Конспект лекций курса «Математическая криптография»

М. И. Анохин

15 июля 2019 г.

0. Основные обозначения, определения и соглашения

Мы будем всюду использовать следующие обозначения (кроме общепринятых):

$\mathbb{N}$ — множество всех целых неотрицательных чисел;

$\mathbb{Z}_m$ — множество $\{0, \ldots, m-1\}$, являющееся ассоциативным коммутативным кольцом с единицей относительно арифметических операций по модулю m ($m \in \mathbb{N} \setminus \{0\}$); аддитивная группа этого кольца также будет обозначаться через $\mathbb{Z}_m$;

$\mathbb{Z}_m^\star$ — группа всех обратимых элементов кольца $\mathbb{Z}_m$, состоящая из всех $x \in \mathbb{Z}_m$, взаимно простых с m ($m \in \mathbb{N} \setminus \{0\}$);

$\mathbb{R}_+$ — множество всех вещественных неотрицательных чисел;

$n \bmod m$ — остаток от деления числа $n \in \mathbb{Z}$ на число $m \in \mathbb{N} \setminus \{0\}$;

$g^k \bmod m$ — результат возведения элемента $g \in \mathbb{Z}_m^\star$ в степень $k \in \mathbb{Z}$ в группе $\mathbb{Z}_m^\star$;

$\operatorname{ord} g$ — порядок элемента g некоторой группы;

X^n — n-я декартова степень множества X (в частности, $\{0,1\}^n$ — множество всех двоичных строк длины n, а X^0 — множество, состоящее лишь из пустого набора);

$X^{\leqslant m} = \bigcup_{n=0}^{m} X^n$ (в частности, $\{0,1\}^{\leqslant m}$ — множество всех двоичных строк длины не более m);

$X^{\geqslant m} = \bigcup_{n=m}^{\infty} X^n$ (в частности, $\{0,1\}^{\geqslant m}$ — множество всех двоичных строк длины не менее m);

$X^* = \bigcup_{n=0}^{\infty} X^n$ (в частности, $\{0,1\}^*$ — множество всех двоичных строк);

$|X|$ — мощность конечного множества X;

$\operatorname{Fun}(X, Y)$ — множество всех функций из множества X в множество Y;

$\operatorname{Per}(X)$ — множество всех перестановок множества X (т. е. биекций из X на X);

$g(f)$ — композиция функций f и g (в этом порядке), т. е. $x \mapsto g(f(x))$ ($f\colon X \to Y$, $g\colon Y \to Z$);

$f(n) = \Omega(g(n))$ при $n \in N$ означает, что $g = O(f)$ при $n \in N$ ($f, g\colon N \to \mathbb{R}_+$, где N — бесконечное подмножество $\mathbb{N}$);

$f(n) = \omega(g(n))$ при $n \in N$ означает, что $g = o(f)$ при $n \in N$ ($f, g\colon N \to \mathbb{R}_+$, где N — бесконечное подмножество $\mathbb{N}$);

$\log$ — логарифм по произвольному основанию, большему 1 (в случаях, когда выбор такого основания несущественен);

$\oplus$ и $\odot$ — операции побитового сложения по модулю 2 и побитового умножения по модулю 2 соответственно (для двоичных строк одинаковой длины);

0^n и 1^n — строки из n нулей и n единиц соответственно ($n \in \mathbb{N}$);

$x_{[i]}$ — i-й бит строки $x \in \{0,1\}^*$;

$x_{[i,\ldots,j]}$ — строка $x_{[i]}x_{[i+1]}\ldots x_{[j]}$, если $1 \leqslant i \leqslant j \leqslant |x|$, и пустая строка в противном случае ($x \in \{0,1\}^*$);

$|x|$ — длина строки $x \in \{0,1\}^*$;

xy — конкатенация двоичных строк x и y;

$\overline{b}$ — бит, противоположный биту b;

$\operatorname{supp} \mathcal{X}$ — носитель распределения вероятностей $\mathcal{X}$ на некотором конечном или счетном множестве;

$\operatorname{supp} \widetilde{x}$ — носитель случайной величины $\widetilde{x}$, принимающей значения в некотором конечном или счетном множестве;

$\widetilde{v}_1, \ldots, \widetilde{v}_k \leftarrow \mathcal{X}$ означает, что $\widetilde{v}_1, \ldots, \widetilde{v}_k$ — независимые случайные величины, имеющие распределение вероятностей $\mathcal{X}$;

$x_1, \ldots, x_k \leftarrow \mathcal{X}$ означает, что $x_1, \ldots, x_k$ — случайные элементы, выбранные относительно распределения вероятностей $\mathcal{X}$ независимо друг от друга;

$x_1, \ldots, x_k \leftarrow \widetilde{x}$ означает, что $x_1, \ldots, x_k$ — случайные значения случайной величины $\widetilde{x}$, выбранные независимо друг от друга;

$\mathcal{U}(X)$ — равномерное распределение на непустом конечном множестве X;

$\widetilde{v}_1, \ldots, \widetilde{v}_k \in_{\mathcal{U}} X$ означает, что $\widetilde{v}_1, \ldots, \widetilde{v}_k$ — независимые случайные величины, распределенные равномерно на непустом конечном множестве X (другими словами, это обозначение эквивалентно $\widetilde{v}_1, \ldots, \widetilde{v}_k \leftarrow \mathcal{U}(X)$);

$x_1, \ldots, x_k \in_{\mathcal{U}} X$ означает, что $x_1, \ldots, x_k$ — случайные элементы непустого конечного множества X, выбранные относительно равномерного распределения вероятностей на этом множестве независимо друг от друга (другими словами, это обозначение эквивалентно $x_1, \ldots, x_k \leftarrow \mathcal{U}(X)$);

$\widetilde{u}_n$ — случайная величина, распределенная равномерно на $\{0,1\}^n$ (т. е. $\widetilde{u}_n \in_{\mathcal{U}} \{0,1\}^n$), где $n \in \mathbb{N}$;

$\mathcal{X}_1 \times \cdots \times \mathcal{X}_k$ — произведение распределений вероятностей $\mathcal{X}_1, \ldots, \mathcal{X}_k$ на конечных или счетных множествах, т. е. распределение случайной величины $(\widetilde{v}_1, \ldots, \widetilde{v}_k)$, где случайные величины $\widetilde{v}_1, \ldots, \widetilde{v}_k$ независимы и имеют распределения вероятностей $\mathcal{X}_1, \ldots, \mathcal{X}_k$ соответственно;

$\mathcal{X}^n$ — n-я степень распределения вероятностей $\mathcal{X}$ на некотором конечном или счетном множестве, т. е. $\mathcal{X} \times \cdots \times \mathcal{X}$ (n множителей); в частности, $\mathcal{X}^0$ — единственное распределение на одноэлементном множестве $(\operatorname{supp} \mathcal{X})^0$.

Как обычно, мы предполагаем, что все дискретные объекты, с которыми работают алгоритмы, отождествляются со строками из $\{0,1\}^*$. При этом *длиной* объекта называется длина соответствующей строки. В частности, если не оговорено противное, целые числа предполагаются заданными своими двоичными записями наименьшей длины.

Для удобства мы называем *полиномом* произвольную функцию $p\colon \mathbb{N} \to \mathbb{N} \setminus \{0\}$ такую, что $p(n) = cn^d$ для всех $n \in \mathbb{N} \setminus \{0\}$, где $c \in \mathbb{N} \setminus \{0\}$ и $d \in \mathbb{N}$ ($p(0)$ может быть произвольным целым положительным числом). Запись poly в (не)равенствах обозначает некоторый полином, существование которого подразумевается. Например, запись $q(n) \leqslant \operatorname{poly}(n)$ означает, что существует полином p (разумеется, не зависящий от n) такой, что $q(n) \leqslant p(n)$. Отметим, что записи poly в разных местах, вообще говоря, обозначают разные полиномы.

Случайные величины, как правило, обозначаются символами с тильдой, а распределения вероятностей — рукописными буквами. Если не оговорено противное, то вводимые случайные величины предполагаются независимыми от всех других. Выбор случайных элементов также осуществляется независимо, если противное не оговорено явно и не вытекает из контекста.

Под словом «алгоритм» мы понимаем машину Тьюринга. *Оракулом* в теории вычислений называется внешнее (по отношению к алгоритмам) устройство, которое в ответ на запрос произвольного алгоритма выдает ему значение некоторой функции на этом запросе. При этом как обращение к оракулу, так и получение от него ответа занимают один такт работы алгоритма. Ответ оракула на запрос может не определяться этим запросом однозначно, а представлять собой значение некоторой случайной величины, зависящей от запроса как от параметра. В этом случае оракул называется *вероятностным*.

Алгоритм A называется *вероятностным*, если он имеет доступ к вероятностному оракулу RB, дающему независимые случайные биты $b \in_{\mathcal{U}} \{0,1\}$. Говоря неформально, вероятностный алгоритм в процессе вычисления может подбрасывать симметричную монету и продолжать вычисления в зависимости от результата подбрасывания. Таким образом, выходное значение вероятностного алгоритма A при вычислении на входе x является случайной величиной, обозначаемой через $A(x)$. Если алгоритм не делает запросов к RB, то он называется *детерминированным*. Детерминированные алгоритмы мы также относим к числу вероятностных. Для детерминированного алгоритма A значение $A(x)$ при произвольном входе x определено однозначно.

Если число запросов к RB вероятностного алгоритма A при вычислении на произвольном входе x не превосходит некоторого числа $\rho(x) \in \mathbb{N}$, то алгоритм A можно рассматривать как детерминированный, но имеющий дополнительный вход, на который подается случайная строка $r \in_{\mathcal{U}} \{0,1\}^{\rho(x)}$, где x — основной вход. А именно, алгоритм A использует $r_{[i]}$ вместо ответа оракула RB на i-й по счету запрос. Разумеется, на дополнительный вход можно подавать произвольную двоичную строку

длины $\rho(x)$. Выходное значение вероятностного алгоритма A при вычислении на входе x с использованием строки r, поданной на дополнительный вход, будет обозначаться через $A(x;r)$.

Алгоритм A (вообще говоря, вероятностный) называется *полиномиальным* или *работающим за полиномиальное время* (polynomial-time algorithm), если время работы (т. е. число тактов до остановки) алгоритма A на любом входе x не превосходит $\mathrm{poly}(|x|)$. Если A — произвольный полиномиальный вероятностный алгоритм, то очевидно, что $|y| \leqslant \mathrm{poly}(|x|)$ для любого входа x и любого $y \in \operatorname{supp} A(x)$. (Здесь полиномы poly могут зависеть от A, но не от x и y.) Функция $f\colon X \to \{0,1\}^*$, где $X \subseteq \{0,1\}^*$, называется *полиномиально вычислимой* или *вычислимой за полиномиальное время* (polynomial-time computable), если существует полиномиальный детерминированный алгоритм E такой, что $E(x) = f(x)$ для всех $x \in X$. Семейство функций $(f_i\colon X_i \to \{0,1\}^* \mid i \in I)$, где $I \subseteq \{0,1\}^*$ и $X_i \subseteq \{0,1\}^*$ для каждого $i \in I$, называется *полиномиально вычислимым* или *вычислимым за полиномиальное время* (polynomial-time computable), если полиномиально вычислима функция $(i,x) \mapsto f_i(x)$, где $i \in I$ и $x \in X_i$. Для семейств функций вида $(g_{n,d}\colon Y_{n,d} \to \{0,1\}^* \mid n \in N,\ d \in D_n)$, где $N \subseteq \mathbb{N}$ и $D_n, Y_{n,d} \subseteq \{0,1\}^*$ при всех $n \in N$ и $d \in D_n$, *полиномиальная вычислимость* или *вычислимость за полиномиальное время* (polynomial-time computability) означает, что полиномиально вычислима функция $(1^n, d, y) \mapsto g_{n,d}(y)$, где $n \in \mathbb{N}$, $d \in D_n$ и $y \in Y_{n,d}$.

Если A — алгоритм (вообще говоря, вероятностный), то запись вида $A^{O_1,\ldots,O_k}$ означает, что A имеет доступ к оракулам $O_1, \ldots, O_k$. При этом вышеуказанный оракул RB не будет указываться таким образом. Кроме того, оракул, возвращающий в ответ на запрос y значение некоторой (вообще говоря, вероятностной) функции на y, обозначается так же, как эта функция.

Если в одну формулу входят несколько одинаковых символов случайной величины, то они обозначают *одну и ту же* случайную величину. Например, пусть R — бинарный предикат на некотором конечном или счетном множестве X, а $\widetilde{x}$ — случайная величина, принимающая значения в X. Тогда $\Pr[R(\widetilde{x},\widetilde{x}) = 1]$ обозначает то же самое, что $\Pr[\widetilde{x} \in \{x \in X \mid R(x,x) = 1\}]$.

Пусть $\mathcal{D}$ и $\mathcal{E}$ — распределения вероятностей на некотором конечном или счетном множестве X. Тогда *статистическим расстоянием* (statistical distance) или *расстоянием по вариации* (variation distance) между этими распределениями называется число

$$\frac{1}{2}\sum_{x\in X}|\Pr_{\mathcal{D}}[\{x\}] - \Pr_{\mathcal{E}}[\{x\}]|,$$

совпадающее с

$$\max_{S\subseteq X}|\Pr_{\mathcal{D}}[S] - \Pr_{\mathcal{E}}[S]| = \max_{S\subseteq X}\left(\Pr_{\mathcal{D}}[S] - \Pr_{\mathcal{E}}[S]\right) = \max_{S\subseteq X}\left(\Pr_{\mathcal{E}}[S] - \Pr_{\mathcal{D}}[S]\right).$$

Статистическое расстояние между распределениями вероятностей $\mathcal{D}$ и $\mathcal{E}$ будет обозначаться через $\Delta(\mathcal{D},\mathcal{E})$. *Статистическим расстоянием* (statistical distance) или *расстоянием по вариации* (variation distance) между случайными величинами $\widetilde{v}$ и $\widetilde{w}$, принимающими значения в X, называется статистическое расстояние между распределениями вероятностей этих случайных величин. Это расстояние обозначается через $\Delta(\widetilde{v},\widetilde{w})$. Легко получить следующие свойства статистического расстояния:

- Δ является метрикой на множестве всех распределений вероятностей на множестве X.
- Если $\widetilde{v}$ и $\widetilde{w}$ — случайные величины, принимающие значения в X, а $\widetilde{r}$ — случайная величина, принимающая значения в некотором конечном или счетном множестве Y, причем $\widetilde{v}$ и $\widetilde{r}$ независимы и $\widetilde{w}$ и $\widetilde{r}$ независимы, то $\Delta((\widetilde{v},\widetilde{r}),(\widetilde{w},\widetilde{r})) = \Delta((\widetilde{r},\widetilde{v}),(\widetilde{r},\widetilde{w})) = \Delta(\widetilde{v},\widetilde{w})$. На языке распределений вероятностей этот факт формулируется короче: если $\mathcal{F}$ — распределение вероятностей на Y, то $\Delta(\mathcal{D}\times\mathcal{F},\mathcal{E}\times\mathcal{F}) = \Delta(\mathcal{F}\times\mathcal{D},\mathcal{F}\times\mathcal{E}) = \Delta(\mathcal{D},\mathcal{E})$.
- Если $\widetilde{v}$ и $\widetilde{w}$ — случайные величины, принимающие значения в X, а f — функция на X, то $\Delta(f(\widetilde{v}),f(\widetilde{w})) \leqslant \Delta(\widetilde{v},\widetilde{w})$.

Пусть N — бесконечное подмножество $\mathbb{N}$. Функция $\nu\colon N \to \mathbb{R}_+$ называется *пренебрежимо малой* (negligible), если для любого полинома p существует число $n_0 \in \mathbb{N}$ такое, что $\nu(n) \leqslant 1/p(n)$ при всех $n \in N$, удовлетворяющих неравенству $n \geqslant n_0$. Другими словами, вышеуказанная функция ν пренебрежимо мала, если и только если для любого полинома p неравенство $\nu(n) \leqslant 1/p(n)$ выполняется при всех достаточно больших $n \in N$. Говоря неформально, функция ν пренебрежимо мала, если она убывает быстрее, чем $1/p$ для любого полинома p. Если функция ν пренебрежимо мала, то говорят также, что величина $\nu(n)$ пренебрежимо мала как функция от $n \in N$. Запись

negl в (не)равенствах обозначает некоторую пренебрежимо малую функцию, существование которой подразумевается. (Не)равенства, содержащие $\mathrm{negl}(n)$, предполагаются выполненными для всех возможных значений n. Например, запись $\mu(n) \leqslant 1/2 + \mathrm{negl}(n)$ означает, что существует пренебрежимо малая функция ν (разумеется, не зависящая от n) такая, что $\mu(n) \leqslant 1/2+\nu(n)$ для всех $n \in N$. Отметим, что записи negl в разных местах, вообще говоря, обозначают разные пренебрежимо малые функции. Примером пренебрежимо малой функции является $n \mapsto 2^{-cn^d}$, где c и d — произвольные положительные константы.

Пусть $I \subseteq \{0,1\}^*$. Семейство распределений вероятностей $(\mathcal{D}_i \,|\, i \in I)$ на множестве $\{0,1\}^*$ называется *полиномиально конструируемым* (polynomial-time samplable или polynomial-time constructible), если существует полиномиальный вероятностный алгоритм G такой, что для любого $i \in I$ случайная величина $G(i)$ имеет распределение $\mathcal{D}_i$. Семейство случайных величин $(\widetilde{x}_i \,|\, i \in I)$, принимающих значения в множестве $\{0,1\}^*$, называется *полиномиально конструируемым* (polynomial-time samplable или polynomial-time constructible), если полиномиально конструируемым является семейство распределений вероятностей случайных величин $\widetilde{x}_i$ $(i \in I)$.

В описаниях полиномиальных алгоритмов в литературе часто встречаются действия вида $x \in_{\mathcal{U}} X$, где X — непустое конечное множество, заданное в некотором смысле эффективно (например, $X = \{1, \ldots, m\}$, $X = \mathbb{Z}_m^\star$ и т. п.). Но если $|X|$ не является степенью двойки, то это действие невыполнимо с помощью вероятностного алгоритма, использующего ограниченное (для данного множества X) число случайных битов (см. также [Sho08, упр. 9.4]). В то же время во многих важных случаях за полиномиальное от $\log|X|$ время можно породить распределение вероятностей такое, что статистическое расстояние между этим распределением и равномерным распределением на X не превосходит $2^{-q(\log|X|)}$, где q — произвольный наперед заданный полином (см. также [Sho08, разд. 9.3]). В этих случаях, допуская вольность речи, мы считаем, что вышеуказанные действия выполнимы за полиномиальное время.

Пусть I — бесконечное подмножество $\{0,1\}^*$, а $(\widetilde{x}_i \,|\, i \in I)$ и $(\widetilde{y}_i \,|\, i \in I)$ — семейства случайных величин, принимающих значения в одном и том же конечном или счетном множестве X. Тогда эти семейства называются *статистически неотличимыми* (statistically close), если $\Delta(\widetilde{x}_i, \widetilde{y}_i) \leqslant \mathrm{negl}(|i|)$.

Предположим теперь, что $\widetilde{x}_i$ и $\widetilde{y}_i$ принимают значения в $\{0,1\}^*$ для любого $i \in I$. Тогда семейства случайных величин $(\widetilde{x}_i \,|\, i \in I)$ и $(\widetilde{y}_i \,|\, i \in I)$ называются *вычислительно неотличимыми* (computationally indistinguishable) или *неотличимыми за полиномиальное время* (polynomial-time indistinguishable), если для любого полиномиального вероятностного алгоритма D

$$|\Pr[D(i, \widetilde{x}_i) = 1] - \Pr[D(i, \widetilde{y}_i) = 1]| \leqslant \mathrm{negl}(|i|).$$

Определение статистической (вычислительной) неотличимости естественным образом переносится на семейства $(\mathcal{R}_i \,|\, i \in I)$ и $(\mathcal{S}_i \,|\, i \in I)$ распределений вероятностей на множестве X ($\{0,1\}^*$). А именно, указанные семейства распределений называются *статистически* или *вычислительно неотличимыми* (statistically close или computationally indistinguishable), если соответственно статистически или вычислительно неотличимы семейства случайных величин $(\widetilde{r}_i \,|\, i \in I)$ и $(\widetilde{s}_i \,|\, i \in I)$, где $\widetilde{r}_i \leftarrow \mathcal{R}_i$ и $\widetilde{s}_i \leftarrow \mathcal{S}_i$ для любого $i \in I$. Очевидно, что статистическая (вычислительная) неотличимость семейств случайных величин зависит только от распределений этих случайных величин, так что определение статистической (вычислительной) неотличимости распределений корректно.

Пусть $(\widetilde{x}_i \,|\, i \in I)$ и $(\widetilde{y}_i \,|\, i \in I)$ — семейства случайных величин, принимающих значения в $\{0,1\}^*$. Тогда легко видеть, что

- если семейства $(\widetilde{x}_i \,|\, i \in I)$ и $(\widetilde{y}_i \,|\, i \in I)$ вычислительно неотличимы, то для любого полиномиального вероятностного алгоритма A семейства случайных величин $(A(i, \widetilde{x}_i) \,|\, i \in I)$ и $(A(i, \widetilde{y}_i) \,|\, i \in I)$ также вычислительно неотличимы;
- если семейства $(\widetilde{x}_i \,|\, i \in I)$ и $(\widetilde{y}_i \,|\, i \in I)$ статистически неотличимы, то они также вычислительно неотличимы.

Следующая лемма полезна в доказательствах.

Лемма 0.1. *Пусть семейства случайных величин $(\widetilde{x}_i \,|\, i \in I)$ и $(\widetilde{y}_i \,|\, i \in I)$ не являются вычислительно неотличимыми. Тогда существуют полиномиальный вероятностный алгоритм D, полином p и бесконечное множество $J \subseteq I$ такие, что для всех $j \in J$*

$$\Pr[D(j, \widetilde{x}_j) = 1] - \Pr[D(j, \widetilde{y}_j) = 1] > \frac{1}{p(|j|)}.$$

Доказательство. По определению вычислительной неотличимости существуют полиномиальный вероятностный алгоритм E, полином p и бесконечное множество $K \subseteq I$ такие, что $|\Pr[E(k, \widetilde{x}_k) = 1] - \Pr[E(k, \widetilde{y}_k) = 1]| > 1/p(|k|)$ для всех $k \in K$. Очевидно, что $K = K_1 \cup K_2$, где

$$K_1 = \{k \in K \mid \Pr[E(k, \widetilde{x}_k) = 1] - \Pr[E(k, \widetilde{y}_k) = 1] > 1/p(|k|)\}$$

и

$$K_2 = \{k \in K \mid \Pr[E(k, \widetilde{x}_k) = 1] - \Pr[E(k, \widetilde{y}_k) = 1] < -1/p(|k|)\}.$$

Если множество K_1 бесконечно, то полагаем $J = K_1$ и $D = E$. В противном случае в качестве J выбираем K_2 (которое бесконечно, так как бесконечно K), а в качестве D — полиномиальный вероятностный алгоритм, который выполняет алгоритм E на заданном входе и возвращает 0, если E возвратил 1, и 1 в противном случае. Легко видеть, что в обоих случаях D, p и J удовлетворяют требованиям леммы. □

Рассмотрим теперь случай, когда множество индексов I является подмножеством множества $\mathbb{N}$. Если не оговорено противное, то в определениях полиномиальной конструируемости, а также статистической и вычислительной неотличимости семейств распределений вероятностей и случайных величин подразумевается, что индексы заданы в бинарной записи. Если же эти индексы заданы в унарной записи, то мы указываем это явно. В качестве примера приведем некоторые определения такого типа.

- Семейство распределений вероятностей $(\mathcal{D}_i \mid i \in I)$ на множестве $\{0,1\}^*$ называется *полиномиально конструируемым, когда индексы заданы в унарной записи*, если существует полиномиальный вероятностный алгоритм G такой, что для любого $i \in I$ случайная величина $G(1^i)$ имеет распределение $\mathcal{D}_i$.
- Семейства $(\widetilde{x}_i \mid i \in I)$ и $(\widetilde{y}_i \mid i \in I)$ случайных величин, принимающих значения в одном и том же конечном или счетном множестве X, называются *статистически неотличимыми, когда индексы заданы в унарной записи*, если $\Delta(\widetilde{x}_i, \widetilde{y}_i) = \mathrm{negl}(i)$ (предполагается, что I бесконечно).
- Семейства $(\widetilde{x}_i \mid i \in I)$ и $(\widetilde{y}_i \mid i \in I)$ случайных величин, принимающих значения в $\{0,1\}^*$, называются *вычислительно неотличимыми, когда индексы заданы в унарной записи*, если для любого полиномиального вероятностного алгоритма D

  $$|\Pr[D(1^i, \widetilde{x}_i) = 1] - \Pr[D(1^i, \widetilde{y}_i) = 1]| = \mathrm{negl}(i)$$

  (предполагается, что I бесконечно).

Лемма 0.1 может быть легко переформулирована на случай, когда индексы заданы в унарной записи. Именно этот случай понадобится нам в дальнейшем, поэтому мы приводим соответствующую формулировку.

Следствие 0.2. *Пусть I — бесконечное подмножество $\mathbb{N}$, а $(\widetilde{x}_i \mid i \in I)$ и $(\widetilde{y}_i \mid i \in I)$ — семейства случайных величин, принимающих значения в $\{0,1\}^*$. Предположим, что эти семейства не являются вычислительно неотличимыми, когда индексы заданы в унарной записи. Тогда существуют полиномиальный вероятностный алгоритм D, полином p и бесконечное множество $J \subseteq I$ такие, что для всех $j \in J$*

$$\Pr[D(1^j, \widetilde{x}_j) = 1] - \Pr[D(1^j, \widetilde{y}_j) = 1] > \frac{1}{p(j)}.$$

Функция $\pi\colon N \to \mathbb{N}$ называется *полиномиальным параметром* (polynomial parameter), если функция $1^n \mapsto 1^{\pi(n)}$ $(n \in N)$ полиномиально вычислима. Это условие выполнено тогда и только тогда, когда функция π полиномиально ограничена (т. е. $\pi(n) \leqslant \mathrm{poly}(n)$ для всех $n \in N$) и вычислима детерминированным алгоритмом за полиномиальное от n время. См. [Lub96, предвар. лекция]. Примером полиномиального параметра является ограничение произвольного полинома (в вышеуказанном смысле) на N.

Лемма 0.3. *Пусть N — бесконечное подмножество $\mathbb{N}$, а π — полиномиальный параметр на N. Пусть также $(\mathcal{D}_n \mid n \in N)$ и $(\mathcal{E}_n \mid n \in N)$ — полиномиально конструируемые (когда индексы заданы в унарной записи) семейства распределений вероятностей на множестве $\{0,1\}^*$. Предположим, что эти семейства вычислительно неотличимы, когда индексы заданы в унарной записи. Тогда семейства $(\mathcal{D}_n^{\pi(n)} \mid n \in N)$ и $(\mathcal{E}_n^{\pi(n)} \mid n \in N)$ распределений вероятностей также вычислительно неотличимы, когда индексы заданы в унарной записи.*

Доказательство. Предположим, что $(\mathcal{D}_n^{\pi(n)} \,|\, n \in N)$ и $(\mathcal{E}_n^{\pi(n)} \,|\, n \in N)$ не являются вычислительно неотличимыми, когда индексы заданы в унарной записи. Для любого $n \in \mathbb{N}$ пусть $\widetilde{v}_1, \ldots, \widetilde{v}_{\pi(n)} \leftarrow \mathcal{D}_n$ и $\widetilde{w}_1, \ldots, \widetilde{w}_{\pi(n)} \leftarrow \mathcal{E}_n$. Ввиду следствия 0.2 существуют полиномиальный вероятностный алгоритм A, полином p и бесконечное множество $M \subseteq \mathbb{N}$ такие, что для всех $n \in M$

$$\Pr[A(1^n, \widetilde{v}_1, \ldots, \widetilde{v}_{\pi(n)}) = 1] - \Pr[A(1^n, \widetilde{w}_1, \ldots, \widetilde{w}_{\pi(n)}) = 1] > \frac{1}{p(n)}. \tag{1}$$

Пусть B — полиномиальный вероятностный алгоритм, работающий на произвольном входе $(1^n, y)$, где $n \in M$ и $y \in \{0,1\}^*$, следующим образом:

1. Выбрать $i \in_{\mathcal{U}} \{1, \ldots, \pi(n)\}$ (из неравенства (1) вытекает, что $\pi(n) \geqslant 1$).
2. Выбрать $v_1, \ldots, v_{i-1} \leftarrow \mathcal{D}_n$ и $w_{i+1}, \ldots, w_{\pi(n)} \leftarrow \mathcal{E}_n$ (это возможно сделать за полиномиальное от n время ввиду полиномиальной конструируемости $(\mathcal{D}_n \,|\, n \in N)$ и $(\mathcal{E}_n \,|\, n \in N)$, когда индексы заданы в унарной записи).
3. Вычислить $A(1^n, v_1, \ldots, v_{i-1}, y, w_{i+1}, \ldots, w_{\pi(n)})$ и возвратить результат (если он есть).

Фиксируем $n \in M$. Для каждого $i \in \{0, \ldots, \pi(n)\}$ положим для краткости

$$\delta_i(n) = \Pr[A(1^n, \widetilde{v}_1, \ldots, \widetilde{v}_i, \widetilde{w}_{i+1}, \ldots, \widetilde{w}_{\pi(n)}) = 1].$$

Тогда если $\widetilde{v} \leftarrow \mathcal{D}_n$ и $\widetilde{w} \leftarrow \mathcal{E}_n$, то

$$\Pr[B(1^n, \widetilde{v}) = 1] = \frac{1}{\pi(n)} \sum_{i=1}^{\pi(n)} \delta_i(n) \quad \text{и} \quad \Pr[B(1^n, \widetilde{w}) = 1] = \frac{1}{\pi(n)} \sum_{i=1}^{\pi(n)} \delta_{i-1}(n).$$

Кроме того,

$$\Pr[A(1^n, \widetilde{v}_1, \ldots, \widetilde{v}_{\pi(n)}) = 1] = \delta_{\pi(n)}(n) \quad \text{и} \quad \Pr[A(1^n, \widetilde{w}_1, \ldots, \widetilde{w}_{\pi(n)}) = 1] = \delta_0(n).$$

Следовательно,

$$\Pr[B(1^n, \widetilde{v}) = 1] - \Pr[B(1^n, \widetilde{w}) = 1] = \frac{\delta_{\pi(n)}(n) - \delta_0(n)}{\pi(n)} > \frac{1}{p(n)\pi(n)} \geqslant \frac{1}{\mathrm{poly}(n)}$$

для всех $n \in M$ (см. неравенство (1)), что противоречит вычислительной неотличимости семейств $(\mathcal{D}_n \,|\, n \in N)$ и $(\mathcal{E}_n \,|\, n \in N)$, когда индексы заданы в унарной записи. □

В доказательстве леммы 0.3 использован так называемый *гибридный метод* (hybrid technique) или *гибридный аргумент* (hybrid argument). Название связано с тем, что мы переходим от $(\widetilde{v}_1, \ldots, \widetilde{v}_{\pi(n)})$ к $(\widetilde{w}_1, \ldots, \widetilde{w}_{\pi(n)})$ с помощью последовательности гибридных случайных величин (гибридов) $\widetilde{z}_i = (\widetilde{v}_1, \ldots, \widetilde{v}_i, \widetilde{w}_{i+1}, \ldots, \widetilde{w}_{\pi(n)})$, где $i \in \{0, \ldots, \pi(n)\}$. Здесь $\widetilde{z}_{\pi(n)} = (\widetilde{v}_1, \ldots, \widetilde{v}_{\pi(n)})$, $\widetilde{z}_0 = (\widetilde{w}_1, \ldots, \widetilde{w}_{\pi(n)})$, а $\widetilde{z}_{i-1}$ и $\widetilde{z}_i$ отличаются лишь i-м элементом ($i \in \{1, \ldots, \pi(n)\}$). Гибридный метод будет широко использоваться и в дальнейшем.

Семейство $(G_d \,|\, d \in D)$ групп такое, что $D \subseteq \{0,1\}^*$ и $G_d \subseteq \{0,1\}^{\leqslant \mathrm{poly}(|d|)}$ для всех $d \in D$, называется *полиномиально вычислимым* (polynomial-time computable), если функции

- $(d, x, y) \mapsto xy$ в G_d ($d \in D$, $x, y \in G_d$),
- $(d, x) \mapsto x^{-1}$ в G_d ($d \in D$, $x \in G_d$),
- $d \mapsto 1$ в G_d ($d \in D$)

полиномиально вычислимы. В качестве примера заметим, что $(\mathbb{Z}_m \,|\, m \in \mathbb{N}\backslash\{0\})$ и $(\mathbb{Z}_m^\star \,|\, m \in \mathbb{N}\backslash\{0\})$ — полиномиально вычислимые семейства групп (здесь $\mathbb{Z}_m$ обозначает аддитивную группу кольца $\mathbb{Z}_m$).

1. Неформальное определение общих понятий математической криптографии

Настоящий курс посвящен теории математических моделей криптографических примитивов и криптографических протоколов. *Криптографический протокол* (cryptographic protocol) — это распределенный алгоритм с несколькими участниками, предназначенный для решения по крайней мере одной из трех задач криптографии, к которым относится обеспечение конфиденциальности, целостности и неотслеживаемости. Примеры видов криптографических протоколов:

- *Системы шифрования* (encryption systems), называемые также *криптосистемами* (cryptosystems) или *шифрами* (ciphers). Протоколы этого вида предназначены для обеспечения конфиденциальности пересылаемых сообщений.
- *Протоколы электронной подписи* (electronic signature protocols, digital signature protocols). Протоколы этого вида предназначены для обеспечения целостности подписываемых сообщений.

Протоколы, предназначенные для обеспечения неотслеживаемости, в настоящем курсе не рассматриваются ввиду сложности как самих этих протоколов, так и их математической теории.

Особенностью криптографических протоколов является то, что они должны противостоять атакам противника; способность к этому называется *стойкостью* (security) криптографических протоколов. *Противник* (adversary) — это субъект, выполняющий не предписанные протоколом действия. Противник может быть как внешним, так и представлять собой нечестного участника протокола или коалицию таких участников. Под *атакой* (attack) понимается совокупность предположений о возможностях противника. Целью противника является «взлом» (в каком-либо смысле) защиты, которую должен обеспечивать протокол. Задача по осуществлению такого «взлома» называется *угрозой* (threat) стойкости протокола. Говоря упрощенно, атака — это то, что противник может делать, а угроза — то, что он хочет сделать.

Большинство криптографических протоколов оперируют с объектами из конечных множеств. Чтобы обеспечить массовость вычислительной задачи по осуществлению угрозы, предполагается, что эти множества зависят от некоторого числа n, пробегающего какое-либо бесконечное подмножество $\mathbb{N}$ и называемого *параметром стойкости* (security parameter). Этот параметр является общедоступным и используется в работе протокола. Обычно протокол строится так, чтобы его участники имели дело с объектами, представленными двоичными строками длины не более $\mathrm{poly}(n)$, где n — параметр стойкости.

Стойкость криптографического протокола определяется только против конкретной угрозы на основе конкретной атаки. Кроме того, эта стойкость зависит и от модели противника. В настоящем курсе (за исключением изложения элементов теории Шеннона) предполагается, что противник может использовать только вероятностные алгоритмы, работающие за полиномиальное время от параметра стойкости. Говоря упрощенно, в большинстве случаев криптографический протокол называется стойким против угрозы X на основе атаки Y, если произвольный противник (в данной модели) может после проведения атаки Y осуществить угрозу X лишь с вероятностью не более $t(n)+\mathrm{negl}(n)$, где $t(n)$ — некоторая вероятность, с которой угроза X заведомо может быть осуществлена за полиномиальное от n время, а n — параметр стойкости.

В качестве примера рассмотрим классический протокол Диффи—Хеллмана распределения ключей. Мы опишем его в несколько другом виде, чем в пионерской работе [DH76]. Пусть $n \in \mathbb{N} \setminus \{0\}$ — число, играющее роль параметра стойкости. Общедоступными параметрами протокола являются число $m \in \mathbb{N}$, удовлетворяющее неравенствам $2^n \leqslant m \leqslant 2^{n+1}-1$ (т. е. такое, что длина его двоичной записи со старшим битом 1 равна $n+1$), и элемент $g \in \mathbb{Z}_m^\star$. Эти параметры выбираются по 1^n с помощью некоторого полиномиального вероятностного алгоритма G. В протоколе имеются два участника, традиционно называемые Алисой и Бобом, целью которых является выработка общего секретного ключа. Протокол выполняется следующим образом:

Общий вход: $(1^n, m, g)$.

1. Алиса выбирает $x \in_{\mathcal{U}} \mathbb{Z}_m$, вычисляет $a = g^x \bmod m$ и посылает a Бобу.
2. Боб выбирает $y \in_{\mathcal{U}} \mathbb{Z}_m$, вычисляет $b = g^y \bmod m$ и посылает b Алисе.
3. Алиса вычисляет общий секретный ключ, равный $g^{xy} \bmod m$, как $b^x \bmod m$.

4. Боб вычисляет общий секретный ключ, равный $g^{xy} \bmod m$, как $a^y \bmod m$.

Предположим, что противник может проводить атаку, заключающуюся в перехвате сообщений $g^x \bmod m$ и $g^y \bmod m$, пересылаемых Алисой и Бобом друг другу. В качестве угрозы рассмотрим нахождение общего секретного ключа $g^{xy} \bmod m$. Тогда вышеописанный протокол называется стойким против этой угрозы на основе вышеуказанной атаки, если для любого полиномиального вероятностного алгоритма A

$$\Pr[A(1^n, \widetilde{m}, \widetilde{g}, \widetilde{g}^{\widetilde{x}} \bmod \widetilde{m}, \widetilde{g}^{\widetilde{y}} \bmod \widetilde{m}) = \widetilde{g}^{\widetilde{x}\widetilde{y}} \bmod \widetilde{m}] = \operatorname{negl}(n),$$

где $(\widetilde{m}, \widetilde{g}) = G(1^n)$ и $\widetilde{x}, \widetilde{y} \in_{\mathcal{U}} \mathbb{Z}_m$. (Здесь $t(n) = 0$.) Отметим, что 1^n можно удалить из числа входов, так как $n + 1$ равно длине двоичной записи числа m со старшим битом 1.

Криптографические примитивы (cryptographic primitives) — это математические объекты, используемые в качестве «строительных блоков» при построении криптографических протоколов. Под *стойкостью* (security) криптографического примитива понимается выполнение для него основного криптографического условия, указанного в определении. Поэтому определение стойкости криптографического примитива не требует указания ни угрозы, ни атаки, ни модели противника. Однако для многих криптографических примитивов естественно определен параметр стойкости. Классическими примерами криптографических примитивов являются односторонняя функция и псевдослучайный генератор (генератор псевдослучайных последовательностей), которые будут подробно рассмотрены ниже.

2. Односторонние и слабо односторонние функции и перестановки

Бесконечное множество $N \subseteq \mathbb{N}$ называется *полиномиально перечислимым* (polynomial-time enumerable), если функция $i \mapsto \min\{n \in N \mid n > i\}$ $(i \in \mathbb{N})$ является полиномиальным параметром (см. [Gol04, п. 2.2.3.1]). Пусть N — полиномиально перечислимое множество. Заметим, что по произвольному $i \in \mathbb{N}$ можно за полиномиальное от i время определить, входит ли i в N. Действительно, пусть s — полиномиальный параметр, отображающий произвольное число $i \in \mathbb{N}$ в $\min\{n \in N \mid n > i\}$. Тогда $i \in N \iff s(i-1) = i$ для произвольного $i \in \mathbb{N} \setminus \{0\}$. См. также [Gol04, доказательство предложения 2.2.3]),

Определение 2.1 (односторонняя и слабо односторонняя функция; см. также [Gol04, определение 2.2.1]). Полиномиально вычислимая функция $f\colon \bigcup_{n\in N}\{0,1\}^n \to \{0,1\}^*$ называется

- *односторонней* (one-way) или *сильно односторонней* (strongly one-way), если для любого полиномиального вероятностного алгоритма A

$$\Pr[A(1^n, f(\widetilde{u}_n)) \in f^{-1}(f(\widetilde{u}_n))] = \operatorname{negl}(n);$$

- *слабо односторонней* (weakly one-way), если существует полином p такой, что для любого полиномиального вероятностного алгоритма A неравенство

$$\Pr[A(1^n, f(\widetilde{u}_n)) \in f^{-1}(f(\widetilde{u}_n))] \leqslant 1 - \frac{1}{p(n)}$$

выполняется при всех достаточно больших $n \in N$.

Определение 2.1 формализует (двумя разными способами) интуитивное понятие функции, которая легко вычислима, но трудно инвертируема. В настоящее время односторонняя и даже слабо односторонняя функция — гипотетические объекты.

Без подачи на вход алгоритму A строки 1^n определение 2.1, вообще говоря, бессодержательно. Например, рассмотрим функцию l, заданную равенством $l(x) = |x|$ для всех $x \in \{0,1\}^*$ (см. также [Gol04, подразд. 2.2.1]; напомним, что число $|x|$ здесь задано своей двоичной записью наименьшей длины). Тогда l полиномиально вычислима и для любого полиномиального вероятностного алгоритма A равенство $\Pr[A(l(x)) \in l^{-1}(l(x))] = 0$ выполняется при всех $x \in \{0,1\}^*$ достаточно большой длины (так как $\operatorname{supp} A(l(x)) \subseteq \{0,1\}^{\leqslant \operatorname{poly}(\lfloor \log_2 |x| \rfloor)}$ и $l^{-1}(l(x)) = \{0,1\}^{|x|}$ для всех $x \in \{0,1\}^{\geqslant 1}$). Однако имеется альтернативное определение односторонней и слабо односторонней функции, в котором не требуется подавать 1^n на вход алгоритма A, но нужно предполагать, что функция f честна. Функция $f\colon X \to \{0,1\}^*$, где $X \subseteq \{0,1\}^*$, называется *честной* (honest), если $|x| \leqslant \operatorname{poly}(|f(x)|)$ для всех $x \in X$. Говоря неформально, условие честности означает, что функция не очень сильно уменьшает длину своего аргумента.

Определение 2.2 (односторонняя и слабо односторонняя функция, альтернативный вариант)**.** Честная полиномиально вычислимая функция $f\colon \bigcup_{n\in N}\{0,1\}^n \to \{0,1\}^*$ называется

- *односторонней* (one-way) или *сильно односторонней* (strongly one-way), если для любого полиномиального вероятностного алгоритма A

$$\Pr[A(f(\widetilde{u}_n)) \in f^{-1}(f(\widetilde{u}_n))] = \operatorname{negl}(n);$$

- *слабо односторонней* (weakly one-way), если существует полином p такой, что для любого полиномиального вероятностного алгоритма A неравенство

$$\Pr[A(f(\widetilde{u}_n)) \in f^{-1}(f(\widetilde{u}_n))] \leqslant 1 - \frac{1}{p(n)}$$

  выполняется при всех достаточно больших $n \in N$.

Между определениями 2.1 и 2.2 имеется тесная связь. А именно, если функция $f\colon \bigcup_{n\in N}\{0,1\}^n \to \{0,1\}^*$ является односторонней (слабо односторонней) в смысле определения 2.2, то она также является односторонней (слабо односторонней) в смысле определения 2.1. Наоборот, если эта функция f является односторонней (слабо односторонней) в смысле определения 2.1, то функция $x \mapsto 1^{|x|}0f(x)$ ($x \in \bigcup_{n\in N}\{0,1\}^n$) является односторонней (слабо односторонней) в смысле определения 2.2. Для честных функций эти определения эквивалентны. В дальнейшем мы под односторонней (слабо односторонней) функцией понимаем одностороннюю (слабо одностороннюю) функцию в смысле определения 2.1.

Односторонняя или слабо односторонняя функция $f\colon \bigcup_{n\in N}\{0,1\}^n \to \bigcup_{n\in N}\{0,1\}^n$ называется соответственно *односторонней* (one-way) или *слабо односторонней перестановкой* (weakly one-way permutation), если она биективна и сохраняет длину (т. е. $|f(x)| = |x|$ для всех $x \in \bigcup_{n\in N}\{0,1\}^n$). Более подробное определение односторонней и слабо односторонней перестановки имеет следующий вид.

Определение 2.3 (односторонняя и слабо односторонняя перестановка)**.** Полиномиально вычислимая биекция $f\colon \bigcup_{n\in N}\{0,1\}^n \to \bigcup_{n\in N}\{0,1\}^n$, сохраняющая длину, называется

- *односторонней* (one-way) или *сильно односторонней перестановкой* (strongly one-way permutation), если для любого полиномиального вероятностного алгоритма A

$$\Pr[A(f(\widetilde{u}_n)) = \widetilde{u}_n] = \operatorname{negl}(n);$$

- *слабо односторонней перестановкой* (weakly one-way permutation), если существует полином p такой, что для любого полиномиального вероятностного алгоритма A неравенство

$$\Pr[A(f(\widetilde{u}_n)) = \widetilde{u}_n] \leqslant 1 - \frac{1}{p(n)}$$

  выполняется при всех достаточно больших $n \in N$.

