Записки научных
семинаров ПОМИ
Том 399, 2012 г.

А. П. Давыдов, С. И. Николенко

СХЕМНАЯ СЛОЖНОСТЬ ЛИНЕЙНЫХ ФУНКЦИЙ: МЕТОД ИСКЛЮЧЕНИЯ ГЕЙТОВ И НАДЕЖНОСТЬ В СЛАБОМ СМЫСЛЕ

§1. Введение

В современной криптографии практически нет доказуемо надёжных конструкций с открытым ключом. Ни самые первые криптографические примитивы – протокол согласования ключа Диффи–Хеллмана [8] и криптосистема RSA [27], – ни более поздние конструкции не привели к доказанной надёжности ни одного криптографического протокола с открытым ключом. Существуют надёжные криптографические примитивы с секретным ключом; в частности, современная криптография фактически началась с одного из таких примитивов – одноразовых блокнотов [29, 34]; но даже доказуемо надёжного протокола с секретным ключом, где длина сообщения превышала бы длину ключа, построить не удаётся.

Конечно, безусловное доказательство надёжности было бы трудно получить, ведь из него неизбежно следовало бы, что $\mathrm{P} \neq \mathrm{NP}$: в любой криптосистеме с открытым ключом противник всегда может угадать случайные биты отправителя и текст сообщения, после чего подтвердить догадку, получив тот же самый шифр. Но дела обстоят ещё хуже: нет и условных доказательств, которые могли бы установить связь между какими-либо естественными структурными предположениями (такими, как $\mathrm{P}{\neq}\mathrm{NP}$ или $\mathrm{BPP}{\neq}\mathrm{NP}$) и криптографической надёжностью. Недавно появившиеся разработки основанных на решётках криптосистем, связывающих криптографическую надёжность

Ключевые слова: надёжность в слабом смысле, схемная сложность, функции с секретом, доказуемая надёжность.

Работа первого автора была поддержана стипендиальной программой “Яндекса” и грантом РФФИ 11-01-12135-офи-м-2011. Работа второго автора была поддержана грантами РФФИ 11-01-12135-офи-м-2011 и 11-01-00760-а, а также грантом Президента РФ для ведущих научных школ НШ-3229.2012.1 и грантом Президента РФ для молодых кандидатов наук МК-6628.2012.1.

со сложностью в худшем случае, в действительности имеют дело с задачами, NP-трудность которых не доказана и, более того, маловероятна [2, 9, 25, 26].

Известны полные конструкции криптографических примитивов: как односторонних функций [11, 23, 39, 40], так и криптосистем с открытым ключом [12, 13]. Однако они тоже не позволяют связать криптографическую надёжность с ключевыми предположениями традиционной структурной теории сложности. Кажется, что классической современной криптографии ещё очень далеко до каких-либо *доказуемо* надёжных конструкций. Более того, асимптотическая природа имеющихся утверждений о полноте не позволяет утверждать, что ту или иную криптографическую конструкцию трудно взломать для ключей какой-либо фиксированной длины. Однако именно ключи фиксированной длины интересуют конечных пользователей криптографических конструкций, ведь их протокол защищён ключом одной конкретной длины. Асимптотические оценки сложности алгоритмов, как правило, действительно имеют прямое отношение к их практической сложности, но доказательства асимптотической надёжности может оказаться недостаточно для практической надёжности: почему трудно взломать RSA для 2048-битных чисел? Может быть, кто-то сможет построить устройство не слишком большого размера, которое взламывает тот или иной протокол для всех ключей одной и той же реально использующейся длины? Никаких теоретических препятствий к этому формальные криптографические определения не представляют: конкретные длины ключей вообще не рассматриваются в принципиально асимптотической теории. Сложность задач, которые уже долгое время исследовались в криптографическом контексте (разложение на множители и дискретный логарифм), действительно представляется "равномерной" по длинам ключей, но для этих задач никаких суперполиномиальных нижних оценок сложности не известно, и получение их в ближайшем будущем крайне маловероятно. Другие опасности асимптотического подхода заключаются в неаккуратном использовании сведéний в доказательствах надёжности (см. [19–21] и комментарии к этим вызвавшим противоречивые отклики публикациям).

