Протоколом аутентификации называется последовательность доступных случайных величин $\langle d_n, e_n \rangle$ (закрытый и открытый ключи) и пара полиномиальных вероятностных интерактивных алгоритмов P, V, алгоритм Доказывающего и алгоритм проверки. Оба алгоритма получают на вход параметр безопасности n и работают время, ограниченное полиномом от n. Алгоритм P получает на вход, кроме параметра безопасности, ключ d, алгоритм V — ключ e.

Требования таковы:

(1) С вероятностью, приблизительно равной 1, выполнено $V^{P_d}(e) = 1$. Вероятность берется по распределению на паре ключей (e, d) и по исходам случайных бросаний, выполняемых P и V.

(2) Алгоритм P_d разглашает только пару $\langle n, e \rangle$. (Мы предполагаем, что e есть функция от d.)

(3) Для любой неравномерной стратегии F_n вероятность события $V^{F_n(e)}(e) = 1$ пренебрежимо мала. Вероятность берется по случайному выбору e и по исходам случайных бросаний, выполняемых V.

Теорема 26. *Если функция Рабина необратима, то существует протокол аутентификации.*

Доказательство. Пусть $\langle m_n, x_n \rangle$ трудная случайная величина для функции Рабина (напомним, что m_n, x_n — n-битовые натуральные числа, а функция Рабина f_n определена как $f_n(m, x) = (x^2 \bmod m)m$. Закрытый ключ d_n равен паре $\langle m_n, x_n \rangle$ а открытый ключ e_n равен паре $\langle m_n, (x_n^2 \bmod m_n) \rangle$, где пара $\langle m_n, x_n \rangle$ есть трудная случайная величина для функции Рабина.

Алгоритмы P, V работают так.

1) Алгоритм P посылает случайный квадратичный вычет $z = y^2$ по модулю m (с вероятностью пренебрежимо малой он не сможет найти такого вычета, в этом случае он останавливает игру).

2) Алгоритм V посылает случайный бит α.
3) Алгоритм P проверяет, является ли присланное ему одним битом, и, если это так, то посылает $u = yx^\alpha$ (а иначе останавливает игру). Заметим, что $u^2 = zx^{2\alpha} \pmod m$.
4) Алгоритм V проверяет равенство $u^2 = zx^{2\alpha} \pmod m$ (он может это сделать, поскольку знает z, x^2, m) и проверяет, взаимно просты ли u и m. Если что либо из этого не выполнено, он останавливает игру досрочно и выдает 0. Иначе повторяются 1)–4), причем каждая из сторон использует новые случайные биты. Если после n повторений игра не остановлена досрочно, алгоритм V останавливает игру и выдает 1.

Условие (1) очевидно выполнено. Условие (2) проверяется следущим образом. По лемме 24 нам достаточно доказать, что в ходе однократного выполнения пунктов 1)–3) разглашается только m и x^2. Пусть Проверяющий выполняет какую-то неравномерно полиномиальную стратегию F, которая по квадратичному вычету z сообщает бит $F(z)$ (будем сначала предполагать, что $F(z)$ всегда есть один бит). Нам надо, используя F, как оракул, сгенерировать случайную величину, вычислительно неотличимую от случайной величины $\langle y^2, F(y^2), yx^{2F(y^2)}\rangle$. Здесь y выбирается по некоторому распределению, статистически неотличимому от равномерного распределения среди всех вычетов по модулю m. Если мы заменим распределение y на равномерное, то новое распределение на тройках $\langle y^2, F(y^2), yx^{2F(y^2)}\rangle$ будет статистически неотличимо от старого, поэтому достаточно сгенерировать случайную величину, вычислительно неотличимую от $\langle y^2, F(y^2), yx^{2F(y^2)}\rangle$ при равномерном распределении на y.

Назовем тройку $\langle z, \alpha, u\rangle$ удачной, если $F(z) = \alpha$ и $u^2 = z(x^2)^\alpha$ и u взаимно просто с m. Все тройки вида, появляющиеся в беседе удачны: каждому y, выбранному Доказывающим на первом шаге соответствует ровно одна удачная тройка. С другой стороны, никаких других удачных троек нет, поскольку любая удачная тройка $\langle z, \alpha, u\rangle$ появится в беседе, если на первом шаге Доказывающий выберет $y = ux^{-\alpha}$. Более того, такое y очевидно единственно, поэтому количество удачных троек равно функции Эйлера $\phi(m)$ от m (столько разных y, взаимно простых с m), и все удачные тройки появляются в беседе с одинаковыми вероятностями.

Следовательно, нам нужно сгенерировать равномерное распределение на множестве удачных троек (или вычислительно неотличимое от него распределение). Заметим, что в удачной тройке первая компонента однозначна определяется второй и третьей компонетной: $z = u^2x^{-2\alpha}$. Значит, если выбирать случайно пару α, u и выдавать тройку $\langle u^2x^{-2\alpha}, \alpha, u\rangle$,

если она оказалось удачной (то есть, $F(z) = \alpha$ и $(u, m) = 1$), то все удачные тройки будут иметь равные шансы быть выданными.