Разумеется, в определении 2.3 нет необходимости подавать 1^n на вход алгоритма A, так как $n = |f(x)|$ для любого $x \in \{0,1\}^n = \operatorname{supp}\widetilde{u}_n$. Кроме того, в этом определении учтено, что $f^{-1}(f(x)) = \{x\}$ для любой инъективной функции f и любого x из ее области определения.

В настоящее время существование односторонних (слабо односторонних) перестановок не доказано даже в предположении существования односторонних (слабо односторонних) функций. Поэтому предположение о существовании односторонних (слабо односторонних) перестановок считается более сильным, чем предположение о существовании односторонних (слабо односторонних) функций.

***Замечание* 2.4** (см. [Gol04, предложение 2.2.3])**.** Пусть $f\colon \bigcup_{n\in N}\{0,1\}^n \to \{0,1\}^*$ — односторонняя (слабо односторонняя) функция. Представим каждую строку $x \in \{0,1\}^{\geqslant \min N}$ в виде $x'x''$, где x' — префикс строки x, имеющий наибольшую длину, принадлежащую N. Пусть функции $g, h\colon \{0,1\}^* \to \{0,1\}^*$ таковы, что $g(x) = f(x')$ и $h(x) = f(x')x''$ для всех $x \in \{0,1\}^{\geqslant \min N}$. (На множестве $\{0,1\}^{\leqslant \min N - 1}$ функции g и h могут быть определены произвольно.) Тогда функции g и h являются односторонними (слабо односторонними). Кроме того, ограничения g и h на $\bigcup_{n\in N}\{0,1\}^n$ совпадают с f. Очевидно также, что если f сохраняет длину, то $h|_{\{0,1\}^{\geqslant \min N}}$ сохраняет длину.

Легко видеть, что если f из замечания 2.4 инъективна и сохраняет длину, то h из этого замечания (при подходящем определении на $\{0,1\}^{\leqslant \min N-1}$) биективна и сохраняет длину. Поэтому из замечания 2.4 вытекает

Следствие 2.5. *Всякая односторонняя (слабо односторонняя) перестановка, определенная на $\bigcup_{n\in N}\{0,1\}^n$ для некоторого полиномиально перечислимого множества $N \subseteq \mathbb{N}$, может быть продолжена до односторонней (слабо односторонней) перестановки, определенной на $\{0,1\}^*$.*

***Замечание* 2.6** (см. [Gol04, предложение 2.2.5])**.** Пусть $f\colon \bigcup_{n\in N}\{0,1\}^n \to \{0,1\}^*$ — односторонняя (слабо односторонняя) функция. Пусть также m — какой-либо полиномиальный параметр на N, удовлетворяющий неравенству $|f(x)| \leqslant m(|x|)$ для всех $x \in \bigcup_{n\in N}\{0,1\}^n$. Тогда функция $x \mapsto f(x)10^{m(|x|)-|f(x)|}$ ($x \in \bigcup_{n\in N}\{0,1\}^n$) является односторонней (слабо односторонней) и отображает $\{0,1\}^n$ в $\{0,1\}^{m(n)+1}$ для всех $n \in N$.

***Замечание* 2.7** (см. [Gol04, предложение 2.2.5])**.** Пусть функция f отображает $\{0,1\}^n$ в $\{0,1\}^{m(n)}$ для всех $n \in N$, где m — инъективный полиномиальный параметр, удовлетворяющий неравенству $m(n) \geqslant n$ для всех $n \in N$. Предположим, что f является односторонней (слабо односторонней). Представим каждую строку $x \in \{0,1\}^{m(n)}$ ($n \in N$) в виде $x'x''$, где $x' \in \{0,1\}^n$. Тогда функция $x \mapsto f(x')$ ($x \in \bigcup_{n\in N}\{0,1\}^{m(n)}$) является односторонней (слабо односторонней) и сохраняет длину. Легко также видеть, что множество $m(N)$ полиномиально перечислимо, так как N полиномиально перечислимо.

Из замечаний 2.4, 2.6 и 2.7 вытекает

Предложение 2.8. *Если существует односторонняя (слабо односторонняя) функция, определенная на $\bigcup_{n\in N}\{0,1\}^n$ для некоторого полиномиально перечислимого множества $N \subseteq \mathbb{N}$, то существует односторонняя (слабо односторонняя) функция, определенная на $\{0,1\}^*$ и сохраняющая длину.*

3. Построение односторонней функции (односторонней перестановки) из слабо односторонней функции (слабо односторонней перестановки)

Очевидно, что всякая односторонняя функция является слабо односторонней. Следующий пример показывает, что обратное в предположении существования слабо односторонних функций неверно (см. также [Gol04, разд. 2.3]).

Пример 3.1. Пусть $f\colon \{0,1\}^* \to \{0,1\}^*$ — слабо односторонняя функция. Для произвольной строки $x \in \{0,1\}^*$ положим $g(x0) = x0$ и $g(x1) = f(x)1$. Тогда g — полиномиально вычислимая функция на $\{0,1\}^{\geqslant 1}$. Эта функция не является односторонней, так как если M — полиномиальный алгоритм такой, что $M(1^{n+1}, y) = y$ для всех $n \in \mathbb{N}$ и $y \in \{0,1\}^*$, то $\Pr[M(1^{n+1}, g(\widetilde{u}_{n+1})) \in g^{-1}(g(\widetilde{u}_{n+1}))] \geqslant 1/2$ для всех $n \in \mathbb{N}$. Покажем теперь, что g является слабо односторонней. Действительно, выберем какой-либо полином p такой, что для любого полиномиального вероятностного алгоритма A неравенство $\Pr[A(1^n, f(\widetilde{u}_n)) \in f^{-1}(f(\widetilde{u}_n))] \leqslant 1 - 1/p(n)$ выполняется при всех достаточно больших $n \in \mathbb{N}$. Пусть B — полиномиальный вероятностный алгоритм. Определим полиномиальный вероятностный алгоритм A, который на произвольном входе вида $(1^n, y)$, где $n \in \mathbb{N}$ и $y \in \{0,1\}^*$, работает следующим образом:

1. Вычислить $w \leftarrow B(1^{n+1}, y1)$.
2. Если $w = z1$, где $z \in \{0,1\}^*$, то возвратить z.

Тогда

$$\begin{aligned}
&\Pr[B(1^{n+1}, g(\widetilde{u}_{n+1})) \in g^{-1}(g(\widetilde{u}_{n+1}))] \\
&\quad = \frac{\Pr[B(1^{n+1}, g(\widetilde{u}_n 0)) \in g^{-1}(g(\widetilde{u}_n 0))] + \Pr[B(1^{n+1}, g(\widetilde{u}_n 1)) \in g^{-1}(g(\widetilde{u}_n 1))]}{2} \\
&\quad \leqslant \frac{1 + \Pr[A(1^n, f(\widetilde{u}_n)) \in f^{-1}(f(\widetilde{u}_n))]}{2} \leqslant 1 - \frac{1}{2p(n)} \leqslant 1 - \frac{1}{2p(n+1)}.
\end{aligned}$$

для всех достаточно больших $n \in \mathbb{N}$. (Здесь использовано то, что $p(n) \leqslant p(n+1)$ при каждом $n \in \mathbb{N} \setminus \{0\}$.) Таким образом, слабая односторонность g доказана. Заметим также, что если f — слабо односторонняя перестановка, то g — слабо односторонняя перестановка, не являющаяся односторонней.

В то же время имеет место следующий результат (см. [Gol04, разд. 2.3], [Lub96, лекция 3]).

Теорема 3.2. *Если существуют слабо односторонние функции (слабо односторонние перестановки), то существуют и односторонние функции (односторонние перестановки).*

Доказательство. Пусть $f\colon \{0,1\}^* \to \{0,1\}^*$ — слабо односторонняя функция. Согласно замечанию 2.6, без ограничения общности можно считать, что $f(\{0,1\}^n) \subseteq \{0,1\}^{m(n)}$ для всех $n \in \mathbb{N}$, где m — некоторый возрастающий полиномиальный параметр на $\mathbb{N}$. Выберем какой-либо полином p такой, что для любого полиномиального вероятностного алгоритма A неравенство

$$\Pr[A(1^n, f(\widetilde{u}_n)) \in f^{-1}(f(\widetilde{u}_n))] \leqslant 1 - \frac{1}{p(n)} \tag{2}$$

выполняется при всех достаточно больших $n \in \mathbb{N}$. Положим $t(n) = np(n)$ для всех $n \in \mathbb{N}$ и определим функцию $g\colon \bigcup_{n\in\mathbb{N}}\{0,1\}^{n^2p(n)} \to \{0,1\}^*$ следующим образом:

$$g(x_1 \dots x_{t(n)}) = f(x_1) \dots f(x_{t(n)}) \quad (x_1, \dots, x_{t(n)} \in \{0,1\}^n,\ n \in \mathbb{N}).$$

Корректность определения функции g вытекает из того, что функция $n \mapsto n^2p(n)$ ($n \in \mathbb{N}$) инъективна (так как она возрастает). Покажем, что функция g является односторонней. Очевидно, что g полиномиально вычислима. Поэтому если g не является односторонней, то существуют полиномиальный вероятностный алгоритм B, полином q и бесконечное множество $N \subseteq \mathbb{N}$ такие, что

$$\Pr\left[B\left(1^{n^2p(n)}, g(\widetilde{u}_{n^2p(n)})\right) \in g^{-1}(g(\widetilde{u}_{n^2p(n)}))\right] > \frac{1}{q(n^2p(n))} \tag{3}$$

для всех $n \in N$. Пусть A_0 — полиномиальный вероятностный алгоритм, работающий на входе $(1^n, y)$ ($n \in \mathbb{N} \setminus \{0\}$, $y \in \{0,1\}^{m(n)}$) следующим образом:

Для i от 1 до $t(n)$:

1) выбрать $x_j \in_{\mathcal{U}} \{0,1\}^n$ для каждого $j \in \{1, \dots, t(n)\} \setminus \{i\}$;

2) вычислить $B\left(1^{n^2p(n)}, f(x_1) \dots f(x_{i-1}) y f(x_{i+1}) \dots f(x_{t(n)})\right) = z$;

3) если $z = z_1 \dots z_{t(n)}$, где $z_1, \dots, z_{t(n)} \in \{0,1\}^n$ и $f(z_i) = y$, то возвратить z_i и закончить работу.

Пусть также полиномиальный вероятностный алгоритм A на входе $(1^n, y)$, где $n \in \mathbb{N} \setminus \{0\}$ и $y \in \{0,1\}^{m(n)}$, выполняет алгоритм A_0 на этом входе не более $k(n) = 2nt(n)q(nt(n)) = 2n^2p(n)q(n^2p(n))$ раз. Если на некоторой итерации алгоритм A_0 возвратил некоторое выходное значение (которое, очевидно, принадлежит $f^{-1}(y)$), то A возвращает это значение и прекращает работу. Если же ни на одной из $k(n)$ итераций этого не произошло, то A заканчивает работу без выходного значения.

Для произвольного $n \in \mathbb{N} \setminus \{0\}$ положим

$$E_n = \{x \in \{0,1\}^n \mid \Pr[A_0(1^n, f(x)) \in f^{-1}(f(x))] > n/k(n)\}.$$

Для доказательства односторонности g достаточно показать, что

$$\Pr[A(1^n, f(x)) \in f^{-1}(f(x))] > 1 - e^{-n} \quad \text{при всех } n \in \mathbb{N} \setminus \{0\} \text{ и } x \in E_n \tag{4}$$

и

$$\Pr[\widetilde{u}_n \in E_n] > 1 - \frac{1}{2p(n)} \quad \text{при всех достаточно больших } n \in N. \tag{5}$$

Действительно, из неравенств (4) и (5) вытекает, что при всех достаточно больших $n \in N$

$$\begin{aligned}&\Pr[A(1^n, f(\widetilde{u}_n)) \in f^{-1}(f(\widetilde{u}_n))]\\ &\quad\geqslant \Pr[A(1^n, f(\widetilde{u}_n)) \in f^{-1}(f(\widetilde{u}_n)) \mid \widetilde{u}_n \in E_n] \Pr[\widetilde{u}_n \in E_n] > (1 - e^{-n})\left(1 - \frac{1}{2p(n)}\right).\end{aligned}$$

Но тогда ввиду (2)

$$1 - \frac{1}{p(n)} > (1 - e^{-n}) \left(1 - \frac{1}{2p(n)}\right)$$

и, следовательно,

$$\frac{1}{p(n)} < e^{-n} + \frac{1}{2p(n)} - \frac{e^{-n}}{2p(n)} < e^{-n} + \frac{1}{2p(n)}$$

при всех достаточно больших $n \in N$, что неверно, так как $e^{-n} < 1/2p(n)$ при всех достаточно больших $n \in \mathbb{N}$.

Докажем сначала неравенство (4). Пусть $n \in \mathbb{N}\setminus\{0\}$ и $x \in E_n$. Тогда по определениям множества E_n и алгоритма A

$$\Pr[A(1^n, f(x)) \notin f^{-1}(f(x))] < \left(1 - \frac{n}{k(n)}\right)^{k(n)} = e^{k(n)\ln(1-n/k(n))} \leqslant e^{-n}. \tag{6}$$

Здесь использовано то, что

$$\ln r \leqslant r - 1 \quad \text{для всех } r > 0. \tag{7}$$

Неравенство (4) непосредственно вытекает из неравенства (6).

Докажем теперь неравенство (5). Предположим, что

$$\Pr[\widetilde{u}_n \in E_n] \leqslant 1 - \frac{1}{2p(n)} \tag{8}$$

при всех n из некоторого бесконечного множества $M \subseteq N\setminus\{0\}$. Пусть $n \in M$. Представим случайную величину $\widetilde{u}_{n^2p(n)}$ в виде $\widetilde{u}_n^1 \ldots \widetilde{u}_n^{t(n)}$, где $\widetilde{u}_n^1, \ldots, \widetilde{u}_n^{t(n)} \in_{\mathcal{U}} \{0,1\}^n$. Положим

$$s_1(n) = \Pr\left[B\left(1^{n^2p(n)}, g(\widetilde{u}_{n^2p(n)})\right) \in g^{-1}(g(\widetilde{u}_{n^2p(n)})),\, \exists\, i \in \{1, \ldots, t(n)\}\ \widetilde{u}_n^i \notin E_n\right]$$

и

$$s_2(n) = \Pr\left[B\left(1^{n^2p(n)}, g(\widetilde{u}_{n^2p(n)})\right) \in g^{-1}(g(\widetilde{u}_{n^2p(n)})),\, \forall\, i \in \{1, \ldots, t(n)\}\ \widetilde{u}_n^i \in E_n\right].$$

Тогда

$$\begin{aligned} s_1(n) &\leqslant \sum_{i=1}^{t(n)} \Pr\left[B\left(1^{n^2p(n)}, g(\widetilde{u}_{n^2p(n)})\right) \in g^{-1}(g(\widetilde{u}_{n^2p(n)})),\, \widetilde{u}_n^i \notin E_n\right] \\ &\leqslant \sum_{i=1}^{t(n)} \Pr\left[B\left(1^{n^2p(n)}, g(\widetilde{u}_{n^2p(n)})\right) \in g^{-1}(g(\widetilde{u}_{n^2p(n)})) \,\middle|\, \widetilde{u}_n^i \notin E_n\right] \\ &\leqslant \sum_{i=1}^{t(n)} \Pr[A_0(1^n, f(\widetilde{u}_n)) \in f^{-1}(f(\widetilde{u}_n)) \,|\, \widetilde{u}_n \notin E_n] \leqslant \frac{t(n)n}{k(n)} = \frac{1}{2q(n^2p(n))}. \end{aligned} \tag{9}$$

Здесь мы воспользовались неравенством $\Pr[X \cap Y] \leqslant \Pr[X \,|\, Y]$ (верным для любых событий X и Y при условии, что $\Pr[Y] \neq 0$), а также определением алгоритма A_0 и тем, что функция $n \mapsto m(n)t(n)$ ($n \in \mathbb{N}$) инъективна (так как она возрастает). Кроме того,

$$\begin{aligned} s_2(n) &\leqslant \Pr[\forall\, i \in \{1, \ldots, t(n)\}\ \widetilde{u}_n^i \in E_n] = \prod_{i=1}^{t(n)} \Pr[\widetilde{u}_n^i \in E_n] \\ &\leqslant \left(1 - \frac{1}{2p(n)}\right)^{t(n)} = e^{np(n)\ln(1-1/2p(n))} \leqslant e^{-n/2} \end{aligned} \tag{10}$$

ввиду неравенств (8) и (7). Очевидно, что $s_1(n) + s_2(n)$ совпадает с левой частью неравенства (3). Следовательно, из неравенств (3), (9) и (10) вытекает, что

$$\frac{1}{q(n^2p(n))} < \frac{1}{2q(n^2p(n))} + e^{-n/2}$$

для всех $n \in M$. Но это неравенство неверно при всех достаточно больших n. Полученное противоречие доказывает неравенство (5) и вместе с ним односторонность функции g.

Напомним, что функция g определена на множестве $\bigcup_{n\in\mathbb{N}}\{0,1\}^{n^2p(n)}$. Так как множество $\{n^2p(n)\,|\,n\in\mathbb{N}\}$ полиномиально перечислимо, эта функция может быть продолжена на $\{0,1\}^*$ с сохранением односторонности (см. замечание 2.4). Если f — слабо односторонняя перестановка, то g является односторонней перестановкой, отображающей $\{0,1\}^{n^2p(n)}$ на себя для любого $n\in\mathbb{N}$. Из следствия 2.5 вытекает, что g может быть продолжена до односторонней перестановки, определенной на $\{0,1\}^*$. □

Теорема 3.2 (для слабо односторонних функций) приписывается Яо. Однако работа [Yao82], на которую обычно ссылаются в литературе в связи с этой теоремой, не содержит последней.

4. Односторонние и слабо односторонние семейства функций и перестановок

Наличие этого раздела мотивировано тем, что гипотетически односторонние функции и перестановки, основанные на алгебраических и теоретико-числовых конструкциях и широко применяемые в математической криптографии, имеют вид семейств.

Пусть I — бесконечное подмножество $\{0,1\}^*$ и для каждого $i\in I$ определено множество $X_i\subseteq\{0,1\}^*$. Пусть также для каждого $n\in\mathbb{N}$ задано распределение вероятностей $\mathcal{I}_n$ на множестве I, а для каждого $i\in I$ — распределение вероятностей $\mathcal{X}_i$ на множестве X_i. Предположим, что семейство $(\mathcal{I}_n\,|\,n\in\mathbb{N})$ распределений вероятностей полиномиально конструируемо, когда индексы заданы в унарной записи, а семейство $(\mathcal{X}_i\,|\,i\in I)$ — полиномиально конструируемо.

Определение 4.1 (одностороннее и слабо одностороннее семейство функций; см. также [Gol04, определение 2.4.3])**.** Полиномиально вычислимое семейство функций $(f_i\colon X_i\to\{0,1\}^*\,|\,i\in I)$ называется

- *односторонним* (one-way) или *сильно односторонним* (strongly one-way) относительно семейств распределений вероятностей $(\mathcal{I}_n\,|\,n\in\mathbb{N})$ и $(\mathcal{X}_i\,|\,i\in I)$, если для любого полиномиального вероятностного алгоритма A

$$\Pr[A(1^n,\widetilde{i},f_{\widetilde{i}}(\widetilde{x}))\in f_{\widetilde{i}}^{-1}(f_{\widetilde{i}}(\widetilde{x}))]=\operatorname{negl}(n),$$

  где $\widetilde{i}\leftarrow\mathcal{I}_n$ и $\widetilde{x}\leftarrow\mathcal{X}_{\widetilde{i}}$;

- *слабо односторонним* (weakly one-way) относительно этих семейств распределений вероятностей, если существует полином p такой, что для любого полиномиального вероятностного алгоритма A неравенство

$$\Pr[A(1^n,\widetilde{i},f_{\widetilde{i}}(\widetilde{x}))\in f_{\widetilde{i}}^{-1}(f_{\widetilde{i}}(\widetilde{x}))]\leqslant 1-\frac{1}{p(n)},$$

  где $\widetilde{i}\leftarrow\mathcal{I}_n$ и $\widetilde{x}\leftarrow\mathcal{X}_{\widetilde{i}}$, выполняется при всех достаточно больших $n\in\mathbb{N}$.

***Замечание* 4.2.** Пусть $(f_i\,|\,i\in I)$ — одностороннее (слабо одностороннее) семейство функций относительно семейств распределений вероятностей $(\mathcal{I}_n\,|\,n\in\mathbb{N})$ и $(\mathcal{X}_i\,|\,i\in I)$. Выберем полиномиальные вероятностные алгоритмы G и H такие, что случайная величина $G(1^n)$ имеет распределение $\mathcal{I}_n$ для любого $n\in\mathbb{N}$, а случайная величина $H(i)$ — распределение $\mathcal{X}_i$ для любого $i\in I$. Пусть также k и m — возрастающие полиномиальные параметры на $\mathbb{N}$ такие, что алгоритм G при вычислении на входе 1^n (где $n\in\mathbb{N}$) использует не более $k(n)$ случайных битов, а алгоритм H при вычислении на входе $i\in I$ — не более $m(|i|)$ случайных битов. Тогда легко видеть, что функция

$$x\mapsto(i,f_i(H(i;s)))\quad\left(x\in\bigcup_{n\in\mathbb{N}}\{0,1\}^{k(n)+m(n)}\right),$$

где $x=rs$, $r\in\{0,1\}^{k(n)}$, $s\in\{0,1\}^{m(n)}$ $(n=(k+m)^{-1}(|x|))$, а $i=G(1^n;r)$, является односторонней (слабо односторонней).

***Замечание* 4.3.** Пусть $f\colon\{0,1\}^*\to\{0,1\}^*$ — односторонняя (слабо односторонняя) функция. Положим $I=\{1^n\,|\,n\in\mathbb{N}\}$, $X_i=\{0,1\}^{|i|}$, $f_i=f|_{X_i}$ $(i\in I)$ и выберем в качестве $\mathcal{I}_n$ распределение вероятностей, сосредоточенное на строке 1^n $(n\in\mathbb{N})$, а в качестве $\mathcal{X}_i$ — равномерное распределение вероятностей на множестве X_i $(i\in I)$. Тогда семейство $(f_i\,|\,i\in I)$ является односторонним (слабо односторонним) относительно семейств распределений вероятностей $(\mathcal{I}_n\,|\,n\in\mathbb{N})$ и $(\mathcal{X}_i\,|\,i\in I)$.

Таким образом, по всякому одностороннему семейству функций можно естественно построить одностороннюю функцию, а по всякой односторонней функции — одностороннее семейство функций. См. также [Gol04, подразд. 2.7.4, упр. 18].

Одностороннее или слабо одностороннее относительно семейств распределений вероятностей $(\mathcal{I}_n \mid n \in \mathbb{N})$ и $(\mathcal{X}_i \mid i \in I)$ семейство функций $(f_i \colon X_i \to \{0,1\}^* \mid i \in I)$ называется соответственно *односторонним* (one-way) или *слабо односторонним* (weakly one-way) относительно тех же семейств распределений вероятностей *семейством перестановок* (family of permutations, permutation family), если f_i является перестановкой множества X_i для любого $i \in I$. Более подробное определение одностороннего и слабо одностороннего семейства перестановок имеет следующий вид.

Определение 4.4 (одностороннее и слабо одностороннее семейство перестановок)**.** Полиномиально вычислимое семейство перестановок $(f_i \colon X_i \to X_i \mid i \in I)$ называется

- *односторонним* (one-way) или *сильно односторонним* (strongly one-way) относительно семейств распределений вероятностей $(\mathcal{I}_n \mid n \in \mathbb{N})$ и $(\mathcal{X}_i \mid i \in I)$, если для любого полиномиального вероятностного алгоритма A

$$\Pr[A(1^n, \widetilde{i}, f_{\widetilde{i}}(\widetilde{x})) = \widetilde{x}] = \operatorname{negl}(n),$$

  где $\widetilde{i} \leftarrow \mathcal{I}_n$ и $\widetilde{x} \leftarrow \mathcal{X}_{\widetilde{i}}$;

- *слабо односторонним* (weakly one-way) относительно этих семейств распределений вероятностей, если существует полином p такой, что для любого полиномиального вероятностного алгоритма A неравенство

$$\Pr[A(1^n, \widetilde{i}, f_{\widetilde{i}}(\widetilde{x})) = \widetilde{x}] \leqslant 1 - \frac{1}{p(n)},$$

  где $\widetilde{i} \leftarrow \mathcal{I}_n$ и $\widetilde{x} \leftarrow \mathcal{X}_{\widetilde{i}}$, выполняется при всех достаточно больших $n \in \mathbb{N}$.

Приведем теперь несколько примеров семейств функций $(f_i \mid i \in I)$, которые гипотетически являются односторонними (при подходящем выборе параметров) и широко используются в математической криптографии. При этом мы будем также указывать соответствующие семейства распределений вероятностей $(\mathcal{I}_n \mid n \in \mathbb{N})$ и $(\mathcal{X}_i \mid i \in I)$.

Пример 4.5 (семейство дискретных экспонент)**.** Пусть $(G_d \mid d \in D)$, где $D \subseteq \{0,1\}^*$, — полиномиально вычислимое семейство групп. Пусть также для каждого $n \in \mathbb{N}$ определено распределение вероятностей $\mathcal{D}_n$ на D, а для каждого $d \in D$ — распределение вероятностей $\mathcal{G}_d$ на G_d. Предположим, что семейство $(\mathcal{D}_n \mid n \in \mathbb{N})$ полиномиально конструируемо, когда индексы заданы в унарной записи, а семейство $(\mathcal{G}_d \mid d \in D)$ — полиномиально конструируемо. В качестве I выберем множество всех пар (d, g), где $d \in D$ и $g \in G_d$, а в качестве $\mathcal{I}_n$ — распределение случайной величины $(\widetilde{d}, \widetilde{g})$, где $\widetilde{d} \leftarrow \mathcal{D}_n$ и $\widetilde{g} \leftarrow \mathcal{G}_{\widetilde{d}}$ $(n \in \mathbb{N})$. Кроме того, предположим, что заданы полиномиально вычислимые функции $\alpha, \beta \colon I \to \mathbb{Z}$ такие, что $\alpha(i) \leqslant \beta(i)$ для всех $i \in I$.

Пусть $(d, g) \in I$. Положим $X_{d,g} = \{\alpha(d,g), \ldots, \beta(d,g)\}$. Определим функцию $f_{d,g} \colon X_{d,g} \to G_d$ равенством $f_{d,g}(x) = g^x$ $(x \in X_{d,g})$, а в качестве $\mathcal{X}_{d,g}$ возьмем равномерное распределение на $X_{d,g}$. Задача вычисления по $(1^n, i, f_i(x))$, где $n \in \mathbb{N}$, $i \in \operatorname{supp} \mathcal{I}_n$ и $x \in X_i$, некоторого элемента множества $f_i^{-1}(f_i(x))$ называется *задачей дискретного логарифмирования* (discrete logarithm problem) для данных семейств $(G_d \mid d \in D)$, $(\mathcal{D}_n \mid n \in \mathbb{N})$, $(\mathcal{G}_d \mid d \in D)$ и функций α, β. Другими словами, эта задача заключается в нахождении по $(1^n, d, g, g^x)$, где $n \in \mathbb{N}$, $d \in \operatorname{supp} \mathcal{D}_n$, $g \in \operatorname{supp} \mathcal{G}_d$ и $x \in \{\alpha(d,g), \ldots, \beta(d,g)\}$, некоторого числа $y \in \{\alpha(d,g), \ldots, \beta(d,g)\}$ такого, что $g^y = g^x$.

Пример 4.6 (семейство рюкзачных функций)**.** Пусть $(G_d \mid d \in D)$, $(\mathcal{D}_n \mid n \in \mathbb{N})$ и $(\mathcal{G}_d \mid d \in D)$ — те же, что в примере 4.5, а k — функция из D в $\mathbb{N}$ такая, что функция $d \mapsto 1^{k(d)}$ $(d \in D)$ полиномиально вычислима. В качестве I выберем множество всех наборов $(d, g_1, \ldots, g_{k(d)})$, где $d \in D$ и $g_1, \ldots, g_{k(d)} \in G_d$, а в качестве $\mathcal{I}_n$ — распределение случайной величины $(\widetilde{d}, \widetilde{g}_1, \ldots, \widetilde{g}_{k(d)})$, где $\widetilde{d} \leftarrow \mathcal{D}_n$ и $\widetilde{g}_1, \ldots, \widetilde{g}_{k(\widetilde{d})} \leftarrow \mathcal{G}_{\widetilde{d}}$ $(n \in \mathbb{N})$.

Пусть $i = (d, g_1, \ldots, g_{k(d)}) \in I$. Положим $X_i = \{0,1\}^{k(d)}$. Определим функцию $f_i \colon X_i \to G_d$ равенством $f_i(x) = g_1^{x_{[1]}} \ldots g_{k(d)}^{x_{[k(d)]}}$ $(x \in X_i)$, а в качестве $\mathcal{X}_i$ возьмем равномерное распределение на X_i. Очевидно, что если $S_x = \{j \in \{1, \ldots, k(d)\} \mid x_{[j]} = 1\}$, то $f_i(x) = \prod_{j \in S_x} g_j$ для произвольного $x \in X_i$, где произведение g_j берется в порядке возрастания индексов. По этой причине задача вычисления по $(1^n, i, f_i(x))$, где $n \in \mathbb{N}$, $i \in \operatorname{supp} \mathcal{I}_n$ и $x \in X_i$, некоторого элемента множества $f_i^{-1}(f_i(x))$ называется

задачей о произведении подмножества (subset product problem) для данных семейств $(G_d \mid d \in D)$, $(\mathcal{D}_n \mid n \in \mathbb{N})$, $(\mathcal{G}_d \mid d \in D)$ и функции k.

В случае, когда G_d являются аддитивно записываемыми абелевыми группами, задачу о произведении подмножества естественно называть *задачей о сумме подмножества* (subset sum problem).

Пример 4.7 (семейство функций, вычисляющих произведения простых чисел)**.** Пусть $(\mathcal{P}_n \mid n \in \mathbb{N})$ — семейство распределений вероятностей на множестве двухэлементных множеств простых чисел. Предположим, что это семейство полиномиально конструируемо, когда индексы заданы в унарной записи. В качестве I выберем множество $\{1^n \mid n \in \mathbb{N}\}$, а в качестве $\mathcal{I}_n$ — распределение, сосредоточенное на 1^n $(n \in \mathbb{N})$.

Пусть $n \in \mathbb{N}$. Положим $X_{1^n} = \operatorname{supp} \mathcal{P}_n$. Определим функцию $f_{1^n} \colon X_{1^n} \to \mathbb{N}$ равенством $f_{1^n}(\{p,q\}) = pq$ $(\{p,q\} \in X_{1^n})$, а в качестве $\mathcal{X}_{1^n}$ возьмем распределение $\mathcal{P}_n$ на X_{1^n}. Задача вычисления по $(i, f_i(x))$, где $i = 1^n \in \operatorname{supp} \mathcal{I}_n$ $(n \in \mathbb{N})$ и $x \in X_i$, некоторого элемента множества $f_i^{-1}(f_i(x))$ называется *задачей факторизации целых чисел* (integer factoring problem) для данного семейства $(\mathcal{P}_n \mid n \in \mathbb{N})$; здесь из входных данных исключена строка 1^n, так как она совпадает с i. Другими словами, эта задача заключается в нахождении по $(1^n, pq)$, где $n \in \mathbb{N}$ и $\{p,q\} \in \operatorname{supp} \mathcal{P}_n$, множества $\{p,q\}$ (или, что эквивалентно, одного из его элементов).

5. Трудные и трудно аппроксимируемые функции и предикаты

Пусть f — функция из $\{0,1\}^*$ в $\{0,1\}^*$, а h — полиномиально вычислимая функция, отображающая $\{0,1\}^n$ в $\{0,1\}^{m(n)}$ при любом $n \in \mathbb{N}$, где $m \colon \mathbb{N} \to \mathbb{N} \setminus \{0\}$ — полиномиальный параметр.

Определение 5.1 (трудная функция; см. также [Gol04, определение 2.5.5])**.** Функция h называется *трудной* (hard-core) для функции f, если семейства случайных величин

$$((f(\widetilde{u}_n), h(\widetilde{u}_n)) \mid n \in \mathbb{N}) \quad \text{и} \quad ((f(\widetilde{u}_n), \widetilde{v}_{m(n)}) \mid n \in \mathbb{N}),$$

где $\widetilde{v}_{m(n)} \in_{\mathcal{U}} \{0,1\}^{m(n)}$, вычислительно неотличимы, когда индексы заданы в унарной записи.

Говоря неформально, условие определения 5.1 означает, что из $f(x)$, где x — случайный аргумент, вычислительно трудно извлекать информацию об $h(x)$.

Определение 5.2 (трудно аппроксимируемая (hard to approximate) функция; см. также [Gol04, подразд. 2.7.4, упр. 31])**.** Функция h называется *трудно аппроксимируемой* (hard to approximate) по функции f, если для любого полиномиального вероятностного алгоритма A

$$\Pr[A(1^n, f(\widetilde{u}_n)) = h(\widetilde{u}_n)] \leqslant \frac{1}{2^{m(n)}} + \operatorname{negl}(n).$$

Неформально условие определения 5.2 означает трудную вычислимость $h(x)$ по $f(x)$, где x — случайный аргумент. Очевидно, что значение $h(x)$ для любого $x \in \{0,1\}^n$ можно легко вычислить с вероятностью $2^{-m(n)}$; для этого достаточно выбрать $y \in_{\mathcal{U}} \{0,1\}^{m(n)}$. Отметим, что в отличие от [Gol04, подразд. 2.7.4, упр. 31] мы определяем трудную аппроксимируемость (по f) лишь для полиномиально вычислимых функций.

Трудная для f или трудно аппроксимируемая по f функция, принимающая значения в $\{0,1\}$, называется соответственно *трудным* (hard-core) для f или *трудно аппроксимируемым* (hard to approximate) по f *предикатом* (predicate).

Следующее предложение указывает на тесную связь понятий трудно аппроксимируемой и трудной функции.

Предложение 5.3 ([Gol04, подразд. 2.7.4, упр. 31])**.** *Если h трудна для f, то h трудно аппроксимируема по f. Наоборот, если h трудно аппроксимируема по f и $m(n) = O(\log n)$ при $n \in \mathbb{N} \setminus \{0\}$, то h трудна для f. Таким образом, если $m(n) = O(\log n)$ при $n \in \mathbb{N} \setminus \{0\}$, то h трудна для f тогда и только тогда, когда h трудно аппроксимируема по f.*

Доказательство. Пусть h трудна для f и A — произвольный полиномиальный вероятностный алгоритм. Выберем полиномиальный вероятностный алгоритм D такой, что $D(1^n, y, z) = 1$ тогда и только тогда, когда $A(1^n, y) = z$ при любых $n \in \mathbb{N}$, $y \in \{0,1\}^*$ и $z \in \{0,1\}^{m(n)}$. Пусть $\widetilde{r}_{m(n)} \in_{\mathcal{U}} \{0,1\}^{m(n)}$. Тогда

$$\Pr[A(1^n, f(\widetilde{u}_n)) = h(\widetilde{u}_n)] = \Pr[D(1^n, f(\widetilde{u}_n), h(\widetilde{u}_n)) = 1]$$

и

$$\frac{1}{2^{m(n)}} \geqslant \Pr[A(1^n, f(\widetilde{u}_n)) = \widetilde{r}_{m(n)}] = \Pr[D(1^n, f(\widetilde{u}_n), \widetilde{r}_{m(n)}) = 1].$$

Следовательно,

$$\begin{aligned}\Pr[A(1^n, f(\widetilde{u}_n)) = h(\widetilde{u}_n)] &= \Pr[D(1^n, f(\widetilde{u}_n), \widetilde{r}_{m(n)}) = 1] \\ &\quad + (\Pr[D(1^n, f(\widetilde{u}_n), h(\widetilde{u}_n)) = 1] - \Pr[D(1^n, f(\widetilde{u}_n), \widetilde{r}_{m(n)}) = 1]) \\ &\leqslant \frac{1}{2^{m(n)}} + |\Pr[D(1^n, f(\widetilde{u}_n), h(\widetilde{u}_n)) = 1] - \Pr[D(1^n, f(\widetilde{u}_n), \widetilde{r}_{m(n)}) = 1]| \\ &\leqslant \frac{1}{2^{m(n)}} + \operatorname{negl}(n)\end{aligned}$$

для всех $n \in \mathbb{N}$. Таким образом, h трудно аппроксимируема по f.

Пусть теперь $m(n) = O(\log n)$ при $n \in \mathbb{N} \setminus \{0\}$ и h трудно аппроксимируема по f. Предположим, что h не является трудной для f. Тогда согласно следствию 0.2 существуют полиномиальный вероятностный алгоритм D, полином p и бесконечное множество $N \subseteq \mathbb{N}$ такие, что если

$$q_1(n) = \Pr[D(1^n, f(\widetilde{u}_n), h(\widetilde{u}_n)) = 1] \quad \text{и} \quad q_2(n) = \Pr[D(1^n, f(\widetilde{u}_n), \widetilde{r}_{m(n)}) = 1],$$

где $\widetilde{r}_{m(n)}$ — та же случайная величина, что и выше, то $q_1(n) - q_2(n) > 1/p(n)$ для всех $n \in N$. Пусть A — полиномиальный вероятностный алгоритм, который на произвольном входе вида $(1^n, y)$, где $n \in N$ и $y \in \{0,1\}^*$, работает следующим образом:

1. Выбрать $r \in_{\mathcal{U}} \{0,1\}^{m(n)}$ и $s \in_{\mathcal{U}} \{0,1\}^{m(n)} \setminus \{r\}$.
2. Вычислить $D(1^n, y, r)$.
3. Если на предыдущем шаге получен выход 1, то возвратить r, а в противном случае — s.

Пусть $\widetilde{s}_{m(n)} \in_{\mathcal{U}} \{0,1\}^{m(n)} \setminus \{\widetilde{r}_{m(n)}\}$. Тогда

$$\begin{aligned}&\Pr[A(1^n, f(\widetilde{u}_n)) = h(\widetilde{u}_n)] \\ &\quad = \Pr[D(1^n, f(\widetilde{u}_n), \widetilde{r}_{m(n)}) = 1,\, \widetilde{r}_{m(n)} = h(\widetilde{u}_n)] + \Pr[D(1^n, f(\widetilde{u}_n), \widetilde{r}_{m(n)}) \neq 1,\, \widetilde{s}_{m(n)} = h(\widetilde{u}_n)] \\ &\quad = \Pr[D(1^n, f(\widetilde{u}_n), h(\widetilde{u}_n)) = 1,\, \widetilde{r}_{m(n)} = h(\widetilde{u}_n)] \\ &\qquad + \Pr[D(1^n, f(\widetilde{u}_n), \widetilde{r}_{m(n)}) \neq 1,\, \widetilde{r}_{m(n)} \neq h(\widetilde{u}_n),\, \widetilde{s}_{m(n)} = h(\widetilde{u}_n)] \\ &\quad = \frac{\Pr[D(1^n, f(\widetilde{u}_n), h(\widetilde{u}_n)) = 1]}{2^{m(n)}} + \frac{\Pr[D(1^n, f(\widetilde{u}_n), \widetilde{r}_{m(n)}) \neq 1,\, \widetilde{r}_{m(n)} \neq h(\widetilde{u}_n)]}{2^{m(n)} - 1} \\ &\quad = \frac{q_1(n)}{2^{m(n)}} + \left(1 - q_2(n) - \frac{1}{2^{m(n)}} + \frac{q_1(n)}{2^{m(n)}}\right) \frac{1}{2^{m(n)} - 1} = \frac{1}{2^{m(n)}} + \frac{q_1(n) - q_2(n)}{2^{m(n)} - 1} \\ &\quad > \frac{1}{2^{m(n)}} + \frac{1}{\operatorname{poly}(n)}\end{aligned}$$

для всех $n \in N$. Здесь мы воспользовались тем, что $\Pr[\overline{X} \cap \overline{Y}] = 1 - \Pr[X] - \Pr[Y] + \Pr[X \cap Y]$ для любых событий X и Y ($\overline{X}$ и $\overline{Y}$ обозначают дополнительные события соответственно к X и Y), а также тем, что $2^{m(n)} - 1 \leqslant \operatorname{poly}(n)$ для всех $n \in \mathbb{N}$. Таким образом, получено противоречие с трудной аппроксимируемостью h по f. Следовательно, h трудна для f. □

6. Трудные предикаты для односторонних функций

Если не требовать от функции f никаких свойств, то трудные предикаты могут быть построены легко. Например, если $f(x) = x_{[2,\ldots,|x|]}$ и $b(x) = x_{[1]}$ для всех $x \in \{0,1\}^{\geqslant 1}$, то b — трудный предикат для функции f. Причина этого факта состоит в том, что $f(x)$ не содержит никакой информации о $b(x)$. См. также [Gol04, подразд. 2.5.1]. В то же время имеет место

Предложение 6.1 (см. [Gol04, подразд. 2.7.4, упр. 25]). *Пусть $f\colon \{0,1\}^* \to \{0,1\}^*$ — полиномиально вычислимая инъективная функция, для которой существует трудный предикат. Тогда f — односторонняя функция.*

Доказательство. Пусть b — трудный предикат для f. Пусть также A — произвольный полиномиальный вероятностный алгоритм. Выберем полиномиальный вероятностный алгоритм B, работающий на произвольном входе $(1^n, y)$, где $n \in \mathbb{N}$ и $y \in \{0,1\}^*$, следующим образом:

1. Вычислить $x \leftarrow A(1^n, y)$.

2. Если $f(x) = y$, то возвратить $b(x)$. В противном случае возвратить $\beta \in_{\mathcal{U}} \{0,1\}$.

Тогда ввиду трудной аппроксимируемости b по f (см. предложение 5.3)

$$\frac{1}{2} + \operatorname{negl}(n) \geqslant \Pr[B(1^n, f(\widetilde{u}_n)) = b(\widetilde{u}_n)] = \Pr[A(1^n, f(\widetilde{u}_n)) = \widetilde{u}_n] + \frac{1 - \Pr[A(1^n, f(\widetilde{u}_n)) = \widetilde{u}_n]}{2}.$$

Следовательно,

$$\Pr[A(1^n, f(\widetilde{u}_n)) = \widetilde{u}_n] \leqslant 2\operatorname{negl}(n) = \operatorname{negl}(n).$$

Односторонность функции вытекает теперь из того, что $f^{-1}(f(x)) = \{x\}$ для любого $x \in \{0,1\}^*$. □

Из доказательства предложения 6.1 не следует, что если для полиномиально вычислимой функции f существует трудный предикат, то $\Pr[A(1^n, f(\widetilde{u}_n)) = \widetilde{u}_n] = \operatorname{negl}(n)$ для любого полиномиального вероятностного алгоритма A. Контрпримером к этому утверждению является функция, определенная перед формулировкой предложения 6.1. Кроме того, отметим, что в алгоритме B из доказательства предложения 6.1 при вычислении на входе $(1^n, f(x))$ ($n \in \mathbb{N}$, $x \in \{0,1\}^n$) необходимо отличать случай, когда $A(1^n, f(x)) = x$, от случая, когда это не так. А если $|f^{-1}(f(x))| \geqslant 2$, то это невозможно, так как $f(x)$ не содержит информации о том, какой из элементов $f^{-1}(f(x))$ был выбран изначально в качестве x.

Следующая теорема Гольдрайха—Левина (см. [GL89], [Gol04, теорема 2.5.2], [Lub96, лекция 7]) дает трудный предикат для любой односторонней функции, имеющей вид $xy \mapsto (f(x), y)$ ($x, y \in \{0,1\}^n$, $n \in \mathbb{N}$), где f — односторонняя функция.

Теорема 6.2. *Пусть $f\colon \{0,1\}^* \to \{0,1\}^*$ — односторонняя функция. Пусть также функция $g\colon \bigcup_{n\in\mathbb{N}}\{0,1\}^{2n} \to \{0,1\}^*$ определена равенством $g(xy) = (f(x), y)$ для всех $x, y \in \{0,1\}^n$ и $n \in \mathbb{N}$. (Очевидно, что g — односторонняя функция.) Тогда предикат b, определенный равенством*

$$b(xy) = \bigoplus_{i=1}^{n} (x_{[i]} \odot y_{[i]}) \quad (x, y \in \{0,1\}^n,\ n \in \mathbb{N}),$$

является трудным для функции g.

Доказательство. Пусть $\widetilde{v}_n \in_{\mathcal{U}} \{0,1\}^n$. Предположим, что b не является трудным предикатом для функции g. Тогда по предложению 5.3 b не является трудно аппроксимируемым по этой функции предикатом. Это значит, что существуют полиномиальный вероятностный алгоритм A, полином p и бесконечное множество $N \subseteq \mathbb{N} \setminus \{0\}$ такие, что

$$\varepsilon(n) = \Pr[A(1^{2n}, f(\widetilde{u}_n), \widetilde{v}_n) = b(\widetilde{u}_n\widetilde{v}_n)] - \frac{1}{2} > \frac{1}{p(n)} \tag{11}$$

для всех $n \in N$. (Очевидно, что для любого полинома q существует полином p такой, что $q(2n) = p(n)$ для всех $n \in \mathbb{N}$.) Не ограничивая общности, мы считаем, что алгоритм A всегда возвращает бит.