В связи с этим представляется, что идеальной моделью вычислений для криптографических нужд является *схемная сложность* – одна из немногих моделей вычислений, в которых возможны доказательства *конкретных*, а не асимптотических оценок сложности. Например,

Л. Стокмайер в своей диссертации привёл функцию, любая реализация которой при помощи бинарной булевой схемы на входах размера $\leqslant 616$ должна иметь не менее 10^{123} гейтов [3,31,33]. Данная модель достаточно хорошо изучена: основные результаты в классической схемной сложности относятся к 1980-м гг. и раньше; в них значительна заслуга отечественных учёных [5, 24, 32, 33, 41, 42, 44–46, 48–53]. Однако, хотя схемная сложность выглядит перспективной моделью вычислений, очевидным недостатком такого подхода является то, что, несмотря на многолетние исследования, мы до сих пор очень плохо понимаем, как доказывать нижние оценки в этой модели. Наилучшие известные нижние оценки схемной сложности в произвольном базисе для явных функций без дополнительных ограничений линейны от числа входов, да и константы там совсем небольшие: так, лучшая нижняя оценка для булевой функции $\mathbb{B}^n \to \mathbb{B}$ составляет всего лишь $3n - o(n)$ [5,24,32,36]. В последние годы центр усилий в теории схемной сложности сместился на результаты, связанные со схемами ограниченной глубины и/или ограниченным набором вычисляемых в них функций [1, 3, 6, 10, 14, 30, 35, 37, 38, 47], однако в криптографическом контексте мы вряд ли сможем убедить противника использовать, скажем, схемы только глубины 64 и не больше.

В этой работе мы продолжаем исследование вопроса о том, как доказать наличие *хоть какой-нибудь* (пусть линейной) разницы между схемной сложностью честного декодирования и взлома. В 1992 году Ален Хильтген [15] сконструировал функцию, для которой такое доказательство найти удалось: функцию Хильтгена почти *вдвое* (в $2 - o(1)$ раз) труднее обратить, чем вычислить. Его пример является линейной функцией над $GF(2)$ с матрицей, в которой мало ненулевых элементов, в то время как в обратной к ней матрице ненулевых элементов много. Сложность обращения при этом следует из простого наблюдения Ламаньи и Сэвиджа [22, 28]: каждый бит выхода нетривиально зависит от многих переменных, и эти биты все разные, следовательно, можно доказать нижнюю оценку на сложность вычисления их всех вместе взятых. Хильтген продолжил исследования доказуемо надёжных односторонних функций в [16, 17].

Эта работа была продолжена вторым автором, Э. А. Гиршем и О. Меланич [18,43]. В [18] была построена первая (линейная) конструкция функций с секретом (trapdoor functions), доказуемо надёжных в

слабом смысле, с порядком надёжности $\frac{25}{22}$, а в [43] было построено семейство нелинейных функций с секретом с порядком надёжности $\frac{7}{5}$.

Настоящая работа посвящена схемной сложности линейных функций и новым конструкциям линейных функций с секретом, доказуемо надёжных в слабом смысле. В разделе 2 мы вводим основные определения. В разделе 3 приводится (неконструктивное) доказательство того, что среди линейных функций существуют функции нелинейной схемной сложности, т.е. того, что линейные функции действительно представляют интерес с точки зрения нижних оценок схемной сложности. В разделе 4 подробно анализируется метод исключения гейтов (gate elimination) для случая схем, реализующих линейные функции, а в разделе 5 результаты раздела 4 применяются для получения новых конструкций линейных функций с секретом, доказуемо надёжных в слабом смысле, и уточнения их оценок надёжности. В результате мы улучшаем результат [18], доказывая следующую теорему.

Теорема 1. *Существует последовательность $\lambda_n \to 2$ и линейная функция с секретом, доказуемо надёжная в слабом смысле, с длиной ключа $\mathrm{pi}(n) = \mathrm{ti}(n) = n$, длиной входа и выхода $c(n) = m(n) = \lceil \lambda_n n \rceil$ и порядком надёжности $\frac{5}{4}$.*

Следует сразу отметить, что и в этой работе, и в других работах о примитивах, доказуемо надёжных в слабом смысле, строится одноразовая криптосистема (функция с секретом) с длиной сообщения, лишь в два раза превышающей длину ключа, и оценки сложности тоже линейны. Поэтому, конечно, практического криптографического значения результаты о доказуемо надёжных в слабом смысле криптографических примитивах не имеют, и наши результаты относятся скорее к области теории схемной сложности, чем к криптографии. Эта работа является журнальной версией статьи [7].

§2. Определения и обозначения

Прежде всего дадим точное определение схемы. Обозначим через $\mathbb{B}_{n,m}$ множество всех 2^{m2^n} функций $f : \mathbb{B}^n \to \mathbb{B}^m$, где $\mathbb{B} = \{0,1\}$ – поле из двух элементов.