Какова вероятность в одной попытке найти удачную тройку? Пусть u выбирается равномерно среди всех вычетов, взаимно простых с m, Тогда вероятность того, что тройка $\langle u^2 x^{-2\alpha}, \alpha, u\rangle$ удачна равна

$$\Pr[V(u^2) = 0]/2 + \Pr[V(u^2 x^{-2}) = 1]/2.$$

Второе слагаемое можно заменить на слагаемое $\Pr[V(u^2)) = 1]/2$, так как u^2 и $u^2 x^{-2}$ имеют одинаковое распределение. Поэтому наша вероятность равна

$$\Pr[V(u^2) = 0]/2 + \Pr[V(u^2) = 1]/2 = 1/2.$$

При замене равномерного распределения среди всех вычетов, взаимно простых с m, на статистически неотличимое от него равномерное распределение вероятность изменится незначительно, а значит, будет приблизительно равна $1/2$.

Повторяя n раз попытку найти удачную тройку, мы преуспеем хотя бы в одной попытке с вероятностью, примерно равной 1. В случае неудачи выдадим любую тройку.

Если не предполагать, что F всегда выдает один бит, то можно применить тот же трюк, что и в доказательстве теоремы 25.

Осталось проверить условие (3). Пусть для некоторой неравномерно полиномиальной стратегии F вероятность события $V^{F_n(m,x^2)}(m, x^2) = 1$ не пренебрежимо мала. Напомним, что вероятность берется по случайному выбору m, x по распределению трудному для функции Рабина. То есть, нам известно, что не существует неравномерно полиномиальной функции D_n которая по этому распределению для бесконечно многих n по паре m, x^2 выдает $\sqrt{x^2}$ с вероятностью не менее $1/\mathrm{poly}(n)$. Чтобы получить противоречие, определим такую функцию D_n.

Нам дано, что для бесконечно многих n вероятность события $V^{F_n(m,x^2)}(m, x^2) = 1$ больше $\varepsilon = 1/\mathrm{poly}(n)$. Фиксируем одно из таких n. Для разминки рассмотрим совсем простой случай: вероятность события $V^{F_n(m,x^2)}(m, x^2) = 1$ равна 1. Стратегия F для любых данных m, x^2 сначала посылает Проверяющему некоторый вычет $z = z(m, x^2)$. Затем в зависимости от присланного ему α она присылает некоторый вычет $u_0 = u_0(m, x^2)$ (если $\alpha = 0$) или $u_1 = u_1(m, x^2)$. Причем с вероятностью 1 выполнено $u_\alpha^2 = z x^{2\alpha}$ и u_α взаимно просто с m. Это означает, что с вероятностью 1 выполнено $u_0^2 = z$ и $u_1^2 = z x^2$ и u_0, u_1 взаимно просты с m, а следовательно,

$x^2 = (u_1/u_0)^2$. Таким образом, применив стратегию F к обоим битам $\alpha = 0, 1$ для одного и того же z, мы можем найти $\sqrt{x^2}$. Заметим, что реальный Доказывающий отказался бы в одном раунде отвечать на два разных α. Когда мы обращаем функцию Рабина, мы используем просто схему, реализующую его стратегию, и может делать это так, как запрещено в процессе аутентификации.

Теперь рассмотрим чуть более сложный случай: вероятность события $V^{F_n(m,x^2)}(m, x^2) = 1$ не меньше $3/4$. В этом случае опять достаточно использовать что с вероятностью не менее $3/4$ игра не будет остановлена в первом раунде. Среднее арифметическое вероятностей событий "$u_0^2 = z$ и u_0 взаимно просто с m" и "$u_1^2 = zx^2$ и u_1 взаимно просто с m" не меньше $3/4$. Значит сумма их вероятностей не меньше $3/2$, поэтому вероятность их пересечения не меньше $1/2$. Таким образом, с вероятностью не меньше $1/2$ будет выполнено $x^2 = (u_1/u_0)^2$.

Теперь рассмотрим общий случай. Обозначим через p_1 вероятность того, что игра не остановлена в первом раунде, а через p_i — вероятность того, что игра не остановлена в i-ом раунде при условии, что она не остановлена раньше. Нам дано, что $p_1 p_2 \cdots p_n \geq \varepsilon = 1/\mathrm{poly}(n)$. Значит, существует такое i, для которого $p_i \geq (1/\mathrm{poly}(n))^{1/n}$. Последовательность $(1/\mathrm{poly}(n))^{1/n}$ стремится к 1, поэтому при достаточно больших n она больше $3/4$. Зафиксируем любое такое i, что $p_i > 3/4$ и рассмотрим следующую вероятностную процедуру обращения функции Рабина. Выполняем $i-1$ первых раундов процедуры аутентификации, а в i-ом раунде запускаем F дважды, для $\alpha = 0$ и для $\alpha = 1$. Получив u_0, u_1 проверяем, будут ли они взаимно просты с m и верны ли равенства $u_0^2 = z$, $u_1^2 = zx^2$. Если все это так, то корнем из x^2 будет u_1/u_0. Нам дано, что вероятность этого события при условии, что игра дойдет до i-ого раунда, не меньше $1/2$. Значит безусловная вероятность того, что мы найдем корень из x^2, не меньше $\varepsilon/2$.

В приведенном рассуждении есть неточность — мы не знаем, чему равно i. Ее легко исправить: например, можно пробовать все i. Для каждого i применяем F к к $\alpha = 0$ и к $\alpha = 1$. Если нам не повезло и корень из x^2 не найден, то выбераем α случайно и переходим к следующему раунду. □

11 Протоколы электронной подписи

11.1 Подпись одного бита

Протоколом подписи одного бита называется последовательность доступных случайных величин $\langle d_n, e_n\rangle$ (закрытый и открытый ключи) и пара детерминированных (неинтерактивных) алгоритмов S, V, называемыми алгоритмом подписи и алгоритм проверки, соответственно. Оба алгоритма получают на вход параметр безопасности n и работают время, ограниченное полиномом от n. Алгоритм P получает на вход, кроме параметра безопасности, ключ d и один бит σ и выдает подпись s. Алгоритм V получает на вход, кроме параметра безопасности, ключ e, бит σ и подпись s и выдает 0 или 1 (отвергает или принимает подпись). Требования таковы:

(1) С вероятностью, приблизительно равной 1, для всех $\sigma = 0, 1$ выполнено $V(e, \sigma, S(d, \sigma)) = 1$ (параметр n мы опускаем). Вероятность берется по распределению на паре ключей (e, d).