Пусть $n \in N$ и $x \in \{0,1\}^n$. Положим

$$t(x) = \Pr[A(1^{2n}, f(x), \widetilde{v}_n) = b(x\widetilde{v}_n)] \quad \text{и} \quad E_n = \left\{ x \in \{0,1\}^n \;\middle|\; t(x) \geqslant \frac{1}{2} + \frac{\varepsilon(n)}{2} \right\}.$$

Тогда

$$\begin{aligned}\Pr\left[t(\widetilde{u}_n) < \frac{1}{2} + \frac{\varepsilon(n)}{2}\right] &= \Pr\left[1 - t(\widetilde{u}_n) > \frac{1}{2} - \frac{\varepsilon(n)}{2}\right] \\ &\leqslant \frac{\mathsf{E}[1 - t(\widetilde{u}_n)]}{1/2 - \varepsilon(n)/2} = \frac{1/2 - \varepsilon(n)}{1/2 - \varepsilon(n)/2} = 1 - \frac{\varepsilon(n)}{1 - \varepsilon(n)} < 1 - \varepsilon(n).\end{aligned}$$

Здесь мы воспользовались неравенством Чебышёва (очевидно, что $\varepsilon(n) \leqslant 1/2$) и равенством $\mathsf{E}[t(\widetilde{u}_n)] = 1/2 + \varepsilon(n)$. Следовательно,

$$\Pr[\widetilde{u}_n \in E_n] = \Pr\left[t(\widetilde{u}_n) \geqslant \frac{1}{2} + \frac{\varepsilon(n)}{2}\right] > \varepsilon(n) > \frac{1}{p(n)} \tag{12}$$

ввиду неравенства (11).

Для произвольного $i \in \{1,\dots,n\}$ положим $e_i = 0^{i-1}10^{n-i} \in \{0,1\}^n$ (т. е. i-й бит строки e_i равен 1, а остальные биты — 0).

Для завершения доказательства теоремы достаточно построить полиномиальный вероятностный алгоритм B такой, что $\Pr[B(1^n, f(x)) = x] \geqslant 1/\operatorname{poly}(n)$ для всех $n \in N$ и $x \in E_n$. Действительно, тогда из последнего неравенства и неравенства (12) следует, что

$$\Pr[B(1^n, f(\widetilde{u}_n)) \in f^{-1}(f(\widetilde{u}_n))] \geqslant \Pr[B(1^n, f(\widetilde{u}_n)) = \widetilde{u}_n \,|\, \widetilde{u}_n \in E_n] \Pr[\widetilde{u}_n \in E_n] > \frac{1}{\operatorname{poly}(n)}$$

для всех $n \in N$, а это противоречит односторонности функции f.

Алгоритм B на входе $(1^n, f(x))$, где $n \in N$ и $x \in E_n$, будет искать каждый бит $x_{[i]}$ отдельно. Для этого алгоритм B

- выбирает случайные строки $r_1,\dots,r_{\pi(n)} \in \{0,1\}^n$, где π — некоторый полиномиальный параметр на N, принимающий лишь нечетные значения;
- для каждых $i \in \{1,\dots,n\}$ и $j \in \{1,\dots,\pi(n)\}$ вычисляет биты $\beta_{i,j}$ и ρ_j, являющиеся предполагаемыми значениями $b(x(r_j \oplus e_i))$ и $b(xr_j)$ соответственно;
- выбирает в качестве предполагаемого значения $x_{[i]}$ бит, который встречается в последовательности $(\beta_{i,j} \oplus \rho_j \,|\, j \in \{1,\dots,\pi(n)\})$ более $\pi(n)/2$ раз.

Очевидно, что если $\beta_{i,j} = b(x(r_j \oplus e_i))$ и $\rho_j = b(xr_j)$ для более чем половины индексов $j \in \{1,\dots,\pi(n)\}$, то $x_{[i]}$ будет найден правильно, так как $b(x(r_j \oplus e_i)) \oplus b(xr_j) = b(xe_i) = x_{[i]}$.

Бит $\beta_{i,j}$ вычисляется как $A(1^{2n}, f(x), r_j \oplus e_i)$. Мы не получим нужную оценку вероятности успеха алгоритма B, если будем вычислять ρ_j как $A(1^{2n}, f(x), r_j)$. Вместо этого алгоритм пытается угадать значения $b(xr_j)$ для всех j. Но если просто выбрать $\rho_j \in_{\mathcal{U}} \{0,1\}$, то вероятность того, что $\rho_j = b(xr_j)$ для всех $j \in \{1,\dots,\pi(n)\}$, будет равна $2^{-\pi(n)}$, а эта величина при нужном для нас росте $\pi(n)$ будет пренебрежимо малой как функция от n. Чтобы обойти это препятствие, алгоритм B

- полагает $\pi(n) = 2^{l(n)} - 1$, где $l(n) = \lceil \log_2(2np(n)^2 + 1) \rceil$;
- выбирает $s_k \in_{\mathcal{U}} \{0,1\}^n$ и $\sigma_k \in_{\mathcal{U}} \{0,1\}$ для всех $k \in \{1,\dots,l(n)\}$;
- вычисляет $r_J = \bigoplus_{j \in J} s_j$ и $\rho_J = \bigoplus_{j \in J} \sigma_j$ для всех непустых множеств $J \subseteq \{1,\dots,l(n)\}$.

В этом случае если $\sigma_k = b(xs_k)$ для всех $k \in \{1,\dots,l(n)\}$ (что выполнено с вероятностью $2^{-l(n)} \geqslant 1/\operatorname{poly}(n)$), то и $\rho_J = b(xr_J)$ для всех непустых множеств $J \subseteq \{1,\dots,l(n)\}$. В то же время r_J, рассматриваемые как случайные величины, лишь попарно независимы. Этого достаточно для наших целей.

Приведем теперь формальное описание работы полиномиального вероятностного алгоритма B на произвольном входе вида $(1^n, z)$, где $n \in \mathbb{N}$ и $z \in \{0,1\}^*$.

1. Вычислить $l(n) = \lceil \log_2(2np(n)^2 + 1) \rceil$.
2. Для каждого $k \in \{1,\dots,l(n)\}$ выбрать $s_k \in_{\mathcal{U}} \{0,1\}^n$ и $\sigma_k \in_{\mathcal{U}} \{0,1\}$.
3. Для каждого непустого множества $J \subseteq \{1,\dots,l(n)\}$ вычислить $r_J = \bigoplus_{j \in J} s_j$ и $\rho_J = \bigoplus_{j \in J} \sigma_j$.
4. Для каждого $i \in \{1,\dots,n\}$ и каждого непустого множества $J \subseteq \{1,\dots,l(n)\}$ вычислить $y_{i,J} \leftarrow A(1^{2n}, z, r_J \oplus e_i) \oplus \rho_J$.
5. Возвратить $y = y_1 \dots y_n$, где y_i — бит, встречающийся более $(2^{l(n)} - 1)/2$ раз среди $y_{i,J}$, когда J пробегает множество всех непустых подмножеств $\{1,\dots,l(n)\}$ $(i \in \{1,\dots,n\})$.

Очевидно, что если

- для любого $i \in \{1,\dots,n\}$ число непустых множеств $J \subseteq \{1,\dots,l(n)\}$ таких, что $A(1^{2n}, f(x), r_J \oplus e_i) \oplus b(xr_J) = x_{[i]}$, больше $(2^{l(n)} - 1)/2$ и
- $\sigma_k = b(xs_k)$ для всех $k \in \{1,\dots,l(n)\}$,

то $B(1^n, f(x)) = x$.

Обозначим для краткости через Γ_n множество всех непустых подмножеств $\{1, \ldots, l(n)\}$. Кроме того, введем для каждого $k \in \{1, \ldots, l(n)\}$ случайные величины $\widetilde{s}_k \in_{\mathcal{U}} \{0,1\}^n$ и $\widetilde{\sigma}_k \in_{\mathcal{U}} \{0,1\}$ и положим $\widetilde{r}_J = \bigoplus_{j \in J} \widetilde{s}_j$ и $\widetilde{\rho}_J = \bigoplus_{j \in J} \widetilde{\sigma}_j$ для всех $J \in \Gamma_n$.

Фиксируем $n \in N$, $x \in E_n$ и $i \in \{1, \ldots, n\}$. Пусть $J \in \Gamma_n$. Непосредственно проверяется, что

$$A(1^{2n}, f(x), \widetilde{r}_J \oplus e_i) \oplus b(x\widetilde{r}_J) = x_{[i]} \iff A(1^{2n}, f(x), \widetilde{r}_J \oplus e_i) = b(x(\widetilde{r}_J \oplus e_i)). \tag{13}$$

Определим случайную величину $\widetilde{\zeta}_J$ как принимающую значение 1, если выполнено любое из эквивалентных равенств в (13), и 0 в противном случае. Очевидно, что случайная величина $\widetilde{r}_J$ (а поэтому и $\widetilde{r}_J \oplus e_i$) распределена равномерно на $\{0,1\}^n$. Следовательно, $\mathsf{E}[\widetilde{\zeta}_J] = \Pr[\widetilde{\zeta}_J = 1] = t(x)$. Кроме того, при различных $J, K \in \Gamma_n$ случайные величины $\widetilde{\zeta}_J$ и $\widetilde{\zeta}_K$ независимы, так как независимы $\widetilde{r}_J$ и $\widetilde{r}_K$. Поэтому

$$\begin{aligned}\mathsf{E}\left[\left(\sum_{J \in \Gamma_n} (\widetilde{\zeta}_J - t(x))\right)^2\right] &= \sum_{J \in \Gamma_n} \mathsf{E}[(\widetilde{\zeta}_J - t(x))^2] + \sum_{J,K \in \Gamma_n \mid J \neq K} \mathsf{E}[\widetilde{\zeta}_J - t(x)]\,\mathsf{E}[\widetilde{\zeta}_K - t(x)] \\ &= |\Gamma_n|(t(x)^2(1-t(x)) + (1-t(x))^2 t(x)) = (2^{l(n)} - 1)t(x)(1-t(x)).\end{aligned} \tag{14}$$

Обозначим через M_i событие, состоящее в том, что число множеств $J \in \Gamma_n$, для которых $A(1^{2n}, f(x), \widetilde{r}_J \oplus e_i) \oplus b(x\widetilde{r}_J) = x_{[i]}$, больше $(2^{l(n)} - 1)/2$. Другими словами, M_i состоит в том, что $\sum_{J \in \Gamma_n} \widetilde{\zeta}_J > (2^{l(n)} - 1)/2$. Пусть также $\overline{M}_i$ — дополнительное событие к M_i. Тогда, применив неравенство Чебышёва и равенство (14), мы получаем, что

$$\begin{aligned}\Pr[\overline{M}_i] &= \Pr\left[\sum_{J \in \Gamma_n} \widetilde{\zeta}_J \leqslant \frac{2^{l(n)} - 1}{2}\right] = \Pr\left[\sum_{J \in \Gamma_n} (\widetilde{\zeta}_J - t(x)) \leqslant (2^{l(n)} - 1)\left(\frac{1}{2} - t(x)\right)\right] \\ &\leqslant \Pr\left[\sum_{J \in \Gamma_n} (\widetilde{\zeta}_J - t(x)) \leqslant -\frac{2^{l(n)} - 1}{2p(n)}\right] \leqslant \Pr\left[\left(\sum_{J \in \Gamma_n} (\widetilde{\zeta}_J - t(x))\right)^2 \geqslant \left(\frac{2^{l(n)} - 1}{2p(n)}\right)^2\right] \\ &\leqslant \frac{\mathsf{E}\left[\left(\sum_{J \in \Gamma_n} (\widetilde{\zeta}_J - t(x))\right)^2\right]}{(2^{l(n)} - 1)^2/4p(n)^2} = \frac{(2^{l(n)} - 1)t(x)(1-t(x))}{(2^{l(n)} - 1)^2/4p(n)^2} \leqslant \frac{1}{2n}.\end{aligned}$$

Здесь мы также воспользовались неравенствами $t(x) \geqslant 1/2 + \varepsilon(n)/2 > 1/2 + 1/2p(n)$ (см. (11)), $t(x)(1-t(x)) \leqslant 1/4$ и $2^{l(n)} - 1 \geqslant 2np(n)^2$. Следовательно,

$$\Pr\left[\bigcap_{i=1}^n M_i\right] = 1 - \Pr\left[\bigcup_{i=1}^n \overline{M}_i\right] \geqslant 1 - \frac{1}{2} = \frac{1}{2}.$$

и

$$\begin{aligned}\Pr[B(1^n, f(x)) = x] &\geqslant \Pr\left[\bigcap_{i=1}^n M_i,\ \forall\, k \in \{1, \ldots, l(n)\}\ \widetilde{\sigma}_j = b(x\widetilde{s}_j)\right] \\ &= \Pr\left[\bigcap_{i=1}^n M_i\right] \prod_{j=1}^{l(n)} \Pr[\widetilde{\sigma}_j = b(x\widetilde{s}_j)] \geqslant \frac{1}{2 \cdot 2^{l(n)}} \geqslant \frac{1}{\mathrm{poly}(n)}\end{aligned}$$

(см. замечание после описания алгоритма B), что и требовалось. □

***Замечание* 6.3.** Функция g и предикат b из теоремы 6.2 определены на множестве $\bigcup_{n \in \mathbb{N}} \{0,1\}^{2n}$. Продолжим функцию g до односторонней функции $g' \colon \{0,1\}^* \to \{0,1\}^*$ любым из способов, указанных в замечании 2.4. Пусть также предикат b' определен на множестве $\{0,1\}^*$ следующим образом: $b'(x) = b(x)$, если $|x|$ четна, и $b'(x) = b(x_{[1,\ldots,|x|-1]})$ в противном случае. Тогда легко видеть (используя теорему 6.2), что b' — трудный предикат для g'. Очевидно также, что если f — односторонняя перестановка, то g можно определить как одностороннюю перестановку $xy \mapsto f(x)y$ ($x, y \in \{0,1\}^n$, $n \in \mathbb{N}$). В этом случае продолжение g на $\{0,1\}^*$ вторым из способов, указанных в замечании 2.4, является односторонней перестановкой.

***Замечание* 6.4.** Приведенное выше доказательство теоремы 6.2 показывает, что она верна не только для односторонних функций f, но и для таких полиномиально вычислимых функций $f\colon \{0,1\}^* \to \{0,1\}^*$, что $\Pr[A(1^n, f(\widetilde{u}_n)) = \widetilde{u}_n] = \mathrm{negl}(n)$ для любого полиномиального вероятностного алгоритма A. Очевидно, что любая односторонняя функция $f\colon \{0,1\}^* \to \{0,1\}^*$ полиномиально вычислима и удовлетворяет последнему условию, но обратное неверно.

7. Псевдослучайные генераторы

Определение 7.1 (псевдослучайный генератор)**.** Функция $g\colon \{0,1\}^* \to \{0,1\}^*$, отображающая $\{0,1\}^n$ в $\{0,1\}^{m(n)}$ для всех $n \in \mathbb{N}$, называется *псевдослучайным генератором* (pseudorandom generator) или, более подробно, *генератором псевдослучайных последовательностей* (pseudorandom sequence generator), если

1) g полиномиально вычислима;

2) функция m удовлетворяет неравенству $m(n) > n$ для всех $n \in \mathbb{N}$;

3) семейства случайных величин $(g(\widetilde{u}_n) \mid n \in \mathbb{N})$ и $(\widetilde{u}_{m(n)} \mid n \in \mathbb{N})$ вычислительно неотличимы, когда индексы заданы в унарной записи.

Очевидно, что если не требовать выполнения условия 2 в этом определении, то понятие псевдослучайного генератора будет бессодержательным. Легко также видеть, что если функция g отображает $\{0,1\}^n$ в $\{0,1\}^{m(n)}$ для всех $n \in \mathbb{N}$, причем выполнено условие 2 определения 7.1, то статистическое расстояние между случайными величинами $g(\widetilde{u}_n)$ и $\widetilde{u}_{m(n)}$ ограничивается снизу $1/2$ и, следовательно, семейства случайных величин из условия 3 определения 7.1 не могут быть статистически неотличимыми (независимо от того, заданы индексы в бинарной или унарной записи).

В определении 7.1 можно заменить условие 3 на условие непредсказуемости следующего бита (см. ниже), которое иногда более удобно, чем условие 3. Полученное определение будет эквивалентно исходному. Этот результат приписывается Яо.

Пусть $(\widetilde{x}_n \mid n \in \mathbb{N})$ — семейство случайных величин такое, что $\widetilde{x}_n$ принимает значения в множестве $\{0,1\}^{m(n)}$ для любого $n \in \mathbb{N}$, где $m\colon \mathbb{N} \to \mathbb{N} \setminus \{0\}$ — полиномиальный параметр.

Определение 7.2 (непредсказуемость следующего бита)**.** Семейство $(\widetilde{x}_n \mid n \in \mathbb{N})$ удовлетворяет условию *непредсказуемости следующего бита* (next bit unpredictability), если для любого полиномиального вероятностного алгоритма A

$$\Pr[A(1^n, (\widetilde{x}_n)_{[1,\ldots,\widetilde{i}-1]}) = (\widetilde{x}_n)_{[\widetilde{i}]}] \leqslant \frac{1}{2} + \mathrm{negl}(n),$$

где $\widetilde{i} \in_{\mathcal{U}} \{1, \ldots, m(n)\}$.

Теорема 7.3. *Семейство $(\widetilde{x}_n \mid n \in \mathbb{N})$ удовлетворяет условию непредсказуемости следующего бита тогда и только тогда, когда это семейство и $(\widetilde{u}_{m(n)} \mid n \in \mathbb{N})$ вычислительно неотличимы, когда индексы заданы в унарной записи.*

Доказательство. Предположим, что $(\widetilde{x}_n \mid n \in \mathbb{N})$ и $(\widetilde{u}_{m(n)} \mid n \in \mathbb{N})$ не являются вычислительно неотличимыми, когда индексы заданы в унарной записи. Ввиду следствия 0.2 существуют полиномиальный вероятностный алгоритм D, полином p и бесконечное множество $N \subseteq \mathbb{N}$ такие, что для всех $n \in N$

$$\Pr[D(1^n, \widetilde{x}_n) = 1] - \Pr[D(1^n, \widetilde{u}_{m(n)}) = 1] > \frac{1}{p(n)}. \tag{15}$$

Выберем полиномиальный вероятностный алгоритм A, работающий на произвольном входе $(1^n, x)$, где $n \in N$ и $x \in \{0,1\}^{\leqslant m(n)-1}$, следующим образом:

1. Выбрать $y \in_{\mathcal{U}} \{0,1\}^{m(n)-|x|}$.
2. Вычислить $b \leftarrow D(1^n, xy)$.
3. Возвратить $y_{[1]}$, если $b = 1$, и $\overline{y}_{[1]}$ в противном случае.

Фиксируем $n \in N$. Для каждого $i \in \{0, \dots, m(n)\}$ положим для краткости

$$\delta_i(n) = \Pr[D(1^n, (\widetilde{x}_n)_{[1,\dots,i]}(\widetilde{u}_{m(n)})_{[i+1,\dots,m(n)]}) = 1].$$

Тогда

$$\begin{aligned}
\Pr[A(1^n, (\widetilde{x}_n)_{[1,\dots,i-1]}) = (\widetilde{x}_n)_{[i]}] &= \Pr[A(1^n, (\widetilde{x}_n)_{[1,\dots,i-1]}) = (\widetilde{x}_n)_{[i]},\ (\widetilde{u}_{m(n)})_{[i]} = (\widetilde{x}_n)_{[i]}] \\
&\quad + \Pr[A(1^n, (\widetilde{x}_n)_{[1,\dots,i-1]}) = (\widetilde{x}_n)_{[i]},\ (\widetilde{u}_{m(n)})_{[i]} = \overline{(\widetilde{x}_n)}_{[i]}] \\
&= \Pr[A(1^n, (\widetilde{x}_n)_{[1,\dots,i-1]}) = (\widetilde{u}_{m(n)})_{[i]},\ (\widetilde{u}_{m(n)})_{[i]} = (\widetilde{x}_n)_{[i]}] \\
&\quad + \Pr[A(1^n, (\widetilde{x}_n)_{[1,\dots,i-1]}) = \overline{(\widetilde{u}_{m(n)})}_{[i]},\ (\widetilde{u}_{m(n)})_{[i]} = \overline{(\widetilde{x}_n)}_{[i]}] \\
&= \Pr[D(1^n, (\widetilde{x}_n)_{[1,\dots,i-1]}(\widetilde{u}_{m(n)})_{[i,\dots,m(n)]}) = 1,\ (\widetilde{u}_{m(n)})_{[i]} = (\widetilde{x}_n)_{[i]}] \\
&\quad + \Pr[D(1^n, (\widetilde{x}_n)_{[1,\dots,i-1]}(\widetilde{u}_{m(n)})_{[i,\dots,m(n)]}) \neq 1,\ (\widetilde{u}_{m(n)})_{[i]} = \overline{(\widetilde{x}_n)}_{[i]}] \\
&= \Pr[D(1^n, (\widetilde{x}_n)_{[1,\dots,i]}(\widetilde{u}_{m(n)})_{[i+1,\dots,m(n)]}) = 1,\ (\widetilde{u}_{m(n)})_{[i]} = (\widetilde{x}_n)_{[i]}] \\
&\quad + \Pr[D(1^n, (\widetilde{x}_n)_{[1,\dots,i-1]}(\widetilde{u}_{m(n)})_{[i,\dots,m(n)]}) \neq 1,\ (\widetilde{u}_{m(n)})_{[i]} = \overline{(\widetilde{x}_n)}_{[i]}] = \frac{\delta_i(n) + 1 - \delta_{i-1}(n)}{2}
\end{aligned}$$

при любом $i \in \{1, \dots, m(n)\}$. Здесь мы воспользовались тем, что если $(\widetilde{u}_{m(n)})_{[i]} = \overline{(\widetilde{x}_n)}_{[i]}$, то

$$(\widetilde{u}_{m(n)})_{[i,\dots,m(n)]} = \overline{(\widetilde{x}_n)}_{[i]}(\widetilde{u}_{m(n)})_{[i+1,\dots,m(n)]}$$

не зависит от $(\widetilde{u}_{m(n)})_{[i]}$. Следовательно, если $\widetilde{i} \in_{\mathcal{U}} \{1, \dots, m(n)\}$, то

$$\begin{aligned}
\Pr[A(1^n, (\widetilde{x}_n)_{[1,\dots,\widetilde{i}-1]}) = (\widetilde{x}_n)_{[\widetilde{i}]}] &= \frac{1}{m(n)} \sum_{i=1}^{m(n)} \Pr[A(1^n, (\widetilde{x}_n)_{[1,\dots,i-1]}) = (\widetilde{x}_n)_{[i]}] \\
&= \frac{1}{2} + \frac{1}{2m(n)} \sum_{i=1}^{m(n)} (\delta_i(n) - \delta_{i-1}(n)) = \frac{1}{2} + \frac{\delta_{m(n)}(n) - \delta_0(n)}{2m(n)} \\
&= \frac{1}{2} + \frac{\Pr[D(1^n, \widetilde{x}_n) = 1] - \Pr[D(1^n, \widetilde{u}_{m(n)}) = 1]}{2m(n)} > \frac{1}{2} + \frac{1}{2p(n)m(n)} \geqslant \frac{1}{2} + \frac{1}{\mathrm{poly}(n)}
\end{aligned}$$

для всех $n \in N$ (см. неравенство (15)). Это показывает, что $(\widetilde{x}_n \,|\, n \in \mathbb{N})$ не удовлетворяет условию непредсказуемости следующего бита.

Предположим теперь, что $(\widetilde{x}_n \,|\, n \in \mathbb{N})$ и $(\widetilde{u}_{m(n)} \,|\, n \in \mathbb{N})$ вычислительно неотличимы, когда индексы заданы в унарной записи. Пусть A — произвольный полиномиальный вероятностный алгоритм. Выберем полиномиальный вероятностный алгоритм D, работающий на произвольном входе $(1^n, x)$, где $n \in \mathbb{N}$ и $x \in \{0,1\}^{m(n)}$, следующим образом:

1. Выбрать $i \in_{\mathcal{U}} \{1, \dots, m(n)\}$.
2. Вычислить $b \leftarrow A(1^n, x_{[1,\dots,i-1]})$.
3. Возвратить 1, если $b = x_{[i]}$, и 0 в противном случае.

Пусть $n \in \mathbb{N}$ и $\widetilde{i} \in_{\mathcal{U}} \{1, \dots, m(n)\}$. Тогда

$$\Pr[D(1^n, \widetilde{x}_n) = 1] = \Pr[A(1^n, \widetilde{x}_{[1,\dots,\widetilde{i}-1]}) = \widetilde{x}_{[\widetilde{i}]}]$$

и

$$\begin{aligned}
\Pr[D(1^n, \widetilde{u}_{m(n)}) = 1] &= \Pr[A(1^n, (\widetilde{u}_{m(n)})_{[1,\dots,\widetilde{i}-1]}) = (\widetilde{u}_{m(n)})_{[\widetilde{i}]}] \\
&= \frac{1}{m(n)} \sum_{i=1}^{m(n)} \Pr[A(1^n, (\widetilde{u}_{m(n)})_{[1,\dots,i-1]}) = (\widetilde{u}_{m(n)})_{[i]}] \leqslant \frac{1}{2}.
\end{aligned}$$

Следовательно,

$$\begin{aligned}
\Pr[A(1^n, \widetilde{x}_{[1,\dots,\widetilde{i}-1]}) = \widetilde{x}_{[\widetilde{i}]}] - \frac{1}{2} &\leqslant \Pr[D(1^n, \widetilde{x}_n) = 1] - \Pr[D(1^n, \widetilde{u}_{m(n)}) = 1] \\
&\leqslant |\Pr[D(1^n, \widetilde{x}_n) = 1] - \Pr[D(1^n, \widetilde{u}_{m(n)}) = 1]| = \mathrm{negl}(n).
\end{aligned}$$

Это показывает, что $(\widetilde{x}_n \,|\, n \in \mathbb{N})$ удовлетворяет условию непредсказуемости следующего бита. □

Отметим, что доказательство первой части теоремы 7.3 использует гибридный метод. Гибридами здесь являются случайные величины $\widetilde{z}_i = (\widetilde{x}_n)_{[1,\dots,i]}(\widetilde{u}_{m(n)})_{[i+1,\dots,m(n)]}$, где $i \in \{0,\dots,m(n)\}$.

Из теоремы 7.3 непосредственно вытекает

Следствие 7.4. *Функция $g\colon \{0,1\}^* \to \{0,1\}^*$, отображающая $\{0,1\}^n$ в $\{0,1\}^{m(n)}$ для всех $n \in \mathbb{N}$, является псевдослучайным генератором тогда и только тогда, когда она удовлетворяет условиям 1 и 2 определения 7.1 и семейство случайных величин $(g(\widetilde{u}_n) \mid n \in \mathbb{N})$ удовлетворяет условию непредсказуемости следующего бита.*

***Замечание* 7.5.** Из свойств вычислительной неотличимости следует, что если g — псевдослучайный генератор, то $x \mapsto g(x)_{[1,\dots,n+1]}$ $(x \in \{0,1\}^n,\ n \in \mathbb{N})$ — псевдослучайный генератор, отображающий $\{0,1\}^n$ в $\{0,1\}^{n+1}$ для всех $n \in \mathbb{N}$.

Пусть g — произвольная функция, отображающая $\{0,1\}^n$ в $\{0,1\}^{n+1}$ при любом $n \in \mathbb{N}$. Для произвольных $n, k \in \mathbb{N}$ определим функцию $g_{n,k}$ на множестве $\{0,1\}^n$ индукцией по k следующим образом:

$$g_{n,0}(x) = x, \quad g_{n,k+1}(x) = g(x)_{[1]}g_{n,k}(g(x)_{[2,\dots,n+1]}) \quad (n,k \in \mathbb{N},\ x \in \{0,1\}^n).$$

Очевидно, что $g_{n,k}$ отображает $\{0,1\}^n$ в $\{0,1\}^{n+k}$ при любых $n, k \in \mathbb{N}$. Легко также видеть, что функция $(1^k, x) \mapsto g_{n,k}(x)$, где $n, k \in \mathbb{N}$ и $x \in \{0,1\}^n$, полиномиально вычислима.

Лемма 7.6 (см. [Lub96, лекция 4])**.** *Пусть g — псевдослучайный генератор, отображающий $\{0,1\}^n$ в $\{0,1\}^{n+1}$ при любом $n \in \mathbb{N}$, и $k\colon \mathbb{N} \to \mathbb{N} \setminus \{0\}$ — полиномиальный параметр. Тогда функция $g_k\colon \{0,1\}^* \to \{0,1\}^*$ такая, что $g_k(x) = g_{n,k(n)}(x)$ для каждых $n \in \mathbb{N}$ и $x \in \{0,1\}^n$, является псевдослучайным генератором, отображающим $\{0,1\}^n$ в $\{0,1\}^{n+k(n)}$ при всех $n \in \mathbb{N}$.*

Доказательство. Легко видеть, что условия 1 и 2 определения 7.1 выполняются для функции g_k. Для доказательства условия 3 этого определения воспользуемся одной из разновидностей гибридного метода.

Предположим, что условие 3 определения 7.1 не выполнено для функции g_k. Тогда согласно следствию 0.2 существуют полиномиальный вероятностный алгоритм D, полином p и бесконечное множество $N \subseteq \mathbb{N}$ такие, что для всех $n \in N$

$$\Pr[D(1^n, g_{n,k(n)}(\widetilde{u}_n) = 1)] - \Pr[D(1^n, \widetilde{u}_{n+k(n)}) = 1)] > \frac{1}{p(n)}. \tag{16}$$

Выберем полиномиальный вероятностный алгоритм E, работающий на произвольном входе $(1^n, y)$, где $n \in N$ и $y \in \{0,1\}^{n+1}$, следующим образом:

1. Выбрать $i \in_{\mathcal{U}} \{1,\dots,k(n)\}$ и $w \in_{\mathcal{U}} \{0,1\}^{k(n)-i}$.
2. Вычислить $D(1^n, wy_{[1]}g_{n,i-1}(y_{[2,\dots,n+1]}))$ и возвратить результат (если он есть).

Пусть $n \in N$, $\widetilde{v}_{k(n)} \in_{\mathcal{U}} \{0,1\}^{k(n)}$, $\widetilde{i} \in_{\mathcal{U}} \{1,\dots,k(n)\}$ и

$$\delta_i(n) = \Pr[D(1^n, (\widetilde{v}_{k(n)})_{[1,\dots,k(n)-i]}g_{n,i}(\widetilde{u}_n)) = 1]$$

для каждого $i \in \{0,\dots,k(n)\}$. Тогда легко видеть, что

$$\Pr[E(1^n, g(\widetilde{u}_n)) = 1] = \Pr[D(1^n, (\widetilde{v}_{k(n)})_{[1,\dots,k(n)-\widetilde{i}]}g_{n,\widetilde{i}}(\widetilde{u}_n)) = 1] = \frac{1}{k(n)}\sum_{i=1}^{k(n)}\delta_i(n)$$

и

$$\Pr[E(1^n, \widetilde{u}_{n+1}) = 1] = \Pr[D(1^n, (\widetilde{v}_{k(n)})_{[1,\dots,k(n)-\widetilde{i}+1]}g_{n,\widetilde{i}-1}(\widetilde{u}_n)) = 1] = \frac{1}{k(n)}\sum_{i=1}^{k(n)}\delta_{i-1}(n).$$

Поэтому

$$\Pr[E(1^n, g(\widetilde{u}_n)) = 1] - \Pr[E(1^n, \widetilde{u}_{n+1}) = 1] = \frac{\delta_{k(n)}(n) - \delta_0(n)}{k(n)} > \frac{1}{p(n)k(n)} \geqslant \frac{1}{\operatorname{poly}(n)}$$

для всех $n \in N$ ввиду неравенства (16) и того, что

$$\Pr[D(1^n, g_{n,k(n)}(\widetilde{u}_n) = 1)] = \delta_{k(n)}(n) \quad \text{и} \quad \Pr[D(1^n, \widetilde{u}_{n+k(n)}) = 1)] = \delta_0(n).$$

Таким образом, получено противоречие с тем, что g — псевдослучайный генератор. □

Конструкция, аналогичная описанной перед леммой 7.6, приводится в [Gol04, подразд. 3.3.2]; см. также [Gol04, подразд. 3.8.4, упр. 19].

Из замечания 7.5 и леммы 7.6 непосредственно вытекает

Следствие 7.7. *Если существует какой-либо псевдослучайный генератор, то для любого полиномиального параметра m на $\mathbb{N}$, удовлетворяющего неравенству $m(n) > n$ при любом $n \in \mathbb{N}$, существует псевдослучайный генератор, отображающий $\{0,1\}^n$ в $\{0,1\}^{m(n)}$ при всех $n \in \mathbb{N}$.*

Вопрос о существовании псевдослучайных генераторов решается следующей теоремой, принадлежащей Хостаду, Импальяццо, Левину и Луби [HILL99]. См. также [Hås90, ILL89, HILL91, IL89], [Gol04, разд. 3.5], [Lub96, лекция 10]. Более простое, чем в [HILL99], доказательство этой теоремы см. в [Hol06].

Теорема 7.8. *Псевдослучайные генераторы существуют тогда и только тогда, когда существуют односторонние функции.*

Мы не будем приводить в настоящем курсе полного доказательства этой теоремы (ввиду сложности этого доказательства). Будут доказаны лишь следующие факты:

- если существуют псевдослучайные генераторы, то существуют и односторонние функции;
- если существуют односторонние перестановки, то существуют и псевдослучайные генераторы.

Первый из этих фактов вытекает из замечания 7.5 и следующего предложения:

Предложение 7.9. *Пусть g — псевдослучайный генератор, отображающий $\{0,1\}^n$ в $\{0,1\}^{m(n)}$ при любом $n \in \mathbb{N}$. Предположим, что функция m инъективна. Тогда g — односторонняя функция.*

Доказательство. Полиномиальная вычислимость функции g имеет место ввиду определения псевдослучайного генератора. Пусть теперь A — произвольный полиномиальный вероятностный алгоритм. Выберем полиномиальный вероятностный алгоритм D такой, что $D(1^n, y) = 1$ тогда и только тогда, когда $A(1^n, y) \in g^{-1}(y)$ при любых $n \in \mathbb{N}$ и $y \in \{0,1\}^{m(n)}$.

Пусть $n \in \mathbb{N}$. Положим для краткости $p_y = \Pr[A(1^n, y) \in g^{-1}(y)]$ и $k_y = |g^{-1}(y)|$ для произвольного $y \in \{0,1\}^{m(n)}$. Здесь $g^{-1}(y) \subseteq \{0,1\}^n$ при любом $y \in \{0,1\}^{m(n)}$ ввиду инъективности m. Тогда

$$\Pr[D(1^n, g(\widetilde{u}_n)) = 1] = \Pr[A(1^n, g(\widetilde{u}_n)) \in g^{-1}(g(\widetilde{u}_n))] = \sum_{y \in \{0,1\}^{m(n)}} p_y \frac{k_y}{2^n} = \sum_{y \in g(\{0,1\}^n)} p_y \frac{k_y}{2^n} \tag{17}$$

и

$$\Pr[D(1^n, \widetilde{u}_{m(n)}) = 1] = \Pr[A(1^n, \widetilde{u}_{m(n)}) \in g^{-1}(\widetilde{u}_{m(n)})] = \sum_{y \in \{0,1\}^{m(n)}} p_y \frac{1}{2^{m(n)}} = \sum_{y \in g(\{0,1\}^n)} p_y \frac{1}{2^{m(n)}},$$

так как $g^{-1}(y) = \varnothing$ и $p_y = 0$ при любом $y \in \{0,1\}^{m(n)} \setminus g(\{0,1\}^n)$. Следовательно,

$$\Pr[D(1^n, g(\widetilde{u}_n)) = 1] - \Pr[D(1^n, \widetilde{u}_{m(n)}) = 1] = \sum_{y \in g(\{0,1\}^n)} p_y \left(\frac{k_y}{2^n} - \frac{1}{2^{m(n)}} \right). \tag{18}$$

Пусть $y \in g(\{0,1\}^n)$. Тогда

$$\frac{1}{2^{m(n)}} \leqslant \frac{1}{2 \cdot 2^n} \leqslant \frac{1}{2} \frac{k_y}{2^n},$$

так как $m(n) \geqslant n + 1$ и $k_y \geqslant 1$. Поэтому

$$\frac{k_y}{2^n} - \frac{1}{2^{m(n)}} \geqslant \frac{1}{2} \frac{k_y}{2^n}. \tag{19}$$

Из (17)–(19) вытекает, что

$$\begin{aligned} \Pr[A(1^n, g(\widetilde{u}_n)) \in g^{-1}(g(\widetilde{u}_n))] &= \sum_{y \in g(\{0,1\}^n)} p_y \frac{k_y}{2^n} \leqslant 2 \sum_{y \in g(\{0,1\}^n)} p_y \left(\frac{k_y}{2^n} - \frac{1}{2^{m(n)}} \right) \\ &= 2(\Pr[D(1^n, g(\widetilde{u}_n)) = 1] - \Pr[D(1^n, \widetilde{u}_{m(n)}) = 1]) = \operatorname{negl}(n). \end{aligned}$$

Таким образом, g — односторонняя функция. □

Второй из приведенных выше фактов вытекает из теоремы Гольдрайха—Левина (см. теорему 6.2), замечания 6.3 и следующего предложения:

Предложение 7.10. *Если $f\colon \{0,1\}^* \to \{0,1\}^*$ — односторонняя перестановка, а b — трудный предикат для f, то функция $x \mapsto f(x)b(x)$ ($x \in \{0,1\}^*$) является псевдослучайным генератором.*

Доказательство. Очевидно, что функция $x \mapsto f(x)b(x)$ ($x \in \{0,1\}^*$) удовлетворяет условиям 1 и 2 определения 7.1. Ввиду следствия 7.4 для завершения доказательства предложения достаточно показать, что семейство случайных величин $(f(\widetilde{u}_n)b(\widetilde{u}_n) \,|\, n \in \mathbb{N})$ удовлетворяет условию непредсказуемости следующего бита.

Пусть $n \in \mathbb{N}$ и $\widetilde{i} \in \{1, \ldots, n+1\}$. Тогда

$$\begin{aligned}
&\Pr[A(1^n, (f(\widetilde{u}_n)b(\widetilde{u}_n))_{[1,\ldots,\widetilde{i}-1]}) = (f(\widetilde{u}_n)b(\widetilde{u}_n))_{[\widetilde{i}]}] \\
&\quad= \frac{1}{n+1}\sum_{i=1}^{n+1} \Pr[A(1^n, (f(\widetilde{u}_n)b(\widetilde{u}_n))_{[1,\ldots,i-1]}) = (f(\widetilde{u}_n)b(\widetilde{u}_n))_{[i]}] \\
&\quad= \frac{1}{n+1}\left(\sum_{i=1}^{n} \Pr[A(1^n, f(\widetilde{u}_n)_{[1,\ldots,i-1]}) = f(\widetilde{u}_n)_{[i]}] + \Pr[A(1^n, f(\widetilde{u}_n)) = b(\widetilde{u}_n)]\right) \\
&\quad\leqslant \frac{1}{n+1}\left(\frac{n}{2} + \frac{1}{2} + \operatorname{negl}(n)\right) \leqslant \frac{1}{2} + \operatorname{negl}(n).
\end{aligned}$$

Здесь мы воспользовались тем, что случайные величины $f(\widetilde{u}_n)_{[1,\ldots,i-1]}$ и $f(\widetilde{u}_n)_{[i]}$ для любого $i \in \{1, \ldots, n\}$ независимы (так как $f(\widetilde{u}_n)$ распределена равномерно на $\{0,1\}^n$) и тем, что предикат b трудно аппроксимируем по f (ввиду предложения 5.3). Таким образом, семейство случайных величин $(f(\widetilde{u}_n)b(\widetilde{u}_n) \,|\, n \in \mathbb{N})$ удовлетворяет условию непредсказуемости следующего бита, что и требовалось. □

Приведем явную конструкцию псевдослучайного генератора исходя из произвольной односторонней перестановки. А именно, из теоремы 6.2 (теоремы Гольдрайха—Левина), замечания 6.3 и предложения 7.10 вытекает

Следствие 7.11. *Пусть f — односторонняя перестановка. Тогда функция*

$$x \mapsto f(x_{[1,\ldots,\lfloor n/2\rfloor]})x_{[\lfloor n/2\rfloor+1,\ldots,n]}\left(\bigoplus_{i=1}^{\lfloor n/2\rfloor}(x_{[i]} \odot x_{[\lfloor n/2\rfloor+i]})\right) \quad (x \in \{0,1\}^n,\ n \in \mathbb{N})$$

является псевдослучайным генератором.

8. Псевдослучайные семейства функций

Пусть для каждого $n \in \mathbb{N}$ определено непустое множество $D_n \subseteq \{0,1\}^*$ и распределение вероятностей $\mathcal{D}_n$ на D_n, причем семейство $(\mathcal{D}_n \,|\, n \in \mathbb{N})$ полиномиально конструируемо, когда индексы заданы в унарной записи. Пусть также k и m — полиномиальные параметры на $\mathbb{N}$. Предположим, что каждой паре (n, d), где $n \in \mathbb{N}$ и $d \in D_n$, поставлена в соответствие функция $f_{n,d}\colon \{0,1\}^{k(n)} \to \{0,1\}^{m(n)}$.

Определение 8.1 (псевдослучайное семейство функций)**.** Семейство функций $F = (f_{n,d} \,|\, n \in \mathbb{N},\ d \in D_n)$ называется *псевдослучайным* (pseudorandom) относительно семейства распределений вероятностей $(\mathcal{D}_n \,|\, n \in \mathbb{N})$, если F полиномиально вычислимо и для любого полиномиального вероятностного алгоритма A

$$|\Pr[A^{f_{n,\widetilde{d}}}(1^n) = 1] - \Pr[A^{\widetilde{\rho}}(1^n) = 1]| = \operatorname{negl}(n),$$

где $\widetilde{d} \leftarrow \mathcal{D}_n$ и $\widetilde{\rho} \in_{\mathcal{U}} \operatorname{Fun}(\{0,1\}^{k(n)}, \{0,1\}^{m(n)})$.

Последнее условие этого определения означает, что никакой полиномиальный вероятностный алгоритм не может отличить $f_{n,d}$ при $d \leftarrow \mathcal{D}_n$ от функции, выбранной случайно и равновероятно из множества $\operatorname{Fun}(\{0,1\}^{k(n)}, \{0,1\}^{m(n)})$. При этом алгоритм получает доступ к функциям посредством оракулов, возвращающих значение функции на заданном алгоритмом аргументе.

***Замечание* 8.2.** Если $\widetilde{\rho} \in_{\mathcal{U}} \mathrm{Fun}(X, Y)$, где X и Y — непустые конечные множества, то $\widetilde{\rho}(x)$ при $x \in X$ — независимые случайные величины, распределенные равномерно на Y.

***Замечание* 8.3.** Пусть семейство функций $F = (f_{n,d}\colon \{0,1\}^{k(n)} \to \{0,1\}^{m(n)} \,|\, n \in \mathbb{N},\, d \in D_n)$ полиномиально вычислимо. Выберем полиномиальный вероятностный алгоритм G такой, что случайная величина $G(1^n)$ имеет распределение $\mathcal{D}_n$ для любого $n \in \mathbb{N}$. Пусть также l — полиномиальный параметр на $\mathbb{N}$ такой, что алгоритм G при вычислении на входе 1^n (где $n \in \mathbb{N}$) использует не более $l(n)$ случайных битов. Тогда вместо исходного семейства F можно рассматривать семейство $F' = (f_{n,G(1^n;r)} \,|\, n \in \mathbb{N},\, r \in \{0,1\}^{l(n)})$. Легко видеть, что F псевдослучайно относительно семейства распределений $(\mathcal{D}_n \,|\, n \in \mathbb{N})$ тогда и только тогда, когда F' псевдослучайно относительно семейства распределений $(\mathcal{U}(\{0,1\}^{l(n)}) \,|\, n \in \mathbb{N})$. В дальнейшем, если речь идет о псевдослучайности семейств функций вида $(f_{n,d} \,|\, n \in \mathbb{N},\, d \in \{0,1\}^{l(n)})$ относительно семейства распределений $(\mathcal{U}(\{0,1\}^{l(n)}) \,|\, n \in \mathbb{N})$, мы будем говорить просто о псевдослучайности.

Следующая теорема принадлежит Гольдрайху, Гольдвассер и Микали [GGM84, GGM86] (см. также [Gol04, разд. 3.6], [Lub96, лекция 12]).

Теорема 8.4. *Предположим, что существуют псевдослучайные генераторы. Тогда для любых полиномиальных параметров k и m, определенных на множестве $\mathbb{N}$, существует псевдослучайное семейство функций $(f_{n,d} \,|\, n \in \mathbb{N},\, d \in \{0,1\}^n)$, в котором $f_{n,d} \in \mathrm{Fun}(\{0,1\}^{k(n)}, \{0,1\}^{m(n)})$ при всех $n \in \mathbb{N}$ и $d \in \{0,1\}^n$.*

Для доказательства этой теоремы нам потребуется следующая

Лемма 8.5. *Пусть g — псевдослучайный генератор, отображающий $\{0,1\}^n$ в $\{0,1\}^{2n}$ для всех $n \in \mathbb{N}$. Положим $g_0(x) = g(x)_{[1,\ldots,n]}$ и $g_1(x) = g(x)_{[n+1,\ldots,2n]}$ для произвольных $n \in \mathbb{N}$ и $x \in \{0,1\}^n$. Для произвольного полиномиального параметра $k\colon \mathbb{N} \to \mathbb{N}$ определим функцию $f_{n,d}\colon \{0,1\}^{k(n)} \to \{0,1\}^n$ равенством*

$$f_{n,d}(x) = g_{x_{[k(n)]}}(\ldots g_{x_{[2]}}(g_{x_{[1]}}(d))\ldots) \quad (x \in \{0,1\}^{k(n)}).$$

Тогда семейство функций $F = (f_{n,d} \,|\, n \in \mathbb{N},\, d \in \{0,1\}^n)$ псевдослучайно.