Определение 1. *Пусть Ω – некоторое множество булевых функций $f : \mathbb{B}^m \to \mathbb{B}$ (m может быть разным для разных f). Тогда* Ω-схема *– это ациклический направленный граф с метками, состоящий из вершин трех типов:*

- *двух выделенных вершин входящей степени* 0, *маркированных как* 0 *и* 1,
- *вершин входящей степени* 0 (*вершин, в которые не входят рёбра*), *маркированных одной из переменных* $x_1, \ldots, x_n$,
- *вершин, маркированных одной из функций* $f \in \Omega$, *в которые входит столько рёбер, какова арность этой функции.*

Вершины первого типа называются константами, *второго* – входами, *а третьего* – гейтами[1]. Размер *схемы – это число гейтов в ней.*

Каждый гейт Ω-схемы вычисляет некоторую булеву функцию, константы вычисляют тождественные 0 и 1.

Соответственно, схемная сложность функции $f : \mathbb{B}^n \to \mathbb{B}^m$ в базисе Ω определяется как размер минимальной Ω-схемы, которая *вычисляет* функцию f, т.е. схемы, в которой есть m вершин (гейтов, входов или констант), вычисляющих результат применения компонент функции f ко входным битам. Чтобы можно было без оговорок устранить унарные гейты, будем считать, что в гейте (переменной) вычисляется как сам результат его функции (значение входной переменной), так и его отрицание. Наша модель вычислений – это булевы схемы с произвольными бинарными гейтами; иными словами, каждый гейт схемы маркируется одной из 16 булевых функций из $\mathbb{B}_{2,1}$. В дальнейшем через $C(f)$ будет обозначаться схемная сложность f в базисе $\mathbb{B}_{2,1}$, состоящем из всех бинарных булевых функций. Мы будем предполагать, что каждый гейт в этой схеме зависит от обоих входов, т.е. нет гейтов, маркированных константами и унарными функциями Id и $\neg$. Это не умаляет общности, потому что такие гейты легко исключить из нетривиальной схемы, не увеличивая её размер. Через $C_\alpha(f)$ будем обозначать минимальный размер схемы, которая вычисляет функцию $f \in \mathbb{B}_{n,m}$ на более чем доле α её входов (длины n). Очевидно, $C_\alpha(f) \leqslant C_\beta(f)$ для всех функций f и всех $\alpha \leqslant \beta$.

А. Хильтген ввёл для каждой *обратимой* функции n переменных $f \in \mathbb{B}_{n,n}$ понятие *меры необратимости* (measure of one-wayness)

$$M_F(f) = \frac{C(f^{-1})}{C(f)}$$

(напомним, что $C(f)$ – размер минимальной схемы с бинарными гейтами, реализующей f). Результаты Хильтгена заключались в том,

[1] В русскоязычной литературе встречается термин "*вентиль*"; лет сорок назад он был общеупотребителен.

что он нашёл последовательности функций $\{f_n\}_{n=1}^{\infty}$ с нетривиальными (большими единицы) константами

$$\lim_{n\to\infty} \inf M_F(f_n),$$

которые Хильтген называет *порядком необратимости* (order of one-wayness).

От односторонних функций перейдём к функциям с секретом.

Определение 2. *Зафиксируем функции* $\mathrm{pi}, \mathrm{ti}, m, c : \mathbb{N} \to \mathbb{N}$. Семейство кандидатов в функции с секретом *представляет собой последовательность троек* $\mathcal{C} = \{(\mathrm{Seed}_n, \mathrm{Eval}_n, \mathrm{Inv}_n)\}_{n=1}^{\infty}$, *где:*

- $\{\mathrm{Seed}_n\}_{n=1}^{\infty}$ – *это семейство схем порождения ключей*
$$\mathrm{Seed}_n : \mathbb{B}^n \to \mathbb{B}^{\mathrm{pi}(n)} \times \mathbb{B}^{\mathrm{ti}(n)},$$
- $\{\mathrm{Eval}_n\}_{n=1}^{\infty}$ – *это семейство вычисляющих функцию схем*
$$\mathrm{Eval}_n : \mathbb{B}^{\mathrm{pi}(n)} \times \mathbb{B}^{m(n)} \to \mathbb{B}^{c(n)},\ a$$
- $\{\mathrm{Inv}_n\}_{n=1}^{\infty}$ – *это семейство обращающих функцию схем*
$$\mathrm{Inv}_n : \mathbb{B}^{\mathrm{ti}(n)} \times \mathbb{B}^{c(n)} \to \mathbb{B}^{m(n)},$$