(2) Мы требуем защищенности протокола от следующей атаки. Противник, изучив открытый ключ e, просит Подписывающего подписать некоторый бит σ. Подписывающий (не подозревая подвоха) дает ему требуемую подпись $s = S(d, \sigma)$. Противник изучает s и выдает поддельную подпись s'. Атака считается успешной, если s' признается правильной подписью под противоположным битом $\overline{\sigma}$, то есть, $V(e, \overline{\sigma}, s') = 1$ Итак, мы требуем, чтобы для любой последовательности схем C_n, D_n полиномиального от n размера вероятность события $V(e, \overline{\sigma}, s') = 1$ было пренебрежимо мала. Здесь $\sigma = C(e)$ и $s' = D(e, s)$, где $s = S(d, \sigma)$ (мы опускаем параметр n).

Теорема 27. *Если существует односторонняя функция, то существует протокол подписи одного бита.*

Доказательство. Пусть f_n односторонняя функция с трудной случайной величиной α_n. В качестве закрытого ключа возьмем пару $\langle x^0, x^1\rangle$, где x^0, x^1 получены независимыми испытаниями случайной величины α_n. В качестве открытого ключа возьмем $\langle y^0, y^1\rangle$, где $y^0 = f_n(x^0)$ и $y^1 = f_n(x^1)$.

Подпись под битом $\sigma = 0$ есть x^0 (первый компонент закрытого ключа), а подпись под $\sigma = 1$ есть x^1 (второй компонент закрытого ключа). Проверяющий применяет f_n к подписи s и принимает подпись, если $f_n(s) = y^\sigma$. Этот протокол удовлетворяет требованию (1). Проверим требование (2).

Допустим, что существуют последовательности схем C_n и D_n полиномиального от n размера, атакующие для бесконечно многих n построенную систему подписи с вероятностью успеха не меньше $\varepsilon = 1/\mathrm{poly}(n)$. Зафиксируем любое из таких n. Противник, применяя схему C_n к открытому ключу y^0, y^1 вычисляет бит $\sigma = C(y^0, y^1)$. Затем, получив от подписывающего x^σ, он применяет схему D_n к открытому ключу и x^σ, получая поддельную подпись $\tilde{s} = D(y^0, y^1, x^\sigma)$. Атака будет успешной, если $y^{\overline{\sigma}} = f(\tilde{s})$. Представим событие $y^{\overline{\sigma}} = f(\tilde{s})$ в виде объединения двух событий

$$C(y^0, y^1) = 1 \ \wedge \ y^0 = f(D(y^0, y^1, x^1)), \tag{2}$$
$$C(y^0, y^1) = 0 \ \wedge \ y^1 = f(D(y^0, y^1, x^0)), \tag{3}$$

соответствующих $\sigma = 0$ и $\sigma = 1$. Вероятность хотя бы одного из этих событий не меньше $\varepsilon/2$. Пусть, скажем, это верно для первого события. Тогда вероятностная схема $D(y, f(x^1), x^1)$ обращает $f(\alpha)$ с вероятностью не менее $\varepsilon/2$. Действительно, вероятность того, что схема $D(y, f(x^1), x^1)$ обращает $f(\alpha)$ равна вероятности того, что $D(y^0, f(x^1), x^1)$ обращает y^0, поскольку y^0 и $f(\alpha)$ одинаково распределены (при любом фиксированном x^1). А вероятность этого не меньше вероятности события (2). Осталось заметить, что вероятностная схема $D(y, f(x^1), x^1)$ имеет полиномиальный от n размер, поскольку функция f вычислима за полиномиальное время. □

11.2 Подпись одного сообщения фиксированной длины

Протоколом подписи одного сообщения длины $p(n)$ (где p полином), называется последовательность доступных случайных величин $\langle d_n, e_n \rangle$ (закрытый и открытый ключи) и пара детерминированных алгоритмов S, V, называемыми алгоритмом подписи и алгоритм проверки, соответственно. Оба алгоритма получают на вход параметр безопасности n и работают время, ограниченное полиномом от n. Алгоритм P получает на вход, кроме параметра безопасности, ключ d и последовательность битов x длины $p(n)$ и выдает подпись s. Алгоритм V получает на вход, кроме параметра безопасности, ключ e, слово x и подпись s и выдает 0 или 1 (отвергает или принимает подпись). Требования таковы:

(1) Для всех x длины $p(n)$ с вероятностью, приблизительно равной 1, выполнено $V(e, x, S(d, x)) = 1$ (параметр n мы опускаем). Вероятность берется по распределению на паре ключей (e, d).