Доказательство. Очевидно, что семейство F полиномиально вычислимо. Поэтому если F не псевдослучайно, то существуют полиномиальный вероятностный алгоритм A, полином p и бесконечное множество $N \subseteq \mathbb{N}$ такое, что $|\Pr[A^{f_{n,\widetilde{d}}}(1^n) = 1] - \Pr[A^{\widetilde{\rho}}(1^n) = 1]| > 1/p(n)$ при всех $n \in N$, где $\widetilde{d} \in_{\mathcal{U}} \{0,1\}^n$ и $\widetilde{\rho} \in_{\mathcal{U}} \mathrm{Fun}(\{0,1\}^{k(n)}, \{0,1\}^n)$. Рассуждая аналогично доказательству леммы 0.1, можно считать, что

$$\Pr[A^{f_{n,\widetilde{d}}}(1^n) = 1] - \Pr[A^{\widetilde{\rho}}(1^n) = 1] > \frac{1}{p(n)} \tag{20}$$

для любого $n \in N$. Выберем полином q такой, что для любой функции $\varphi \in \mathrm{Fun}(\{0,1\}^{k(n)}, \{0,1\}^n)$ алгоритм A при вычислении на входе 1^n ($n \in N$) с использованием оракула φ делает не более $q(n)$ запросов к этому оракулу. Для всех $n \in N$ положим также $l(n) = k(n)q(n)$. Из неравенства (20) вытекает, что $k(n) \geqslant 1$ и, следовательно, $l(n) \geqslant 1$ при любом $n \in N$.

Пусть B — полиномиальный вероятностный алгоритм, который на произвольном входе $(1^n, y_0y_1)$, где $n \in N$ и $y_0, y_1 \in \{0,1\}^n$, выполняет алгоритм A на входе 1^n и возвращает результат (если он есть). При этом в качестве ответа на произвольный запрос $x \in \{0,1\}^{k(n)}$ алгоритма A к оракулу алгоритм B дает алгоритму A значение $h(x)$, где h — функция, определенная на некотором множестве $V \subseteq \{0,1\}^{\leqslant k(n)}$ и принимающая значения в $\{0,1\}^n$. Множество V и функция h изменяются (в сторону расширения) в процессе вычисления.

Опишем теперь подробнее работу алгоритма B на входе $(1^n, y_0y_1)$, где $n \in N$ и $y_0, y_1 \in \{0,1\}^n$. Вначале алгоритм B выбирает $t \in_{\mathcal{U}} \{1,\ldots,l(n)\}$, $s \in_{\mathcal{U}} \{0,1\}^n$ и $r_{i,b} \in_{\mathcal{U}} \{0,1\}^n$ для всех $i \in \{1,\ldots,t-1\}$ и $b \in \{0,1\}$ и полагает $V = \{\lambda\}$ и $h(\lambda) = s$, где λ — пустая строка. Затем B запускает алгоритм A на входе 1^n. Получив от алгоритма A запрос $x \in \{0,1\}^{k(n)}$ к оракулу, алгоритм B выполняет следующие действия:

1. Выбрать в строке x префикс v наибольшей длины, принадлежащий множеству V (такой существует, так как $\lambda \in V$). Пусть $x = vw$, где $w \in \{0,1\}^*$.

2. Для i от 1 до $|w|$:

   - положить $z_i = vw_{[1,\ldots,i-1]}$;

- добавить в V элементы множества $\{z_i0, z_i1\}$ и определить функцию h на z_i0 и z_i1 следующим образом: если множество $\{z_i0, z_i1\}$ было объединено с V j-м по счету, считая с 1 от начала работы алгоритма A (т. е. учитывая и предыдущие запросы к оракулу), то

$$h(z_ib) = \begin{cases} r_{j,b}, & \text{если } j < t, \\ y_b, & \text{если } j = t, \\ g_b(h(z_i)), & \text{если } j > t \end{cases}$$

для каждого $b \in \{0,1\}$. Очевидно, что к началу данной операции $z_i \in V$, но $z_ib \notin V$ для любого $b \in \{0,1\}$. Последнее верно ввиду того, что строки (за исключением λ) помещаются в множество V парами вида $(z0, z1)$.

3. Дать алгоритму A значение $h(x)$ в качестве ответа на запрос (к данному моменту $x \in V$).

После завершения работы A алгоритм B возвращает выходное значение алгоритма A (если оно есть).

Фиксируем $n \in N$. Для каждого $j \in \{1, \ldots, l(n)\}$ положим для краткости

$$\varepsilon_j(n) = \Pr[B(1^n, g(\widetilde{u}_n)) = 1 \,|\, t = j] \quad \text{и} \quad \delta_j(n) = \Pr[B(1^n, \widetilde{u}_{2n}) = 1 \,|\, t = j],$$

где t рассматривается как случайная величина, распределенная равномерно на $\{1, \ldots, l(n)\}$. Тогда легко видеть, что

$$\Pr[B(1^n, g(\widetilde{u}_n)) = 1] = \frac{1}{l(n)} \sum_{j=1}^{l(n)} \varepsilon_j(n) \quad \text{и} \quad \Pr[B(1^n, \widetilde{u}_{2n}) = 1] = \frac{1}{l(n)} \sum_{j=1}^{l(n)} \delta_j(n),$$

где $\varepsilon_j(n) = \delta_{j-1}(n)$ для всех $j \in \{2, \ldots, l(n)\}$. Поэтому

$$\Pr[B(1^n, g(\widetilde{u}_n)) = 1] - \Pr[B(1^n, \widetilde{u}_{2n}) = 1] = \frac{\varepsilon_1(n) - \delta_{l(n)}(n)}{l(n)}. \tag{21}$$

Если $y_0y_1 = g(d)$ (где $d \in \{0,1\}^n$) и $t = 1$ в алгоритме B, то $h(x) = f_{n,d}(x)$ при любом $x \in \{0,1\}^{k(n)}$. Следовательно,

$$\varepsilon_1(n) = \Pr[A^{f_{n,\tilde{d}}}(1^n) = 1]. \tag{22}$$

Кроме того, если $y_0, y_1 \in_{\mathcal{U}} \{0,1\}^n$ и $t = l(n)$ в алгоритме B, то значения $h(x)$ при $x \in V$, рассматриваемые как случайные величины, независимы и распределены равномерно на $\{0,1\}^n$, так как число выполненных операций объединения V с множествами вида $\{z0, z1\}$ в процессе работы алгоритма B на входе $(1^n, y_0y_1)$ не превосходит $k(n)q(n) = l(n)$. Поэтому ввиду замечания 8.2

$$\delta_{l(n)}(n) = \Pr[A^{\widetilde{\rho}}(1^n) = 1]. \tag{23}$$

Из равенств (21)–(23) и неравенства (20) вытекает неравенство

$$\Pr[B(1^n, g(\widetilde{u}_n)) = 1] - \Pr[B(1^n, \widetilde{u}_{2n}) = 1] > \frac{1}{p(n)l(n)} \geqslant \frac{1}{\mathrm{poly}(n)},$$

для всех $n \in N$, которое противоречит тому, что g — псевдослучайный генератор. □

Таким образом, для доказательства леммы 8.5 мы применили некоторую разновидность гибридного метода.

Доказательство теоремы 8.4. Пусть k и m — полиномиальные параметры на множестве $\mathbb{N}$. Ввиду следствия 7.7 существует псевдослучайный генератор, отображающий $\{0,1\}^n$ в $\{0,1\}^{2n}$ при всех $n \in \mathbb{N}$. Тогда из леммы 8.5 следует существование псевдослучайного семейства функций $(f_{n,d} \,|\, n \in \mathbb{N}, d \in \{0,1\}^n)$, в котором $f_{n,d} \in \mathrm{Fun}(\{0,1\}^{k(n)}, \{0,1\}^n)$ для любых $n \in \mathbb{N}$ и $d \in \{0,1\}^n$. Выберем какую-либо полиномиально вычислимую функцию h, отображающую $\{0,1\}^n$ в $\{0,1\}^{m(n)}$ для всех $n \in \mathbb{N}$ и такую, что семейства случайных величин $(h(\widetilde{u}_n) \,|\, n \in \mathbb{N})$ и $(\widetilde{u}_{m(n)} \,|\, n \in \mathbb{N})$ вычислительно неотличимы, когда индексы заданы в унарной записи (в отличие от определения псевдослучайного генератора, здесь не требуется выполнения неравенства $m(n) > n$). В качестве такой функции можно взять $x \mapsto g(x)_{[1,\ldots,m(n)]}$ ($n \in \mathbb{N}$, $x \in \{0,1\}^n$), где g — псевдослучайный генератор, отображающий $\{0,1\}^n$ в $\{0,1\}^{n+1+m(n)}$ при всех $n \in \mathbb{N}$; такой псевдослучайный генератор существует ввиду

следствия 7.7. Покажем, что семейство функций $(h(f_{n,d}) \mid n \in \mathbb{N},\ d \in \{0,1\}^n)$ является искомым. Очевидно, что этой семейство полиномиально вычислимо и что $h(f_{n,d}) \in \mathrm{Fun}(\{0,1\}^{k(n)}, \{0,1\}^{m(n)})$ для любых $n \in \mathbb{N}$ и $d \in \{0,1\}^n$. Псевдослучайность данного семейства следует из того, что для любого полиномиального вероятностного алгоритма A

$$|\Pr[A^{h(f_{n,\widetilde{d}})}(1^n) = 1] - \Pr[A^{h(\widetilde{\rho})}(1^n) = 1]| = \mathrm{negl}(n) \tag{24}$$

и

$$|\Pr[A^{h(\widetilde{\rho})}(1^n) = 1] - \Pr[A^{\widetilde{\sigma}}(1^n) = 1]| = \mathrm{negl}(n), \tag{25}$$

где $\widetilde{d} \in_{\mathcal{U}} \{0,1\}^n$, $\widetilde{\rho} \in_{\mathcal{U}} \mathrm{Fun}(\{0,1\}^{k(n)}, \{0,1\}^n)$ и $\widetilde{\sigma} \in_{\mathcal{U}} \mathrm{Fun}(\{0,1\}^{k(n)}, \{0,1\}^{m(n)})$.

Пусть A — произвольный полиномиальный вероятностный алгоритм. Докажем сначала (24). Выберем полиномиальный вероятностный алгоритм B такой, что $B^{\varphi}(1^n) = A^{h(\varphi)}(1^n)$ для всех $n \in \mathbb{N}$ и всех функций $\varphi \in \mathrm{Fun}(\{0,1\}^{k(n)}, \{0,1\}^n)$. А именно, алгоритм B при вычислении на входе 1^n запускает алгоритм A на этом входе. Получив от алгоритма A запрос $x \in \{0,1\}^{k(n)}$ к оракулу, алгоритм B передает этот запрос своему оракулу. После получения ответа (скажем, $y = \varphi(x)$) от своего оракула алгоритм B вычисляет значение $h(y)$ и дает это значение алгоритму A в качестве ответа на запрос. После завершения работы A алгоритм B возвращает выходное значение алгоритма A (если оно есть). Тогда

$$|\Pr[A^{h(f_{n,\widetilde{d}})}(1^n) = 1] - \Pr[A^{h(\widetilde{\rho})}(1^n) = 1]| = |\Pr[B^{f_{n,\widetilde{d}}}(1^n) = 1] - \Pr[B^{\widetilde{\rho}}(1^n) = 1]| = \mathrm{negl}(n),$$

так как семейство функций $(f_{n,d} \mid n \in \mathbb{N},\ d \in \{0,1\}^n)$ псевдослучайно.

Докажем теперь (25). Выберем полином q такой, что для любой функции $\psi \in \mathrm{Fun}(\{0,1\}^{k(n)}, \{0,1\}^{m(n)})$ алгоритм A при вычислении на входе 1^n ($n \in N$) с использованием оракула ψ делает не более $q(n)$ запросов к этому оракулу. Пусть C — полиномиальный вероятностный алгоритм, который на произвольном входе $(1^n, y_1, \ldots, y_{q(n)})$, где $n \in N$ и $y_1, \ldots, y_{q(n)} \in \{0,1\}^{m(n)}$, выполняет алгоритм A на входе 1^n и возвращает результат (если он есть). Не ограничивая общности, считаем, что алгоритм не повторяет свои запросы к оракулу (он может запоминать ответы на свои запросы и вместо повторного запроса использовать полученный ранее ответ на данный запрос). Тогда C дает алгоритму A y_i в качестве ответа на i-й по счету запрос к оракулу алгоритма A (независимо от вида запроса). В этом случае из замечания 8.2 вытекает, что

$$\Pr[A^{h(\widetilde{\rho})}(1^n) = 1] = \Pr[C(1^n, h(\widetilde{v}_1), \ldots, h(\widetilde{v}_{q(n)})) = 1]$$

и

$$\Pr[A^{\widetilde{\sigma}}(1^n) = 1] = \Pr[C(1^n, \widetilde{w}_1, \ldots, \widetilde{w}_{q(n)}) = 1]$$

для любого $n \in \mathbb{N}$, где $\widetilde{v}_1, \ldots, \widetilde{v}_{q(n)} \in_{\mathcal{U}} \{0,1\}^n$ и $\widetilde{w}_1, \ldots, \widetilde{w}_{q(n)} \in_{\mathcal{U}} \{0,1\}^{m(n)}$. Из этих двух равенств и леммы 0.3 следует (25). □

Предложение 8.6 (см. также [Gol04, подразд. 3.8.4, упр. 28], [Lub96, упр. 49])**.** *Пусть существует псевдослучайное семейство функций $(f_{n,d} \mid n \in \mathbb{N},\ d \in \{0,1\}^n)$, в котором $f_{n,d} \in \mathrm{Fun}(\{0,1\}^{k(n)}, \{0,1\}^{m(n)})$ при всех $n \in \mathbb{N}$ и $d \in \{0,1\}^n$. (Здесь k и m — некоторые полиномиальные параметры на $\mathbb{N}$.) Предположим, что $n < 2^{k(n)}m(n)$ при всех достаточно больших $n \in \mathbb{N}$. Тогда существует псевдослучайный генератор.*

Доказательство. Выберем $t \in \mathbb{N}$ такое, что $n < 2^{k(n)}m(n)$ при всех $n \geqslant t$. Мы определим псевдослучайный генератор g лишь на $\{0,1\}^{\geqslant t}$; на $\{0,1\}^* \setminus \{0,1\}^{\geqslant t}$ он может быть определен произвольно при условии, что для всех $n < t$ верно включение $g(\{0,1\}^n) \subseteq \{0,1\}^{p(n)}$, где $p(n) > n$.

Пусть $n \geqslant t$. Тогда из неравенства $n < 2^{k(n)}m(n)$ следует, что $m(n) \geqslant 1$. Положим $s(n) = \lfloor n/m(n) \rfloor + 1$. Так как $\lfloor n/m(n) \rfloor \leqslant n/m(n) < 2^{k(n)}$, имеет место неравенство $s(n) \leqslant 2^{k(n)}$. Поэтому мы можем выбрать различные строки $v_{n,1}, \ldots, v_{n,s(n)} \in \{0,1\}^{k(n)}$ так, чтобы функция $1^n \mapsto (v_{n,1}, \ldots, v_{n,s(n)})$ была полиномиально вычислимой. Например, для каждого $i \in \{1, \ldots, s(n)\}$ в качестве $v_{n,i}$ можно взять строку из $\{0,1\}^{k(n)}$, являющуюся двоичной записью числа $i-1$. Для произвольного $x \in \{0,1\}^n$ положим

$$g(x) = f_{n,x}(v_{n,1}) \ldots f_{n,x}(v_{n,s(n)}).$$

Покажем, что g является псевдослучайным генератором. Полиномиальная вычислимость g очевидна. Кроме того, g отображает $\{0,1\}^n$ в $\{0,1\}^{m(n)s(n)}$, где $m(n)s(n) > n$, так как $s > n/m(n)$.

Пусть теперь D — произвольный полиномиальный вероятностный алгоритм. Выберем полиномиальный вероятностный алгоритм A такой, что $A^{\varphi}(1^n) = D(1^n, \varphi(v_{n,1}) \dots \varphi(v_{n,s(n)}))$ для любого $n \geqslant t$ и любой функции $\varphi \in \mathrm{Fun}(\{0,1\}^{k(n)}, \{0,1\}^{m(n)})$. Из замечания 8.2 вытекает, что если $\widetilde{\rho} \in_{\mathcal{U}} \mathrm{Fun}(\{0,1\}^{k(n)}, \{0,1\}^{m(n)})$, то случайная величина $\widetilde{\rho}(v_{n,1}) \dots \widetilde{\rho}(v_{n,s(n)})$ распределена равномерно на $\{0,1\}^{m(n)s(n)}$. Следовательно,

$$|\Pr[D(1^n, g(\widetilde{u}_n)) = 1] - \Pr[D(1^n, \widetilde{u}_{m(n)s(n)}) = 1]| = |\Pr[A^{f_{n,\widetilde{u}_n}}(1^n) = 1] - \Pr[A^{\widetilde{\rho}}(1^n) = 1]| = \mathrm{negl}(n)$$

ввиду псевдослучайности семейства функций $(f_{n,d} \mid n \in \mathbb{N},\, d \in \{0,1\}^n)$. □

В связи с теоремой 8.4 и предложением 8.6 напомним, что псевдослучайные генераторы существуют тогда и только тогда, когда существуют односторонние функции (см. теорему 7.8).

9. Псевдослучайные и сильно псевдослучайные семейства перестановок

Пусть, как и в предыдущем разделе, для каждого $n \in \mathbb{N}$ определено непустое множество $D_n \subseteq \{0,1\}^*$ и распределение вероятностей $\mathcal{D}_n$ на D_n, причем семейство $(\mathcal{D}_n \mid n \in \mathbb{N})$ полиномиально конструируемо, когда индексы заданы в унарной записи. Пусть также k — полиномиальный параметр на $\mathbb{N}$. Предположим, что каждой паре (n, d), где $n \in \mathbb{N}$ и $d \in D_n$, поставлена в соответствие перестановка $f_{n,d}$ множества $\{0,1\}^{k(n)}$.

Определение 9.1 (псевдослучайное семейство перестановок)**.** Семейство перестановок $F = (f_{n,d} \mid n \in \mathbb{N},\, d \in D_n)$ называется *псевдослучайным* (pseudorandom) относительно семейства распределений вероятностей $(\mathcal{D}_n \mid n \in \mathbb{N})$, если F полиномиально вычислимо и для любого полиномиального вероятностного алгоритма A

$$|\Pr[A^{f_{n,\widetilde{d}}}(1^n) = 1] - \Pr[A^{\widetilde{\pi}}(1^n) = 1]| = \mathrm{negl}(n),$$

где $\widetilde{d} \leftarrow \mathcal{D}_n$ и $\widetilde{\pi} \in_{\mathcal{U}} \mathrm{Per}(\{0,1\}^{k(n)})$.

Последнее условие этого определения означает, что никакой полиномиальный вероятностный алгоритм не может отличить $f_{n,d}$ при $d \leftarrow \mathcal{D}_n$ от перестановки, выбранной случайно и равновероятно из множества $\mathrm{Per}(\{0,1\}^{k(n)})$. При этом алгоритм получает доступ к перестановкам посредством оракулов, возвращающих значение перестановки на заданном алгоритмом аргументе.

Определение 9.2 (сильно псевдослучайное семейство перестановок)**.** Семейство перестановок $F = (f_{n,d} \mid n \in \mathbb{N},\, d \in D_n)$ называется *сильно псевдослучайным* (strongly pseudorandom, super pseudorandom) относительно семейства распределений вероятностей $(\mathcal{D}_n \mid n \in \mathbb{N})$, если F полиномиально вычислимо и для любого полиномиального вероятностного алгоритма A

$$|\Pr[A^{f_{n,\widetilde{d}}, f^{-1}_{n,\widetilde{d}}}(1^n) = 1] - \Pr[A^{\widetilde{\pi}, \widetilde{\pi}^{-1}}(1^n) = 1]| = \mathrm{negl}(n),$$

где $\widetilde{d} \leftarrow \mathcal{D}_n$ и $\widetilde{\pi} \in_{\mathcal{U}} \mathrm{Per}(\{0,1\}^{k(n)})$.

Отличие этого определения от определения псевдослучайного семейства перестановок (см. определение 9.1) состоит в том, что алгоритм A имеет также доступ к оракулу, возвращающему значение обратной перестановки на заданном алгоритмом аргументе.

***Замечание* 9.3.** Пусть $\widetilde{\pi} \in_{\mathcal{U}} \mathrm{Per}(X)$, где X — непустое конечное множество. Тогда совместное распределение случайных величин $\widetilde{\pi}(x)$ при $x \in X$ описывается следующим образом: для любых различных $x_1, \dots, x_m \in X$ и любых $y_1, \dots, y_m \in X$

$$\Pr[\widetilde{\pi}(x_1) = y_1, \dots, \widetilde{\pi}(x_m) = y_m] = \begin{cases} (|X| - m)!/|X|!, & \text{если } y_1, \dots, y_m \text{ различны,} \\ 0 & \text{в противном случае.} \end{cases}$$

***Замечание* 9.4** (аналог замечания 8.3)**.** Пусть семейство перестановок $F = (f_{n,d} \colon \{0,1\}^{k(n)} \to \{0,1\}^{k(n)} \mid n \in \mathbb{N},\, d \in D_n)$ полиномиально вычислимо. Выберем полиномиальный вероятностный алгоритм G такой, что случайная величина $G(1^n)$ имеет распределение $\mathcal{D}_n$ для любого $n \in \mathbb{N}$. Пусть также l — полиномиальный параметр на $\mathbb{N}$ такой, что алгоритм G при вычислении на входе 1^n (где

$n \in \mathbb{N}$) использует не более $l(n)$ случайных битов. Тогда вместо исходного семейства F можно рассматривать семейство $F' = (f_{n,G(1^n;r)} \,|\, n \in \mathbb{N},\, r \in \{0,1\}^{l(n)})$. Легко видеть, что F псевдослучайно (сильно псевдослучайно) относительно семейства распределений $(\mathcal{D}_n \,|\, n \in \mathbb{N})$ тогда и только тогда, когда F' псевдослучайно (сильно псевдослучайно) относительно семейства распределений $(\mathcal{U}(\{0,1\}^{l(n)}) \,|\, n \in \mathbb{N})$. В дальнейшем, если речь идет о псевдослучайности (сильной псевдослучайности) семейств перестановок вида $(f_{n,d} \,|\, n \in \mathbb{N},\, d \in \{0,1\}^{l(n)})$ относительно семейства распределений $(\mathcal{U}(\{0,1\}^{l(n)}) \,|\, n \in \mathbb{N})$, мы будем говорить просто о псевдослучайности (сильной псевдослучайности).

Лемма 9.5 (см. также [Gol04, предложение 3.7.3], [Lub96, упр. 51])**.** *Пусть A — вероятностный алгоритм. Предположим, что для любой функции $\varphi\colon \{0,1\}^k \to \{0,1\}^k$, где k — некоторое фиксированное число из $\mathbb{N}$, алгоритм A при вычислении без входа с использованием оракула φ делает не более $t \in \mathbb{N}$ запросов к этому оракулу и всегда завершает работу. Тогда*

$$|\Pr[A^{\widetilde{\rho}} = 1] - \Pr[A^{\widetilde{\pi}} = 1]| \leqslant \frac{t(t-1)}{2 \cdot 2^k}, \tag{26}$$

где $\widetilde{\rho} \in_{\mathcal{U}} \mathrm{Fun}(\{0,1\}^k, \{0,1\}^k)$ и $\widetilde{\pi} \in_{\mathcal{U}} \mathrm{Per}(\{0,1\}^k)$.

Доказательство. Не ограничивая общности, наложим на работу алгоритма A при вычислении без входа с использованием произвольного оракула $\varphi \in \mathrm{Fun}(\{0,1\}^k, \{0,1\}^k)$ некоторые дополнительные требования. Во-первых, мы считаем, что A при этом вычислении не повторяет свои запросы к оракулу. Действительно, A может запоминать ответы на свои запросы и вместо повторного запроса использовать полученный ранее ответ на данный запрос; при этом число запросов не увеличивается. Во-вторых, предполагается, что A при указанном вычислении всегда делает m запросов к оракулу, где m — наибольшее число таких запросов по всем $\varphi \in \mathrm{Fun}(\{0,1\}^k, \{0,1\}^k)$ и всем строкам случайных битов алгоритма A (очевидно, что $m \leqslant t$). А именно, если A сделал перед завершением работы $m' < m$ запросов к оракулу (m может быть встроено в описание алгоритма A), то он делает еще $m - m'$ таких запросов, выбрав их произвольным образом с соблюдением требования различности запросов (это возможно, так как $m \leqslant 2^k$) и только после этого завершает работу. В-третьих, считаем, что A при данном вычислении не использует случайные биты. Общий случай сводится к этому частному следующим образом. Пусть A использует при рассматриваемом вычислении не более l случайных битов. Если требуемое неравенство (26) верно для алгоритмов, не использующих случайные биты при вышеуказанном вычислении, то

$$\begin{aligned}|\Pr[A^{\widetilde{\rho}} = 1] - \Pr[A^{\widetilde{\pi}} = 1]| &= |\mathsf{E}_r[\Pr[A^{\widetilde{\rho}}(;r) = 1] - \Pr[A^{\widetilde{\pi}}(;r) = 1]]| \\ &\leqslant \mathsf{E}_r[|\Pr[A^{\widetilde{\rho}}(;r) = 1] - \Pr[A^{\widetilde{\pi}}(;r) = 1]|] \leqslant \frac{t(t-1)}{2 \cdot 2^k},\end{aligned}$$

где математическое ожидание берется по r, распределенной равномерно на $\{0,1\}^l$.

Пусть $\widetilde{y}_i$ — ответ на i-й запрос алгоритма A к оракулу при вычислении без входа с использованием оракула $\widetilde{\rho}$ ($i \in \{1, \ldots, m\}$). Очевидно, что каждый запрос к оракулу однозначно определяется ответами оракула на предыдущие запросы. Поэтому для любых $y_1, \ldots, y_m \in \{0,1\}^k$

$$\Pr[\widetilde{y}_1 = y_1, \ldots, \widetilde{y}_m = y_m] = \Pr[\widetilde{\rho}(x_1) = y_1, \ldots, \widetilde{\rho}(x_m) = y_m] = \frac{1}{2^{km}} \tag{27}$$

ввиду замечания 8.2, где x_i — i-й запрос алгоритма A к оракулу при условии, что на предыдущие запросы были даны ответы $y_1, \ldots, y_{i-1}$ (напомним, что, согласно нашему предположению, $x_1, \ldots, x_m$ различны). Следовательно, случайные величины $\widetilde{y}_1, \ldots, \widetilde{y}_m$ независимы и распределены равномерно на $\{0,1\}^k$.

Обозначим через E событие, заключающееся в том, что $\widetilde{y}_1, \ldots, \widetilde{y}_m$ различны. Тогда

$$\Pr[E] = \frac{|\{(y_1, \ldots, y_m) \,|\, y_1, \ldots, y_m \text{ — различные элементы } \{0,1\}^k\}|}{2^{km}} = \frac{2^k!}{(2^k - m)!2^{km}} \tag{28}$$

и

$$\Pr[\overline{E}] \leqslant \sum_{1 \leqslant i < j \leqslant m} \Pr[\widetilde{y}_i = \widetilde{y}_j] = \frac{m(m-1)}{2 \cdot 2^k} \leqslant \frac{t(t-1)}{2 \cdot 2^k} \tag{29}$$

где $\overline{E}$ — дополнительное событие к E.

Пусть $\widetilde{y}'_i$ — ответ на i-й запрос алгоритма A к оракулу при вычислении без входа с использованием оракула $\widetilde{\pi}$ ($i \in \{1, \ldots, m\}$). Пусть также $y_1, \ldots, y_m \in \{0,1\}^k$. Тогда если $y_1, \ldots, y_m$ различны, то

$$\Pr[\widetilde{y}_1 = y_1, \ldots, \widetilde{y}_m = y_m \mid E] = \frac{\Pr[\widetilde{y}_1 = y_1, \ldots, \widetilde{y}_m = y_m]}{\Pr[E]} = \frac{(2^k - m)!}{2^k!}$$
$$= \Pr[\widetilde{\pi}(x_1) = y_1, \ldots, \widetilde{\pi}(x_m) = y_m] = \Pr[\widetilde{y}'_1 = y_1, \ldots, \widetilde{y}'_m = y_m]$$

ввиду равенств (27) и (28) и замечания 9.3, где x_i определяются по $y_1, \ldots, y_m$ так же, как и выше. Если же среди $y_1, \ldots, y_m$ есть совпадающие, то

$$\Pr[\widetilde{y}_1 = y_1, \ldots, \widetilde{y}_m = y_m \mid E] = 0 = \Pr[\widetilde{y}'_1 = y_1, \ldots, \widetilde{y}'_m = y_m].$$

Поэтому случайная величина $(\widetilde{y}_1, \ldots, \widetilde{y}_m)$ при условии E распределена так же, как случайная величина $(\widetilde{y}'_1, \ldots, \widetilde{y}'_m)$ (без какого-либо условия). Следовательно, $\Pr[A^{\widetilde{\rho}} = 1 \mid E] = \Pr[A^{\widetilde{\pi}} = 1]$. Из этого равенства и неравенств $0 \leqslant \Pr[A^{\widetilde{\rho}} = 1, \overline{E}] \leqslant \Pr[\overline{E}]$ и (29) вытекает, что

$$|\Pr[A^{\widetilde{\rho}} = 1] - \Pr[A^{\widetilde{\pi}} = 1]| = |\Pr[A^{\widetilde{\rho}} = 1 \mid E] \Pr[E] + \Pr[A^{\widetilde{\rho}} = 1, \overline{E}] - \Pr[A^{\widetilde{\pi}} = 1]|$$
$$= |\Pr[A^{\widetilde{\rho}} = 1, \overline{E}] - \Pr[A^{\widetilde{\pi}} = 1] \Pr[\overline{E}]| \leqslant \Pr[\overline{E}] \leqslant \frac{t(t-1)}{2 \cdot 2^k}. \quad \square$$

Отметим, что в лемме 9.5 мы имеем дело не с бесконечными семействами функций, а с отдельными функциями.

Предложение 9.6. *Пусть k — полиномиальный параметр на $\mathbb{N}$ такой, что $k(n) = \omega(\log n)$ при $n \in \mathbb{N} \setminus \{0\}$ (т. е. $k(n)/\log n \to +\infty$ при $n \to +\infty$). Тогда произвольное семейство перестановок $(f_{n,d} \colon \{0,1\}^{k(n)} \to \{0,1\}^{k(n)} \mid n \in \mathbb{N},\ d \in D_n)$ является псевдослучайным семейством перестановок, если и только если оно является псевдослучайным семейством функций.*

Доказательство. Непосредственно проверяется, что $p(n)/2^{k(n)} = \operatorname{negl}(n)$ для любого полинома p. Поэтому из леммы 9.5 следует, что если $\widetilde{\rho} \in_{\mathcal{U}} \operatorname{Fun}(\{0,1\}^{k(n)}, \{0,1\}^{k(n)})$ и $\widetilde{\pi} \in_{\mathcal{U}} \operatorname{Per}(\{0,1\}^{k(n)})$, то $|\Pr[A^{\widetilde{\rho}}(1^n) = 1] - \Pr[A^{\widetilde{\pi}}(1^n) = 1]| = \operatorname{negl}(n)$ для любого полиномиального вероятностного алгоритма A. Отсюда непосредственно вытекает требуемое утверждение. $\square$

Часто на псевдослучайные и сильно псевдослучайные семейства перестановок накладывают дополнительное условие полиномиальной инвертируемости.

Определение 9.7 (полиномиально инвертируемое семейство перестановок)**.** Семейство перестановок $(f_{n,d} \mid n \in \mathbb{N},\ d \in D_n)$ называется *полиномиально инвертируемым* (polynomial-time invertible), если полиномиально вычислима функция $(1^n, d, y) \mapsto f_{n,d}^{-1}(y)$, где $n \in \mathbb{N}$, $d \in D_n$ и $y \in \{0,1\}^{k(n)}$.

Для формулировки следующей теоремы нам потребуется понятие *преобразования Файстеля* (Feistel transformation), которое преобразует произвольную функцию $f \colon \{0,1\}^n \to \{0,1\}^n$ (где $n \in \mathbb{N}$) в перестановку $\Phi_f \colon \{0,1\}^{2n} \to \{0,1\}^{2n}$, определенную следующей формулой:

$$\Phi_f(xy) = y(x \oplus f(y)) \quad (x, y \in \{0,1\}^n).$$

Обратное преобразование к Φ_f задается следующей формулой:

$$\Phi_f^{-1}(xy) = (y \oplus f(x))x \quad (x, y \in \{0,1\}^n).$$

Теорема 9.8 (см. [LR86, LR88], [Gol04, теорема 3.7.7], [Lub96, лекция 14, упр. 56])**.** *Пусть $(f_{n,d} \mid n \in \mathbb{N},\ d \in \{0,1\}^n)$ — псевдослучайное семейство функций, в котором $f_{n,d} \in \operatorname{Fun}(\{0,1\}^n, \{0,1\}^n)$ для всех $n \in \mathbb{N}$ и $d \in \{0,1\}^n$.*

- *Для произвольных $n \in \mathbb{N}$ и $d_1, d_2, d_3 \in \{0,1\}^n$ определим перестановку $g_{n,(d_1,d_2,d_3)} \in \operatorname{Per}(\{0,1\}^{2n})$ равенством*

$$g_{n,(d_1,d_2,d_3)}(x) = \Phi_{f_{n,d_3}}(\Phi_{f_{n,d_2}}(\Phi_{f_{n,d_1}}(x))) \quad (x \in \{0,1\}^{2n}).$$

Тогда

$$(g_{n,(d_1,d_2,d_3)} \mid n \in \mathbb{N},\ d_1, d_2, d_3 \in \{0,1\}^n)$$

является псевдослучайным полиномиально инвертируемым семейством перестановок.

- *Для произвольных* $n \in \mathbb{N}$ *и* $d_1, d_2, d_3, d_4 \in \{0,1\}^n$ *определим перестановку* $h_{n,(d_1,d_2,d_3,d_4)} \in \mathrm{Per}(\{0,1\}^{2n})$ *равенством*

$$h_{n,(d_1,d_2,d_3,d_4)}(x) = \Phi_{f_{n,d_4}}(\Phi_{f_{n,d_3}}(\Phi_{f_{n,d_2}}(\Phi_{f_{n,d_1}}(x)))) \quad (x \in \{0,1\}^{2n}).$$

Тогда

$$(h_{n,(d_1,d_2,d_3,d_4)} \mid n \in \mathbb{N},\ d_1, d_2, d_3, d_4 \in \{0,1\}^n)$$

является сильно псевдослучайным полиномиально инвертируемым семейством перестановок.

В этой теореме Φ_f обозначает образ функции f при преобразовании Файстеля.

***Замечание* 9.9.** В связи с теоремой 9.8 отметим, что псевдослучайные семейства функций, требуемые в ней, существуют тогда и только тогда, когда существуют односторонние функции. Это следует из теорем 7.8 и 8.4 и предложения 8.6.

Доказательство теоремы 9.8 остается за рамками настоящего курса ввиду сложности этого доказательства.

10. Семейства хэш-функций с трудно обнаружимыми коллизиями

Коллизией (collision) произвольной функции $h\colon X \to Y$ называется всякая пара (x_1, x_2) различных элементов X такая, что $h(x_1) = h(x_2)$. Говоря неформально, в криптографии хэш-функцией называется полиномиально вычислимая и уменьшающая длину входа функция, для которой задача нахождения коллизий вычислительно трудна. Одной из формализаций этого понятия является понятие семейства хэш-функций с трудно обнаружимыми коллизиями, введенное Дамгордом в [Dam87]. Поэтому семейства хэш-функций с трудно обнаружимыми коллизиями уместно называть семействами хэш-функций Дамгорда.

Понятие семейства хэш-функций с трудно обнаружимыми коллизиями моделирует ситуацию, когда противник сначала получает случайное описание хэш-функции, а затем пытается найти коллизию этой хэш-функции. В определении семейства хэш-функций с трудно обнаружимыми коллизиями требуется, чтобы любой полиномиально ограниченный противник такого вида мог достичь цели лишь с пренебрежимо малой вероятностью.

Приведем формальное определение. Пусть для каждого $n \in \mathbb{N}$ определено непустое множество $D_n \subseteq \{0,1\}^*$ и распределение вероятностей $\mathcal{D}_n$ на D_n, причем семейство $(\mathcal{D}_n \mid n \in \mathbb{N})$ полиномиально конструируемо, когда индексы заданы в унарной записи. Пусть также k и m — полиномиальные параметры на $\mathbb{N}$ такие, что $m(n) < k(n)$ для всех $n \in \mathbb{N}$. Предположим, что каждой паре (n, d), где $n \in \mathbb{N}$ и $d \in D_n$, поставлена в соответствие функция $h_{n,d}\colon \{0,1\}^{k(n)} \to \{0,1\}^{m(n)}$.

Определение 10.1 (семейство хэш-функций с трудно обнаружимыми коллизиями)**.** Семейство $H = (h_{n,d} \mid n \in \mathbb{N},\ d \in D_n)$ называется *семейством хэш-функций с трудно обнаружимыми коллизиями* (collision-intractable, collision-resistant, or collision-free hash function family) относительно семейства распределений вероятностей $(\mathcal{D}_n \mid n \in \mathbb{N})$, если H полиномиально вычислимо и для любого полиномиального вероятностного алгоритма A

$$\Pr[A(1^n, \widetilde{d}) \text{ — коллизия } h_{n,\widetilde{d}}] = \mathrm{negl}(n),$$

где $\widetilde{d} \leftarrow \mathcal{D}_n$.

Следует отметить, что англоязычный термин «collision-free hash function family» неточно отражает сущность этого понятия. А именно, коллизии заведомо существуют, но их трудно находить.

Пусть для каждого $i \in \mathbb{N} \setminus \{0\}$ определено семейство $(h_{n,i,d_i}\colon \{0,1\}^{k_i(n)} \to \{0,1\}^{k_{i-1}(n)} \mid n \in \mathbb{N},\ d_i \in D_{n,i})$ хэш-функций с трудно обнаружимыми коллизиями относительно (полиномиально конструируемого, когда индексы заданы в унарной записи) семейства распределений $(\mathcal{D}_{n,i} \mid n \in \mathbb{N})$. Пусть также $l\colon \mathbb{N} \to \mathbb{N}\setminus\{0\}$ — полиномиальный параметр. Предположим, что выполнены следующие условия:

1) функция $n \mapsto k_{l(n)}(n)$ $(n \in \mathbb{N})$ — полиномиальный параметр;

2) функция $(1^n, 1^i, d_i, x) \mapsto h_{n,i,d_i}(x)$, где $n \in \mathbb{N}$, $i \in \{1, \dots, l(n)\}$, $d_i \in D_{n,i}$ и $x \in \{0,1\}^{k_i(n)}$, полиномиально вычислима;

3) существует полиномиальный вероятностный алгоритм G такой, что для любых $n \in \mathbb{N}$ и $i \in \{1, \dots, l(n)\}$ случайная величина $G(1^n, 1^i)$ имеет распределение $\mathcal{D}_{n,i}$;

4) для любого полиномиального вероятностного алгоритма A

$$\max_{i \in \{1,\dots,l(n)\}} \Pr[A(1^n, 1^i, \widetilde{d}_i) \text{ — коллизия } h_{n,i,\widetilde{d}_i}] = \operatorname{negl}(n),$$

где $\widetilde{d}_i \leftarrow \mathcal{D}_{n,i}$.

Отметим, что если полиномиальный параметр l ограничен сверху, то условия 1–4 выполняются автоматически.

Для каждых $n \in \mathbb{N}$ и $d_j \in D_{n,j}$ при всех $j \in \{i+1, \dots, l(n)\}$ (где $i \in \{0, \dots, l(n)\}$) обозначим через $\chi_{n,(d_{i+1},\dots,d_{l(n)})}$ композицию $l(n) - i$ функций $h_{n,l(n),d_{l(n)}}, \dots, h_{n,i+1,d_{i+1}}$ (в этом порядке). В частности, если $i = l(n)$, то $\chi_{n,()}$ есть тождественная функция множества $\{0,1\}^{k_{l(n)}(n)}$. Таким образом, $\chi_{n,(d_{i+1},\dots,d_{l(n)})}$ отображает $\{0,1\}^{k_{l(n)}(n)}$ в $\{0,1\}^{k_i(n)}$.

Теорема 10.2 (о композиции)**.** *Если выполнены условия 1–4, то семейство*

$$(\chi_{n,d} \colon \{0,1\}^{k_{l(n)}(n)} \to \{0,1\}^{k_0(n)} \mid n \in \mathbb{N},\ d \in D_{n,1} \times \dots \times D_{n,l(n)}) \tag{30}$$

является семейством хэш-функций с трудно обнаружимыми коллизиями относительно (полиномиально конструируемого, когда индексы заданы в унарной записи) семейства распределений $(\mathcal{D}_{n,1} \times \dots \times \mathcal{D}_{n,l(n)} \mid n \in \mathbb{N})$.

Доказательство. Полиномиальная вычислимость семейства функций (30) следует из условия 2, а полиномиальная конструируемость семейства распределений $(\mathcal{D}_{n,1} \times \dots \times \mathcal{D}_{n,l(n)} \mid n \in \mathbb{N})$, когда индексы заданы в унарной записи, — из условия 3. Пусть теперь A — произвольный полиномиальный вероятностный алгоритм. Выберем полиномиальный вероятностный алгоритм B, работающий на произвольном входе $(1^n, 1^i, d_i)$, где $n \in \mathbb{N}$, $i \in \{1, \dots, l(n)\}$ и $d_i \in D_{n,i}$, следующим образом:

1. Выбрать $d_j \leftarrow \mathcal{D}_{n,j}$ для каждого $j \in \{1, \dots, l(n)\} \setminus \{i\}$ (это возможно сделать за полиномиальное время ввиду условия 3).
2. Вычислить $v \leftarrow A(1^n, (d_1, \dots, d_{l(n)}))$.
3. Если $v = (x, y)$, где $x, y \in \{0,1\}^{k_{l(n)}(n)}$, то возвратить $(\chi_{n,(d_{i+1},\dots,d_{l(n)})}(x), \chi_{n,(d_{i+1},\dots,d_{l(n)})}(y))$ (эту пару можно вычислить за полиномиальное время ввиду условия 2).

Легко видеть, что в обозначениях алгоритма B

$$\begin{aligned} &A(1^n, (d_1, \dots, d_{l(n)})) \text{ — коллизия } \chi_{n,(d_1,\dots,d_{l(n)})} \\ &\quad \iff \exists\, i \in \{1, \dots, l(n)\}\ B(1^n, 1^i, d_i) \text{ — коллизия } h_{n,i,d_i}, \end{aligned}$$

причем если такое i существует, то оно единственно. Поэтому если $\widetilde{d}_j \leftarrow \mathcal{D}_{n,j}$ для всех $j \in \{1, \dots, l(n)\}$ и $\widetilde{d} = (\widetilde{d}_1, \dots, \widetilde{d}_{l(n)})$, то

$$\Pr[A(1^n, \widetilde{d}) \text{ — коллизия } \chi_{n,\widetilde{d}}] = \sum_{i=1}^{l(n)} \Pr[B(1^n, 1^i, \widetilde{d}_i) \text{ — коллизия } h_{n,i,\widetilde{d}_i}] \leqslant l(n) \operatorname{negl}(n) = \operatorname{negl}(n)$$

ввиду условия 4. □

***Замечание* 10.3.** Пусть $H = (h_{n,d} \colon \{0,1\}^{n+1} \to \{0,1\}^n \mid n \in \mathbb{N},\ d \in D_n)$ — семейство хэш-функций с трудно обнаружимыми коллизиями относительно семейства распределений вероятностей $(\mathcal{D}_n \mid n \in \mathbb{N})$. Для каждых $n \in \mathbb{N}$ и $i \in \mathbb{N} \setminus \{0\}$ положим $D_{n,i} = D_{n+i-1}$, $\mathcal{D}_{n,i} = \mathcal{D}_{n+i-1}$, $k_i(n) = n + i$ и $k_0(n) = n$. Кроме того, если $d_i \in D_{n,i} = D_{n+i-1}$, то пусть $h_{n,i,d_i} = h_{n+i-1,d_i} \colon \{0,1\}^{k_i(n)} \to \{0,1\}^{k_{i-1}(n)}$. Тогда непосредственно проверяется, что условия 1–4 выполнены. Поэтому из теоремы 10.2 следует, что если $l \colon \mathbb{N} \to \mathbb{N} \setminus \{0\}$ — полиномиальный параметр и $\chi_{n,(d_1,\dots,d_{l(n)})}$ — композиция $l(n)$ функций $h_{n+l(n)-1,d_{l(n)}}, \dots, h_{n,d_1}$ (в этом порядке), то

$$(\chi_{n,d} \colon \{0,1\}^{n+l(n)} \to \{0,1\}^n \mid n \in \mathbb{N},\ d \in D_n \times \dots \times D_{n+l(n)-1})$$

является семейством хэш-функций с трудно обнаружимыми коллизиями относительно семейства распределений $(\mathcal{D}_n \times \dots \times \mathcal{D}_{n+l(n)-1} \mid n \in \mathbb{N})$.