причём для каждого n, *каждого* $s \in \mathbb{B}^n$ (*начального числа генератора*) *и каждого* $t \in \mathbb{B}^{m(n)}$ (*сообщения*)

$$\mathrm{Inv}_n(\mathrm{Seed}_{n,2}(s), \mathrm{Eval}_n(\mathrm{Seed}_{n,1}(s), t)) = t,$$

где $\mathrm{Seed}_{n,1}(s)$ *и* $\mathrm{Seed}_{n,2}(s)$ – *первые* $\mathrm{pi}(n)$ *бит* (*"публичная информация", public information*) *и последние* $\mathrm{ti}(n)$ *бит* (*"секрет", trapdoor information*) *выхода схемы* $\mathrm{Seed}_n(s)$, *соответственно.*

Неформально говоря, здесь число n – это параметр надёжности функции с секретом, длина начального числа генератора случайных чисел. Длину входа функции с секретом мы обозначили через $m(n)$, через $c(n)$ – длину её выхода, а через $\mathrm{pi}(n)$ и $\mathrm{ti}(n)$ – длину публичной информации и секрета соответственно. Мы называем такое семейство функций "кандидатом", потому что в определении 2 ничего не говорится о надёжности, а только вводятся обозначения для размерностей и устанавливается корректность обращения.

Чтобы понять, насколько функция надёжна, нужно ввести понятие взлома функции. Неформально говоря, противник должен обратить функцию, не зная секрета.

Определение 3. *Схема* Adv_n успешно обращает *семейство кандидатов в функции с секретом* $\{\mathrm{Seed}_n, \mathrm{Eval}_n, \mathrm{Inv}_n\}$ *на входах длины* n, *если для равномерного распределения* U, *взятого по* $s \in \mathbb{B}^n$ *и* $m \in \mathbb{B}^{m(n)}$ (*по начальным числам генератора и входам*),

$$\Pr_{(s,m)\in U}[\mathrm{Adv}_n(\mathrm{Seed}_{n,1}(s), \mathrm{Eval}_n(\mathrm{Seed}_{n,1}(s), m)) = m] > \frac{1}{2}.$$

Заметим, что в действительности мы докажем даже более сильный факт, а именно надёжность конструкций в слабом смысле для *любого* начального числа генератора s, а не только для большинства таких чисел.

Определение 4. *Будем говорить, что семейство кандидатов в функции с секретом* $\mathcal{C} = \{(\mathrm{Seed}_n, \mathrm{Eval}_n, \mathrm{Inv}_n)\}_{n=1}^{\infty}$ *имеет* порядок надёжности $k \in \mathbb{R}$, *если для каждой такой последовательности схем* $\{\mathrm{Adv}_n\}_{n=1}^{\infty}$, *в которой* Adv_n *успешно обращает* $\mathcal{C}$ *на входах длины* n,

$$\liminf_{n\to\infty} \min\left\{\frac{C(\mathrm{Adv}_n)}{C(\mathrm{Seed}_n)}, \frac{C(\mathrm{Adv}_n)}{C(\mathrm{Eval}_n)}, \frac{C(\mathrm{Adv}_n)}{C(\mathrm{Inv}_n)}\right\} \geqslant k.$$

Иначе говоря,

$$\liminf_{n\to\infty} \min\left\{\frac{C_{1/2}(f_{\mathrm{pi}(n)+c(n)})}{C(\mathrm{Seed}_n)}, \frac{C_{1/2}(f_{\mathrm{pi}(n)+c(n)})}{C(\mathrm{Eval}_n)}, \frac{C_{1/2}(f_{\mathrm{pi}(n)+c(n)})}{C(\mathrm{Inv}_n)}\right\} \geqslant k,$$

где функция $f_{\mathrm{pi}(n)+c(n)} \in \mathbb{B}_{\mathrm{pi}(n)+c(n),m(n)}$ *отображает*

$$(\mathrm{Seed}_{n,1}(s), \mathrm{Eval}_n(\mathrm{Seed}_{n,1}(s), m)) \mapsto m.$$

Приведём несколько простых примеров. Если секретного ключа нет вовсе ($\mathrm{pi}(n) = 0$), то каждое семейство кандидатов в функции с секретом $\{(\mathrm{Seed}_n, \mathrm{Eval}_n, \mathrm{Inv}_n)\}_{n=1}^{\infty}$ имеет порядок надёжности 1, т.к. последовательность схем $\{\mathrm{Inv}_n\}_{n=1}^{\infty}$ успешно его обращает. Если

$$\{(\mathrm{Seed}_n, \mathrm{Eval}_n, \mathrm{Inv}_n)\}_{n=1}^{\infty}$$

реализуют функцию с секретом в обычном криптографическом, не в слабом смысле, то $k = \infty$. Более того, $k = \infty$ даже если оценки на размер схем противника просто более чем линейно превышают оценки на размер схем честных участников протокола, например, если противнику требуется $O(n \log n)$ гейтов, а честное участие линейно. К сожалению, даже таких оценок на размер схем из произвольных бинарных гейтов, реализующих конкретные булевы функции, пока не известно.