(2) Мы требуем защищенности протокола от следующей атаки. Противник, изучив открытый ключ e, просит Подписывающего подписать некоторое сообщение x. Подписывающий дает ему требуемую подпись $s = S(d, x)$. Противник изучает s и печатает некоторое сообщение x', отличное от x и выдает поддельную подпись s' под ним. Итак, требование состоит в следующем. Для любой последовательности схем C_n, D_n, E_n полиномиального от n размера вероятность события

$$x' \neq x, \qquad V(e, x', s') = 1$$

пренебрежимо мала (опускаем параметр n). Здесь $x = C(e)$, $x' = E(e, s)$ и $s' = D(e, s)$, где $s = S(d, x)$.

Теорема 28. *Если существует протокол подписи одного бита, то для любого полинома p существует протокол подписи одного сообщения длины $p(n)$.*

Доказательство. Пусть $\langle e_n, d_n\rangle$ и S, V данный нам протокол подписи одного бита. Рассмотрим "$p(n)$-ую степень этого протокола", определяемую так: $\langle \tilde{e}_n, \tilde{d}_n\rangle = \langle e_n^1 \dots e_n^{p(n)}, d_n^1 \dots d_n^{p(n)}\rangle$ есть случайная величина, полученая $p(n)$ независимыми испытаниями случайной величины $\langle e_n, d_n\rangle$. Подпись $\tilde{s}$ под сообщением $x_1 \dots x_{p(n)}$ состоит из конкатенаций подписей $s_1, \dots, s_{p(n)}$ с помощью S под битами $x_1, \dots, x_{p(n)}$, причем при подписывании бита x_i используется ключ d_n^i. Алгоритм проверки заключается в проверке всех подписей $s_1, \dots, s_{p(n)}$ с открытыми ключами $e_n^1, \dots, e_n^{p(n)}$. Если все они выдержали проверку, то подпись признается, иначе отвергается.

Условие (1) очевидно выполнено. Условие (2) проверяется так. Нам нужно атакующего $\tilde{C}_n, \tilde{D}_n, \tilde{E}_n$ новую систему подписи преобразовать в атакущего исходную систему. Пусть вероятность успешной атаки $\tilde{C}_n, \tilde{D}_n, \tilde{E}_n$ равна $\varepsilon = 1/\mathrm{poly}(n)$ для бесконечно многих n. Очевидно существует $i \leq p(n)$, для которого с вероятностью не меньше $\varepsilon/p(n)$ атака будет успешной и при этом i-ые биты x' и x различны. Успешность атаки означает, в частности, что $V(e^i, x'[i], s_i) = 1$ (параметр n опускаем). Будем атаковать исходную систему подписи одного бита следующим образом. Нам дан открытый ключ e, причем закрытый ключ, соответствующий

ему, неизвестен. Выбираем случайным образом $p(n)-1$ пару ключей. Даем полученную последовательность из $p(n)-1$ открытых ключей на вход схеме $\tilde{C}$, вставив в нее на i-ое место данный нам открытый ключ e. Схема $\tilde{C}$ выдаст некоторое сообщение x. Подписываем все его биты, кроме i-ого, с помощью известных закрытых ключей. А i-ый бит просим подписать Подписывающего один бит с ключом d (неизвестным нам). Полученную последовательность из n подписей даем на вход схеме D. Она выдаст n поддельных подписей. Оставляем в этой последовательности только i-ую подпись s'_i. Эта подпись с вероятностью не менее $\varepsilon/p(n)$ будет принята Проверяющим V, как подпись под $x'[i] = \overline{x[i]}$ с ключом e.

Описанная атака использует случайные биты (при выборе $p(n)-1$ пар ключей). Зафиксировав их подходящим образом, мы можем получить детерминированные схемы с не меньшей вероятностью успеха. □

Эта система подписи является "одноразовой" в следующем смысле. Имея подписи под сообщениями $00\ldots0$ и $11\ldots1$, можно изготовить подпись под любым сообщением. Как построить схему подписи, которую можно использовать любое полиномиальное число раз (и что это означает), мы обсудим позже.

11.3 Подпись одного сообщения произвольной длины

Протоколом подписи одного сообщения произвольной полиномиальной длины называется последовательность доступных случайных величин $\langle d_n, e_n\rangle$ (закрытый и открытый ключи) и пара алгоритмов S, V, называемыми алгоритмом подписи и алгоритм проверки, соответственно. Алгоритм S вероятностный, он получает на вход параметр безопасности n, ключ d и двоичную строку x выдает за время $\mathrm{poly}(n+|x|)$ подпись s. Алгоритм V детерминированный, он получает на вход параметр безопасности n, ключ e, слово x и подпись s и за время, ограниченное полиномом от $n+|x|$, выдает 0 или 1 (отвергает или принимает подпись). Требования таковы:

(1) Для любой последовательности слов x_n (длина слова x_n должна быть ограничена полиномом от n) с вероятностью, приблизительно равной 1, выполнено $V(e, x, S(d, x)) = 1$ (параметр n мы опускаем). Вероятность берется по распределению на паре ключей (e, d) и по исходам случайных бросаний алгоритма S.

(2) Мы требуем защищенности протокола от следующей атаки. Противник, изучив открытый ключ e, просит Подписывающего подписать некоторое сообщение $x = C(e)$ полиномиальной от n длины. Подписывающий дает ему требуемую подпись $s = S(d, x)$. Противник изучает ее и открытый ключ e и выдает сообщение x' и фальшивую подпись s'. Атака считается успешной, если x' не равно x и подпись s' под x' принята алгоритмом V. Итак, требование состоит в следующем. Для любой последовательности схем C_n, D_n, E_n полиномиального от n размера вероятность события

$$x' \neq x, \qquad V(e, x', s') = 1$$

пренебрежимо мала. Здесь $x = C(e)$, $x' = E(e, s)$, и $s' = D(e, s)$, где $s = S(d, x)$.