Таким образом, в замечании 10.3 из семейства $(h_{n,d}\colon \{0,1\}^{n+1} \to \{0,1\}^n \mid n \in \mathbb{N},\ d \in D_n)$ хэш-функций с трудно обнаружимыми коллизиями строится семейство $(\chi_{n,e}\colon \{0,1\}^{n+l(n)} \to \{0,1\}^n \mid n \in \mathbb{N},\ e \in E_n)$ хэш-функций с трудно обнаружимыми коллизиями, где $l\colon \mathbb{N} \to \mathbb{N} \setminus \{0\}$ — произвольный полиномиальный параметр. Увеличить разность между длинами входа и выхода хэш-функций (даже в более общей ситуации) можно также с помощью следующего варианта конструкции Меркле—Дамгорда (см. например, [Gol04, п. 6.2.3.2]). Пусть $H = (h_{n,d}\colon \{0,1\}^{k(n)} \to \{0,1\}^{m(n)} \mid n \in \mathbb{N},\ d \in D_n)$ — семейство функций, где k и m — полиномиальные параметры на $\mathbb{N}$ такие, что $m(n) < k(n)$ для всех $n \in \mathbb{N}$. Пусть также l — полиномиальный параметр на $\mathbb{N}$ такой, что $m(n) < l(n)$ при любом $n \in \mathbb{N}$. Для каждого $n \in \mathbb{N}$ положим

$$q(n) = \left\lceil \frac{l(n)}{k(n) - m(n)} \right\rceil \quad \text{и} \quad r(n) = q(n)(k(n) - m(n)) - l(n).$$

Очевидно, что q и r — полиномиальные параметры.

Пусть $n \in \mathbb{N}$, $d \in D_n$ и $x \in \{0,1\}^{l(n)}$. Представим $x0^{r(n)}$ в виде $x_1 \dots x_{q(n)}$, где $x_1, \dots, x_{q(n)} \in \{0,1\}^{k(n)-m(n)}$. Определим $g^i_{n,d}(x) \in \{0,1\}^{m(n)}$ индуктивно следующим образом:

$$g^0_{n,d}(x) = 0^{m(n)}, \quad g^i_{n,d}(x) = h_{n,d}(x_i g^{i-1}_{n,d}(x)) \text{ для } i \in \{1, \dots, q(n)\}.$$

Положим теперь $h'_{n,d}(x) = g^{q(n)}_{n,d}(x)$. Эти обозначения будут использоваться в следующем предложении и его доказательстве.

Предложение 10.4. *Предположим, что H — семейство хэш-функций с трудно обнаружимыми коллизиями относительно семейства распределений $(\mathcal{D}_n \mid n \in \mathbb{N})$. Тогда $H' = (h'_{n,d}\colon \{0,1\}^{l(n)} \to \{0,1\}^{m(n)} \mid n \in \mathbb{N},\ d \in D_n)$ также является семейством хэш-функций с трудно обнаружимыми коллизиями относительно семейства $(\mathcal{D}_n \mid n \in \mathbb{N})$.*

Доказательство. Полиномиальная вычислимость семейства функций H' следует из полиномиальной вычислимости семейства функций H. Пусть теперь A — произвольный полиномиальный вероятностный алгоритм. Выберем полиномиальный вероятностный алгоритм B, работающий на произвольном входе $(1^n, d)$, где $n \in \mathbb{N}$ и $d \in D$, следующим образом:

1. Вычислить $v \leftarrow A(1^n, d)$.
2. Предположим, что $v = (x, y)$ — коллизия функции $h'_{n,d}$. Тогда
   - вычислить $x_i, y_i \in \{0,1\}^{k(n)-m(n)}$ $(i \in \{1, \dots, q(n)\})$ такие, что $x0^{r(n)} = x_1 \dots x_{q(n)}$ и $y0^{r(n)} = y_1 \dots y_{q(n)}$;
   - выбрать наибольшее $i \in \{1, \dots, q(n)\}$, для которого $x_i g^{i-1}_{n,d}(x) \neq y_i g^{i-1}_{n,d}(y)$ (такие i существуют ввиду того, что $x0^{r(n)} \neq y0^{r(n)}$);
   - возвратить $(x_i g^{i-1}_{n,d}(x), y_i g^{i-1}_{n,d}(y))$ (легко видеть, что эта пара является коллизией функции $h_{n,d}$).

Следовательно,

$$\Pr[A(1^n, \widetilde{d}) \text{ — коллизия } h'_{n,\widetilde{d}}] \leqslant \Pr[B(1^n, \widetilde{d}) \text{ — коллизия } h_{n,\widetilde{d}}] = \operatorname{negl}(n),$$

где $\widetilde{d} \leftarrow \mathcal{D}_n$. □

Построим теперь семейство хэш-функций с трудно обнаружимыми коллизиями в предположении трудности задачи факторизации целых чисел. Для этого нам потребуются некоторые теоретико-числовые факты.

***Замечание* 10.5.** Пусть p и q — различные простые числа и $N = pq$. Предположим, что y и z — целые числа, для которых $y \not\equiv z \pmod N$, $y \not\equiv -z \pmod N$ и $y^2 \equiv z^2 \pmod N$. Тогда хорошо известно, что $\text{НОД}(y - z, N), \text{НОД}(y + z, N) \in \{p, q\}$. Действительно, пусть $\sigma \in \{-1, 1\}$. Тогда если $\text{НОД}(y + \sigma z, N) = N$, то $y + \sigma z \equiv 0 \pmod N$, что противоречит предположению $y \not\equiv -\sigma z \pmod N$. Если же $\text{НОД}(y + \sigma z, N) = 1$, то из сравнения $(y + \sigma z)(y - \sigma z) = y^2 - z^2 \equiv 0 \pmod N$ вытекает сравнение $y - \sigma z \equiv 0 \pmod N$, которое противоречит предположению $y \not\equiv \sigma z \pmod N$. Следовательно $\text{НОД}(y + \sigma z, N) \in \{p, q\}$, так как положительными делителями числа N являются только 1, p, q и N.

Мы будем применять доказанное утверждение в случае, когда $y \in \mathbb{Z}^\star_N$ и $z = 1$. А именно, если $y \in \mathbb{Z}^\star_N \setminus \{-1 \bmod N\}$ и $\operatorname{ord} y = 2$, то $\text{НОД}(y - 1, N), \text{НОД}(y + 1, N) \in \{p, q\}$.

Лемма 10.6. *Пусть p и q — различные нечетные простые числа и $N = pq$. Пусть также $\widetilde{g} \in_{\mathcal{U}} \mathbb{Z}_N^\star$. Тогда*

$$\Pr[\operatorname{ord}\widetilde{g} \text{ четен и } -1 \bmod N \notin \langle\widetilde{g}\rangle] \geqslant \frac{1}{2}. \tag{31}$$

Доказательство. По китайской теореме об остатках отображение $x \mapsto (x \bmod p, x \bmod q)$ является кольцевым изоморфизмом $\mathbb{Z}_N$ на $\mathbb{Z}_p \times \mathbb{Z}_q$. В частности, его ограничение на $\mathbb{Z}_N^\star$ будет групповым изоморфизмом $\mathbb{Z}_N^\star$ на $\mathbb{Z}_p^\star \times \mathbb{Z}_q^\star$. Пусть $p-1 = 2^s p'$, где $s \in \mathbb{N}\setminus\{0\}$, а p' — нечетное число. Тогда легко видеть, что $\mathbb{Z}_p^\star$ (являющаяся циклической группой порядка $p-1$) разлагается во внутреннее прямое произведение своей силовской 2-подгруппы A_p (циклической группы порядка 2^s) и подгруппы $B_p = \{x \in \mathbb{Z}_p^\star \mid \operatorname{ord} x \text{ нечетен}\}$ (циклической группы порядка p'). Аналогично, если $q-1 = 2^t q'$, где $t \in \mathbb{N}\setminus\{0\}$, а q' — нечетное число, то $\mathbb{Z}_q^\star$ (являющаяся циклической группой порядка $q-1$) разлагается во внутреннее прямое произведение своей силовской 2-подгруппы A_q (циклической группы порядка 2^t) и подгруппы $B_q = \{y \in \mathbb{Z}_q^\star \mid \operatorname{ord} y \text{ нечетен}\}$ (циклической группы порядка q'). Поэтому требуемое неравенство (31) эквивалентно неравенству

$$\Pr[\operatorname{ord}\widetilde{v} \text{ четен и } (-1 \bmod p, -1 \bmod q) \notin \langle\widetilde{v}\rangle] \geqslant \frac{1}{2}. \tag{32}$$

где $\widetilde{v} = (\widetilde{a}_p\widetilde{b}_p \bmod p, \widetilde{a}_q\widetilde{b}_q \bmod q)$, $\widetilde{a}_p \in_{\mathcal{U}} A_p$, $\widetilde{b}_p \in_{\mathcal{U}} B_p$, $\widetilde{a}_q \in_{\mathcal{U}} A_q$ и $\widetilde{b}_q \in_{\mathcal{U}} B_q$.

Пусть $v = (a_p b_p \bmod p, a_q b_q \bmod q)$, $a_p \in A_p$, $b_p \in B_p$, $a_q \in A_q$ и $b_q \in B_q$. Покажем, что

$$\operatorname{ord} v \text{ нечетен или } (-1 \bmod p, -1 \bmod q) \in \langle v\rangle \iff \operatorname{ord} a_p = \operatorname{ord} a_q. \tag{33}$$

Действительно, если $\operatorname{ord} v$ нечетен, то $a_p = 1$, $a_q = 1$ (так как $v \in B_p \times B_q$), поэтому $\operatorname{ord} a_p = \operatorname{ord} a_q = 1$. Если же $v^m = (-1 \bmod p, -1 \bmod q)$ для некоторого $m = 2^l m'$ (где $l \in \mathbb{N}$ и m' — нечетное число из $\mathbb{N}$), то $(a_p^{m'})^{2^l} \equiv -1 \pmod p$ и, следовательно, $(a_p^{m'})^{2^{l+1}} \equiv 1 \pmod p$. Поэтому $\operatorname{ord} a_p = \operatorname{ord}(a_p^{m'} \bmod p) = 2^{l+1}$. Равенство порядков элементов a_p и $a_p^{m'} \bmod p$ вытекает из того, что $x \mapsto x^{m'} \bmod p$ — изоморфизм группы A_p (так как m' и $|A_p|$ взаимно просты). Рассуждая аналогично, получим, что $\operatorname{ord} a_q = \operatorname{ord}(a_q^{m'} \bmod q) = 2^{l+1}$. Таким образом, $\operatorname{ord} a_p = \operatorname{ord} a_q$. Обратно, пусть $\operatorname{ord} a_p = \operatorname{ord} a_q = 2^r$ для некоторого $r \in \mathbb{N}$. Если $r = 0$, то $a_p = 1$, $a_q = 1$ и, следовательно, $\operatorname{ord} v$ нечетен. Если же $r \geqslant 1$, то $a_p^{2^{r-1}} \equiv -1 \pmod p$, так как $-1 \bmod p$ — единственный элемент порядка 2 в A_p (и даже в $\mathbb{Z}_p^\star$). Аналогично, $a_q^{2^{r-1}} \equiv -1 \pmod q$. Поэтому

$$v^{2^{r-1}p'q'} = ((-1)^{p'q'}(b_p^{p'})^{2^{r-1}q'} \bmod p, (-1)^{p'q'}(b_q^{q'})^{2^{r-1}p'} \bmod q) = (-1 \bmod p, -1 \bmod q).$$

Напомним, что при $1 \leqslant i \leqslant n$ число элементов порядка 2^i в циклической группе порядка 2^n равно 2^{i-1}. Из эквивалентности (33) и этого факта вытекает, что

$$\begin{aligned}\Pr[\operatorname{ord}\widetilde{v} \text{ нечетен или } (-1 \bmod p, -1 \bmod q) \in \langle\widetilde{v}\rangle] &= \Pr[\operatorname{ord}\widetilde{a}_p = \operatorname{ord}\widetilde{a}_q] \\ = \frac{1 + \sum_{i=1}^{\min\{s,t\}}(2^{i-1})^2}{2^{s+t}} &\leqslant \frac{1 + \sum_{j=0}^{2(\min\{s,t\}-1)} 2^j}{2^{s+t}} = \frac{2^{2\min\{s,t\}-1}}{2^{s+t}} \leqslant \frac{1}{2},\end{aligned}$$

откуда непосредственно следует требуемое неравенство (32). □

Лемма 10.7. *Существует полиномиальный детерминированный алгоритм A, удовлетворяющий следующему условию. Пусть p и q — различные нечетные простые числа и $N = pq$. Пусть также $g \in \mathbb{Z}_N^\star$ таков, что $\operatorname{ord} g$ четен и $-1 \bmod N \notin \langle g\rangle$. Тогда если для $m \in \mathbb{Z}\setminus\{0\}$ имеет место равенство $g^m \bmod N = 1$, то $A(N, g, m) \in \{p, q\}$.*

Доказательство. Требуемый алгоритм A на произвольном входе (N, g, m), где N, g и m удовлетворяют вышеуказанным требованиям, работает следующим образом:

1. Пусть $m = 2^l m'$, где $l \in \mathbb{N}\setminus\{0\}$ и m' нечетно. Пусть также $g_i = g^{2^i m'} \bmod N$ при $i \in \{0, \ldots, l\}$. Алгоритм находит (последовательно вычисляя $g_0, g_1, \ldots$) такое единственное $j \in \{1, \ldots, l\}$, что $g_{j-1} \neq 1$ и $g_j = 1$. (Оно существует, поскольку $g_0 \neq 1$ и $g_l = 1$. Кроме того, $g_{j-1} \neq -1 \bmod N$, так как $g_{j-1} \in \langle g\rangle$).
2. Возвратить НОД$(g_{j-1}-1, N)$. (Из замечания 10.5 следует, что НОД$(g_{j-1}-1, N) \in \{p, q\}$.) □

Предположим, что $(\mathcal{N}_n \mid n \in \mathbb{N})$ — семейство распределений вероятностей, удовлетворяющее следующим условиям:

- для любого $n \in \mathbb{N}$ каждый элемент множества $\operatorname{supp}\mathcal{N}_n$ является произведением двух различных нечетных простых чисел;
- семейство $(\mathcal{N}_n \mid n \in \mathbb{N})$ — полиномиально конструируемо, когда индексы заданы в унарной записи;
- для любого полиномиального вероятностного алгоритма A

$$\Pr[A(1^n, \widetilde{N}) \text{ — нетривиальный делитель } \widetilde{N}] = \operatorname{negl}(n),$$

где $\widetilde{N} \leftarrow \mathcal{N}_n$.

(Делитель d целого положительного числа m называется *нетривиальным*, если $2 \leqslant d \leqslant m-1$.) Предположение о существовании такого семейства распределений вероятностей мы называем *предположением о сложности задачи факторизации целых чисел* (integer factoring intractability assumption).

Пусть $n \in \mathbb{N}$. Обозначим через η_n биекцию $\mathbb{Z}_{2^n}$ на $\{0,1\}^n$, ставящую в соответствие произвольному числу $i \in \mathbb{Z}_{2^n}$ строку из $\{0,1\}^n$, являющуюся двоичной записью числа i. Положим $D_n = \{(N, g_1, \ldots, g_n) \mid N \in \operatorname{supp} N_n,\ g_1, \ldots, g_n \in \mathbb{Z}_N^\star\}$. Кроме того, обозначим через $\mathcal{D}_n$ распределение случайной величины $(\widetilde{N}, \widetilde{g}_1, \ldots, \widetilde{g}_n)$, где $\widetilde{N} \leftarrow \mathcal{N}_n$ и $\widetilde{g}_1, \ldots, \widetilde{g}_n \in_{\mathcal{U}} \mathbb{Z}_{\widetilde{N}}^\star$.

Выберем полиномиальный параметр $\pi\colon \mathbb{N} \to \mathbb{N}$ такой, что $\max \operatorname{supp}\mathcal{N}_n \leqslant 2^{\pi(n)}$ для всех $n \in \mathbb{N}$. Пусть также k — полиномиальный параметр на $\mathbb{N}$, удовлетворяющий неравенству $k(n) > n\pi(n)$ при любом $n \in \mathbb{N}$. Очевидно, что $\pi(n) \geqslant 1$ и, следовательно, $k(n) \leqslant \operatorname{poly}(n) \leqslant \operatorname{poly}(n\pi(n))$ для всех $n \in \mathbb{N}$. Для каждых $n \in \mathbb{N}$, $N \in \operatorname{supp}\mathcal{N}_n$ и $g_1, \ldots g_n \in \mathbb{Z}_N^\star$ определим функцию $h_{n,(N,g_1,\ldots g_n)}\colon \{0,1\}^{k(n)} \to \{0,1\}^{n\pi(n)}$ равенством

$$h_{n,(N,g_1,\ldots g_n)}(\eta_{k(n)}(x)) = \eta_{\pi(n)}(g_1^x \bmod N) \ldots \eta_{\pi(n)}(g_n^x \bmod N) \quad (x \in \mathbb{Z}_{2^{k(n)}}).$$

Здесь использовано то, что $N \leqslant 2^{\pi(n)}$ (по выбору π) и, следовательно, $\mathbb{Z}_N \subseteq \mathbb{Z}_{2^{\pi(n)}}$.

Теорема 10.8. *Семейство $H = (h_{n,d} \mid n \in \mathbb{N},\ d \in D_n)$ построенных выше функций является семейством хэш-функций с трудно обнаружимыми коллизиями относительно семейства $(\mathcal{D}_n \mid n \in \mathbb{N})$ вышеуказанных распределений вероятностей.*

Доказательство. Полиномиальная вычислимость семейства H очевидна. Пусть теперь A — произвольный полиномиальный вероятностный алгоритм. Выберем полиномиальный вероятностный алгоритм B, работающий на произвольном входе вида $(1^n, N)$, где $n \in \mathbb{N}$ и $N \in \operatorname{supp}\mathcal{N}_n$, следующим образом:

1. Выбрать $g_1, \ldots, g_n \in_{\mathcal{U}} \mathbb{Z}_N^\star$.
2. Вычислить $A(1^n, (N, g_1, \ldots, g_n))$.
3. Предположим, что на предыдущем шаге получен выход вида $(\eta_{k(n)}(x), \eta_{k(n)}(y))$, где $x, y \in \mathbb{Z}_{2^{k(n)}}$ и $x \neq y$. Тогда для i от 1 до n выполнять полиномиальный детерминированный алгоритм, удовлетворяющий условию леммы 10.7, на входе $(N, g_i, x-y)$. Если для некоторого i последний алгоритм выдаст нетривиальный делитель числа N, то возвратить этот делитель и закончить работу.

Пусть $n \in \mathbb{N}$ и $N \in \operatorname{supp}\mathcal{N}_n$. Обозначим через E_N множество всех $g \in \mathbb{Z}_N^\star$ таких, что $\operatorname{ord} g$ четен и $-1 \bmod N \notin \langle g \rangle$. Используя обозначения описания алгоритма B, легко видеть, что если $A(1^n, (N, g_1, \ldots, g_n)) = (\eta_{k(n)}(x), \eta_{k(n)}(y))$ — коллизия $h_{n,(N,g_1,\ldots,g_n)}$ (т. е. $g_1^{x-y} \bmod N = \cdots = g_n^{x-y} \bmod N = 1$) и $g_i \in E_N$ при некотором $i \in \{1, \ldots, n\}$, то $B(1^n, N)$ — нетривиальный делитель числа N. Поэтому если $\widetilde{g}_1, \ldots, \widetilde{g}_n \in_{\mathcal{U}} \mathbb{Z}_N^\star$, то

$$\begin{aligned}
&\Pr[A(1^n, (N, \widetilde{g}_1, \ldots, \widetilde{g}_n)) \text{ — коллизия } h_{n,(N,\widetilde{g}_1,\ldots\widetilde{g}_n)}] \\
&\quad = \Pr[A(1^n, (N, \widetilde{g}_1, \ldots, \widetilde{g}_n)) \text{ — коллизия } h_{n,(N,\widetilde{g}_1,\ldots,\widetilde{g}_n)},\ \exists\, i \in \{1, \ldots, n\}\ \widetilde{g}_i \in E_N] \\
&\qquad + \Pr[A(1^n, (N, \widetilde{g}_1, \ldots, \widetilde{g}_n)) \text{ — коллизия } h_{n,(N,\widetilde{g}_1,\ldots,\widetilde{g}_n)},\ \forall\, i \in \{1, \ldots, n\}\ \widetilde{g}_i \notin E_N] \\
&\quad \leqslant \Pr[B(1^n, N) \text{ — нетривиальный делитель } N] + \Pr[\forall\, i \in \{1, \ldots, n\}\ \widetilde{g}_i \notin E_N],
\end{aligned}$$

где $\Pr[\forall\, i \in \{1, \ldots, n\}\ \widetilde{g}_i \notin E_N] \leqslant 2^{-n}$, так как $\Pr[\widetilde{g}_i \in E_N] \geqslant 1/2$ по лемме 10.6. Следовательно,

$$\Pr[A(1^n, \widetilde{d}) \text{ — коллизия } h_{n,\widetilde{d}}] \leqslant \Pr[B(1^n, \widetilde{N}) \text{ — нетривиальный делитель } \widetilde{N}] + \frac{1}{2^n} = \operatorname{negl}(n),$$

где $\widetilde{d} \leftarrow \mathcal{D}_n$ и $\widetilde{N} \leftarrow \mathcal{N}_n$. □

11. Универсальные односторонние семейства хэш-функций

Рассмотрим теперь другую формализацию неформального понятия хэш-функции, а именно, понятие универсального одностороннего семейства хэш-функций, введенное Наором и Юнгом в [NY89]. Поэтому универсальные односторонние семейства хэш-функций уместно называть семействами хэш-функций Наора—Юнга. К универсальным односторонним семействам хэш-функций предъявляются более слабые требования, чем к семействам хэш-функций с трудно обнаружимыми коллизиями. Однако этих требований достаточно для некоторых важных приложений семейств хэш-функций, например, для построения протоколов электронной подписи, стойких в некотором сильном смысле. Другим преимуществом универсальных односторонних семейств хэш-функций перед семействами хэш-функций с трудно обнаружимыми коллизиями является то, что первые существуют тогда и только тогда, когда существуют односторонние функции (см. теорему 11.4 ниже), в то время как для вторых аналогичный результат не доказан. Более того, известен результат, сильно ограничивающий класс методов, с помощью которых можно было бы построить семейство хэш-функций с трудно обнаружимыми коллизиями на основе произвольной односторонней перестановки (см. [Sim98]).

Понятие универсального одностороннего семейства хэш-функций моделирует ситуацию, когда противник сначала выбирает значение x аргумента хэш-функций, а затем, получив случайное описание хэш-функции h, пытается найти значение y аргумента хэш-функций такое, что (x, y) — коллизия функции h, т. е. $x \neq y$ и $h(x) = h(y)$. В определении универсального одностороннего семейства хэш-функций требуется, чтобы любой полиномиально ограниченный противник такого вида мог достичь цели лишь с пренебрежимо малой вероятностью. Таким образом, принципиальное отличие универсального одностороннего семейства хэш-функций от семейства хэш-функций с трудно обнаружимыми коллизиями состоит в том, что противник должен найти коллизию (x, y) хэш-функции h такую, что x выбран *до* получения им описания функции h и, следовательно, независимо от этого описания.

Пусть h — функция, определенная на множестве X, а $x \in X$. Тогда элемент $y \in X$ называется *специфической коллизией* (specific collision) для функции h и элемента x, если (x, y) — коллизия функции h. Этот термин использовался в работе [Sim98]. Отметим, что в этой работе [Sim98] коллизия в смысле предыдущего раздела называлась *экзистенциальной коллизией* (existential collision). Таким образом, можно сказать, что понятие универсального одностороннего семейства хэш-функций формализует условие трудности нахождения специфических коллизий, а понятие семейства хэш-функций с трудно обнаружимыми коллизиями — условие трудности нахождения экзистенциальных коллизий.

Перейдем к формальному определению. Пусть для каждого $n \in \mathbb{N}$ определено непустое множество $D_n \subseteq \{0,1\}^*$ и распределение вероятностей $\mathcal{D}_n$ на D_n, причем семейство $(\mathcal{D}_n \mid n \in \mathbb{N})$ полиномиально конструируемо, когда индексы заданы в унарной записи. Пусть также k и m — полиномиальные параметры на $\mathbb{N}$ такие, что $m(n) < k(n)$ для всех $n \in \mathbb{N}$. Предположим, что каждой паре (n, d), где $n \in \mathbb{N}$ и $d \in D_n$, поставлена в соответствие функция $h_{n,d}\colon \{0,1\}^{k(n)} \to \{0,1\}^{m(n)}$.

Назовем алгоритм A *алгоритмом поиска специфических коллизий* семейства $(h_{n,d} \mid n \in \mathbb{N},\ d \in D_n)$, если он работает следующим образом. Сначала, получив на вход 1^n при произвольном $n \in \mathbb{N}$, алгоритм A вычисляет некоторую строку $x \in \{0,1\}^{k(n)}$, обозначаемую нами через $A(1^n)$. Затем этому алгоритму подается на вход строка $d \in D_n$, после чего он вычисляет еще одну строку $y \in \{0,1\}^{k(n)}$, обозначаемую нами через $A(1^n, d)$. В случае успеха строка y должна быть специфической коллизией для функции $h_{n,d}$ и строки x. Другими словами, пара (x, y) должна быть коллизией функции $h_{n,d}$.

Определение 11.1 (универсальное одностороннее семейство хэш-функций)**.** Семейство $H = (h_{n,d} \mid n \in \mathbb{N},\ d \in D_n)$ называется *универсальным односторонним семейством хэш-функций* (universal one-way hash function family, UOWHF family) относительно семейства распределений вероятностей $(\mathcal{D}_n \mid n \in \mathbb{N})$, если H полиномиально вычислимо и для любого полиномиального вероятностного алгоритма A поиска специфических коллизий семейства H

$$\Pr[(A(1^n), A(1^n, \widetilde{d})) — \text{коллизия } h_{n,\widetilde{d}}] = \mathrm{negl}(n),$$

где $\widetilde{d} \leftarrow \mathcal{D}_n$.

Отметим, что иногда слово «универсальное» в названии этого понятия опускается. Кроме того, очевидно, что всякое семейство хэш-функций с трудно обнаружимыми коллизиями относительно

какого-либо семейства распределений вероятностей является универсальным односторонним семейством хэш-функций относительно того же семейства распределений.

Теорема о композиции, аналогичная теореме 10.2 справедлива и для универсальных односторонних семейств хэш-функций. Мы приведем подробную формулировку этой теоремы для универсальных односторонних семейств хэш-функций; доказательство полностью аналогично доказательству теоремы 10.2. См. также [NY89].

Пусть для каждого $i \in \mathbb{N} \setminus \{0\}$ определено универсальное одностороннее семейство хэш-функций $(h_{n,i,d_i} \colon \{0,1\}^{k_i(n)} \to \{0,1\}^{k_{i-1}(n)} \mid n \in \mathbb{N},\ d_i \in D_{n,i})$ относительно (полиномиально конструируемого, когда индексы заданы в унарной записи) семейства распределений $(\mathcal{D}_{n,i} \mid n \in \mathbb{N})$. Пусть также $l \colon \mathbb{N} \to \mathbb{N} \setminus \{0\}$ — полиномиальный параметр. Предположим, что выполнены следующие условия:

1) функция $n \mapsto k_{l(n)}(n)$ $(n \in \mathbb{N})$ — полиномиальный параметр;

2) функция $(1^n, 1^i, d_i, x) \mapsto h_{n,i,d_i}(x)$, где $n \in \mathbb{N}$, $i \in \{1, \ldots, l(n)\}$, $d_i \in D_{n,i}$ и $x \in \{0,1\}^{k_i(n)}$, полиномиально вычислима;

3) существует полиномиальный вероятностный алгоритм G такой, что для любых $n \in \mathbb{N}$ и $i \in \{1, \ldots, l(n)\}$ случайная величина $G(1^n, 1^i)$ имеет распределение $\mathcal{D}_{n,i}$;

4) для любого полиномиального вероятностного алгоритма A, который на входе $(1^n, 1^i)$ (при произвольных $n \in \mathbb{N}$ и $i \in \{1, \ldots, l(n)\}$) сначала вычисляет некоторую строку $A(1^n, 1^i) \in \{0,1\}^{k_i(n)}$, а потом, получив на вход $d_i \in D_{n,i}$, вычисляет еще одну строку $A(1^n, 1^i, d_i) \in \{0,1\}^{k_i(n)}$, имеет место равенство

$$\max_{i \in \{1,\ldots,l(n)\}} \Pr[(A(1^n, 1^i), A(1^n, 1^i, \widetilde{d_i})) \text{ — коллизия } h_{n,i,\widetilde{d_i}}] = \operatorname{negl}(n),$$

где $\widetilde{d_i} \leftarrow \mathcal{D}_{n,i}$.

Отметим, что если полиномиальный параметр l ограничен сверху, то условия 1–4 выполняются автоматически.

Для каждых $n \in \mathbb{N}$ и $d_j \in D_{n,j}$ при всех $j \in \{i+1, \ldots, l(n)\}$ (где $i \in \{0, \ldots, l(n)\}$) обозначим через $\chi_{n,(d_{i+1},\ldots,d_{l(n)})}$ композицию $l(n) - i$ функций $h_{n,l(n),d_{l(n)}}, \ldots, h_{n,i+1,d_{i+1}}$ (в этом порядке). В частности, если $i = l(n)$, то $\chi_{n,()}$ есть тождественная функция множества $\{0,1\}^{k_{l(n)}(n)}$. Таким образом, $\chi_{n,(d_{i+1},\ldots,d_{l(n)})}$ отображает $\{0,1\}^{k_{l(n)}(n)}$ в $\{0,1\}^{k_i(n)}$.

Теорема 11.2 (о композиции)**.** *Если выполнены условия 1–4, то семейство*

$$(\chi_{n,d} \colon \{0,1\}^{k_{l(n)}(n)} \to \{0,1\}^{k_0(n)} \mid n \in \mathbb{N},\ d \in D_{n,1} \times \cdots \times D_{n,l(n)})$$

является универсальным односторонним семейством хэш-функций относительно (полиномиально конструируемого, когда индексы заданы в унарной записи) семейства распределений $(\mathcal{D}_{n,1} \times \cdots \times \mathcal{D}_{n,l(n)} \mid n \in \mathbb{N})$.

***Замечание* 11.3** (аналог замечания 10.3)**.** Пусть $H = (h_{n,d} \colon \{0,1\}^{n+1} \to \{0,1\}^n \mid n \in \mathbb{N},\ d \in D_n)$ — универсальное одностороннее семейство хэш-функций относительно семейства распределений вероятностей $(\mathcal{D}_n \mid n \in \mathbb{N})$. Для каждых $n \in \mathbb{N}$ и $i \in \mathbb{N} \setminus \{0\}$ положим $D_{n,i} = D_{n+i-1}$, $\mathcal{D}_{n,i} = \mathcal{D}_{n+i-1}$, $k_i(n) = n + i$ и $k_0(n) = n$. Кроме того, если $d_i \in D_{n,i} = D_{n+i-1}$, то пусть $h_{n,i,d_i} = h_{n+i-1,d_i} \colon \{0,1\}^{k_i(n)} \to \{0,1\}^{k_{i-1}(n)}$. Тогда непосредственно проверяется, что условия 1–4 выполнены. Поэтому из теоремы 11.2 следует, что если $l \colon \mathbb{N} \to \mathbb{N} \setminus \{0\}$ — полиномиальный параметр и $\chi_{n,(d_1,\ldots,d_{l(n)})}$ — композиция $l(n)$ функций $h_{n+l(n)-1,d_{l(n)}}, \ldots, h_{n,d_1}$ (в этом порядке), то

$$(\chi_{n,d} \colon \{0,1\}^{n+l(n)} \to \{0,1\}^n \mid n \in \mathbb{N},\ d \in D_n \times \cdots \times D_{n+l(n)-1})$$

является универсальным односторонним семейством хэш-функций относительно семейства распределений $(\mathcal{D}_n \times \cdots \times \mathcal{D}_{n+l(n)-1} \mid n \in \mathbb{N})$.

Вопрос о существовании универсальных односторонних семейств хэш-функций решается следующей теоремой, принадлежащей Ромпелю [Rom90]. Полное доказательство см. в [KK05].

Теорема 11.4. *Универсальные односторонние семейства хэш-функций (относительно некоторого семейства распределений вероятностей, полиномиально конструируемого, когда индексы заданы в унарной записи) существуют тогда и только тогда, когда существуют односторонние функции.*

Доказательство теоремы 11.4 остается за рамками настоящего курса ввиду сложности этого доказательства. Мы приведем лишь конструкцию универсального одностороннего семейства хэш-функций на основе произвольной односторонней перестановки (см. [NY89], [Lub96, лекция 16]). В этой конструкции для каждого $n \in \mathbb{N} \setminus \{0\}$ на множестве $\{0,1\}^n$ вводится структура поля (из 2^n элементов) таким образом, чтобы сложение в этом поле совпадало с $\oplus$, а умножение (обозначаемое точкой) выполнялось за полиномиальное время. Для этого достаточно иметь полиномиальный детерминированный алгоритм, который на входе 1^n при произвольном $n \in \mathbb{N} \setminus \{0\}$ строит какой-либо неприводимый многочлен $p_n(X) \in \mathbb{Z}_2[X]$ степени n. Тогда двоичная строка $b_0 \dots b_{n-1}$ отождествляется с элементом $b_0 + b_1 X + \dots b_{n-1} X^{n-1} + \mathbb{Z}_2[X] p_n(X)$ факторкольца $\mathbb{Z}_2[X]/\mathbb{Z}_2[X]p_n(X)$, которое является полем из 2^n элементов. Такие алгоритмы существуют (см. [Чис84, Сем88, Sho90]), но мы не будем их приводить здесь.

Теорема 11.5. *Пусть f — односторонняя перестановка. Для каждого $n \in \mathbb{N}$ положим $D_n = (\{0,1\}^{n+1} \setminus \{0^{n+1}\}) \times \{0,1\}^{n+1}$ и $\mathcal{D}_n = \mathcal{U}((\{0,1\}^{n+1} \setminus \{0^{n+1}\}) \times \{0,1\}^{n+1}) = \mathcal{U}(\{0,1\}^{n+1} \setminus \{0^{n+1}\}) \times \mathcal{U}(\{0,1\}^{n+1})$. Пусть также для всех $n \in \mathbb{N}$ и $(a,b) \in D_n$ функция $h_{n,(a,b)} \colon \{0,1\}^{n+1} \to \{0,1\}^n$ определена равенством*

$$h_{n,(a,b)}(x) = (a \cdot f(x) \oplus b)_{[1,\dots,n]} \quad (x \in \{0,1\}^{n+1}).$$

Тогда семейство $(h_{n,(a,b)} \mid n \in \mathbb{N},\ (a,b) \in D_n)$, является универсальным односторонним семейством хэш-функций относительно семейства распределений вероятностей $(\mathcal{D}_n \mid n \in \mathbb{N})$.

Доказательство. Полиномиальная вычислимость семейства $(h_{n,(a,b)} \mid n \in \mathbb{N},\ (a,b) \in D_n)$ очевидна. Пусть теперь A — произвольный полиномиальный вероятностный алгоритм поиска специфических коллизий этого семейства. Выберем полиномиальный вероятностный алгоритм B, работающий на произвольном входе $w \in \{0,1\}^{\geqslant 1}$ следующим образом:

1. Положить $n = |w| - 1$ и вычислить $x \leftarrow A(1^n)$.
2. Если $f(x) = w$, то возвратить x и закончить работу.
3. Если $f(x) \neq w$, то вычислить $a = (f(x) \oplus w)^{-1} \cdot e$, где $e = 0^n 1$ (обращение осуществляется в поле с носителем $\{0,1\}^{n+1}$), выбрать $b \in_{\mathcal{U}} \{0,1\}^{n+1}$, вычислить $y \leftarrow A(1^n, (a,b))$ и возвратить y.

Используя обозначения, введенные при описании этого алгоритма, легко видеть, что если $f(x) \neq w$ и (x,y) — коллизия функции $h_{n,(a,b)}$, то $a \cdot f(x) \oplus b = a \cdot f(y) \oplus b \oplus e$ и, следовательно, $f(y) = f(x) \oplus a^{-1} \cdot e = f(x) \oplus f(x) \oplus w = w$, так как $a^{-1} = (f(x) \oplus w) \cdot e^{-1}$.

Выберем полиномиальный параметр l на $\mathbb{N}$ такой, что алгоритм A при вычислении на входе $(1^n, d)$ (где $n \in \mathbb{N}$ и $d \in \operatorname{supp} \mathcal{D}_n$) использует не более $l(n)$ случайных битов. Пусть $n \in \mathbb{N}$, $\widetilde{v}, \widetilde{b} \in_{\mathcal{U}} \{0,1\}^{n+1}$, $\widetilde{r} \in_{\mathcal{U}} \{0,1\}^{l(n)}$, $\widetilde{x} = A(1^n; \widetilde{r})$, $\widetilde{d} \leftarrow \mathcal{D}_n$ и $\widetilde{a} = (f(\widetilde{x}) \oplus f(\widetilde{v}))^{-1} \cdot e$, где $e = 0^n 1$, как и выше (случайная величина $\widetilde{a}$ определена при условии $f(\widetilde{x}) \neq f(\widetilde{v})$ или, что эквивалентно, $\widetilde{x} \neq \widetilde{v}$). Непосредственно проверяется, что для любых $r \in \{0,1\}^{l(n)}$, $a \in \{0,1\}^{n+1} \setminus \{0^{n+1}\}$ и $b \in \{0,1\}^{n+1}$

$$\Pr[\widetilde{r} = r,\ \widetilde{a} = a,\ \widetilde{b} = b \mid \widetilde{x} \neq \widetilde{v}] = \frac{\Pr[\widetilde{r} = r,\ \widetilde{v} = f^{-1}(f(A(1^n; r)) \oplus a^{-1} \cdot e),\ \widetilde{b} = b]}{1 - 2^{-(n+1)}} = \frac{1}{2^{l(n)}} \frac{1}{2^{n+1} - 1} \frac{1}{2^{n+1}}.$$

Это показывает, что случайная величина $(\widetilde{r}, (\widetilde{a}, \widetilde{b}))$ при условии $\widetilde{x} \neq \widetilde{v}$ имеет то же самое распределение, что и случайная величина $(\widetilde{r}, \widetilde{d})$ без каких-либо условий. Поэтому

$$\Pr[(\widetilde{x}, A(1^n, \widetilde{d}; \widetilde{r})) \text{ — коллизия } h_{n,\widetilde{d}}] = \Pr[(\widetilde{x}, A(1^n, (\widetilde{a}, \widetilde{b}); \widetilde{r})) \text{ — коллизия } h_{n,(\widetilde{a},\widetilde{b})} \mid \widetilde{x} \neq \widetilde{v}].$$

В то же время

$$\Pr[(\widetilde{x}, A(1^n, (\widetilde{a}, \widetilde{b}); \widetilde{r})) \text{ — коллизия } h_{n,(\widetilde{a},\widetilde{b})} \mid \widetilde{x} \neq \widetilde{v}] \leqslant \frac{\Pr[B(f(\widetilde{v})) = \widetilde{v}]}{1 - 2^{-(n+1)}} \leqslant 2 \operatorname{negl}(n+1) = \operatorname{negl}(n)$$

ввиду замечания, приведенного сразу после описания алгоритма B, и неравенства $1 - 2^{-(n+1)} \geqslant 1/2$. Следовательно,

$$\Pr[(A(1^n), A(1^n, \widetilde{d})) \text{ — коллизия } h_{n,\widetilde{d}}] = \Pr[(\widetilde{x}, A(1^n, \widetilde{d}; \widetilde{r})) \text{ — коллизия } h_{n,\widetilde{d}}] = \operatorname{negl}(n). \quad \square$$

12. Семейства функций и перестановок с секретом

Пусть I — подмножество $\{0,1\}^*$ и для каждого $i \in I$ определено множество $X_i \subseteq \{0,1\}^*$. Пусть также для каждого $n \in \mathbb{N}$ задано распределение вероятностей $\mathcal{G}_n$ на множестве $I \times \{0,1\}^*$, а для каждого $i \in I$ — распределение вероятностей $\mathcal{X}_i$ на множестве X_i. Предположим, что семейство $(\mathcal{G}_n \mid n \in \mathbb{N})$ распределений вероятностей полиномиально конструируемо, когда индексы заданы в унарной записи, а семейство $(\mathcal{X}_i \mid i \in I)$ — полиномиально конструируемо. Обозначим через $\mathcal{I}_n$, где $n \in \mathbb{N}$, распределение случайной величины $\widetilde{i}$ такой, что $(\widetilde{i}, \widetilde{t}) \leftarrow \mathcal{G}_n$.

Определение 12.1 (семейство функций с секретом)**.** Полиномиально вычислимое семейство функций $(f_i \colon X_i \to \{0,1\}^* \mid i \in I)$ называется *семейством функций с секретом* (family of trapdoor functions, trapdoor function family) относительно семейств распределений вероятностей $(\mathcal{G}_n \mid n \in \mathbb{N})$ и $(\mathcal{X}_i \mid i \in I)$, если

1) это семейство является односторонним относительно семейств распределений вероятностей $(\mathcal{I}_n \mid n \in \mathbb{N})$ и $(\mathcal{X}_i \mid i \in I)$;

2) существует полиномиальный детерминированный алгоритм M такой, что $M(1^n, i, t, y) \in f_i^{-1}(y)$ для любых $n \in \mathbb{N}$, $(i,t) \in \operatorname{supp} \mathcal{G}_n$ и $y \in f_i(X_i)$.

Говоря упрощенно, семейство функций с секретом — это одностороннее семейство функций, которое тем не менее полиномиально инвертируемо, если известна некоторая дополнительная информация (обозначаемая в условии 2 определения 12.1 через t от слова «trapdoor»), соответствующая инвертируемой функции из семейства. Эта дополнительная информация t должна быть трудновычислима по $(1^n, i)$, где n — параметр стойкости, а i — индекс функции в семействе. Однако по 1^n можно генерировать за полиномиальное время пары вида (i,t), где t — вышеуказанная дополнительная информация, позволяющая инвертировать за полиномиальное время функцию f_i.

Если в определении 12.1 потребовать, чтобы функция f_i была перестановкой множества X_i для всех $i \in I$, то мы получим определение семейства перестановок с секретом. См. также [Gol04, определение 2.4.4].

Определение 12.2 (семейство перестановок с секретом)**.** Полиномиально вычислимое семейство перестановок $(f_i \colon X_i \to X_i \mid i \in I)$ называется *семейством перестановок с секретом* (family of trapdoor permutations, trapdoor permutation family) относительно семейств распределений вероятностей $(\mathcal{G}_n \mid n \in \mathbb{N})$ и $(\mathcal{X}_i \mid i \in I)$, если

1) это семейство является односторонним относительно семейств распределений вероятностей $(\mathcal{I}_n \mid n \in \mathbb{N})$ и $(\mathcal{X}_i \mid i \in I)$;

2) существует полиномиальный детерминированный алгоритм M такой, что $M(1^n, i, t, f_i(x)) = x$ для любых $n \in \mathbb{N}$, $(i,t) \in \operatorname{supp} \mathcal{G}_n$ и $x \in X_i$.

Иногда семейства функций (перестановок) с секретом в смысле определения 12.1 (соответственно, 12.2) называют односторонними или сильно односторонними семействами функций (соответственно, перестановок) с секретом. Можно также определить слабо односторонние семейства функций (перестановок) с секретом, потребовав в определении 12.1 (соответственно, 12.2), чтобы семейство было слабо односторонним. Однако в настоящем курсе слабо односторонние семейства функций (перестановок) с секретом не рассматриваются.

***Замечание* 12.3.** Пусть $(f_i \mid i \in I)$ — семейство функций (перестановок) с секретом относительно семейств распределений вероятностей $(\mathcal{G}_n \mid n \in \mathbb{N})$ и $(\mathcal{X}_i \mid i \in I)$. Выберем полиномиальный вероятностный алгоритм G такой, что случайная величина $G(1^n)$ имеет распределение $\mathcal{G}_n$ для любого $n \in \mathbb{N}$. Пусть также l — полиномиальный параметр на $\mathbb{N}$ такой, что алгоритм G при вычислении на входе 1^n (где $n \in \mathbb{N}$) использует не более $l(n)$ случайных битов. Для каждого $n \in \mathbb{N}$ определим распределение вероятностей $\mathcal{H}_n$ как распределение случайной величины $(\widetilde{i}, \widetilde{r})$ такой, что $(\widetilde{i}, \widetilde{t}) = G(1^n; \widetilde{r})$, где $\widetilde{r} \leftarrow \mathcal{U}(\{0,1\}^{l(n)})$. Тогда $(f_i \mid i \in I)$ является семейством функций (соответственно, перестановок) с секретом относительно семейств распределений вероятностей $(\mathcal{H}_n \mid n \in \mathbb{N})$ и $(\mathcal{X}_i \mid i \in I)$. Таким образом, не ограничивая общности, можно считать, что дополнительная информация (trapdoor) r выбирается случайно и равновероятно из множества $\{0,1\}^{l(n)}$ (где l — некоторый полиномиальный параметр на $\mathbb{N}$), а соответствующий индекс i вычисляется по $(1^n, r)$ детерминированным образом за полиномиальное время.

Пример 12.4 (семейство перестановок RSA)**.** Пусть $(\mathcal{Q}_n \mid n \in \mathbb{N})$ — семейство распределений вероятностей на множестве пар вида $(\{p,q\},e)$, где p и q — различные простые числа, а e — целое число, не меньшее 3 и взаимно простое с $\varphi(pq) = |\mathbb{Z}^\star_{pq}| = (p-1)(q-1)$. Предположим, что это семейство полиномиально конструируемо, когда индексы заданы в унарной записи. Для каждого $n \in \mathbb{N}$ определим распределение вероятностей $\mathcal{G}_n$ как распределение случайной величины $((\widetilde{pq},\widetilde{e}),\widetilde{d})$, где $(\{\widetilde{p},\widetilde{q}\},\widetilde{e}) \leftarrow \mathcal{Q}_n$ и $\widetilde{d} = \widetilde{e}^{-1} \bmod \varphi(\widetilde{pq})$. Для определенности здесь и далее запись $(\{\widetilde{p},\widetilde{q}\},\widetilde{e}) \leftarrow \mathcal{Q}_n$ подразумевает, что $\widetilde{p}$ — меньшее, а $\widetilde{q}$ — большее число в $\{\widetilde{p},\widetilde{q}\}$.

Положим $I = \{(pq,e) \mid n \in \mathbb{N},\ (\{p,q\},e) \in \operatorname{supp}\mathcal{Q}_n\}$. Пусть $(N,e) \in I$. Тогда мы полагаем $X_{(N,e)} = \mathbb{Z}^\star_N$ и $\mathcal{X}_{(N,e)} = \mathcal{U}(\mathbb{Z}^\star_N)$. Определим перестановку $\mathrm{RSA}_{N,e}$ множества $\mathbb{Z}^\star_N$ равенством $\mathrm{RSA}_{N,e}(x) = x^e \bmod N$ $(x \in \mathbb{Z}^\star_N)$. Очевидно, что если $((N,e),d) \in \operatorname{supp}\mathcal{G}_n$ $(n \in \mathbb{N})$, то $\mathrm{RSA}^{-1}_{N,e}(x) = x^d \bmod N$ для любого $x \in \mathbb{Z}^\star_N$. Поэтому семейство $(\mathrm{RSA}_i \mid i \in I)$ удовлетворяет условию 2 определения 12.2.

Семейство $(\mathrm{RSA}_i \mid i \in I)$ называется *семейством перестановок RSA* (RSA permutation family). Название семейства происходит от первых букв фамилий авторов статьи [RSA78], в котором оно было определено. Необходимым условием для того, чтобы это семейство удовлетворяло условию 1 определения 12.2 (и, следовательно, было семейством перестановок с секретом относительно семейств распределений вероятностей $(\mathcal{G}_n \mid n \in \mathbb{N})$ и $(\mathcal{X}_i \mid i \in I)$), является трудность задачи факторизации чисел $\widetilde{pq}$ при известном $\widetilde{e}$, где $(\{\widetilde{p},\widetilde{q}\},\widetilde{e}) \leftarrow \mathcal{Q}_n$. Это условие формализуется следующим образом: $\Pr[A(1^n,\widetilde{pq},\widetilde{e}) \in \{\widetilde{p},\widetilde{q}\}] = \mathrm{negl}(n)$ для любого полиномиального вероятностного алгоритма A. Однако неизвестно, является ли данное условие или какое-либо другое стандартное криптографическое предположение достаточным для того, чтобы семейство перестановок RSA при подходящем выборе $(\mathcal{Q}_n \mid n \in \mathbb{N})$ удовлетворяло условию 1 определения 12.2. См. также [Gol04, п. 2.4.3.1, 2.4.4.2] [Lub96, лекция 15].

Пример 12.5 (семейство функций Рабина)**.** Число N называется *числом Блюма* (Blum integer), если $N = pq$, где p и q — различные простые числа, причем $p \equiv q \equiv 3 \pmod 4$. Для простоты мы рассмотрим семейство функций Рабина лишь в случае, когда модули являются числами Блюма.

Пусть $(\mathcal{P}_n \mid n \in \mathbb{N})$ — семейство распределений вероятностей на множестве двухэлементных множеств простых чисел, сравнимых с 3 по модулю 4. Предположим, что это семейство полиномиально конструируемо, когда индексы заданы в унарной записи. Для каждого $n \in \mathbb{N}$ определим распределение вероятностей $\mathcal{G}_n$ как распределение случайной величины $\widetilde{pq},\{\widetilde{p},\widetilde{q}\})$, где $\{\widetilde{p},\widetilde{q}\} \leftarrow \mathcal{P}_n$. Для определенности здесь и далее запись $\{\widetilde{p},\widetilde{q}\} \leftarrow \mathcal{P}_n$ подразумевает, что $\widetilde{p}$ — меньшее, а $\widetilde{q}$ — большее число в $\{\widetilde{p},\widetilde{q}\}$ (как в примере 12.4).

Положим $I = \{pq \mid n \in \mathbb{N},\ \{p,q\} \in \operatorname{supp}\mathcal{P}_n\}$. Пусть $N \in I$. Тогда N — произведение двух различных простых чисел, сравнимых с 3 по модулю 4. Определим функцию $\mathrm{Rabin}_N \colon \mathbb{Z}^\star_N \to \mathbb{Z}^\star_N$ равенством $\mathrm{Rabin}_N(x) = x^2 \bmod N$ $(x \in \mathbb{Z}^\star_N)$. Семейство $(\mathrm{Rabin}_i \mid i \in I)$ называется *семейством функций Рабина* (Rabin function family). Основное свойство этого семейства состоит в том, что оно является семейством функций с секретом в предположении трудности задачи факторизации чисел $\widetilde{pq}$, где $\{\widetilde{p},\widetilde{q}\} \leftarrow \mathcal{P}_n$.

Напомним, что число $z \in \mathbb{Z}$, взаимно простое с числом $m \in \mathbb{N} \setminus \{0\}$, называется *квадратичным вычетом* (quadratic residue) по модулю m, если $z \bmod m$ является квадратом некоторого элемента группы $\mathbb{Z}^\star_m$, и *квадратичным невычетом* (quadratic nonresidue) по модулю m в противном случае. Таким образом, квадратичным (не)вычетом по модулю m может быть только целое число, взаимно простое с m.

Теорема 12.6 (см. также [Rab79], [Gol04, п. 2.4.3.2, 2.4.4.2, подразд. A.2.2], [Lub96, теорема 15.1])**.** *Предположим, что* $\Pr[A(1^n,\widetilde{pq}) \in \{\widetilde{p},\widetilde{q}\}] = \mathrm{negl}(n)$ *для любого полиномиального вероятностного алгоритма* A*, где* $\{\widetilde{p},\widetilde{q}\} \leftarrow \mathcal{P}_n$*. Тогда* $(\mathrm{Rabin}_i \mid i \in I)$ *является семейством функций с секретом относительно семейств распределений вероятностей* $(\mathcal{G}_n \mid n \in \mathbb{N})$ *и* $(\mathcal{U}(\mathbb{Z}^\star_i) \mid i \in I)$*.*

Доказательство. Полиномиальная вычислимость семейства $(\mathrm{Rabin}_i \mid i \in I)$ очевидна. Пусть теперь A — произвольный полиномиальный вероятностный алгоритм. Выберем полиномиальный вероятностный алгоритм B, работающий на произвольном входе $(1^n,N)$, где $n \in \mathbb{N}$ и $N = pq$ при $\{p,q\} \in \operatorname{supp}\mathcal{P}_n$, следующим образом:

1. Выбрать $x \in_{\mathcal{U}} \mathbb{Z}^\star_N$.
2. Вычислить $y \leftarrow A(1^n,N,\mathrm{Rabin}_N(x))$.
3. Если $y \in \mathrm{Rabin}^{-1}_N(\mathrm{Rabin}_N(x)) \setminus \{x,-x \bmod N\}$, то возвратить НОД$(x-y,N)$. (Согласно замечанию 10.5, в этом случае НОД$(x-y,N) \in \{p,q\}$, так как $x^2 \bmod N = \mathrm{Rabin}_N(x) = y^2 \bmod N$.)

Пусть $n \in \mathbb{N}$ и $N = pq$ при $\{p, q\} \in \operatorname{supp} \mathcal{P}_n$. Обозначим через Z множество всех элементов группы $\mathbb{Z}_N^\star$, являющихся квадратичными вычетами по модулю N. Для каждого $z \in Z$ положим также $\sqrt{z} = \{x \in \mathbb{Z}_N^\star \mid x^2 \bmod N = z\}$, т. е. $\sqrt{z}$ — это множество всех квадратных корней из z в группе $\mathbb{Z}_N^\star$. Из китайской теоремы об остатках (см. доказательство леммы 10.6) следует, что $|\sqrt{z}| = 4$ для любого $z \in Z$ ($\sqrt{z}$ является смежным классом по подгруппе $\sqrt{1}$, состоящей из четырех элементов). Пусть также $\widetilde{x}_0 \in_{\mathcal{U}} \mathbb{Z}_N^\star$ и $\widetilde{y}_0 = A(1^n, N, \operatorname{Rabin}_N(\widetilde{x}_0))$.

Выберем произвольный элемент $z \in Z$ такой, что $\Pr[\widetilde{y}_0 \in \sqrt{z}] \neq 0$. Тогда легко видеть, что при условии $\widetilde{x}_0, \widetilde{y}_0 \in \sqrt{z}$ случайные величины $\widetilde{x}_0$ и $\widetilde{y}_0$ независимы (так как $\widetilde{y}_0 = A(1^n, N, z)$), причем $\widetilde{x}_0$ распределена равномерно на $\sqrt{z}$. Поэтому

$$\Pr[\widetilde{y}_0 \notin \{\widetilde{x}_0, -\widetilde{x}_0 \bmod N\} \mid \widetilde{x}_0, \widetilde{y}_0 \in \sqrt{z}] = \Pr[\widetilde{x}_0 \notin \{\widetilde{y}_0, -\widetilde{y}_0 \bmod N\} \mid \widetilde{x}_0, \widetilde{y}_0 \in \sqrt{z}] = \frac{|\sqrt{z}| - 2}{|\sqrt{z}|} = \frac{1}{2}.$$

Следовательно,

$$\begin{aligned}
\Pr[B(1^n, N) \in \{p, q\}] &= \Pr[\widetilde{y}_0 \in \operatorname{Rabin}_N^{-1}(\operatorname{Rabin}_N(\widetilde{x}_0)) \setminus \{\widetilde{x}_0, -\widetilde{x}_0 \bmod N\}] \\
&= \sum_{z \in Z} \Pr[\widetilde{y}_0 \notin \{\widetilde{x}_0, -\widetilde{x}_0 \bmod N\},\ \widetilde{x}_0, \widetilde{y}_0 \in \sqrt{z}] \\
&= \sum_{z \in Z \mid \Pr[\widetilde{y}_0 \in \sqrt{z}] \neq 0} \Pr[\widetilde{y}_0 \notin \{\widetilde{x}_0, -\widetilde{x}_0 \bmod N\} \mid \widetilde{x}_0, \widetilde{y}_0 \in \sqrt{z}] \Pr[\widetilde{x}_0, \widetilde{y}_0 \in \sqrt{z}] \\
\frac{1}{2} \sum_{z \in Z \mid \Pr[\widetilde{y}_0 \in \sqrt{z}] \neq 0} &\Pr[\widetilde{x}_0, \widetilde{y}_0 \in \sqrt{z}] = \frac{1}{2} \Pr[\widetilde{y}_0 \in \operatorname{Rabin}_N^{-1}(\operatorname{Rabin}_N(\widetilde{x}_0))]
\end{aligned}$$

Это показывает, что если $\{\widetilde{p}, \widetilde{q}\} \leftarrow \mathcal{P}_n$, $\widetilde{N} = \widetilde{p}\widetilde{q}$ и $\widetilde{x} \in_{\mathcal{U}} \mathbb{Z}_{\widetilde{N}}^\star$, то

$$\Pr[A(1^n, \widetilde{N}, \operatorname{Rabin}_{\widetilde{N}}(\widetilde{x})) \in \operatorname{Rabin}_{\widetilde{N}}^{-1}(\operatorname{Rabin}_{\widetilde{N}}(\widetilde{x}))] = 2 \Pr[B(1^n, \widetilde{N}) \in \{\widetilde{p}, \widetilde{q}\}] = \operatorname{negl}(n).$$

Таким образом, семейство $(\operatorname{Rabin}_i \mid i \in I)$ удовлетворяет условию 1 определения 12.1.

Осталось построить полиномиальный детерминированный алгоритм M, удовлетворяющий условию 2 определения 12.1. В качестве такого алгоритма можно взять алгоритм, который на произвольном входе $(1^n, N, \{p, q\}, z)$, где $n \in \mathbb{N}$, $\{p, q\} \in \operatorname{supp} \mathcal{P}_n$, $N = pq$ и $z = y^2 \bmod N$ при некотором $y \in \mathbb{Z}_N^\star$, возвращает $x \in \mathbb{Z}_N^\star$, удовлетворяющее сравнениям $x \equiv z^{(p+1)/4} \pmod p$ и $x \equiv z^{(q+1)/4} \pmod q$ (напомним, что $p \equiv q \equiv 3 \pmod 4$). Такой элемент x существует и единственен ввиду китайской теоремы об остатках (см. доказательство леммы 10.6). Легко также видеть, что $x^2 \equiv z^{(p+1)/2} \equiv y^{p+1} \equiv y^2 \equiv z \pmod p$ (так как $y^{p-1} \equiv 1 \pmod p$), $x^2 \equiv z^{(q+1)/2} \equiv y^{q+1} \equiv y^2 \equiv z \pmod q$ (так как $y^{q-1} \equiv 1 \pmod q$) и, следовательно, $\operatorname{Rabin}_N(x) = x^2 \bmod N = z$. □

В случае, когда модули N являются произведениями двух произвольных различных нечетных простых чисел, семейство функций Рабина также является семейством функций с секретом. Доказательство условия 1 определения 12.1 в этом случае такое же, как приведенное выше. Однако доказательство условия 2 определения 12.1 становится сложнее, чем в теореме 12.6. Дополнительная информация, позволяющая инвертировать функцию Rabin_N ($N = pq$), должна в этом случае содержать не только $\{p, q\}$, но и некоторый элемент $a \in \mathbb{Z}_N^\star$, являющийся квадратичным невычетом как по модулю p, так и по модулю q. Доказательство условия 2 определения 12.1 использует полиномиальный детерминированный алгоритм, извлекающий квадратные корни в группе $\mathbb{Z}_r^\star$ (r — произвольное простое число) с использованием какого-либо квадратичного невычета по модулю r (алгоритм Тонелли—Шенкса; см., например, [Sho08, подразд. 12.5.1], [BS96, разд. 7.1]).

Если ограничить функцию Rabin_N на множество всех элементов нечетного порядка из группы $\mathbb{Z}_N^\star$, то она становится перестановкой этого множества. Кроме того, такие ограничения этих функций образуют семейство перестановок с секретом. При этом для инвертирования ограничения функции Rabin_N ($N = pq$) на указанное множество за полиномиальное время достаточно знать лишь $\{p, q\}$. В частности, для этого нет необходимости предполагать, что N — число Блюма или знать некоторый квадратичный невычет по обоим модулям p и q.

Пример 12.7 (семейство функций Рабина, ограниченных на множества элементов нечетного порядка)**.** Пусть $(\mathcal{P}_n \mid n \in \mathbb{N})$ — семейство распределений вероятностей на множестве двухэлементных множеств нечетных простых чисел. Предположим, что это семейство полиномиально конструируемо, когда индексы заданы в унарной записи. Для каждого $n \in \mathbb{N}$ определим распределение вероятностей $\mathcal{G}_n$ как распределение случайной величины $(\widetilde{p}\widetilde{q}, \{\widetilde{p}, \widetilde{q}\})$, где $\{\widetilde{p}, \widetilde{q}\} \leftarrow \mathcal{P}_n$. Для определенности

здесь и далее запись $\{\widetilde{p},\widetilde{q}\} \leftarrow \mathcal{P}_n$ подразумевает, что $\widetilde{p}$ — меньшее, а $\widetilde{q}$ — большее число в $\{\widetilde{p},\widetilde{q}\}$ (как в примерах 12.4 и 12.5).

Как и в примере 12.5, положим $I = \{pq \mid n \in \mathbb{N},\ \{p,q\} \in \operatorname{supp}\mathcal{P}_n\}$. Пусть $N \in I$. Тогда N — произведение двух различных нечетных простых чисел. Обозначим через Y_N множество всех элементов нечетного порядка из группы $\mathbb{Z}_N^\star$ (являющееся подгруппой этой группы). Очевидно, что если $\widetilde{g} \in_{\mathcal{U}} \mathbb{Z}_N^\star$, то случайная величина $\widetilde{g}^{2^{\lfloor \log_2 N \rfloor}}$ распределена равномерно на Y_N. Поэтому семейство распределений вероятностей $(\mathcal{U}(Y_i) \mid i \in I)$ полиномиально конструируемо. Определим перестановку Rabin'_N множества Y_N равенством $\operatorname{Rabin}'_N(y) = y^2 \bmod N$ $(y \in Y_N)$.

Теорема 12.8 (см. также [Gol04, п. 2.4.3.3, подразд. A.2.2]). *Предположим, что* $\Pr[A(1^n, \widetilde{pq}) \in \{\widetilde{p},\widetilde{q}\}] = \operatorname{negl}(n)$ *для любого полиномиального вероятностного алгоритма A, где* $\{\widetilde{p},\widetilde{q}\} \leftarrow \mathcal{P}_n$. *Тогда* $(\operatorname{Rabin}'_i \mid i \in I)$ *является семейством перестановок с секретом относительно семейств распределений вероятностей* $(\mathcal{G}_n \mid n \in \mathbb{N})$ *и* $(\mathcal{U}(Y_i) \mid i \in I)$.

Доказательство. Полиномиальная вычислимость семейства $(\operatorname{Rabin}'_i \mid i \in I)$ очевидна. Пусть теперь A — произвольный полиномиальный вероятностный алгоритм. Выберем полиномиальный вероятностный алгоритм B, работающий на произвольном входе $(1^n, N)$, где $n \in \mathbb{N}$ и $N = pq$ при $\{p,q\} \in \operatorname{supp}\mathcal{P}_n$, следующим образом:

1. Выбрать $g \in_{\mathcal{U}} \mathbb{Z}_N^\star$.
2. Для k от 0 до $\lfloor \log_2 N \rfloor - 1$:
   - вычислить $g_k = g^{2^k} \bmod N$ и $h_k \leftarrow A(1^n, N, g_k^2 \bmod N)$;
   - если $h_k \in \mathbb{Z}_N^\star \setminus \{g_k, -g_k \bmod N\}$ и $h_k^2 \equiv g_k^2 \pmod N$, то возвратить $\operatorname{НОД}(g_k - h_k, N)$ и закончить работу. (Согласно замечанию 10.5, в этом случае $\operatorname{НОД}(g_k - h_k, N) \in \{p,q\}$.)

Пусть $n \in \mathbb{N}$, $N = pq$ при $\{p,q\} \in \operatorname{supp}\mathcal{P}_n$, $g \in \mathbb{Z}_N^\star$ и $g_k = g^{2^k} \bmod N$ при $k \in \{0, \ldots, \lfloor \log_2 N \rfloor\}$, как в описании алгоритма B. Обозначим через E_N множество всех $x \in \mathbb{Z}_N^\star$ таких, что $\operatorname{ord} x$ четен и $-1 \bmod N \notin \langle x \rangle$. Предположим, что $g \in E_N$. Пусть $l = (\log_2 \operatorname{ord}(gY_N)) - 1$ (здесь порядок берется в 2-группе $\mathbb{Z}_N^\star / Y_N$, поэтому он является степенью двойки). Тогда $0 \leqslant l \leqslant \lfloor \log_2 N \rfloor - 1$ (так как $g_0 = g \notin Y_N$ и $g_{\lfloor \log_2 N \rfloor} \in Y_N$), $g_l \notin Y_N$ и $g_{l+1} = g^{\operatorname{ord}(gY_N)} \bmod N \in Y_N$. Предположим также, что $A(1^n, N, g_{l+1}) = h_l = \operatorname{Rabin}'^{-1}_N(g_{l+1})$. Тогда $h_l \neq g_l$ (так как $h_l \in Y_N$ и $g_l \notin Y_N$), $h_l \neq -g_l \bmod N$ (так как $h_l, g_l \in \langle g \rangle$ и $-1 \bmod N \notin \langle g \rangle$) и $h_l^2 \equiv g_{l+1} \equiv g_l^2 \pmod N$. Следовательно, в этом случае алгоритм B на входе $(1^n, N)$ возвращает p или q.

Хорошо известно, что $\mathbb{Z}_N^\star$ является внутренним прямым произведением своей единственной силовской 2-подгруппы S_N и подгруппы Y_N. Положим $D_N = E_N \cap S_N = \{t \in S_N \setminus \{1\} \mid -1 \bmod N \notin \langle t \rangle\}$. Пусть $g = sy \bmod N$, где $s \in S_N$ и $y \in Y_N$, — произвольный элемент группы $\mathbb{Z}_N^*$. Тогда $\operatorname{ord}(gY_N) = \operatorname{ord} s$. Кроме того, если $h \in Y_N$, то $g^{\operatorname{ord}(gY_N)} \bmod N = h$ в том и только том случае, когда $y^{\operatorname{ord} s} \bmod N = h$, причем для каждого $s \in S_N$ элемент $y \in Y_N$, удовлетворяющий последнему равенству, существует и единственен (так как $\operatorname{ord} s$ является степенью двойки, а $|Y_N|$ нечетно). Легко также видеть, что $g \in E_N$ тогда и только тогда, когда $s \in D_N$ (здесь используется то, что $\langle s \rangle$ — единственная силовская 2-подгруппа группы $\langle g \rangle$).

Пусть теперь $\widetilde{s} \in_{\mathcal{U}} S_N$, $\widetilde{y}_0 \in_{\mathcal{U}} Y_N$ и $\widetilde{g} = \widetilde{s}\widetilde{y}_0 \bmod N$. Тогда случайная величина $\widetilde{g}$ распределена равномерно на $\mathbb{Z}_N^\star$. Из доказанного в предыдущем абзаце вытекает, что

$$\begin{aligned}\Pr[\widetilde{g}^{\operatorname{ord}(\widetilde{g}Y_N)} \bmod N = h,\ \widetilde{g} \in E_N] &= \Pr[\widetilde{y}_0^{\operatorname{ord}\widetilde{s}} \bmod N = h,\ \widetilde{s} \in D_N] \\ &= \frac{|D_N|}{|S_N||Y_N|} = \frac{\Pr[\widetilde{s} \in D_N]}{|Y_N|} = \frac{\Pr[\widetilde{g} \in E_N]}{|Y_N|}\end{aligned}$$

и, следовательно, $\Pr[\widetilde{g}^{\operatorname{ord}(\widetilde{g}Y_N)} \bmod N = h \mid \widetilde{g} \in E_N] = 1/|Y_N|$ для любого $h \in Y_N$. Это показывает, что при условии $\widetilde{g} \in E_N$ случайная величина $\widetilde{g}^{\operatorname{ord}(\widetilde{g}Y_N)} \bmod N$ распределена равномерно на Y_N, т. е. так же, как $\operatorname{Rabin}'_N(\widetilde{y}_0)$. Из доказанного выше и из неравенства $\Pr[\widetilde{g} \in E_N] \geqslant 1/2$ (см. лемму 10.6) вытекает, что

$$\begin{aligned}&\frac{1}{2}\Pr[A(1^n, N, \operatorname{Rabin}'_N(\widetilde{y}_0)) = \widetilde{y}_0] \\ &\quad\leqslant \Pr[A(1^n, N, \widetilde{g}^{\operatorname{ord}(\widetilde{g}Y_N)} \bmod N) \in \operatorname{Rabin}'^{-1}_N(g^{\operatorname{ord}(\widetilde{g}Y_N)} \bmod N) \mid \widetilde{g} \in E_N] \Pr[\widetilde{g} \in E_N] \\ &\quad= \Pr[A(1^n, N, \widetilde{g}^{\operatorname{ord}(\widetilde{g}Y_N)} \bmod N) \in \operatorname{Rabin}'^{-1}_N(g^{\operatorname{ord}(\widetilde{g}Y_N)} \bmod N),\ \widetilde{g} \in E_N] \leqslant \Pr[B(1^n, N) \in \{p,q\}].\end{aligned}$$

Поэтому если $\{\widetilde{p},\widetilde{q}\} \leftarrow \mathcal{P}_n$, $\widetilde{N} = \widetilde{p}\widetilde{q}$ и $\widetilde{y} \in_{\mathcal{U}} Y_{\widetilde{N}}$, то

$$\Pr[A(1^n, \widetilde{N}, \mathrm{Rabin}'_{\widetilde{N}}(\widetilde{y})) = \widetilde{y}] \leqslant 2\Pr[B(1^n, \widetilde{N}) \in \{\widetilde{p},\widetilde{q}\}] = \mathrm{negl}(n).$$

Таким образом, семейство $(\mathrm{Rabin}'_i \mid i \in I)$ удовлетворяет условию 1 определения 12.2.

Покажем теперь, что семейство $(\mathrm{Rabin}'_i \mid i \in I)$ удовлетворяет условию 2 определения 12.2. Для этого достаточно заметить, что $\mathrm{Rabin}'^{-1}_N(y) = y^d \bmod N$ для любого $y \in Y_N$, где $d = 2^{-1} \bmod |Y_N|$. Здесь $|Y_N| = |\mathbb{Z}^\star_N|/2^m$, где $|\mathbb{Z}^\star_N| = \varphi(N) = (p-1)(q-1)$, а 2^m — наибольшая степень двойки, делящая $|\mathbb{Z}^\star_N|$. Поэтому число $|Y_N|$ вычислимо за полиномиальное время по $\{p,q\}$. □

Недостатком семейства $(\mathrm{Rabin}'_i \mid i \in I)$ является то, что неизвестны полиномиальные алгоритмы, проверяющие по произвольным $N \in I$ и $x \in \mathbb{Z}^\star_N$, входит ли x в Y_N.

13. Элементы теории Шеннона систем секретной связи

В работе [Sha49] Шеннон разработал математическую теорию систем секретной связи. Система секретной связи представляет собой одну из математических моделей криптосистемы с секретным ключом. Теория Шеннона носит чисто теоретико-информационный характер; она не интересуется вопросами вычислимости каких-либо функций. В частности, на вычислительные ресурсы противника не накладывается никаких ограничений.

Определение 13.1 (система секретной связи)**.** *Системой секретной связи* (secrecy system) называется пара $((e_i \mid i \in K), \mathcal{K})$, где K — непустое конечное множество, $\mathcal{K}$ — распределение вероятностей на множестве K, а e_i для каждого $i \in K$ — инъективное отображение из непустого конечного множества M в конечное множество C. Множества K, M и C называются *пространствами ключей* (key space), *сообщений* (message space) и *криптограмм* (cryptogram space) соответственно.

Отметим, что в русском переводе работы [Sha49] «secrecy system» переведено как «секретная система», что, по мнению автора настоящего курса, неудачно.

Пусть $S = ((e_i \mid i \in K), \mathcal{K})$ — система секретной связи с пространством ключей K, пространством сообщений M и пространством криптограмм C. Схема применения этой системы секретной связи выглядит следующим образом. Имеется три действующих лица: отправитель, получатель и противник. Отправитель и получатель выбирают общий секретный ключ $k \leftarrow \mathcal{K}$. Чтобы передать получателю сообщение $m \in M$, отправитель вычисляет криптограмму $c = e_k(m)$ и посылает эту криптограмму получателю, который может по ней однозначно восстановить m ввиду инъективности отображения e_k. Обычно считается, что противник в данной теории может проводить лишь атаку, заключающуюся в перехвате криптограммы $e_k(m)$ при $k \leftarrow \mathcal{K}$ и $m \leftarrow \mathcal{M}$, где $\mathcal{M}$ — некоторое распределение вероятностей на M, называемое *априорным распределением вероятностей* (a priori probability distribution) на пространстве сообщений. На основе этой атаки противник пытается получить какую-либо информацию о сообщении m или ключе k. Естественно, предполагается, что $(e_i \mid i \in K)$, $\mathcal{K}$ и $\mathcal{M}$ известны противнику. После проведения вышеуказанной атаки с точки зрения противника передаваемое сообщение m выбрано случайно в соответствии с *апостериорным распределением вероятностей* (a posteriori probability distribution) на пространстве сообщений, которое определяется как условное распределение случайной величины $\widetilde{m} \leftarrow \mathcal{M}$ при условии, что $e_{\widetilde{k}}(\widetilde{m})$ равно перехваченной криптограмме (где $\widetilde{k} \leftarrow \mathcal{K}$). Аналогично, с точки зрения противника секретный ключ k до проведения указанной атаки выбран случайно в соответствии с распределением $\mathcal{K}$ (называемом *априорным распределением вероятностей* на пространстве ключей), а после проведения этой атаки — в соответствии с условным распределением случайной величины $\widetilde{k} \leftarrow \mathcal{K}$ при условии, что $e_{\widetilde{k}}(\widetilde{m})$ равно перехваченной криптограмме (где $\widetilde{m} \leftarrow \mathcal{M}$). Последнее условное распределение называется *апостериорным распределением вероятностей* на пространстве ключей.

Часто предполагается, что все отображения $e_i \colon M \to C$ не только инъективны, но и сюръективны. В этом случае система секретной связи S называется *замкнутой* (closed).

Определение 13.2 (совершенная система секретной связи)**.** Пусть $\mathcal{M}$ — некоторое априорное распределение вероятностей на пространстве сообщений M. Пусть также $\widetilde{k} \leftarrow \mathcal{K}$ и $\widetilde{m} \leftarrow \mathcal{M}$. Тогда система секретной связи S называется *совершенной* (perfect) относительно $\mathcal{M}$, если для любого сообщения $m \in M$ и любой криптограммы $c \in \operatorname{supp} e_{\widetilde{k}}(\widetilde{m})$ справедливо равенство $\Pr[\widetilde{m} = m] = \Pr[\widetilde{m} = m \mid e_{\widetilde{k}}(\widetilde{m}) = c]$. Другими словами, условие совершенности означает, что при любой перехваченной криптограмме c апостериорное распределение вероятностей на пространстве сообщений совпадает с априорным.

Неформальный смысл условия совершенности состоит в том, что перехват противником криптограммы не дает ему никакой дополнительной информации о зашифрованном сообщении.

***Замечание* 13.3.** Непосредственно проверяется, что система секретной связи S совершенна относительно $\mathcal{M}$ тогда и только тогда, когда случайные величины $\widetilde{m}$ и $e_{\widetilde{k}}(\widetilde{m})$ независимы.

***Замечание* 13.4.** Пусть система секретной связи S совершенна относительно $\mathcal{M}$. Тогда $\Pr[e_{\widetilde{k}}(m) = c] = \Pr[e_{\widetilde{k}}(\widetilde{m}) = c \,|\, \widetilde{m} = m] = \Pr[e_{\widetilde{k}}(\widetilde{m}) = c]$ при всех $m \in \operatorname{supp}\mathcal{M}$ и $c \in C$ ввиду замечания 13.3. Поэтому случайные величины $e_{\widetilde{k}}(m)$ при всех $m \in \operatorname{supp}\mathcal{M}$ распределены одинаково, а именно, так же, как и $e_{\widetilde{k}}(\widetilde{m})$.

Обратно, пусть случайные величины $e_{\widetilde{k}}(m)$ при всех $m \in \operatorname{supp}\mathcal{M}$ распределены одинаково. Тогда легко видеть, что каждая из этих случайных величин распределена так же, как и $e_{\widetilde{k}}(\widetilde{m})$. Следовательно, $\Pr[e_{\widetilde{k}}(\widetilde{m}) = c \,|\, \widetilde{m} = m] = \Pr[e_{\widetilde{k}}(m) = c] = \Pr[e_{\widetilde{k}}(\widetilde{m}) = c]$ для любых $m \in \operatorname{supp}\mathcal{M}$ и $c \in C$. Поэтому $\widetilde{m}$ и $e_{\widetilde{k}}(\widetilde{m})$ независимы и система секретной связи S совершенна относительно $\mathcal{M}$ ввиду замечания 13.3.

Таким образом, система секретной связи S совершенна относительно $\mathcal{M}$ тогда и только тогда, когда случайные величины $e_{\widetilde{k}}(m)$ при всех $m \in \operatorname{supp}\mathcal{M}$ распределены одинаково. Это показывает, что совершенность системы секретной связи относительно $\mathcal{M}$ зависит лишь от $\operatorname{supp}\mathcal{M}$. В частности, если потребовать, чтобы $\operatorname{supp}\mathcal{M} = M$, то условие совершенности относительно $\mathcal{M}$ для фиксированной системы секретной связи либо выполняется при любом таком распределении $\mathcal{M}$, либо не выполняется ни при каком.

Предложение 13.5. *Если система секретной связи S совершенна относительно $\mathcal{M}$, то* $|\operatorname{supp}\mathcal{M}| \leqslant |\operatorname{supp}\mathcal{K}|$.

Доказательство. Достаточно построить какое-либо инъективное отображение из $\operatorname{supp}\mathcal{M}$ в $\operatorname{supp}\mathcal{K}$. Выберем какой-либо элемент $c \in \operatorname{supp} e_{\widetilde{k}}(\widetilde{m})$. Тогда, согласно замечанию 13.4, $\Pr[e_{\widetilde{k}}(m) = c] = \Pr[e_{\widetilde{k}}(\widetilde{m}) = c] \neq 0$ для любого $m \in \operatorname{supp}\mathcal{M}$. Следовательно, каждому сообщению $m \in \operatorname{supp}\mathcal{M}$ можно поставить в соответствие ключ $k(m) \in \operatorname{supp}\mathcal{K}$ такой, что $e_{k(m)}(m) = c$. Из инъективности отображений e_i вытекает, что отображение $m \mapsto k(m)$ ($m \in \operatorname{supp}\mathcal{M}$) инъективно. □

Таким образом, ввиду предложения 13.5 совершенные системы секретной связи обладают следующим недостатком (с практической точки зрения): говоря неформально, число ключей должно быть не меньшим, чем число сообщений.

Предположим теперь, что рассматриваемая система секретной связи S замкнута и что $e_i \neq e_j$ при $i \neq j$ ($i, j \in K$).

Определение 13.6 (чистая система секретной связи)**.** Система секретной связи S называется *чистой* (pure), если $\mathcal{K}$ — равномерное распределение на множестве K и для любых $i, j, k \in K$ существует индекс $l \in K$ такой, что $e_i e_j^{-1} e_k = e_l$. Здесь и далее в этом разделе под произведением отображений понимается их композиция, взятая справа налево.

Пусть рассматриваемая система секретной связи S чиста. Тогда непосредственно проверяется, что $G = \{e_i^{-1} e_j \,|\, i, j \in K\}$ является подгруппой группы всех перестановок множества M, причем

$$G = \{e_i^{-1} e_k \,|\, i \in K\} = \{e_k^{-1} e_j \,|\, j \in K\}$$

для любого $k \in K$. Следовательно, $|G| = |K|$. Группа G естественно действует на множестве M; орбиты этого действия называются *остаточными классами сообщений* (message residue classes). Каждому остаточному классу сообщений X ставится в соответствие множество $E(X) = \{e_i(x) \,|\, i \in K,\, x \in X\} \subseteq C$. Тогда легко видеть, что C является объединением попарно не пересекающихся множеств $E(X)$, причем различным остаточным классам сообщений X соответствуют различные множества $E(X)$. Эти множества $E(X)$ называются *остаточными классами криптограмм* (cryptogram residue classes).

***Замечание* 13.7.** Выберем произвольный остаточный класс сообщений X. Тогда $E(X) = e_k(X)$ для любого $k \in K$, так как $e_i(x) = e_k(e_k^{-1} e_i(x))$, где $e_k^{-1} e_i(x) \in X$ ($i \in K$, $x \in X$). Поэтому $|E(X)| = |X|$. В частности, $|E(X)|$ делит $|K|$ ввиду того, что $|X|$ делит $|G| = |K|$.

В дальнейшем через $[m]$ будет обозначаться остаточный класс сообщений, содержащий сообщение $m \in M$.

Предложение 13.8. *Предположим, что система секретной связи S чиста. Пусть $\widetilde{k} \leftarrow \mathcal{K}$. Тогда для любых $m \in M$ и $c \in C$*

$$\Pr[e_{\widetilde{k}}(m) = c] = \begin{cases} 1/|[m]|, & \text{если } c \in E([m]); \\ 0 & \text{в противном случае.} \end{cases}$$

Доказательство. Положим для краткости $K_{m,c} = \{i \in K \mid e_i(m) = c\}$. Если $c \notin E([m])$, то, разумеется $K_{m,c} = \varnothing$ и $\Pr[e_{\widetilde{k}}(m) = c] = |K_{m,c}|/|K| = 0$.

Пусть теперь $c \in E([m])$. Прежде всего заметим, что $K_{m,c} \neq \varnothing$. Действительно, если i — произвольный ключ из K, то $e_i^{-1}(c) \in [m]$ и, следовательно, $e_i^{-1}(c) = e_j^{-1}e_k(m)$ для некоторых $j, k \in K$. Поэтому $c = e_i e_j^{-1} e_k(m) = e_l(m)$ при некотором $l \in K$.

Выберем какой-либо ключ $l \in K_{m,c}$. Обозначим через $\mathrm{St}_G(m)$ стабилизатор m при естественном действии G на M, т. е. $\{g \in G \mid g(m) = m\}$. Каждому $g \in \mathrm{St}_G(m)$ поставим в соответствие единственный ключ $k(g)$ такой, что $e_l g = e_{k(g)}$. Тогда непосредственно проверяется, что $g \mapsto k(g)$ — биекция $\mathrm{St}_G(m)$ на $K_{m,c}$. Следовательно, $|K_{m,c}| = |\mathrm{St}_G(m)| = |G|/|[m]|$ и $\Pr[e_{\widetilde{k}}(m) = c] = |K_{m,c}|/|K| = 1/|[m]|$, так как $|G| = |K|$. □

Пусть X — произвольный остаточный класс сообщений. Тогда можно определить систему секретной связи $S_X = ((e_i|_X \mid i \in K), \mathcal{K})$. Пространством сообщений этой системы секретной связи является X, а пространством криптограмм — $E(X)$. Из предложения 13.8 и замечания 13.4 вытекает, что система секретной связи S_X совершенна относительно любого априорного распределения вероятностей на своем пространстве сообщений X.

Таким образом, рассматриваемая система секретной связи S в некотором смысле разлагается в «дизъюнктное объединение» совершенных (относительно любых априорных распределений вероятностей на своих пространствах сообщений) систем секретной связи S_X по всем остаточным классам сообщений X. Говоря неформально, вся дополнительная информация о зашифрованном сообщении, которую получает противник в результате перехвата криптограммы в чистой системе секретной связи, состоит в том, к какому остаточному классу принадлежит это сообщение.

Пример 13.9 (система секретной связи Вернама)**.** Пусть $n \in \mathbb{N}$. Тогда система секретной связи $((v_i \mid i \in \{0,1\}^n), \mathcal{U}(\{0,1\}^n))$, где отображение $v_i \colon \{0,1\}^n \to \{0,1\}^n$ определено формулой $v_i(x) = i \oplus x$ $(i, x \in \{0,1\}^n)$, называется *системой секретной связи Вернама* [Ver26]. (В этой системе пространства ключей, сообщений и криптограмм совпадают с $\{0,1\}^n$.) Очевидно, что $i \mapsto v_i$ $(i \in \{0,1\}^n)$ — инъективный гомоморфизм группы $\{0,1\}^n$ (относительно операции $\oplus$) в группу всех перестановок множества $\{0,1\}^n$. Следовательно, система секретной связи Вернама чиста. Кроме того, она имеет только один остаточный класс сообщений (совпадающий с $\{0,1\}^n$). Поэтому система секретной связи Вернама является совершенной относительно любого априорного распределения вероятностей на пространстве сообщений.

Подчеркнем, что система секретной связи является математической моделью криптосистемы с *одноразовым* секретным ключом. Это означает, что если противник перехватил $e_k(m)$ и $e_k(m')$ для некоторых $k \in K$ и $m, m' \in M$, то он, вообще говоря, может получить информацию о (m, m') даже при условии совершенности рассматриваемой системы секретной связи S. Это можно проиллюстрировать на примере системы секретной связи Вернама. Действительно, по $v_k(m) = k \oplus m$ и $v_k(m') = k \oplus m'$, где $k, m, m' \in \{0,1\}^n$ при некотором $n \in \mathbb{N}$ (см. пример 13.9), можно вычислить $v_k(m) \oplus v_k(m') = m \oplus m'$ и тем самым узнать, для каких $i \in \{1, \ldots, n\}$ i-е биты сообщений m и m' различны. Таким образом, для обеспечения конфиденциальности пересылаемых сообщений необходимо шифровать каждое сообщение на заново выбранном секретном ключе.

14. Системы шифрования (криптосистемы)

Модель Шеннона системы шифрования, рассмотренная в предыдущем разделе, имеет некоторые ограничения и недостатки как с теоретической, так и с практической точки зрения. К ним относятся, например, непригодность для формализации понятия системы шифрования с открытым ключом, одноразовость секретного ключа, его большая длина. В настоящем разделе рассматривается модель системы шифрования (как с секретным, так и с открытым ключом), основанная на теории сложности вычислений.

Определение 14.1 (система шифрования; см. также [Gol04, определение 5.1.1])**.** Предположим, что каждому числу $n \in \mathbb{N}$ (играющему роль параметра стойкости) поставлено в соответствие множество $M_n \subseteq \{0,1\}^*$, называемое *пространством сообщений* (message space) или *пространством открытых текстов* (plaintext space) при данном n. *Системой шифрования* (encryption system) называется тройка алгоритмов (G, E, D) таких, что

1) G и E — полиномиальные вероятностные алгоритмы, а D — полиномиальный детерминированный алгоритм;

2) $\operatorname{supp} G(1^n) \subseteq \{0,1\}^* \times \{0,1\}^*$ для любого $n \in \mathbb{N}$;

3) при всех $n \in \mathbb{N}$, $m \in M_n$ и $(e,d) \in \operatorname{supp} G(1^n)$ всегда выполняется равенство $D(1^n, d, E(1^n, e, m)) = m$.

Алгоритмы G, E и D называются алгоритмами *генерации ключей* (key generation), *шифрования* (encryption) и *расшифрования* или *дешифрования* (decryption) соответственно.

В качестве синонимов системы шифрования в литературе употребляются термины *криптосистема* (cryptosystem) и *шифр* (cipher). Отметим также, что некоторые отечественные авторы считают дешифрование не синонимом расшифрования (что естественно с лингвистической точки зрения), а разновидностью взлома системы шифрования.

Схема применения системы шифрования (G, E, D) может выглядеть следующим образом. Пусть выбраны параметр стойкости $n \in \mathbb{N}$ и пара ключей $(e,d) \leftarrow G(1^n)$ (см. условие 2 определения 14.1). Ключ e называется *ключом шифрования* (encryption key) и служит для шифрования сообщений, а ключ d, называемый *ключом расшифрования* или *дешифрования* (decryption key), — для их расшифрования. Чтобы конфиденциально передать сообщение $m \in M_n$, называемое в данном контексте также *открытым текстом* (plaintext), отправитель вычисляет *шифртекст* (ciphertext) $w \leftarrow E(1^n, e, m)$ и посылает w получателю. Получатель, получив шифртекст w от отправителя, может восстановить исходное сообщение m как $D(1^n, d, w)$ согласно условию 3 определения 14.1. Наиболее часто рассматриваются следующие два случая:

- Ключи e и d совпадают и этот общий ключ хранится отправителем и получателем в секрете. В этом случае (G, E, D) называется *системой шифрования с секретным ключом* (private-key encryption system), а $e = d$ — *общим секретным ключом* (common secret key).

- Ключ e публикуется, а ключ d хранится получателем в секрете. В этом случае (G, E, D) называется *системой шифрования с открытым ключом* (public-key encryption system), e — *открытым ключом* (public key), а d — *секретным ключом* (private key). Таким образом, в системе шифрования с открытым ключом шифровать произвольные сообщения может каждый, в том числе и противник. Главным преимуществом систем шифрования с открытым ключом по сравнению с системами шифрования с секретным ключом является отсутствие необходимости предварительной выработки общего секретного ключа отправителя и получателя.

Система шифрования может использовать некоторую дополнительную открытую информацию, генерируемую по 1^n за полиномиальное время и публикуемую для всеобщего доступа. Например, если система шифрования использует вычисления в кольце $\mathbb{Z}_N$, то эта информация содержит N. Если есть такая дополнительная открытая информация, то подразумевается, что она подается на дополнительный вход алгоритмов G, E и D.