Схему электронной подписи одного сообщения произвольной полиномиальной длины можно построить, имея универсальное семейство односторонних хэш-функций и протокол подписи одного сообщения любой фиксированной полиномиальной длины.

11.4 Универсальные семейства односторонних хэш-функций

Говоря неформально, хэш-функцией называется функция $h : \{0,1\}^m \to \{0,1\}^k$, где $k < m$, для которой трудно найти такие $x_1 \neq x_2$, что $h(x_1) = h(x_2)$. Любая такая пара слов $\langle x_1, x_2 \rangle$ называется коллизией функции h. Поскольку $k < m$, коллизии обязательно есть, однако найти их трудно. Чтобы формализовать это понятие, будем говорить не об одной хэш-функции, а о семействе хэш-функций. Семейством хэш-функций называется полиномиально вычислимое отображение $h : \{0,1\}^l \times \{0,1\}^m \to \{0,1\}^k$, где $k < m$. При каждом фиксированном t из $\{0,1\}^l$ получаем функцию $x \mapsto h(t, x)$, которую мы будем обозначать через h_t.

Пусть семейство функций h зависит от натурального параметра n (параметра безопасности) и l, m, k являются полиномами от n, то есть задана последовательность отображений $h^n : \{0,1\}^{l(n)} \times \{0,1\}^{m(n)} \to \{0,1\}^{k(n)}$. Пусть еще имеется доступная случайная величина α_n, принимающая значения в $\{0,1\}^{l(n)}$. Пару, состоящую из h^n и α_n и называют семейством хэш-функций. Говорят, что семейство хэш функций свободно от коллизий, если для любой последовательности схем C_n размера

poly(n) вероятность события "$C_n(\alpha_n)$ есть коллизия для $h^n_{\alpha_n}$" пренебрежимо мала.

Для протоколов электронной подписи, на самом деле, достаточно защищенности от более слабой атаки. А именно, первое слово из коллизии противник должен указать, не зная хэш-функции, к которой подбирается коллизия. Семейство хэш-функций, защищенное от такой атаки, называется универсальным семейством односторонних хэш-функций. А именно, требуется, чтобы для любой последовательности схем C_n размера poly(n) и любой последовательности слов $x_n \in \{0,1\}^{m(n)}$ вероятность события "пара слов $\langle x_n, C_n(\alpha_n)\rangle$ есть коллизия для $h^n_{\alpha_n}$" пренебрежимо мала.

Если существует одностороняя перестановка g_n, множества всех слов некоторой длины $L = \text{poly}(n)$, трудное распределение α_n для которой является равномерным распределением на множестве всех слов длины L, то для любых полиномов $m(n)$ и $k(n)$ можно построить универсальное семейство односторонних хэш-функций $h^n : \{0,1\}^{l(n)} \times \{0,1\}^{m(n)} \to \{0,1\}^{k(n)}$ (для некоторого полинома $l(n)$, зависящего от $m(n)$ и $k(n)$).

11.5 Построение схемы подписи одного сообщения произвольной длины

Пусть имеется схема подписи $\langle\langle e_n, d_n\rangle, S, V\rangle$ одного сообщения длины $n+1$. Пусть длина открытого ключа e_n в этой схеме равна полиному $p(n)$. Пусть, кроме того, имеется универсальное семейство семейство односторонних хэш-функций $h^n : \{0,1\}^{q(n)} \times \{0,1\}^{p(n)} \to \{0,1\}^n$. Длина слова в области определения хэш-функций этого семейства равна длине открытого ключа, а хэш-значения имеют n битов.

Сначала построим систему подписи в которой условие $x' \neq x$ успеха атаки заменено на более сильное условие: ни одно из слов x' и x не является началом другого. Открытый ключ состоит из пары $\langle e, \alpha\rangle$, а закрытый ключ — из пары $\langle d, \alpha\rangle$ (опускаем параметр безопасности). Здесь α — случайная величина, входящая в определение универсального семейства односторонних хэш-функций. Подпись под последовательностью битов $x = \sigma_1 \ldots \sigma_l$ устроена так. Подписывающей выбирает случайно и независимо новые пары ключей $\langle e_2, d_2\rangle, \ldots, \langle e_{l+1}, d_{l+1}\rangle$. Затем подписывает (алгоритмом S) последовательность $h_\alpha(e_2)\sigma_1$ с ключом d, последовательность $h_\alpha(e_3)\sigma_2$ с ключом d_2, и так далее. Последовательность $h_\alpha(e_{i+1})\sigma_i$ подписывается с ключом d_i. Последней подписывается последователь-

ность $h_\alpha(e_{l+1})\sigma_l$ с ключом d_l (ключ e_{l+1} использоваться не будет). Обозначим через $s_1, \ldots, s_l$ полученную последовательность подписей. Общей подписью будет последовательность открытых ключей $e_2, \ldots, e_{l+1}$ и последовательность подписей $s_1, \ldots, s_l$. Алгоритм проверки подписи состоит в применении алгоритма V с ключами $e, e_2, \ldots, e_l$ к сообщениям (соответственно) $h_\alpha(e_2)\sigma_1$, $\ldots$, $h_\alpha(e_l)\sigma_{l-1}$, $h_\alpha(e_{l+1})\sigma_l$ и подписям $s_1, \ldots, s_l$.