Наиболее важным случаем в определении 14.1 является случай, когда пространство сообщений M_n есть $\{0,1\}^*$ для всех $n \in \mathbb{N}$. Если же $M_n = \{0,1\}^{l(n)}$ при любом $n \in \mathbb{N}$, где $l\colon \mathbb{N} \to \mathbb{N} \setminus \{0\}$ — полиномиальный параметр, то система шифрования называется *блоковой* (block) с длиной блока l. Ниже в замечании 14.8 будет показано, как блоковую систему шифрования преобразовать в систему шифрования с $M_n = \{0,1\}^*$.

Существует альтернативная терминология (возможно, более естественная с лингвистической точки зрения), согласно которой система шифрования называется блоковой, если в ней алгоритм шифрования разбивает сообщение (произвольной длины) на блоки и шифрует каждый блок отдельно по одному и тому же алгоритму. Пример такой системы шифрования приводится ниже в замечании 14.8. Однако автор настоящего курса предпочитает пользоваться определением блоковой системы шифрования, идейно аналогичным принятым в двух книгах по математической криптографии, служащих стандартными источниками для ссылок (см. [Gol04, определение 5.3.5] и [Lub96, лекции 12, 15]).

До сих пор ничего не говорилось о стойкости систем шифрования. Напомним, что стойкость системы шифрования, как и любого криптографического протокола, определяется против конкретной угрозы на основе конкретной атаки. В связи с эти перечислим некоторые атаки на произвольную систему шифрования (G, E, D) и угрозы стойкости последней. В описаниях этих атак и угроз через n обозначается параметр стойкости (число из $\mathbb{N}$), а через e и d — ключи шифрования и расшифрования соответственно, полученные как части случайного значения $G(1^n)$. При этом предполагается, что n, e и d не меняются в течение работы противника.

Атаки на систему (G, E, D)**:**

- *Атака с открытой информацией*: противник получает только открытую информацию, относящуюся к атакуемой системе шифрования. Эта информация включает 1^n для системы с секретным ключом и 1^n и e для системы с открытым ключом. Когда атакуется система шифрования с открытым ключом, эта атака называется *атакой с открытым ключом* (key-only attack). Данная атака является самой слабой и всегда доступной для противника.

- *Атака с выбором открытых текстов* (chosen-plaintext attack, CPA): противник получает открытую информацию, относящуюся к атакуемой системе шифрования, после чего имеет доступ к оракулу, который в ответ на произвольный запрос $q \in \{0,1\}^*$ выдает случайное значение $E(1^n, e, q)$. Этот оракул будет обозначаться через $E(1^n, e, \cdot)$. Предполагается, что случайные биты, используемые алгоритмом E, при каждом его выполнении выбираются заново. Если атакуется система шифрования с открытым ключом, то e входит в открытую информацию. Поэтому для систем шифрования с открытым ключом при стандартных предположениях о вычислительных возможностях противника данная атака эквивалентна атаке с открытым ключом.

- *Атака с выбором шифртекстов* (chosen-ciphertext attack, CCA): противник получает открытую информацию, относящуюся к атакуемой системе шифрования, после чего имеет доступ к оракулу, который в ответ на произвольный запрос $w \in \{0,1\}^*$ выдает $D(1^n, d, w)$.

- *Атака с выбором открытых текстов и шифртекстов* является комбинацией двух предыдущих атак. А именно, противник получает открытую информацию, относящуюся к атакуемой системе шифрования, после чего имеет доступ как к оракулу из описания атаки с выбором открытых текстов, так и к оракулу из описания атаки с выбором шифртекстов.

Угрозы стойкости системы (G, E, D)**:**

- *Полное раскрытие* (total breaking) для системы шифрования с открытым ключом: нахождение некоторого ключа d' такого, что $(e, d') \in \operatorname{supp} G(1^n)$. Такой ключ d' позволяет расшифровывать сообщения, зашифрованные с помощью открытого ключа e. Отметим, что d' может отличаться от секретного ключа d, используемого получателем сообщений.

- *Нахождение открытого текста по шифртексту*: по случайному значению $E(1^n, e, m)$ найти m. Здесь m может быть как произвольным элементом пространства сообщений M_n системы шифрования (G, E, D), так и случайным элементом, выбранным относительно некоторого априорного распределения вероятностей $\mathcal{M}_n$ на M_n. В последнем случае предполагается, что семейство распределений вероятностей $(\mathcal{M}_n \mid n \in \mathbb{N})$ полиномиально конструируемо, когда индексы заданы в унарной записи.

- *Различение шифртекстов*: выбрать два сообщения $m_0, m_1 \in M_n$ одинаковой длины, получить случайное значение $E(1^n, e, m_b)$, где $b \in_{\mathcal{U}} \{0,1\}$, и найти b. При этом требуется, чтобы m_0 и m_1 были отличны от сообщений, которые противник получает или выбирает при осуществлении атаки (если таковые имеются). Чтобы избежать патологических случаев, мы рассматриваем эту угрозу лишь в случае, когда $M_n \supseteq \{0,1\}^{l(n)}$ при всех $n \in \mathbb{N}$, где $l\colon \mathbb{N} \to \mathbb{N} \setminus \{0\}$ — некоторый полиномиальный параметр. Очевидно, что данную угрозу заведомо можно осуществить с вероятностью $1/2$. Поэтому стойкость против различения шифртекстов означает, что любой полиномиально ограниченный противник может осуществить эту угрозу лишь с вероятностью не более $1/2 + \operatorname{negl}(n)$.

  Эквивалентным образом стойкость против различения шифртекстов можно определить следующим образом. Для бита b и противника A обозначим через $p_b^A(n)$ вероятность того, что A, осуществляя рассматриваемую угрозу, возвратил 1, получив случайное значение $E(1^n, e, m_b)$, где

m_0 и m_1 — сообщения, выбранные этим противником (см. выше). Тогда система шифрования является стойкой против различения шифртекстов, если и только если $|p_1^A(n)-p_0^A(n)| = \mathrm{negl}(n)$ для любого полиномиально ограниченного противника A. Действительно, пусть полиномиально ограниченный противник A вычисляет бит $b \in_{\mathcal{U}} \{0,1\}$ по $E(1^n, e, m_b)$ с вероятностью не менее $1/2 + 1/\operatorname{poly}(n)$ для всех $n \in N$, где N — некоторое бесконечное подмножество $\mathbb{N}$. Легко видеть, что последняя вероятность не превосходит $(p_1^A(n) + 1 - p_0^A(n))/2$, поэтому

$$|p_1^A(n) - p_0^A(n)| = p_1^A(n) - p_0^A(n) \geqslant \frac{2}{\operatorname{poly}(n)} \geqslant \frac{1}{\operatorname{poly}(n)}$$

для всех $n \in N$. Обратно, пусть $|p_1^A(n)-p_0^A(n)| \geqslant 1/\operatorname{poly}(n)$ для всех $n \in N$, где A — некоторый полиномиально ограниченный противник, а N — некоторое бесконечное подмножество $\mathbb{N}$. Рассуждая аналогично доказательству леммы 0.1, можно считать, что $p_1^A(n) - p_0^A(n) \geqslant 1/\operatorname{poly}(n)$ для всех $n \in N$. Пусть противник B действует как A, но возвращает 1, если A возвратил 1, и возвращает 0 в противном случае. Тогда вероятность того, что B вычисляет бит $b \in_{\mathcal{U}} \{0,1\}$ по $E(1^n, e, m_b)$, есть

$$\frac{p_1^A(n) + 1 - p_0^A(n)}{2} \geqslant \frac{1}{2} + \frac{1}{2\operatorname{poly}(n)} \geqslant \frac{1}{2} + \frac{1}{\operatorname{poly}(n)}$$

для всех $n \in N$.

Если система шифрования является стойкой против различения шифртекстов, то говорят также, что эта система удовлетворяет условию *неотличимости шифртекстов* (indistinguishability of encryptions). Для систем шифрования с открытым ключом используется также термин *полиномиальная стойкость* (polynomial security; см. [GM84]) Сокращенно стойкость против различения шифртекстов называется *IND-стойкостью* (IND security). Если IND-стойкость имеет место на основе атаки с выбором открытых текстов (CPA), то говорят об *IND-CPA-стойкости* (IND-CPA security). Аналогично, IND-стойкость на основе атаки с выбором шифртекстов (CCA) называется *IND-CCA-стойкостью* (IND-CCA security).

Пример 14.2 (см. также [Lub96, лекция 14])**.** Предположим, что для каждого $n \in \mathbb{N}$ определено непустое множество $J_n \subseteq \{0,1\}^*$ и распределение вероятностей $\mathcal{J}_n$ на J_n, причем семейство $(\mathcal{J}_n \mid n \in \mathbb{N})$ полиномиально конструируемо, когда индексы заданы в унарной записи. Пусть также $(f_{n,d}\colon \{0,1\}^{l(n)} \to \{0,1\}^{l(n)} \mid n \in \mathbb{N},\ d \in J_n)$ — псевдослучайное относительно семейства $(\mathcal{J}_n \mid n \in \mathbb{N})$ полиномиально инвертируемое семейство перестановок, где l — полиномиальный параметр на $\mathbb{N}$. Известно, что такие семейства перестановок (при $l(n) = 2n$ для всех $n \in \mathbb{N}$) существуют в предположении существования односторонних функций (см. теорему 9.8 и замечание 9.9).

Определим блоковую систему шифрования (G, E, D) с секретным ключом и с длиной блока l следующим образом. Полиномиальный вероятностный алгоритм G (алгоритм генерации ключей) на входе 1^n, где $n \in \mathbb{N}$, выбирает $d \leftarrow \mathcal{J}_n$ и возвращает (d, d). Полиномиальные детерминированные алгоритмы E и D (алгоритмы шифрования и расшифрования соответственно) выбираются так, чтобы для всех $n \in \mathbb{N}$ и $d \in J_n$

$$E(1^n, d, m) = \begin{cases} f_{n,d}(m), & \text{если } m \in \{0,1\}^{l(n)}; \\ \perp & \text{в противном случае} \end{cases} \quad \text{и} \quad D(1^n, d, w) = \begin{cases} f_{n,d}^{-1}(w), & \text{если } w \in \{0,1\}^{l(n)}; \\ \perp & \text{в противном случае.} \end{cases}$$

Тогда легко видеть, что (G, E, D) удовлетворяет определению 14.1 с $M_n = \{0,1\}^{l(n)}$.

Предложение 14.3 (см. также [Lub96, лекция 14, упр. 54])**.** *Блоковая система шифрования (G, E, D), определенная в примере 14.2, является стойкой против различения шифртекстов на основе атаки с выбором открытых текстов, т. е. IND-CPA-стойкой.*

Доказательство. Пусть, напротив, существует полиномиальный вероятностный алгоритм A, осуществляющий различение шифртекстов на основе атаки с выбором открытых текстов для системы шифрования (G, E, D). Это значит, что A работает следующим образом. Сначала, получив на вход 1^n при произвольном $n \in \mathbb{N}$ и имея доступ к оракулу, вычисляющему произвольную перестановку π множества $\{0,1\}^{l(n)}$, алгоритм A вычисляет некоторую пару сообщений $(m_0, m_1) \in \{0,1\}^{l(n)} \times \{0,1\}^{l(n)}$, обозначаемую нами через $A^\pi(1^n)$. При этом требуется, чтобы ни m_0, ни m_1 не использовались в качестве запросов к указанному оракулу. Предполагается также, что в ответ на запросы не из $\{0,1\}^{l(n)}$ оракул возвращает $\perp$ (как оракул $E(1^n, d, \cdot)$ при любом $d \in J_n$). Поэтому мы считаем (не ограничивая общности), что алгоритм A при рассматриваемом вычислении

не делает таких бесполезных для него запросов. Затем алгоритму A подается на вход $\pi(m_b)$ при $b \in_{\mathcal{U}} \{0,1\}$, после чего он, не используя никаких оракулов, вычисляет бит, обозначаемый нами через $A(1^n, \pi(m_b))$. Условие осуществления рассматриваемой угрозы выглядит следующим образом: если $(\widetilde{d}, \widetilde{d}) = G(1^n)$, $(\widetilde{m}_0, \widetilde{m}_1) = A^{E(1^n,\widetilde{d},\cdot)}(1^n)$ и $\widetilde{b} \in_{\mathcal{U}} \{0,1\}$, то $\Pr[A(1^n, E(1^n, \widetilde{d}, \widetilde{m}_{\widetilde{b}})) = \widetilde{b}] \geqslant 1/2 + 1/\operatorname{poly}(n)$ для всех $n \in N$, где N — некоторое бесконечное подмножество $\mathbb{N}$.

Выберем полиномиальный вероятностный алгоритм B, который на произвольном входе вида 1^n (где $n \in N$) с использованием оракула, вычисляющего произвольную перестановку π множества $\{0,1\}^{l(n)}$, работает следующим образом:

1. Вычислить $(m_0, m_1) \leftarrow A^{\pi}(1^n)$, где $m_0, m_1 \in \{0,1\}^{l(n)}$.
2. Выбрать $b \in_{\mathcal{U}} \{0,1\}$.
3. Вычислить $b' \leftarrow A(1^n, \pi(m_b))$
4. Возвратить 1, если и только если $b' = b$.

Пусть $n \in N$, $\widetilde{d} \leftarrow \mathcal{J}_n$ (или $(\widetilde{d}, \widetilde{d}) = G(1^n)$), $(\widetilde{m}_0, \widetilde{m}_1) = A^{f_{n,\tilde{d}}}(1^n) = A^{E(1^n,\widetilde{d},\cdot)}(1^n)$, $\widetilde{\pi} \in_{\mathcal{U}} \operatorname{Per}(\{0,1\}^{l(n)})$, $(\widetilde{m}'_0, \widetilde{m}'_1) = A^{\widetilde{\pi}}(1^n)$ и $\widetilde{b} \in_{\mathcal{U}} \{0,1\}$. Тогда

$$\Pr[B^{f_{n,\tilde{d}}}(1^n) = 1] = \Pr[A(1^n, f_{n,\widetilde{d}}(\widetilde{m}_{\widetilde{b}})) = \widetilde{b}] = \Pr[A(1^n, E(1^n, \widetilde{d}, \widetilde{m}_{\widetilde{b}})) = \widetilde{b}] \geqslant \frac{1}{2} + \frac{1}{\operatorname{poly}(n)}.$$

В то же время информация, получаемая алгоритмом A при вычислении на входе 1^n с доступом к оракулу $\widetilde{\pi}$, распределена одинаково при условии $\widetilde{b} = 0$ и при условии $\widetilde{b} = 1$ (это следует из замечания 9.3). Поэтому $\widetilde{\pi}(\widetilde{m}'_{\widetilde{b}})$ не зависит от $\widetilde{b}$ и

$$\Pr[B^{\widetilde{\pi}}(1^n) = 1] = \Pr[A(1^n, \widetilde{\pi}(\widetilde{m}'_{\widetilde{b}})) = \widetilde{b}] = \frac{1}{2}.$$

Следовательно,

$$|\Pr[B^{f_{n,\tilde{d}}}(1^n) = 1] - \Pr[B^{\widetilde{\pi}}(1^n) = 1]| = \Pr[B^{f_{n,\tilde{d}}}(1^n) = 1] - \Pr[B^{\widetilde{\pi}}(1^n) = 1] \geqslant \frac{1}{\operatorname{poly}(n)}$$

для всех $n \in N$. Таким образом, получено противоречие с псевдослучайностью семейства перестановок $(f_{n,d} \mid n \in \mathbb{N},\ d \in J_n)$ относительно семейства $(\mathcal{J}_n \mid n \in \mathbb{N})$ распределений вероятностей. □

***Замечание* 14.4** (см. также [Lub96, лекция 14, упр. 57])**.** Можно показать, что если в примере 14.2 исходное полиномиально инвертируемое семейство перестановок $(f_{n,d} \mid n \in \mathbb{N},\ d \in J_n)$ является сильно псевдослучайным относительно семейства $(\mathcal{J}_n \mid n \in \mathbb{N})$, то определенная в этом примере блоковая система шифрования является стойкой против различения шифртекстов (т. е. IND-стойкой) на основе атаки с выбором открытых текстов и шифртекстов.

Рассмотрим теперь стойкость систем шифрования с открытым ключом на основе атаки с открытым ключом (или, что эквивалентно, атаки с выбором открытых текстов). В примерах 14.5 и 14.6 (см. ниже) мы будем использовать для построения систем шифрования с открытым ключом следующие объекты. Пусть $(I_n \mid n \in \mathbb{N})$ — семейство непустых попарно непересекающихся подмножеств $\{0,1\}^*$ и $I = \bigcup_{n\in\mathbb{N}} I_n$. Пусть также функция $\nu\colon I \to \mathbb{N}$ ставит в соответствие каждому $i \in I$ единственное число $n \in \mathbb{N}$ такое, что $i \in I_n$. Мы предполагаем, что функция $i \mapsto 1^{\nu(i)}$ $(i \in I)$ полиномиально вычислима. Предположим также, что каждого $n \in \mathbb{N}$ заданы множество $X_n \subseteq \{0,1\}^*$ и распределения вероятностей $\mathcal{G}_n$ и $\mathcal{X}_n$ на множествах $I_n \times \{0,1\}^*$ и X_n соответственно, причем семейства распределений вероятностей $(\mathcal{G}_n \mid n \in \mathbb{N})$ и $(\mathcal{X}_n \mid n \in \mathbb{N})$ полиномиально конструируемы, когда индексы заданы в унарной записи. Через $\mathcal{I}_n$, где $n \in \mathbb{N}$, будет обозначаться распределение случайной величины $\widetilde{i}$ такой, что $(\widetilde{i}, \widetilde{t}) \leftarrow \mathcal{G}_n$. Пусть теперь $F = (f_i\colon X_{\nu(i)} \to X_{\nu(i)} \mid i \in I)$ — семейство перестановок с секретом относительно семейств распределений вероятностей $(\mathcal{G}_n \mid n \in \mathbb{N})$ и $(\mathcal{X}_{\nu(i)} \mid i \in I)$ (очевидно, что последнее семейство полиномиально конструируемо). Таким образом, от общего случая, рассматриваемого в разделе 12, данный случай отличается лишь специальным видом множества I, а также тем, что X_i и $\mathcal{X}_i$ зависят только от $\nu(i)$.

Пример 14.5. Определим систему шифрования (G, E, D) с открытым ключом следующим образом. Пусть $n \in \mathbb{N}$. Полиномиальный вероятностный алгоритм G (алгоритм генерации ключей) на входе 1^n выбирает $(i, t) \leftarrow \mathcal{G}_n$ и возвращает $(i, (i, t))$. Полиномиальный детерминированный алгоритм E

(алгоритм шифрования) на входе $(1^n, i, m)$, где $i \in I_n$ и $m \in X_n$, возвращает $f_i(m)$. Наконец, полиномиальный детерминированный алгоритм D (алгоритм расшифрования) на входе $(1^n, (i,t), w)$, где $(i,t) \in \operatorname{supp} \mathcal{G}_n$ и $w \in X_n$, возвращает $f_i^{-1}(w)$ (такой алгоритм существует ввиду условия 2 определения 12.2). Тогда легко видеть, что (G, E, D) удовлетворяет определению 14.1 с $M_n = X_n$. Кроме того, из условия 1 определения 12.2 следует, что система шифрования (G, E, D) является стойкой против нахождения открытого текста (выбранного случайно относительно априорного распределения вероятностей $\mathcal{X}_n$) по шифртексту на основе атаки с открытым ключом.

Пример 14.6 (см. также [Gol04, конструкция 5.3.13])**.** Пусть $H = (h_i \colon X_{\nu(i)} \to \{0,1\}^{l(\nu(i))} \mid i \in I)$ — семейство функций, трудное для F относительно семейств распределений вероятностей $(\mathcal{I}_n \mid n \in \mathbb{N})$ и $(\mathcal{X}_{\nu(i)} \mid i \in I)$, где $l \colon \mathbb{N} \to \mathbb{N} \setminus \{0\}$ — полиномиальный параметр. Это значит, что H полиномиально вычислимо и семейства случайных величин

$$((\widetilde{i}_n, f_{\widetilde{i}_n}(\widetilde{x}_n), h_{\widetilde{i}_n}(\widetilde{x}_n)) \mid n \in \mathbb{N}) \quad \text{и} \quad ((\widetilde{i}_n, f_{\widetilde{i}_n}(\widetilde{x}_n), \widetilde{u}_{l(n)}) \mid n \in \mathbb{N}),$$

где $\widetilde{i}_n \leftarrow \mathcal{I}_n$ и $\widetilde{x}_n \leftarrow \mathcal{X}_n$, вычислительно неотличимы, когда индексы заданы в унарной записи (здесь на самом деле $\widetilde{x}_n = \widetilde{x}_{\nu(\widetilde{i}_n)} \leftarrow \mathcal{X}_n = \mathcal{X}_{\nu(\widetilde{i}_n)}$). Таким образом, понятие трудного семейства функций определяется аналогично понятию трудной функции (см. определение 5.1). Мы не будем обсуждать здесь вопрос о существовании трудного семейства функций для F; скажем лишь, что во многих случаях для его построения можно воспользоваться некоторой разновидностью теоремы Гольдрайха—Левина (см. теорему 6.2).

Определим блоковую систему шифрования (G, E, D) с открытым ключом и с длиной блока l следующим образом. Пусть $n \in \mathbb{N}$. Полиномиальный вероятностный алгоритм G (алгоритм генерации ключей) на входе 1^n выбирает $(i,t) \leftarrow \mathcal{G}_n$ и возвращает $(i, (i,t))$. Полиномиальный вероятностный алгоритм E (алгоритм шифрования) на входе $(1^n, i, m)$, где $i \in I_n$ и $m \in \{0,1\}^{l(n)}$, выбирает $x \leftarrow \mathcal{X}_n$ и возвращает $(f_i(x), h_i(x) \oplus m)$. Наконец, полиномиальный детерминированный алгоритм D (алгоритм расшифрования) на входе $(1^n, (i,t), (w,z))$, где $(i,t) \in \operatorname{supp} \mathcal{G}_n$, $w \in X_n$ и $z \in \{0,1\}^{l(n)}$, возвращает $h_i(f_i^{-1}(w)) \oplus z$ (здесь вычисление $f_i^{-1}(w)$ возможно за полиномиальное время ввиду условия 2 определения 12.2). Тогда легко видеть, что (G, E, D) удовлетворяет определению 14.1 с $M_n = \{0,1\}^{l(n)}$.

Предложение 14.7 (см. также [Gol04, предложение 5.3.14])**.** *Блоковая система шифрования (G, E, D), определенная в примере 14.6, является стойкой против различения шифртекстов (т. е. IND-стойкой) на основе атаки с открытым ключом.*

Доказательство. Пусть, напротив, существует полиномиальный вероятностный алгоритм A, осуществляющий различение шифртекстов на основе атаки с открытым ключом для системы шифрования (G, E, D). Это значит, что A работает следующим образом. Сначала, получив на вход $(1^n, i)$ при произвольных $n \in \mathbb{N}$ и $i \in \operatorname{supp} \mathcal{I}_n$, алгоритм A вычисляет некоторую пару сообщений $(m_0, m_1) \in \{0,1\}^{l(n)} \times \{0,1\}^{l(n)}$, обозначаемую нами через $A(1^n, i)$. Затем алгоритму A подается на вход $E(1^n, i, m_b) = (f_i(x), h_i(x) \oplus m_b)$ при $b \in_{\mathcal{U}} \{0,1\}$ и $x \leftarrow \mathcal{X}_n$, после чего он вычисляет бит, обозначаемый нами через $A(1^n, i, E(1^n, i, m_b))$. Условие осуществления рассматриваемой угрозы выглядит следующим образом: если $\widetilde{i} \leftarrow \mathcal{I}_n$, $(\widetilde{m}_0, \widetilde{m}_1) = A(1^n, \widetilde{i})$ и $\widetilde{b} \in_{\mathcal{U}} \{0,1\}$, то $\Pr[A(1^n, \widetilde{i}, E(1^n, \widetilde{i}, \widetilde{m}_{\widetilde{b}})) = \widetilde{b}] \geqslant 1/2 + 1/\operatorname{poly}(n)$ для всех $n \in N$, где N — некоторое бесконечное подмножество $\mathbb{N}$.

Выберем полиномиальный вероятностный алгоритм B, который на произвольном входе вида $(1^n, i, w, z)$, где $n \in N$, $i \in \operatorname{supp} \mathcal{I}_n$, $w \in X_n$ и $z \in \{0,1\}^{l(n)}$, работает следующим образом:

1. Вычислить $(m_0, m_1) \leftarrow A(1^n, i)$, где $m_0, m_1 \in \{0,1\}^{l(n)}$.
2. Выбрать $b \in_{\mathcal{U}} \{0,1\}$.
3. Вычислить $b' \leftarrow A(1^n, i, (w, z \oplus m_b))$
4. Возвратить 1, если и только если $b' = b$.

Пусть $n \in N$, $\widetilde{i} \leftarrow \mathcal{I}_n$, $(\widetilde{m}_0, \widetilde{m}_1) = A(1^n, \widetilde{i})$, $\widetilde{x} \leftarrow \mathcal{X}_n$ и $\widetilde{b} \in_{\mathcal{U}} \{0,1\}$. Тогда

$$\Pr[B(1^n, \widetilde{i}, f_{\widetilde{i}}(\widetilde{x}), h_{\widetilde{i}}(\widetilde{x})) = 1] = \Pr[A(1^n, \widetilde{i}, E(1^n, \widetilde{i}, \widetilde{m}_{\widetilde{b}})) = \widetilde{b}] \geqslant \frac{1}{2} + \frac{1}{\operatorname{poly}(n)}.$$

В то же время

$$\Pr[B(1^n, \widetilde{i}, f_{\widetilde{i}}(\widetilde{x}), \widetilde{u}_{l(n)}) = 1] = \Pr[A(1^n, \widetilde{i}, (f_{\widetilde{i}}(\widetilde{x}), \widetilde{u}_{l(n)} \oplus \widetilde{m}_{\widetilde{b}})) = \widetilde{b}] = \frac{1}{2},$$

так как $\widetilde{u}_{l(n)} \oplus \widetilde{m}_{\widetilde{b}}$ не зависит от $\widetilde{b}$. Следовательно,

$$\begin{aligned}&|\Pr[B(1^n, \widetilde{i}, f_{\widetilde{i}}(\widetilde{x}), h_{\widetilde{i}}(\widetilde{x})) = 1] - \Pr[B(1^n, \widetilde{i}, f_{\widetilde{i}}(\widetilde{x}), \widetilde{u}_{l(n)}) = 1]| \\ &\quad = \Pr[B(1^n, \widetilde{i}, f_{\widetilde{i}}(\widetilde{x}), h_{\widetilde{i}}(\widetilde{x})) = 1] - \Pr[B(1^n, \widetilde{i}, f_{\widetilde{i}}(\widetilde{x}), \widetilde{u}_{l(n)}) = 1] \geqslant \frac{1}{\operatorname{poly}(n)}\end{aligned}$$

для всех $n \in N$. Таким образом, получено противоречие с тем, что семейство H трудно для F относительно семейств распределений вероятностей $(\mathcal{I}_n \mid n \in \mathbb{N})$ и $(\mathcal{X}_{\nu(i)} \mid i \in I)$. □

***Замечание* 14.8** (см. также [Gol04, конструкция 5.3.7])**.** Пусть (G, E, D) — блоковая система шифрования с длиной блока l. Тогда можно естественным образом построить систему шифрования, в которой пространство сообщений совпадает с $\{0,1\}^*$. Говоря упрощенно, алгоритм шифрования новой системы разбивает сообщения произвольной длины на блоки длины $l(n)$ и применяет алгоритм E к каждому из этих блоков. Более подробно, пусть $n \in \mathbb{N}$ и $(e, d) \in \operatorname{supp} G(1^n)$. Выберем полиномиальный вероятностный алгоритм E', работающий на произвольном входе $(1^n, e, m)$, где $m \in \{0,1\}^*$, следующим образом:

1. Выбрать единственные $m_1, \ldots, m_k \in \{0,1\}^{l(n)}$ такие, что $m_1 \ldots m_k = m10^s$, где $s = (-|m| - 1) \bmod l(n)$ и $k = (|m| + 1 + s)/l(n)$.
2. Для каждого $i \in \{1, \ldots, k\}$ вычислить $w_i \leftarrow E(1^n, e, m_i)$.
3. Возвратить $(w_1, \ldots, w_k)$.

Пусть также D' — полиномиальный детерминированный алгоритм, который на произвольном входе $(1^n, d, (w_1, \ldots, w_k))$, где $k \in \mathbb{N} \setminus \{0\}$ и $w_1, \ldots, w_k \in \{0,1\}^*$, работает следующим образом:

1. Для каждого $i \in \{1, \ldots, k\}$ вычислить $m_i = D(1^n, d, w_i)$.
2. Если $m_i \in \{0,1\}^{l(n)}$ для всех $i \in \{1, \ldots, k\}$ и $m_k = m'_k 10^s$ при $s \in \{0, \ldots, l(n) - 1\}$, то возвратить $m_1 \ldots m_{k-1} m'_k$.

Тогда легко видеть, что (G, E', D') — система шифрования с пространством сообщений $\{0,1\}^*$. Вопрос о ее стойкости рассматривается в следующем предложении.

Предложение 14.9 (см. также [Gol04, предложение 5.3.8])**.** *Пусть (G, E, D) — блоковая система шифрования с открытым ключом, а (G, E', D') — система шифрования, определенная по ней в замечании 14.8. Тогда если (G, E, D) является IND-стойкой на основе атаки с открытым ключом, то и (G, E', D') является IND-стойкой на основе атаки с открытым ключом.*

Доказательство. Пусть, напротив, существует полиномиальный вероятностный алгоритм A, осуществляющий различение шифртекстов на основе атаки с открытым ключом для системы шифрования (G, E', D'). Это значит, что A работает следующим образом. Сначала, получив на вход $(1^n, e)$ при произвольных $n \in \mathbb{N}$ и $(e, d) \in \operatorname{supp} G(1^n)$, алгоритм A вычисляет некоторую пару сообщений $(m_0, m_1) \in \{0,1\}^* \times \{0,1\}^*$ одинаковой длины, обозначаемую нами через $A(1^n, e)$. Затем алгоритму A подается на вход $E'(1^n, e, m_b)$ при некотором $b \in \{0,1\}$, после чего он вычисляет бит, обозначаемый нами через $A(1^n, e, E'(1^n, e, m_b))$. Условие осуществления рассматриваемой угрозы выглядит следующим образом: если $(\widetilde{e}, \widetilde{d}) = G(1^n)$, $(\widetilde{m}_0, \widetilde{m}_1) = A(1^n, \widetilde{e})$, то $|\Pr[A(1^n, \widetilde{e}, E'(1^n, \widetilde{e}, \widetilde{m}_0)) = 1] - \Pr[A(1^n, \widetilde{e}, E'(1^n, \widetilde{e}, \widetilde{m}_1)) = 1]| \geqslant 1/\operatorname{poly}(n)$ для всех $n \in N$, где N — некоторое бесконечное подмножество $\mathbb{N}$. Рассуждая аналогично доказательству леммы 0.1, можно считать, что

$$\Pr[A(1^n, \widetilde{e}, E'(1^n, \widetilde{e}, \widetilde{m}_0)) = 1] - \Pr[A(1^n, \widetilde{e}, E'(1^n, \widetilde{e}, \widetilde{m}_1)) = 1] \geqslant \frac{1}{\operatorname{poly}(n)} \tag{34}$$

для всех $n \in N$. Кроме того, мы предполагаем (без ограничения общности), что сообщения m_0 и m_1, вычисляемые алгоритмом A, имеют длину $\pi(n)$, где $\pi\colon \mathbb{N} \to \mathbb{N} \setminus \{0\}$ — некоторый полиномиальный параметр. Этого можно добиться, заменяя m_b на $m_b 10^t$ для подходящего $t \geqslant (-|m_b| - 1) \bmod l(n)$; очевидно, что тогда из случайного шифртекста для $m_b 10^t$ можно легко извлечь случайный шифртекст для m_b, зная $|m_b|$ $(b \in \{0,1\})$.

Воспользуемся гибридным методом. Выберем полиномиальный вероятностный алгоритм B, который на произвольном входе вида $(1^n, e)$, где $n \in N$ и $(e, d) \in \operatorname{supp} G(1^n)$, работает следующим образом:

1. Вычислить $(m_0, m_1) \leftarrow A(1^n, e)$, где $m_0, m_1 \in \{0,1\}^{\pi(n)}$. Пусть $s(n) = (-\pi(n) - 1) \bmod l(n)$, $k(n) = (\pi(n) + 1 + s(n))/l(n)$ и $m_b 10^{s(n)} = m_{b,1} \dots m_{b,k(n)}$ для каждого $b \in \{0,1\}$, где $m_{b,1}, \dots, m_{b,k(n)} \in \{0,1\}^{l(n)}$.
2. Выбрать $i \in_{\mathcal{U}} \{1, \dots, k(n)\}$ (очевидно, что $k(n) \geqslant 1$).
3. Возвратить $(m_{0,i}, m_{1,i})$ в качестве $B(1^n, e)$.
4. Получив $z \leftarrow E(1^n, e, m_{b,i})$ для некоторого $b \in \{0,1\}$, вычислить $v_j \leftarrow E(1^n, e, m_{0,j})$ для каждого $j \in \{1, \dots, i-1\}$ и $w_j \leftarrow E(1^n, e, m_{1,j})$ для каждого $j \in \{i+1, \dots, k(n)\}$.
5. Возвратить $A(1^n, e, (v_1, \dots, v_{i-1}, z, w_{i+1}, \dots, w_{k(n)}))$ в качестве $B(1^n, e, z)$.

Пусть $n \in N$, $(\widetilde{e}, \widetilde{d}) = G(1^n)$, $(\widetilde{m}_0, \widetilde{m}_1) = A(1^n, \widetilde{e})$, $s(n)$ и $k(n)$ определяются так же, как в описании алгоритма B, $\widetilde{m}_b 10^{s(n)} = \widetilde{m}_{b,1} \dots \widetilde{m}_{b,k(n)}$ для каждого $b \in \{0,1\}$, где $\widetilde{m}_{b,1}, \dots, \widetilde{m}_{b,k(n)} \in \{0,1\}^{l(n)}$, $\widetilde{i} \in_{\mathcal{U}} \{1, \dots, k(n)\}$, $\widetilde{v}_j = E(1^n, \widetilde{e}, \widetilde{m}_{0,j})$ и $\widetilde{w}_j = E(1^n, \widetilde{e}, \widetilde{m}_{1,j})$ при всех $j \in \{1, \dots, k(n)\}$. Для каждого $i \in \{0, \dots, k(n)\}$ положим

$$\delta_i(n) = \Pr[A(1^n, \widetilde{e}, (\widetilde{v}_1, \dots, \widetilde{v}_i, \widetilde{w}_{i+1}, \dots, \widetilde{w}_{k(n)})) = 1].$$

Тогда если $(\widetilde{q}_0, \widetilde{q}_1) = B(1^n, \widetilde{e})$, то

$$\Pr[B(1^n, \widetilde{e}, E(1^n, \widetilde{e}, \widetilde{q}_0)) = 1] = \Pr[B(1^n, \widetilde{e}, E(1^n, \widetilde{e}, \widetilde{m}_{0,\widetilde{i}})) = 1] = \frac{1}{k(n)} \sum_{i=1}^{k(n)} \delta_i(n)$$

и

$$\Pr[B(1^n, \widetilde{e}, E(1^n, \widetilde{e}, \widetilde{q}_1)) = 1] = \Pr[B(1^n, \widetilde{e}, E(1^n, \widetilde{e}, \widetilde{m}_{1,\widetilde{i}})) = 1] = \frac{1}{k(n)} \sum_{i=1}^{k(n)} \delta_{i-1}(n).$$

Кроме того,

$$\delta_{k(n)}(n) = \Pr[A(1^n, \widetilde{e}, E'(1^n, \widetilde{e}, \widetilde{m}_0)) = 1] \quad \text{и} \quad \delta_0(n) = \Pr[A(1^n, \widetilde{e}, E'(1^n, \widetilde{e}, \widetilde{m}_1)) = 1].$$

Следовательно,

$$\begin{aligned} &|\Pr[B(1^n, \widetilde{e}, E(1^n, \widetilde{e}, \widetilde{q}_0)) = 1] - \Pr[B(1^n, \widetilde{e}, E(1^n, \widetilde{e}, \widetilde{q}_1)) = 1]| \\ &\quad = \Pr[B(1^n, \widetilde{e}, E(1^n, \widetilde{e}, \widetilde{q}_0)) = 1] - \Pr[B(1^n, \widetilde{e}, E(1^n, \widetilde{e}, \widetilde{q}_1)) = 1] = \frac{\delta_{k(n)}(n) - \delta_0(n)}{k(n)} \\ &\quad = \frac{\Pr[A(1^n, \widetilde{e}, E'(1^n, \widetilde{e}, \widetilde{m}_0)) = 1] - \Pr[A(1^n, \widetilde{e}, E'(1^n, \widetilde{e}, \widetilde{m}_1) = 1]}{k(n)} \geqslant \frac{1}{\operatorname{poly}(n)} \end{aligned}$$

для любого $n \in N$ ввиду неравенства (34) и того, что $k(n) \leqslant \operatorname{poly}(n)$ при всех $n \in \mathbb{N}$. Таким образом, получено противоречие с IND-стойкостью системы шифрования (G, E, D) на основе атаки с открытым ключом. □

***Замечание* 14.10.** Если существуют односторонние функции, то аналог предложения 14.9 для систем шифрования с секретным ключом и для IND-CPA-стойкости неверен. Действительно, пусть (G, E, D) — блоковая система шифрования с секретным ключом, определенная в примере 14.2, а (G, E', D') — система шифрования, определенная по ней в замечании 14.8. Тогда (G, E, D) является IND-CPA-стойкой согласно предложению 14.3. Рассмотрим теперь противника, который в течение атаки с выбором открытых текстов на систему (G, E', D') выбирает произвольные различные $q_0, q_1 \in \{0,1\}^{l(n)}$ (такие существуют, так как $l(n) \geqslant 1$) и получает $E'(1^n, d, q_0 q_1) = (f_{n,d}(q_0), f_{n,d}(q_1), f_{n,d}(10^{l(n)-1}))$. Здесь n — параметр стойкости (известный противнику в виде 1^n), а d — общий секретный ключ отправителя и получателя (неизвестный противнику). После этого противник может осуществить различение шифртекстов, выбрав $m_0 = q_0$, $m_1 = q_1$ и, получив $E'(1^n, d, m_b) = (f_{n,d}(q_b), f_{n,d}(10^{l(n)-1}))$ при $b \in_{\mathcal{U}} \{0,1\}$, найти b с помощью сравнения $f_{n,d}(q_b)$ с $f_{n,d}(q_0)$ или $f_{n,d}(q_1)$ (напомним, что $f_{n,d}$ — перестановка множества $\{0,1\}^{l(n)}$). Это показывает, что (G, E', D') не является IND-CPA-стойкой.

15. Протоколы электронной подписи

Определение 15.1 (протокол электронной подписи; см. также [Gol04, определение 6.1.1])**.** Предположим, что каждому числу $n \in \mathbb{N}$ (играющему роль параметра стойкости) поставлено в соответствие множество $M_n \subseteq \{0,1\}^*$, называемое *пространством сообщений* (message space) при данном n. *Протоколом электронной подписи* (electronic signature protocol, digital signature protocol) называется тройка алгоритмов (G, S, V) таких, что

1) G и S — полиномиальные вероятностные алгоритмы, а V — полиномиальный детерминированный алгоритм;

2) $\operatorname{supp} G(1^n) \subseteq \{0,1\}^* \times \{0,1\}^*$ для любого $n \in \mathbb{N}$;

3) при всех $n \in \mathbb{N}$, $m \in M_n$, и $(s, v) \in \operatorname{supp} G(1^n)$ всегда выполняется равенство $V(1^n, v, m, S(1^n, s, m)) = 1$.

Алгоритмы G, S и V называются алгоритмами *генерации ключей* (key generation), *генерации подписей* (signing) и *проверки подписей* (verification) соответственно.

Схема применения протокола электронной подписи (G, S, V) может выглядеть следующим образом. Пусть выбран параметр стойкости $n \in \mathbb{N}$. Подписывающий генерирует пару ключей $(s, v) \leftarrow G(1^n)$ (см. условие 2 определения 15.1). Ключ s служит для генерации подписей, а v — для их проверки. Поэтому s хранится подписывающим в секрете, а v публикуется. В связи с этим s называется *секретным ключом* (private key), а v — *открытым ключом* (public key). Подписью для произвольного сообщения $m \in M_n$, вычисляемой подписывающим, является выходное значение алгоритма S на входе $(1^n, s, m)$. Алгоритм V служит для проверки подписей. А именно, $w \in \{0,1\}^*$ называется *допустимой подписью* (valid signature) для сообщения $m \in M_n$ относительно открытого ключа v, если $V(1^n, v, m, w) = 1$ (это равенство означает, что строка w принята в качестве подписи для сообщения m при открытом ключе v). Таким образом, условие 3 определения 15.1 означает, что если $(s, v) \in \operatorname{supp} G(1^n)$, то подпись для произвольного сообщения из M_n, сгенерированная алгоритмом S с использованием секретного ключа s, является допустимой для этого сообщения относительно открытого ключа v.

Протокол электронной подписи может использовать некоторую дополнительную открытую информацию, генерируемую по 1^n за полиномиальное время и публикуемую для всеобщего доступа. Например, если протокол использует вычисления в кольце $\mathbb{Z}_N$, то эта информация содержит N. Если есть такая дополнительная открытая информация, то подразумевается, что она подается на дополнительный вход алгоритмов G, S и V.

Вообще говоря, для применения протоколов электронной подписи на практике необходимо наличие процедуры арбитража, позволяющей разрешать споры между участниками. Для простоты в настоящем курсе, как и в книге [Gol04], арбитраж не рассматривается.

До сих пор ничего не говорилось о стойкости протоколов электронной подписи. Напомним, что стойкость протокола электронной подписи, как и любого криптографического протокола, определяется против конкретной угрозы на основе конкретной атаки. В связи с эти перечислим некоторые атаки на произвольный протокол электронной подписи (G, S, V) и угрозы стойкости последнего (см. также [GMR88, GB08]). В описаниях этих атак и угроз через n обозначается параметр стойкости (число из $\mathbb{N}$), а через s и v — секретный и открытый ключи соответственно, полученные как части случайного значения $G(1^n)$. При этом предполагается, что n, s и v не меняются в течение работы противника.

Атаки на протокол (G, S, V)**:**

- *Атака с открытым ключом* (key-only attack): противник получает только открытую информацию, относящуюся к атакуемому протоколу. Эта информация включает 1^n и v. Данная атака является самой слабой и всегда доступной для противника.

- *Атака с известными сообщениями* (known-message attack): противник получает открытую информацию, относящуюся к атакуемому протоколу, и множество $\{(m_1, w_1), \ldots, (m_k, w_k)\}$, где $m_i \in M_n$ и $w_i \leftarrow S(1^n, s, m_i)$ для всех $i \in \{1, \ldots, k\}$. При этом противник никак не влияет на выбор сообщений $m_1, \ldots, m_k$.

- *Адаптивная атака с выбором сообщений* (adaptive chosen-message attack): противник получает открытую информацию, относящуюся к атакуемому протоколу, после чего имеет доступ к

оракулу, который в ответ на произвольный запрос $q \in \{0,1\}^*$ выдает случайное значение $S(1^n, s, q)$. Этот оракул будет обозначаться через $S(1^n, s, \cdot)$. Предполагается, что случайные биты, используемые алгоритмом S, при каждом его выполнении выбираются заново.

Угрозы стойкости протокола (G, S, V)**:**

- *Полное раскрытие* (total breaking): нахождение некоторого ключа s' такого, что $(s', v) \in \operatorname{supp} G(1^n)$. Такой ключ s' позволяет подписывать сообщения от имени подписывающего, имеющего секретный ключ s. Отметим, что s' может отличаться от секретного ключа s, используемого подписывающим.
- *Универсальная подделка* (universal forgery): нахождение допустимой подписи для любого сообщения $m \in M_n$ относительно открытого ключа v.
- *Экзистенциальная подделка* (existential forgery): нахождение некоторой пары (m, w), где $m \in M_n$, а w — допустимая подпись для сообщения m относительно открытого ключа v. При этом предполагается, что m отлично от сообщений, которые противник получает или выбирает при осуществлении атак. Вообще говоря, противник не контролирует выбор сообщения m.

Атаки здесь перечислены в порядке возрастания силы, а угрозы — в порядке ее убывания. Чем сильнее атака и слабее угроза, тем сильнее стойкость протокола против данной угрозы на основе данной атаки. Следующее замечание показывает, что если существуют протоколы электронной подписи, стойкие против самой сильной угрозы на основе самой слабой атаки (среди перечисленных атак и угроз), то существуют односторонние функции.