Теорема 29. *Построенные пара алгоритмов $\tilde{S}, \tilde{V}$ и случайная величина $\langle \tilde{e}, \tilde{d} \rangle$ являются протоколом электронной подписи одного сообщения произвольной длины.*

Доказательство. Пусть противник атакует построенную систему подписи, используя схемы $\tilde{C}, \tilde{D}, \tilde{E}$. То есть, сначала он вычисляет

$$x = \sigma_1 \ldots \sigma_l = \tilde{C}(\alpha, e).$$

Затем получает подпись $\langle s_1 s_2 \ldots s_l, e_2 e_3 \ldots e_{l+1} \rangle$ под x, где

$$\begin{aligned} s_1 &= S(d, h_\alpha(e_2)\sigma_1), \\ s_2 &= S(d_2, h_\alpha(e_3)\sigma_2), \\ &\ldots \\ s_l &= S(d_l, h_\alpha(e_{l+1})\sigma_l). \end{aligned}$$

После этого он сам вычисляет

$$x' = \sigma'_1 \ldots \sigma'_{l'} = \tilde{E}(\alpha, e, s_1 s_2 \ldots s_l)$$

и фальшивую подпись

$$s' = \langle s'_1 s'_2 \ldots s'_{l'}, e'_2 e'_3 \ldots e'_{l'+1} \rangle = \tilde{D}(\alpha, e, s_1 s_2 \ldots s_l).$$

Атака успешна, если

$$\begin{aligned} x' &\neq x, \\ V(e, h_\alpha(e'_2)\sigma'_1, s'_1) &= 1, \\ V(e_2, h_\alpha(e'_3)\sigma'_2, s'_2) &= 1, \\ &\ldots \\ V(e_{l'}, h_\alpha(e_{l'+1})\sigma'_{l'}, s'_{l'}) &= 1. \end{aligned}$$

Допустим для бесконечно многих почти всех n вероятность этого события больше $\varepsilon = 1/\mathrm{poly}(n)$.

Лемма 30. *Если атака успешна, то найдется такое* $j \leq \min\{l, l'\}$, *что* $e_j \neq e'_j$, *но* $h_\alpha(e_j) = h_\alpha(e'_j)$, *или найдется такое* $j \leq \min\{l, l'\}$, *что* $e_j = e'_j$, *но* $\sigma_j h_\alpha(e_{j+1}) \neq \sigma'_j h_\alpha(e_{j+1})$.

В формулировке этой леммы мы предполагаем, что $e_1 = e'_1 = e$. Эта лемма утверждает, что у противника есть только две возможности подделать подпись в новой системе: первая заключается в том, чтобы отыскать e'_j отличное от e_j с тем же хэш-значением, что и у e_j. Вторая возможность — подделать подпись в старой системе под сообщением $\sigma'_j h_\alpha(e'_{j+1})$, отличным от сообщения $\sigma_j h_\alpha(e_{j+1})$, подписанного владельцем секретного ключа d_j.

Доказательство. Допустим, что утверждение леммы неверно. В частности, оно неверно для $j = 1$. Поскольку $e_1 = e'_1 = e$, мы можем заключить, что $\sigma_1 h_\alpha(e_2) = \sigma'_1 h_\alpha(e'_2)$, то есть σ_1 и σ'_1 совпадают и $h_\alpha(e_2) = h_\alpha(e'_2)$. Пользуясь тем, что оно неверно для $j = 2$, выводим, что $e_2 = e'_2$. Рассуждая по индукции, мы установим, что первые $m = \min\{l, l'\}$ битов x и x' совпадают и $e_{m+1} = e'_{m+1}$. Это означает, что одно из слов x и x' является началом другого, то есть, атака не является успешной. □

По лемме с вероятностью не меньше $\varepsilon/2$ то найдется такое $j \leq \min\{l, l'\}$, что $e_j \neq e'_j$, но $h_\alpha(e_j) = h_\alpha(e'_j)$, или найдется такое $j \leq \min\{l, l'\}$, что $e_j = e'_j$, но $\sigma_j h_\alpha(e_{j+1}) \neq \sigma'_j h_\alpha(e'_{j+1})$ и $V(e'_j, \sigma'_j h_\alpha(e'_{j+1}, s'_j) = 1$. В первом случае, мы можем успешно атаковать семейство хэш-функций. Поскольку j ограничено некоторым полиномом $r(n)$ для некоторого фиксированного $j \leq r(n)$ с вероятностью не менее $\varepsilon/2r(n)$ будет выполнено $j \leq \min\{l, l'\}$, $e_j \neq e'_j$, но $h_\alpha(e_j) = h_\alpha(e'_j)$. Сгенерируем первый элемент коллизии, взяв случайный ключ e_j. Получив случайный номер хэш-функции α, сгенерируем случайную пару ключей e, d и дадим ее противнику, использующему схемы $\tilde{C}, \tilde{D}, \tilde{E}$. Он попросит нас подписать неоторое 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. Этот ключ и будет с вероятностью не меньше $\varepsilon/2r(n)$ вторым элементом коллизии h_α. Описанная атака на семейство хэш-функций вычисляется вероятностными схемами полиномиального размера. Зафиксировав в этих схемах подходящим образом используемые случайные биты, получим детермирированные атакующие схемы полиномиального размера.