***Замечание* 15.2** (см. [Rom90])**.** Пусть (G, S, V) — протокол электронной подписи, стойкий против полного раскрытия на основе атаки с открытым ключом. Выберем полиномиальный параметр l на $\mathbb{N}$ такой, что алгоритм G при вычислении на входе 1^n (где $n \in \mathbb{N}$) использует не более $l(n)$ случайных битов. Для произвольных $n \in \mathbb{N}$ и $x \in \{0,1\}^{l(n)}$ положим $f_{1^n}(x) = v$, если $G(1^n; x) = (s, v)$. Пусть также $I = \{1^n \mid n \in \mathbb{N}\}$, $\mathcal{I}_n$ — распределение вероятностей, сосредоточенное на 1^n, а $\mathcal{X}_{1^n} = \mathcal{U}(\{0,1\}^{l(n)})$ $(n \in \mathbb{N})$. Тогда легко видеть, что семейство функций $(f_i \mid i \in I)$ является односторонним относительно семейств распределений вероятностей $(\mathcal{I}_n \mid n \in \mathbb{N})$ и $(\mathcal{X}_i \mid i \in I)$. Согласно замечанию 4.2, на основе этого семейства функций можно построить одностороннюю функцию.

Наоборот, если существуют односторонние функции, то существуют даже протоколы электронной подписи, стойкие против самой слабой угрозы на основе самой сильной атаки (среди перечисленных атак и угроз). Этот результат принадлежит Ромпелю [Rom90] (см. также [Gol04, теорема 6.4.1]).

Теорема 15.3. *Если существуют односторонние функции, то существуют протоколы электронной подписи, стойкие против экзистенциальной подделки на основе адаптивной атаки с выбором сообщений.*

Доказательство теоремы 15.3 остается за рамками настоящего курса ввиду сложности этого доказательства.

Пример 15.4 (протокол Лэмпорта; см. также [Lam79], [Gol04, п. 6.4.1.2], [Lub96, лекция 17])**.** Пусть f — односторонняя функция, а $l\colon \mathbb{N} \to \mathbb{N} \setminus \{0\}$ — полиномиальный параметр. Определим протокол электронной подписи (G, S, V) с пространством сообщений $M_n = \{0,1\}^{l(n)}$ для каждого $n \in \mathbb{N}$ следующим образом. Пусть $n \in \mathbb{N}$. Полиномиальный вероятностный алгоритм G (алгоритм генерации ключей) на входе 1^n

- выбирает $s_{i,b} \in_{\mathcal{U}} \{0,1\}^n$ и вычисляет $v_{i,b} = f(s_{i,b})$ для каждых $i \in \{1, \ldots, l(n)\}$ и $b \in \{0,1\}$;
- возвращает на выходе (s, v), где

$$s = \begin{pmatrix} s_{1,0} & \ldots & s_{l(n),0} \\ s_{1,1} & \ldots & s_{l(n),1} \end{pmatrix} \text{ (секретный ключ)} \quad \text{и} \quad v = \begin{pmatrix} v_{1,0} & \ldots & v_{l(n),0} \\ v_{1,1} & \ldots & v_{l(n),1} \end{pmatrix} \text{ (открытый ключ).}$$

Полиномиальный детерминированный алгоритм S (алгоритм генерации подписей) на входе $(1^n, s, m)$, где $s = \left(\begin{smallmatrix} s_{1,0} & \ldots & s_{l(n),0} \\ s_{1,1} & \ldots & s_{l(n),1} \end{smallmatrix}\right)$, $s_{i,b} \in \{0,1\}^n$ и $m \in \{0,1\}^{l(n)}$, выдает в качестве подписи для сообщения m набор $(s_{1,m_{[1]}}, \ldots, s_{l(n),m_{[l(n)]}})$. Наконец, полиномиальный детерминированный алгоритм V (алгоритм проверки подписей) на входе $(1^n, v, m, w)$, где $v = \left(\begin{smallmatrix} v_{1,0} & \ldots & v_{l(n),0} \\ v_{1,1} & \ldots & v_{l(n),1} \end{smallmatrix}\right)$ и $m \in \{0,1\}^{l(n)}$, принимает w в качестве подписи для сообщения m (возвращает 1), если и только если $w = (w_1, \ldots, w_{l(n)})$, $w_i \in \{0,1\}^n$ и $f(w_i) = v_{i,m_{[i]}}$ для всех $i \in \{1, \ldots, l(n)\}$. Легко видеть, что (G, S, V) удовлетворяет определению 15.1 (с $M_n = \{0,1\}^{l(n)}$).

Предложение 15.5. *Протокол Лэмпорта электронной подписи (G, S, V) , определенный в примере 15.4, является стойким против экзистенциальной подделки на основе частного случая адаптивной атаки с выбором сообщений, в котором противник может обратиться к оракулу $S(1^n, s, \cdot)$ только один раз.*

Доказательство. Пусть, напротив, существует полиномиальный вероятностный алгоритм A, осуществляющий экзистенциальную подделку на основе указанной атаки для протокола Лэмпорта (G, S, V). Это значит, что A работает следующим образом. Сначала, получив на вход $(1^n, v)$ при произвольных $n \in \mathbb{N}$ и $(s, v) \in \operatorname{supp} G(1^n)$, алгоритм A вычисляет некоторое сообщение $q \in \{0,1\}^{l(n)} = M_n$, обозначаемое нами через $A(1^n, v)$. Затем алгоритму A подается на вход $S(1^n, s, q)$, после чего он вычисляет пару (m, w) (где $m \in \{0,1\}^{l(n)} \setminus \{q\}$ и $w \in (\{0,1\}^n)^{l(n)}$), обозначаемую нами через $A(1^n, v, S(1^n, s, q))$. Условие осуществления рассматриваемой угрозы выглядит следующим образом: если $(\widetilde{s}, \widetilde{v}) = G(1^n)$, $\widetilde{q} = A(1^n, \widetilde{v})$ и $(\widetilde{m}, \widetilde{w}) = A(1^n, \widetilde{v}, S(1^n, \widetilde{s}, \widetilde{q}))$, то $\Pr[V(1^n, \widetilde{v}, \widetilde{m}, \widetilde{w}) = 1] \geqslant 1/\operatorname{poly}(n)$ для всех $n \in N$, где N — некоторое бесконечное подмножество $\mathbb{N}$.

Выберем полиномиальный вероятностный алгоритм B, который на произвольном входе вида $(1^n, y)$, где $n \in N$ и $y \in f(\{0,1\}^n)$, работает следующим образом:

1. Вычислить $(s, v) \leftarrow G(1^n)$, т. е.

$$s = \begin{pmatrix} s_{1,0} & \dots & s_{l(n),0} \\ s_{1,1} & \dots & s_{l(n),1} \end{pmatrix} \quad \text{и} \quad v = \begin{pmatrix} v_{1,0} & \dots & v_{l(n),0} \\ v_{1,1} & \dots & v_{l(n),1} \end{pmatrix},$$

   где $s_{i,b} \in_{\mathcal{U}} \{0,1\}^n$ и $v_{i,b} = f(s_{i,b})$ для каждых $i \in \{1, \dots, l(n)\}$ и $b \in \{0,1\}$.

2. Выбрать $j \in_{\mathcal{U}} \{1, \dots, l(n)\}$ и $c \in_{\mathcal{U}} \{0,1\}$.

3. Вычислить $q \leftarrow A(1^n, v') \in \{0,1\}^{l(n)}$, где матрица v' получена из v заменой $v_{j,c}$ на y.

4. Если $q_{[j]} = c$, то закончить работу; в этом случае вычисление неуспешно. В противном случае вычислить $z = S(1^n, s, q) = (s_{1,q_{[1]}}, \dots, s_{l(n),q_{[l(n)]}})$.

5. Вычислить $(m, w) = A(1^n, v', z)$, где $m \in \{0,1\}^{l(n)} \setminus \{q\}$, $w = (w_1, \dots, w_{l(n)})$ и $w_i \in \{0,1\}^n$ для всех $i \in \{1, \dots, l(n)\}$.

6. Если $m_{[j]} = c$, то возвратить w_j.

Очевидно, что если $q_{[j]} \neq c$, $m_{[j]} = c$ (или, что эквивалентно, $q_{[j]} \neq c$, $m_{[j]} \neq q_{[j]}$) и $V(1^n, v', m, w) = 1$, то $B(1^n, y) = w_j \in f^{-1}(y)$.

Пусть $n \in N$, $\widetilde{x} \in_{\mathcal{U}} \{0,1\}^n$, $\widetilde{y} = f(\widetilde{x})$, $(\widetilde{s}, \widetilde{v}) = G(1^n)$, $\widetilde{j} \in_{\mathcal{U}} \{1, \dots, l(n)\}$, $\widetilde{c} \in_{\mathcal{U}} \{0,1\}$, матрица $\widetilde{v}'$ получена из $\widetilde{v}$ заменой $\widetilde{v}_{\widetilde{j},\widetilde{c}}$ на $\widetilde{y}$, $\widetilde{q} = A(1^n, \widetilde{v}')$, $\widetilde{z} = S(1^n, \widetilde{s}, \widetilde{q})$ и $(\widetilde{m}, \widetilde{w}) = A(1^n, \widetilde{v}', \widetilde{z})$. Очевидно, что случайные величины $(\widetilde{j}, \widetilde{c})$ и $\widetilde{v}'$ независимы. Поэтому

$$\Pr[B(1^n, \widetilde{y}) \in f^{-1}(\widetilde{y})] \geqslant \Pr[\widetilde{q}_{[\widetilde{j}]} \neq \widetilde{c},\ \widetilde{m}_{[\widetilde{j}]} \neq \widetilde{q}_{[\widetilde{j}]},\ V(1^n, \widetilde{v}', \widetilde{m}, \widetilde{w}) = 1] \geqslant \frac{1}{2l(n)\operatorname{poly}(n)} \geqslant \frac{1}{\operatorname{poly}(n)}$$

для всех $n \in N$. Здесь использовано то, что $\Pr[\widetilde{q}_{[\widetilde{j}]} \neq \widetilde{c}] = 1/2$ и $\Pr[\widetilde{m}_{[\widetilde{j}]} \neq \widetilde{q}_{[\widetilde{j}]}] = \Pr[\widetilde{j} \in \delta(\widetilde{m}, \widetilde{q})] \geqslant \Pr[\widetilde{j} = \min \delta(\widetilde{m}, \widetilde{q})] = 1/l(n)$, где $\delta(\widetilde{m}, \widetilde{q}) = \{i \in \{1, \dots, l(n)\} \mid \widetilde{m}_{[i]} \neq \widetilde{q}_{[i]}\} \neq \varnothing$. Таким образом, получено противоречие с односторонностью функции f. □

***Замечание* 15.6.** Очевидно, что если противник в протоколе Лэмпорта получил допустимые подписи (относительно открытого ключа v) для каких-либо сообщений $m, m' \in \{0,1\}^{l(n)}$ таких, что $m_{[i]} \neq m'_{[i]}$ при всех $i \in \{1, \dots, l(n)\}$, то он может осуществить полное раскрытие. Таким образом, этот протокол является стойким только тогда, когда с использованием одного и того же секретного ключа подписывается лишь одно сообщение.

Отметим, что протокол Лэмпорта используется для построения протоколов электронной подписи, допускающих подписывание многих сообщений с использованием одного и того же секретного ключа (см., например, [NY89, Rom90]).

16. Протоколы интерактивного доказательства

Пусть L — язык, т. е. $L \subseteq \{0,1\}^*$. Говоря неформально, протокол интерактивного доказательства для языка L — это протокол с двумя участниками, называемыми *доказывающим* (prover) и *проверяющим* (verifier), в котором целью доказывающего является доказать проверяющему, что их общий вход $x \in \{0,1\}^*$ принадлежит L. При этом вычислительные возможности доказывающего не ограничиваются. Протокол должен удовлетворять следующим условиям:

- Условие полноты: если $x \in L$, то честный доказывающий успешно докажет это честному проверяющему.
- Условие корректности: если $x \notin L$, то нечестный доказывающий не может доказать честному проверяющему, что $x \in L$.

Мы дадим определение протокола интерактивного доказательства в несколько более общем виде, чем это принято. А именно, в следующем определении протокол интерактивного доказательства может иметь произвольные границы полноты и корректности (см. также [Gol04, определение 4.2.6]). Часто эти границы предполагаются равными $2/3$ и $1/3$ соответственно. См. также пионерские работы [GMR85, GMR89].

Нам потребуется несколько определений. Алгоритм A (вообще говоря, вероятностный) называется *интерактивным* (interactive), если он имеет дополнительный вход R_A и дополнительный выход S_A, предназначенные для взаимодействия с другими интерактивными алгоритмами. Вход R_A служит для приема сообщений, а выход S_A — для их отправки. В любой момент вычисления алгоритм либо активен, либо находится в специальном состоянии, называемом *состоянием ожидания* (idle). В период активности алгоритм производит обычные вычисления, включающие чтение очередного входящего сообщения с R_A и формирование очередного исходящего сообщения для отправки через S_A. Если же алгоритм находится в состоянии ожидания, то никакие вычисления не производятся; алгоритм выходит из этого состояния при получении очередного входящего сообщения на вход R_A.

Интерактивные алгоритмы предназначены для совместных вычислений с другими интерактивными алгоритмами. Пусть (A, B) — пара интерактивных алгоритмов. Чтобы организовать их совместное вычисление, необходимо отождествить R_A с S_B и S_A — с R_B. Часто бывает удобно считать, что A и B имеют общий вход, помимо собственного входа каждого из них, к которому другой алгоритм не имеет доступа. Предположим, что инициирует вычисление алгоритм A. Тогда в начале вычисления A активен, а B находится в состоянии ожидания. В этот период активности алгоритм A вычисляет некоторое сообщение m_1 и посылает его алгоритму B посредством своего выхода S_A, совпадающего с входом R_B алгоритма B. После этого A переходит в состояние ожидания, а B становится активным, читает сообщение m_1 с входа R_B, вычисляет некоторое сообщение m_2 и посылает его алгоритму A посредством своего выхода S_B, совпадающего с входом R_A алгоритма A. Затем B переходит в состояние ожидания, а A снова становится активным, читает сообщение m_2 с входа R_A, вычисляет некоторое сообщение m_3 и посылает его алгоритму B посредством своего выхода S_A, совпадающего с входом R_B алгоритма B, и т. д. Если один из алгоритмов закончил свое вычисление, то заканчивается и совместное вычисление. Каждый период активности A или B называется *раундом* (round). Таким образом, число раундов совпадает с числом пересылаемых сообщений. Это определение не является общепринятым; в некоторых работах раундом считается период времени, когда оба алгоритма обменяются сообщениями.

Рассмотрим совместное вычисление интерактивных вероятностных алгоритмов A и B при условии, что y и z — собственные входы алгоритмов A и B соответственно, а x — общий вход этих алгоритмов (некоторые из этих входов могут отсутствовать). Тогда через $\langle A(y), B(z)\rangle(x)$ мы будем обозначать выходное значение алгоритма B, полученное в результате этого вычисления. Пусть также $\mathrm{view}_{B(z)}^{A(y)}(x)$ — набор, состоящий из случайной строки алгоритма B, используемой при рассматриваемом вычислении, и последовательности сообщений (в хронологическом порядке), полученных этим алгоритмом в процессе данного вычисления. Разумеется, как $\langle A(y), B(z)\rangle(x)$, так и $\mathrm{view}_{B(z)}^{A(y)}(x)$ являются случайными величинами.

Интерактивный алгоритм A называется *полиномиальным* или *работающим за полиномиальное время* (polynomial-time algorithm), если время работы (т. е. число тактов до остановки) алгоритма A при совместном вычислении с любым интерактивным алгоритмом не превосходит $\mathrm{poly}(|x| + |y|)$, где x — произвольный общий вход алгоритмов, а y — произвольный собственный вход алгоритма A. Отметим, что в этом определении не участвуют сообщения, которые получает алгоритм A при

совместном вычислении. В частности, не предполагается, что полиномиальный интерактивный алгоритм может прочитать полностью получаемые им сообщения.

Определение 16.1 (протокол интерактивного доказательства)**.** Пусть α и β — функции из $\mathbb{N}$ в $\mathbb{R}$. Пара интерактивных вероятностных алгоритмов (P, V), в которой V полиномиален, называется *протоколом интерактивного доказательства* (interactive proof protocol) для языка L с *границей полноты* (completeness bound) α и *границей корректности* (soundness bound) β, если

- $\Pr[\langle P, V\rangle(x) = 1] \geqslant \alpha(|x|)$ для любого $x \in L$ (условие *полноты* (completeness) протокола);
- $\Pr[\langle P', V\rangle(x) = 1] \leqslant \beta(|x|)$ для любого $x \in \{0,1\}^* \setminus L$ и для любого интерактивного вероятностного алгоритма P' (условие *корректности* (soundness) протокола).

В этом определении алгоритм P является моделью честного доказывающего, алгоритм V — честного проверяющего, а алгоритм P' — нечестного доказывающего. Алгоритм V возвращает на выходе 1, если и только если доказательство принимается.

Класс сложностей IP определяется как класс всех языков, для которых существует протокол интерактивного доказательства с границей полноты $2/3$ и границей корректности $1/3$. На самом деле границы $2/3$ и $1/3$ здесь могут быть заменены на другие в широких пределах.

Предложение 16.2 (см. также [Gol04, предложение 4.2.7], [AB07, лемма 8.7])**.** *Для произвольного языка $L \subseteq \{0,1\}^*$ следующие условия эквивалентны:*

(i) $L \in \mathrm{IP}$*;*

(ii) *существует вычислимая за полиномиальное от аргумента время функция $\gamma\colon \mathbb{N} \to \mathbb{Q}$ такая, что имеется протокол интерактивного доказательства для языка L с границей полноты $\gamma + 1/\operatorname{poly}$ и границей корректности $\gamma - 1/\operatorname{poly}$;*

(iii) *для любого полинома q существует протокол интерактивного доказательства для языка L с границей полноты $1 - 2^{-q}$ и границей корректности 2^{-q}.*

Доказательство предложения 16.2 остается за рамками настоящего курса ввиду сложности этого доказательства.

Определение 16.3 (абсолютно полный протокол интерактивного доказательства)**.** Протокол интерактивного доказательства (P, V) для языка L называется *абсолютно полным* (perfectly complete), если для любого $x \in L$ всегда выполняется равенство $\langle P, V\rangle(x) = 1$.

Другими словами, абсолютная полнота протокола интерактивного доказательства означает, что если $x \in L$, то честный доказывающий всегда принимает доказательство честного проверяющего.

Определение 16.4 (игра Артура и Мерлина)**.** Протокол интерактивного доказательства называется *игрой Артура и Мерлина* (Arthur-Merlin game), если в нем проверяющий (называемый Артуром) в каждом раунде выбирает строку $v \in_{\mathcal{U}} \{0,1\}^l$ при некотором $l \in \mathbb{N}$ (вообще говоря, l меняется от раунда к раунду) и посылает эту строку доказывающему (называемому Мерлином). Какие-либо другие сообщения от Артура к Мерлину не посылаются.

Имена персонажей игры Артура и Мерлина взяты из британского эпоса, в котором мудрец и волшебник Мерлин был наставником и помощником короля Артура.

Теорема 16.5 (см. также [GS86, GS89, FGM$^+$89], [AB07, лемма 8.7, замечание 8.17], [Gol04, предложение 4.2.7, подразд. 4.2.3])**.** *Пусть $L \in \mathrm{IP}$, а q — произвольный полином. Тогда существует абсолютно полная игра Артура и Мерлина для языка L с границей корректности 2^{-q}.*

Напомним, что PSPACE — это класс языков, распознаваемых детерминированными алгоритмами (т. е. машинами Тьюринга) с использованием объема памяти (т. е. числа ячеек), полиномиального от длины входа. Более подробно, язык $L \subseteq \{0,1\}^*$ принадлежит классу PSPACE, если и только если существует машина Тьюринга A, использующая при вычислении на произвольном входе $x \in \{0,1\}^*$ не более $\operatorname{poly}(|x|)$ ячеек на лентах и такая, что $A(x) = 1$ при всех $x \in L$ и $A(x) = 0$ при всех $x \in \{0,1\}^* \setminus L$. Класс IP может быть охарактеризован следующим образом.

Теорема 16.6 (см. [Sha90, Sha92, She92], [Gol08, теорема 9.4], [AB07, теорема 8.19], [Gol04, подразд. 4.2.3])**.** IP = PSPACE.

Доказательства теорем 16.5 и 16.6 в настоящем курсе не приводятся ввиду сложности этих доказательств.

Следующий пример появился уже в пионерских работах [GMR85, GMR89] (см. также [AB07, пример 8.9]).

Пример 16.7. Пусть язык QNR (QUADRATIC NONRESIDUES или КВАДРАТИЧНЫЕ НЕВЫЧЕТЫ) состоит из всех пар (m, y), где $m \in \mathbb{N}$, $m \geqslant 3$, а $y \in \mathbb{Z}_m^\star$ — квадратичный невычет по модулю m. Определим протокол интерактивного доказательства следующим образом:

Общий вход: (m, y), где $m \in \mathbb{N}$, $m \geqslant 3$, $y \in \mathbb{Z}_m^\star$.

1. Проверяющий выбирает $b \in_{\mathcal{U}} \{0, 1\}$ и $z \in_{\mathcal{U}} \mathbb{Z}_m^\star$, вычисляет $g = y^b z^2 \bmod m$ и посылает g доказывающему (от которого требуется найти b).

2. Доказывающий полагает b' равным 0, если g — квадратичный вычет по модулю m, и 1, если g — квадратичный невычет по этому модулю. Затем доказывающий посылает b' проверяющему.

3. Проверяющий принимает доказательство, если и только если $b' = b$.

Здесь и далее мы не описываем случаи, когда какой-либо нечестный участник отказывается выполнять протокол до конца или посылает сообщения, формат которых не соответствует протоколу. В этом случае честный участник останавливает выполнение протокола.

Если $(m, y) \in$ QNR, т. е. y — квадратичный невычет по модулю m, то очевидно, что честный проверяющий всегда принимает доказательство честного доказывающего. Пусть теперь y — квадратичный вычет по модулю m. Тогда при любом значении бита b вычет g распределен равномерно на множестве всех квадратичных вычетов по модулю m, принадлежащих группе $\mathbb{Z}_m^\star$. Это показывает, что g и b, рассматриваемые как случайные величины, независимы. Следовательно, произвольный нечестный доказывающий может найти b с вероятностью не более $1/2$. Таким образом, приведенный выше протокол является абсолютно полным протоколом интерактивного доказательства для языка QNR с границей корректности $1/2$.

17. Протоколы доказательства с нулевым разглашением

Говоря неформально, протокол интерактивного доказательства (P, V) для языка L называется протоколом *доказательства с нулевым разглашением* (zero-knowledge proof), если для любого интерактивного полиномиального вероятностного алгоритма V' существует полиномиальный вероятностный алгоритм S, называемый *симулятором* (simulator) для взаимодействия V' с P, такой, что семейства случайных величин $(\mathrm{view}_{V'}^P(x) \mid x \in L)$ и $(S(x) \mid x \in L)$ в некотором смысле неотличимы. Смысл условия нулевого разглашения состоит в том, чтобы нечестный проверяющий (моделью которого является V'), выполняя протокол с честным доказывающим, не мог получить никакой информации, кроме доступной ему и без взаимодействия с доказывающим.

Это неформальное определение может быть формализовано несколькими способами. Важнейшими видами нулевого разглашения являются вычислительно нулевое разглашение, статистически нулевое разглашение и абсолютно нулевое разглашение (см. ниже). Отметим, что термин «нулевое разглашение» используется не только в качестве обобщающего термина для различных видов этого понятия (как в настоящем курсе), но и как синоним вычислительно нулевого разглашения.

Определение 17.1 (протокол доказательства с вычислительно нулевым разглашением; см. также [Gol04, определения 4.3.3 и 4.3.2])**.** Протокол интерактивного доказательства (P, V) для бесконечного языка $L \subseteq \{0, 1\}^*$ называется протоколом *доказательства с вычислительно нулевым разглашением* (computational zero-knowledge proof), если для любого интерактивного полиномиального вероятностного алгоритма V' существует полиномиальный вероятностный алгоритм S такой, что семейства случайных величин $(\mathrm{view}_{V'}^P(x) \mid x \in L)$ и $(S(x) \mid x \in L)$ вычислительно неотличимы.

Определение 17.2 (протокол доказательства со статистически нулевым разглашением; см. также [Gol04, определение 4.3.4])**.** Протокол интерактивного доказательства (P, V) для бесконечного языка $L \subseteq \{0, 1\}^*$ называется протоколом *доказательства со статистически нулевым разглашением* (statistical zero-knowledge proof), если для любого интерактивного полиномиального вероятностного алгоритма V' существует полиномиальный вероятностный алгоритм S такой, что семейства случайных величин $(\mathrm{view}_{V'}^P(x) \mid x \in L)$ и $(S(x) \mid x \in L)$ статистически неотличимы.

В качестве синонима доказательства со статистически нулевым разглашением используется термин *доказательство с почти абсолютно нулевым разглашением* (almost perfect zero-knowledge proof).

В определениях 17.1 и 17.2 мы предполагаем, что язык L бесконечен. Это делается для того, чтобы избежать проблем с определением вычислительной и статистической неотличимости семейств случайных величин, индексированных элементами L. Кроме того, конечные языки для рассматриваемой теории не представляют интереса.

Определение 17.3 (протокол доказательства с абсолютно нулевым разглашением; см. также [Gol04, определение 4.3.1])**.** Протокол интерактивного доказательства (P, V) для языка $L \subseteq \{0,1\}^*$ называется протоколом *доказательства с абсолютно нулевым разглашением* (perfect zero-knowledge proof), если для любого интерактивного полиномиального вероятностного алгоритма V' существуют полиномиальный вероятностный алгоритм S такой, что при каждом $x \in L$ выполняются следующие условия:

- $\Pr[S(x) \neq \perp] \geqslant 1/\operatorname{poly}(|x|)$;
- случайная величина $S(x)$ при условии $S(x) \neq \perp$ распределена так же, как и $\operatorname{view}^P_{V'}(x)$.

Символ $\perp$ на выходе алгоритма S означает, что этому алгоритму не удалось смоделировать распределение случайной величины $\operatorname{view}^P_{V'}(x)$.

Как уже отмечалось, полиномиальный вероятностный алгоритм S из определений 17.1–17.3 называется *симулятором* (simulator) для взаимодействия V' с P. Отметим, что соответствующая симулятору машина Тьюринга иногда называется *моделирующей машиной*.

Приведем общие свойства трех видов нулевого разглашения, определенных выше.

- Очевидно, что нулевое разглашение для протокола интерактивного доказательства по существу является лишь свойством честного доказывающего.
- Если в определении нулевого разглашения заменить $\operatorname{view}^P_{V'}(x)$ на $\langle P, V'\rangle(x)$, то получится эквивалентное определение (см. также [Gol04, подразд. 4.12.4, упр. 10]).

Очевидно также, что из условия абсолютно нулевого разглашения (для бесконечного языка) следует условие статистически нулевого разглашения, а из последнего условия — условие вычислительно нулевого разглашения.

Через CZK (SZK и PZK) обозначается класс всех конечных языков вместе со всеми языками, для которых существует протокол интерактивного доказательства с вычислительно (соответственно, статистически и абсолютно) нулевым разглашением, имеющий границу полноты $2/3$ и границу корректности $1/3$. Иногда класс CZK обозначается через ZK (см., например, [Gol04, определение 4.3.5], [Gol08, определение 9.8]), а класс SZK — через APZK (см., например, [For87, For89]). Очевидно, что $\mathrm{PZK} \subseteq \mathrm{SZK} \subseteq \mathrm{CZK}$. Кроме того, справедлива следующая теорема, принадлежащая Бен-Ору и др.

Теорема 17.4 (см. также [Gol04, п. 4.4.1.3 и теорема 4.4.12], [Gol08, теорема 9.12], [BOGG$^+$90])**.** *Предположим, что существуют односторонние функции. Пусть L — бесконечный язык из* IP*, а q — произвольный полином. Тогда существует абсолютно полный протокол интерактивного доказательства с вычислительно нулевым разглашением для языка L, имеющий границу корректности 2^{-q}. В частности,* $\mathrm{CZK} = \mathrm{IP}$.

В связи с этой теоремой напомним, что IP = PSPACE (см. теорему 16.6). В то же время из некоторых результатов теории сложности вычислений (см. [For87, For89, AH87, BHZ87]) следует, что если $\mathrm{PH} \neq \Sigma^p_2$ (или, что эквивалентно, $\mathrm{PH} \neq \Pi^p_2$), то уже в NP и coNP есть языки, не принадлежащие SZK.

Для криптографии имеют особое значение те протоколы интерактивного доказательства для языка L, в которых доказывающий работает за полиномиальное время, если ему на собственный вход подается некоторая дополнительная информация. Такой дополнительной информацией может быть NP-доказательство для общего входа, если $L \in \mathrm{NP}$. Напомним определения. Пусть R — бинарное отношение на множестве $\{0,1\}^*$ (т. е. подмножество $\{0,1\}^* \times \{0,1\}^*$). Для каждого $x \in \{0,1\}^*$ положим $R(x) = \{y \in \{0,1\}^* \mid (x,y) \in R\}$. Отношение R называется NP-*отношением* (NP-relation), если оно распознаваемо некоторым полиномиальным детерминированным алгоритмом и $R(x) \subseteq \{0,1\}^{\leqslant \operatorname{poly}(|x|)}$ для всех $x \in \{0,1\}^*$. Если R — NP-отношение, то элементы множества

$R(x)$ (и только они) называются NP-*доказательствами* (NP-proof, NP-witness) для $x \in \{0,1\}^*$ относительно R. Положим также

$$L_R = \{x \in \{0,1\}^* \mid \exists\, y \in \{0,1\}^*\ (x,y) \in R\} = \{x \in \{0,1\}^* \mid R(x) \neq \varnothing\}.$$

Тогда NP — это класс всех языков вида L_R, где R пробегает всевозможные NP-отношения.

Следующая теорема принадлежит Гольдрайху, Микали и Вигдерсону.

Теорема 17.5 (см. также [Lub96, лекция 18], [Gol04, п. 4.4.1.3 и теорема 4.4.11], [Gol08, теорема 9.11], [GMW86, GMW91])**.** *Предположим, что существуют односторонние функции. Пусть R — NP-отношение такое, что соответствующий ему язык $L_R = \{x \mid \exists\, y\ (x,y) \in R\}$ бесконечен, а q — произвольный полином. Тогда существует абсолютно полный протокол интерактивного доказательства (P,V) с вычислительно нулевым разглашением для языка L_R, имеющий границу корректности 2^{-q}, в котором P работает за полиномиальное время на любом общем входе $x \in L_R$, если ему на собственный вход подается произвольное* NP-*доказательство для x относительно R.*

Доказательства теорем 17.4 и 17.5 остаются за рамками настоящего курса ввиду сложности этих доказательств.

Следующий пример является классическим (см. [GMW86, GMW91], [Gol04, подразд. 4.3.2], [Gol08, п. 9.2.2.1], [AB07, пример 9.15], [Lub96, теорема 18.2]).

Пример 17.6. Под словом «граф» в настоящем примере мы понимаем неориентированный граф без петель и кратных ребер с множеством вершин $\{1,\dots,n\}$ при некотором (не фиксированном) $n \in \mathbb{N}$. Такой граф может быть представлен для алгоритмов как матрицей смежности, так и списком ребер. Пусть язык GI (GRAPH ISOMORPHISM или ИЗОМОРФИЗМ ГРАФОВ) состоит из всех пар (G_0, G_1), где G_0 и G_1 — изоморфные графы. Через S_n в этом примере будет обозначаться группа всех перестановок множества $\{1,\dots,n\}$ ($n \in \mathbb{N}$). Если G — граф с множеством вершин $\{1,\dots,n\}$ и $\pi \in S_n$, то через $\pi(G)$ обозначается граф с тем же множеством вершин, в котором вершины i и j смежны тогда и только тогда, когда вершины $\pi^{-1}(i)$ и $\pi^{-1}(j)$ смежны в G. Другими словами, $\pi(G)$ — это такой единственный граф с множеством вершин $\{1,\dots,n\}$, что π — изоморфизм G на $\pi(G)$. Таким образом, для каждого $n \in \mathbb{N}$ задано действие группы S_n на множестве графов с множеством вершин $\{1,\dots,n\}$.

Определим протокол доказательства следующим образом:

Общий вход: (G_0, G_1), где G_0 и G_1 — графы с одним и тем же множеством вершин $\{1,\dots,n\}$. Если эти графы изоморфны, то через α обозначается произвольный изоморфизм G_0 на G_1.

1. Доказывающий выбирает $\pi \in_{\mathcal{U}} S_n$, вычисляет граф $H = \pi(G_1)$ и посылает H проверяющему.
2. Проверяющий выбирает $b \in_{\mathcal{U}} \{0,1\}$ и посылает b доказывающему (от которого требуется предъявить изоморфизм G_b на H).
3. Доказывающий полагает $\rho = \pi(\alpha)$, если $b = 0$, и $\rho = \pi$ в противном случае. Затем доказывающий посылает проверяющему ρ.
4. Проверяющий принимает доказательство, если и только если H — граф с множеством вершин $\{1,\dots,n\}$, $\rho \in S_n$ и $\rho(G_b) = H$.

Абсолютная полнота протокола очевидна. Кроме того, если изоморфизм α подан доказывающему на собственный вход, то доказывающий может выполнить протокол за полиномиальное время.

Пусть G_0 и G_1 не изоморфны. Тогда какой бы граф H ни послал нечестный доказывающий проверяющему в раунде 1, этот граф не может быть изоморфен обоим графам G_0 и G_1. Поэтому в данном случае проверяющий примет доказательство с вероятностью не более $1/2$. Таким образом, приведенный выше протокол является абсолютно полной игрой Артура и Мерлина для языка GI с границей корректности $1/2$.

Докажем теперь свойство абсолютно нулевого разглашения. Пусть V' — произвольный интерактивный полиномиальный вероятностный алгоритм. Определим вычисление алгоритма S на произвольном входе (G_0, G_1), где G_0 и G_1 — изоморфные графы с множеством вершин $\{1,\dots,n\}$, следующим образом:

1. Выбрать $c \in_{\mathcal{U}} \{0,1\}$ и $\sigma \in_{\mathcal{U}} S_n$ и вычислить граф $H = \sigma(G_c)$.

2. Запустить алгоритм V' на входе (G_0, G_1), используя H в качестве первого сообщения, посланного этому алгоритму (см. раунд 1 протокола). Пусть b — сообщение, посланное алгоритмом V' в ответ.

3. Если $b \notin \{0,1\}$, то положить $b = 1$.

4. Если $b = c$, то продолжить выполнение алгоритма V', послав ему сообщение σ. После окончания работы этого алгоритма возвратить (r, H, σ), где r — строка случайных битов, использованная V' при вышеуказанном вычислении. Если же $b \neq c$, то возвратить $\perp$.

Очевидно, что если $\widetilde{c} \in_{\mathcal{U}} \{0,1\}$ и $\widetilde{\sigma} \in_{\mathcal{U}} S_n$, то случайные величины $\widetilde{c}$ и $\widetilde{\sigma}(G_{\widetilde{c}})$ независимы. Это следует из того, что $\widetilde{\sigma}(G_0)$ и $\widetilde{\sigma}(G_1)$ распределены равномерно на множестве всех графов с множеством вершин $\{1, \ldots, n\}$, изоморфных G_0 (или G_1). Поэтому $\Pr[S(G_0, G_1) \neq \perp] = 1/2$. Легко также видеть, что случайная величина $S(G_0, G_1)$ при условии $S(G_0, G_1) \neq \perp$ распределена так же, как и $\mathrm{view}_{V'}^{P(\alpha)}(G_0, G_1)$. Таким образом, построенный протокол является протоколом доказательства с абсолютно нулевым разглашением.

Список литературы

[Сем88] И. А. Семаев. Построение неприводимых над конечным полем многочленов с линейно независимыми корнями. *Матем. сб.*, 135(4):520–532, 1988.

[Чис84] А. Л. Чистов. Построение конечного поля в полиномиальное время. Тезисы докладов VII Всесоюзной конф. по матем. логике, с. 196, Новосибирск, 1984.

[AB07] S. Arora and B. Barak. *Computational complexity: A modern approach*. Cambridge University Press, 2007.

[AH87] W. Aiello and J. Håstad. Perfect zero-knowledge languages can be recognized in two rounds. In *Proc. 28th Annu. IEEE Symp. on Found. of Comput. Sci.*, pages 439–448, 1987.

[BHZ87] R. B. Boppana, J. Håstad, and S. Zachos. Does co-NP have short interactive proofs? *Information Processing Letters*, 25(2):127–132, 6 May 1987.

[BOGG+90] M. Ben-Or, O. Goldreich, S. Goldwasser, J. Håstad, J. Kilian, S. Micali, and P. Rogaway. Everything provable is provable in zero-knowledge. In *Proc. CRYPTO'88*, volume 403 of *Lecture Notes in Comput. Sci.*, pages 37–56. Springer-Verlag, 1990.

[BS96] E. Bach and J. Shallit. *Algoritmic number theory. Volume 1 (Efficient algorithms)*. MIT Press, 1996.

[Dam87] I. B. Damgård. Collision free hash functions and public key signature schemes. In *Proc. EUROCRYPT'87*, volume 304 of *Lecture Notes in Comput. Sci.*, pages 203–216. Springer-Verlag, 1987.

[DH76] W. Diffie and M. E. Hellman. New directions in cryptography. *IEEE Trans. on Information Theory*, IT-22(6):644–654, November 1976. Русский перевод: У. Диффи, М. Э. Хеллман. Защищенность и имитостойкость. Введение в криптографию. ТИИЭР. 1979. Т. 67. № 3.

[FGM+89] M. Furer, O. Goldreich, Y. Mansour, M. Sipser, and S. Zachos. On completeness and soundness in interactive proof systems. In *Advances in Computing Research: A Research Annual*, volume 5 (Randomness and Computation), pages 429–442. JAI Press, 1989.

[For87] L. Fortnow. The complexity of perfect zero-knowledge. In *Proc. 19th Annu. ACM Symp. on Theory of Computing*, pages 204–209, 1987. Preliminary version of [For89].

[For89] L. Fortnow. The complexity of perfect zero-knowledge. In *Advances in Computing Research: A Research Annual*, volume 5 (Randomness and Computation), pages 327–343. JAI Press, 1989. Full version of [For87].

[GB08] S. Goldwasser and M. Bellare. Lecture notes on cryptography. Available at `http://cseweb.ucsd.edu/users/mihir/papers/gb.pdf`, July 2008.

[GGM84] O. Goldreich, S. Goldwasser, and S. Micali. How to construct random functions. In *Proc. 25th Annu. IEEE Symp. on Found. of Comput. Sci.*, pages 464–479, 1984. Preliminary version of [GGM86].

[GGM86] O. Goldreich, S. Goldwasser, and S. Micali. How to construct random functions. *J. of the ACM*, 33(4):792–807, October 1986. Full version of [GGM84].

[GL89] O. Goldreich and L. A. Levin. A hard-core predicate for all one-way functions. In *Proc. 21st Annu. ACM Symp. on Theory of Computing*, pages 25–32, 1989.

[GM84] S. Goldwasser and S. Micali. Probabilistic encryption. *J. of Computer and System Sci.*, 28(2):270–299, April 1984.

[GMR85] S. Goldwasser, S. Micali, and C. Rackoff. The knowledge complexity of interactive proof systems. In *Proc. 17th Annu. ACM Symp. on Theory of Computing*, pages 291–304, 1985. Preliminary version of [GMR89].

[GMR88] S. Goldwasser, S. Micali, and R. L. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM J. on Computing*, 17(2):281–308, April 1988.

[GMR89] S. Goldwasser, S. Micali, and C. Rackoff. The knowledge complexity of interactive proof systems. *SIAM J. on Computing*, 18(1):186–208, February 1989. Full version of [GMR85].

[GMW86] O. Goldreich, S. Micali, and A. Wigderson. Proofs that yield nothing but their validity and a methodology of cryptographic protocol design. In *Proc. 27th Annu. IEEE Symp. on Found. of Comput. Sci.*, pages 174–187, 1986. Preliminary version of [GMW91].

[GMW91] O. Goldreich, S. Micali, and A. Wigderson. Proofs that yield nothing but their validity or all languages in NP have zero-knowledge proof systems. *J. of the ACM*, 38(3):691–729, July 1991. Full version of [GMW86].

[Gol04] O. Goldreich. *Foundations of cryptography. Volume 1 (Basic tools). Volume 2 (Basic applications).* Cambridge University Press, 2001, 2004.

[Gol08] O. Goldreich. *Computational complexity: A conceptual perspective.* Cambridge University Press, 2008.

[GS86] S. Goldwasser and M. Sipser. Private coins versus public coins in interactive proof systems. In *Proc. 18th Annu. ACM Symp. on Theory of Computing*, pages 59–68, 1986. Preliminary version of [GS89].

[GS89] S. Goldwasser and M. Sipser. Private coins versus public coins in interactive proof systems. In *Advances in Computing Research: A Research Annual*, volume 5 (Randomness and Computation), pages 73–90. JAI Press, 1989. Full version of [GS86].

[Hås90] J. Håstad. Pseudo-random generators under uniform assumptions. In *Proc. 22nd Annu. ACM Symp. on Theory of Computing*, pages 395–404, 1990. Preliminary version of [HILL99].

[HILL91] J. Håstad, R. Impagliazzo, L. A. Levin, and M. Luby. Construction of a pseudo-random generator from any one-way function. Technical Report TR-91-068, International Computer Science Institute, December 1991.

[HILL99] J. Håstad, R. Impagliazzo, L. A. Levin, and M. Luby. Construction of a pseudo-random generator from any one-way function. *SIAM J. on Computing*, 28(4):1364–1396, 1999. Full version of [ILL89] and [Hås90].

[Hol06] T. Holenstein. Pseudorandom generators from one-way functions: A simple construction for any hardness. In *Proc. 3rd Theory of Cryptography Conference*, volume 3876 of *Lecture Notes in Comput. Sci.*, pages 443–461. Springer-Verlag, 2006.

[IL89] R. Impagliazzo and M. Luby. Pseudo-random number generator from any one-way function. Technical Report TR-89-002, International Computer Science Institute, February 1989.

[ILL89] R. Impagliazzo, L. A. Levin, and M. Luby. Pseudo-random generation from one-way functions. In *Proc. 21st Annu. ACM Symp. on Theory of Computing*, pages 12–24, 1989. Preliminary version of [HILL99].

[KK05] J. Katz and C.-Y. Koo. On constructing universal one-way hash functions from arbitrary one-way functions. Cryptology ePrint Archive (`https://eprint.iacr.org/`), Report 2005/328, 2005.

[Lam79] L. Lamport. Constructing digital signatures from one-way functions. Technical Report CSL-98, SRI International, October 1979.

[LR86] M. Luby and C. Rackoff. Pseudo-random permutation generators and cryptographic composition. In *Proc. 18th Annu. ACM Symp. on Theory of Computing*, pages 356–363, 1986. Preliminary version of [LR88].

[LR88] M. Luby and C. Rackoff. How to construct pseudorandom permutations from pseudorandom functions. *SIAM J. on Computing*, 17(2):373–386, April 1988. Full version of [LR86].

[Lub96] M. Luby. *Pseudorandomness and cryptographic applications.* Princeton University Press, 1996.

[NY89] M. Naor and M. Yung. Universal one-way hash functions and their cryptographic applications. In *Proc. 21st Annu. ACM Symp. on Theory of Computing*, pages 33–43, 1989.

[Rab79] M. O. Rabin. Digitalized signatures and public key functions as intractable as factoring. Technical Report TR-212, Lab. Comput. Sci., MIT, Cambridge, Massachusetts, USA, January 1979.

[Rom90] J. Rompel. One-way functions are necessary and sufficient for secure signatures. In *Proc. 22nd Annu. ACM Symp. on Theory of Computing*, pages 387–394, 1990.

[RSA78] R. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public key cryptosystems. *Communications of the ACM*, 21(2):120–126, February 1978.

[Sha49] C. E. Shannon. Communication theory of secrecy systems. *Bell System Techn. J.*, 28(4):656–715, 1949. Русский перевод: Теория связи в секретных системах. В кн.: К. Шеннон. Работы по теории информации и кибернетике. М.: ИЛ, 1963. С. 333–402.

[Sha90] A. Shamir. IP = PSPACE. In *Proc. 31st Annu. IEEE Symp. on Found. of Comput. Sci.*, pages 11–15, 1990. Preliminary version of [Sha92].

[Sha92] A. Shamir. IP = PSPACE. *J. of the ACM*, 39(4):869–877, October 1992. Full version of [Sha90].

[She92] A. Shen. IP = PSPACE: Simplified proof. *J. of the ACM*, 39(4):878–880, October 1992.

[Sho90] V. Shoup. New algorithms for finding irreducible polynomials over finite fields. *Mathematics of computation*, 54(189):435–447, January 1990.

[Sho08] V. Shoup. *A computational introduction to number theory and algebra.* Cambridge University Press, 2nd edition, 2008. Electronic version is available at `http://shoup.net/ntb/`.

[Sim98] D. R. Simon. Finding collisions on a one-way street: Can secure hash functions be based on general assumptions? In *Proc. EUROCRYPT'98*, volume 1403 of *Lecture Notes in Comput. Sci.*, pages 334–345. Springer-Verlag, 1998.

[Ver26] G. S. Vernam. Cipher printing telegraph systems for secret wire and radio telegraphic communications. *J. of the American Institute of Electrical Engineers*, 45:109–115, 1926.

[Yao82] A. C. Yao. Theory and applications of trapdoor functions. In *Proc. 23rd Annu. IEEE Symp. on Found. of Comput. Sci.*, pages 80–91, 1982.