Пусть теперь с вероятностью не меньше $\varepsilon/2$ найдется такое $j \leq \min\{l, l'\}$, что $e_j = e'_j$, но $\sigma_j h_\alpha(e_{j+1}) \neq \sigma'_j h_\alpha(e_{j+1})$. Опять, это событие имеет место с вероятностью не менее $\varepsilon/2r(n)$ для некоторого фиксированного $j \leq r(n)$. Будем атаковать систему подписи одного сообщения длины $p(n)$. Пусть нам дан открытый ключ, обозначим его через e_j (он будет использован как j-ый ключ в подписи). Выберем случайно пару ключей e, d и хэш-функцию h_α. Затем дадим ключ $\tilde{e} = \langle e, \alpha \rangle$ на вход схеме $\tilde{C}$. Подпишем выданное схемой сообщение $\sigma_1 \dots \sigma_l$, генерируя новые пары ключей $e_2, d_2, \dots, e_l, d_l$ (кроме j-ой пары). А подпись под $\sigma_j h_\alpha(e_{j+1})$ получим у Подписывающего одно сообщение длины $p(n)$ закрытим ключом d_j. Все подписи даем на вход схемам $\tilde{D}, \tilde{E}$. Схемы выдадут новое сообщение $\sigma'_1 \dots \sigma'_{l'}$ и фальшивую подпись $\langle s'_1 s'_2 \dots s'_{l'}, e'_2 e'_3 \dots e'_{l'+1} \rangle$, про которые известно, что с вероятностью не менее $\varepsilon/2r(n)$ выполнено $e_j = e'_j$, $\sigma_j h_\alpha(e_{j+1}) \neq \sigma'_j h_\alpha(e'_{j+1})$ и $V(e'_j, \sigma'_j h_\alpha(e'_{j+1}), s'_j) = 1$. То есть, $\sigma'_j h_\alpha(e'_{j+1}$ является сообщением, отличным от того, которое мы просили подписать, и подпись s'_j под которым принимается с ключом $e_j = e'_j$.

Осталось построить схему в которой условие $x' \neq x$ успеха атаки не заменено на более сильное условие. Определим для каждого двоичного слова y его префиксный код $\hat{y}$ следующим образом. Удвоим все биты y и припишем 01 в конце, например, $\widehat{011} = 00111101$. По y легко найти $\hat{y}$, и если одно из слов вида $\hat{y}$ является началом другого слова такого же вида, то эти слова совпадают. Модифицируем построенную систему подписи следующим образом. Будем вместо подписывания слова y подпишем его префиксный код $\hat{y}$ (и при проверке будем проверять, подписано ли $\hat{y}$). Если два слова вида вида $\hat{y}$ различны, то ни одно из них не является началом другого. Поэтому модифицированная система подписи удовлетворяет требованию (2) в его исходном виде. □

11.6 Общий случай: подпись произвольного количества сообщений произвольной длины

Протоколом подписи (произвольного количества сообщений произвольной длины) называется последовательность доступных случайных величин $\langle d_n, e_n \rangle$ (закрытый и открытый ключи) и пара алгоритмов S, V, называемыми алгоритмом подписи и алгоритм проверки, соответственно. Алгоритм S вероятностный, он получает на вход параметр безопасности n, ключ d и двоичную строку x выдает за время $\mathrm{poly}(n + |x|)$ подпись s.

Алгоритм V детерминированный, он получает на вход параметр безопасности n, ключ e, слово x и подпись s и за время, ограниченное полиномом от $n+|x|$, выдает 0 или 1 (отвергает или принимает подпись). Требования таковы:

(1) Для любой последовательности слов x_n (длина слова x_n должна быть ограничена полиномом от n) с вероятностью, приблизительно равной 1, выполнено $V(e, x, S(d, x)) = 1$ (параметр n мы опускаем). Вероятность берется по распределению вероятностей на паре ключей (e, d) и по бросаниям монетки, выполняемым алгоритмом S.

(2) Мы требуем защищенности протокола от следующей атаки. Противник, изучив открытый ключ e, просит Подписывающего подписать некоторое сообщение $x_1 = C(e)$. Подписывающий дает ему требуемую подпись $s_1 = S(d, x_1)$. Противник изучает ее и просит Подписывающего подписать другое сообщение $x_2 = C(e, s_1)$. Подписывающий опять дает ему требуемую подпись $s_2 = S(d, x_2)$. Так продолжается k раз, где k ограничено полиномом от n. Изучив все k подписей $s_1, \ldots, s_k$ противник выдает сообщение $x_{k+1} = C(e, s_1 \ldots s_k)$ и фальшивую подпись $s_{k+1} = D(e, s_1 \ldots s_k)$. Атака считается успешной, если x_{k+1} не равно ни одному из сообщений $x_1, \ldots, x_k$ и подпись s_{k+1} под ним принята алгоритмом V. Итак, требование состоит в следующем. Для любой неравномерно полиномиальной стратегии C_n, любой неравномерно полиномиальной функции D_n и для любого полинома p для почти всех n вероятность события

$$x_{k+1} \notin \{x_1, \ldots, x_k\}, \qquad V(e, x_{k+1}, s_{k+1}) = 1$$

пренебрежимо мала. Здесь $k = p(n)$, а $x_1, \ldots, x_k, x_{k+1}, s_{k+1}$ определяются индуктивно (мы опускаем параметр n):

$$\begin{aligned} x_1 &= C(e), \quad s_1 = S(d, x_1), \\ x_2 &= C(e, s_1), \quad s_2 = S(d, x_2), \\ &\ldots \\ x_k &= C(e, s_1, \ldots, s_{k-1}), \quad s_k = S(d, x_k), \\ x_{k+1} &= C(e, s_1, \ldots, s_k), \quad s_{k+1} = D(e, s_1, \ldots, s_k). \end{aligned}$$

Построить протокол, удовлетворяющий этим требованиям, не удается в каких-либо достаточно естественных общих предположениях. Поэтому мы немного ослабим требования, разрешив подписи под очередным сообщением x_i зависеть не только от d и x_i, но и от всех предыдущих

сообщений и от случайных битов, использованных при подписи предыдущих сообщений. Таким образом, мы будем рассматривать алгоритм подписи S как детерминированный алгоритм с дополнительным входом, на который подается бесконечная равномерно распределенная строка r. Алгоритм S подписывая x_i использует $d, x_1, \ldots, x_i$ и за полиномиальное от $n + |x_1| + \cdots + |x_i|$ время вырабатывает подпись. Тем самым, он использует не более $\mathrm{poly}(n + |x_1| + \cdots + |x_i|)$ первых битов r. Требования (1) и (2) теперь записываются так.

(1) Для любого полинома p и последовательности сообщений $x_1^n, \ldots, x_{k_n}^n$, $k \le p(n)$, $|x_i^n| \le p(n)$, с вероятностью, приблизительно равной 1, для всех $i \le k_n$ выполнено $V(e, x_i, S(r, d, x_1, \ldots, x_i)) = 1$ (параметр n мы опускаем). Вероятность берется по распределению на паре ключей (e, d) и равномерному распределению на r.

(2) Для любой неравномерно полиномиальной стратегии C_n, любой неравномерно полиномиальной функции D_n и для любого полинома p для почти всех n вероятность события

$$x_{k+1} \notin \{x_1, \ldots, x_k\}, \qquad V(e, x_{k+1}, s_{k+1}) = 1$$

пренебрежимо мала. Здесь $k = p(n)$, а $x_1, \ldots, x_k, x_{k+1}, s_{k+1}$ определяются индуктивно (мы опускаем параметр n):

$$\begin{aligned} x_1 &= C(e), \quad s_1 = S(r, d, x_1), \\ x_2 &= C(e, s_1), \quad s_2 = S(r, d, x_1, x_2), \\ &\ldots \\ x_k &= C(e, s_1, \ldots, s_{k-1}), \quad s_k = S(r, d, x_1, \ldots, x_k), \\ x_{k+1} &= C(e, s_1, \ldots, s_k), \quad s_{k+1} = D(e, s_1, \ldots, s_k). \end{aligned}$$

Протокол, удовлетворяющий этим требованиям существует в тех же предположениях, что и протокол подписи одного сообщения произвольной длины. А именно, нам понадобятся схема подписи $\langle e_n, d_n\rangle$, S, V одного сообщения, состоящего из $2n$ битов, схема подписи $\langle e'_n, d'_n\rangle$, S', V' одного бита, и универсальные семейства односторонних хэш-функций $h^n : \{0,1\}^{q(n)} \times \{0,1\}^{p(n)} \to \{0,1\}^n$, $g^n : \{0,1\}^{q'(n)} \times \{0,1\}^{p'(n)} \to \{0,1\}^n$, где $p(n)$ и $p'(n)$ длины ключей e_n и e'_n, соответственно. Модифицируем протокол $\tilde{S}, \tilde{V}$ из предыдущего раздела следующим образом. Исходный открытый ключ теперь является четверкой $\langle \alpha, \alpha', e, e'\rangle$, где α, α' случайные величины из определения семейства хэш-функций. Закрытый ключ равен $\langle \alpha, \alpha', d, d'\rangle$, Алгоритм S' будем использовать, чтобы

подписывать биты сообщений, а алгоритм S — чтобы подписывать хэш-значения $h_\alpha(e_2), h_{\alpha'}(e'_2), h_\alpha(e_3), h_{\alpha'}(e'_3), \ldots$ от новых ключей $e_2, e'_2, e_3, e'_3, \ldots$. Подписывая биты префиксного кода $\widehat{x_1}$ первого сообщения, мы на последнем шаге сгенерируем новую четверку ключей $d_{l+1}, e_{l+1}, d'_{l+1}, e'_{l+1}$. Раньше последняя пара ключей никак не использовалась, но теперь мы будем использовать эту четверку в качестве начальных ключей при подписи второго сообщения. Подпись второго сообщения получается так же, как и первого. Кроме того, к ней добавляются все ключи $e_2, e_3, \ldots$ и подписи под их хэш-значениями $h_\alpha(e_2), h_\alpha(e_3), \ldots$.

Проверка подписи происходит так. Получив сообщение x_i, ключи

$$e_2, \ldots, e_m, e_{m+1}, e'_{m+1}, \ldots, e_{m+l}, e'_{m+l}$$

(где $l = |x_i|$) и подписи

$$s_1, s_2 \ldots s_m, s_{m+1}, s'_{m+1}, \ldots, s_{m+l}, s'_{m+l},$$

мы сначала проверяем алгоритмом V все подписи $s_1, s_2 \ldots s_m, s_{m+1}, \ldots, s_{m+l}$ под хэш-значениями всех ключей. Затем, будучи уверенными в аутентичности ключей $e'_{m+1}, \ldots, e'_{m+l}$, мы проверяем алгоритмом V' подписи под битами $\widehat{x_i}$.

Список литературы

[1] Johan Håstad, Russell Impagliazzo, Leonid A. Levin, Michael Luby: A Pseudorandom Generator from any One-way Function. SIAM J. Comput. (SIAMCOMP) 28(4):1364-1396 (1999)

[2] Eric Bach: How to Generate Factored Random Numbers. SIAM J. Comput. 17(2): 179-193 (1988)