Следует явно отметить, что мы говорим исключительно об *одноразовой* надёжности. Противник может использовать меньшее число гейтов, обращая семейство кандидатов в функции с секретом второй раз для той же пары ключей: например, он может вычислить секретную информацию и успешно использовать её вторично. Таким образом, в наших конструкциях требуется для каждого нового входа выбирать новую пару ключей (фактически, новое начальное число генератора).

§3. Неконструктивные оценки схемной сложности линейных функций

Непосредственным подсчётом можно показать, что среди всех функций от n переменных найдутся функции со схемной сложностью не менее $\frac{1}{n}2^n$. Возникает естественный вопрос: возможны ли нелинейные оценки для линейных функций? Оказывается, что линейные функции с нелинейными нижними оценками действительно существуют. Отсылки к этому результату приводятся в [4, 16], но нам не удалось найти подробное доказательство в литературе; мы приводим доказательство здесь, а также уточняем результат, получая точные значения констант.

Теорема 2.

(1) *Для любого n найдется такая константа δ_n, что схемная сложность всех линейных булевых функций $\phi : \{0,1\}^n \to \{0,1\}^n$ не превосходит $\delta_n \frac{n^2}{\log n}$, и $\lim\limits_{n\to\infty} \delta_n = 1$.*

(2) *Для любого $\epsilon > 0$ существует такое $n_0(\epsilon)$, что для всякого $n > n_0(\epsilon)$ найдётся линейная булева функция $\phi : \{0,1\}^n \to \{0,1\}^n$, чья схемная сложность превосходит $(1-\epsilon)\frac{n^2}{2\log n}$.*

Доказательство. 1. *Верхняя оценка.* Пусть A – матрица, соответствующая функции ϕ. Положим сначала n равным степени двойки. Мы будем вычислять A следующим образом. Введём переменную c_i, которая будет пробегать все возможные строки длины $l = q\log n$, где q – константа, которую мы определим позже. Обозначим входы ϕ как $x = \begin{pmatrix} x_1 & \cdots & x_n \end{pmatrix}$ и предвычислим значения всех возможных комбинаций $c_i \cdot \begin{pmatrix} x_{j+1} \\ \cdots \\ x_{j+l} \end{pmatrix}$, где j кратно l. Чтобы подсчитать каждую такую комбинацию, достаточно $q\log n$ гейтов, так как c_i – фиксированная строка, которую можно жёстко зашить в структуру схемы. Таким образом,

число гейтов, используемых в процессе предвычисления, можно оценить сверху как

$$2^{q\cdot\log n}\cdot\frac{n}{q\cdot\log n}\cdot(q\log n-1)\leqslant n^q\cdot n=n^{q+1}.$$

Пусть $A_1,A_2,\cdots,A_n$ – столбцы A. Для того чтобы найти $A\cdot x$, подсчитаем

$$A\cdot x=\begin{pmatrix}A_1&\cdots&A_l\end{pmatrix}\begin{pmatrix}x_1\\\vdots\\x_l\end{pmatrix}\oplus\begin{pmatrix}A_{l+1}&\cdots&A_{2l}\end{pmatrix}\begin{pmatrix}x_{l+1}\\\vdots\\x_{2l}\end{pmatrix}\oplus\ldots\oplus\begin{pmatrix}A_{n-l+1}&\cdots&A_n\end{pmatrix}\begin{pmatrix}x_{n-l+1}\\\vdots\\x_n\end{pmatrix}.$$

Каждое из произведений, участвующих в этой формуле, было вычислено ранее, на этапе предвычисления, так что всё, что требуется сделать – подать на вход результат соответствующего гейта (никаких дополнительных гейтов не нужно). После этого потребуются еще $n\cdot(\frac{n}{l}-1)$ гейтов для того, чтобы подсчитать значение XOR. Таким образом, общее число гейтов, необходимых для вычисления результата, равно

$$n^{q+1}+n\cdot\left(\frac{n}{q\cdot\log n}-1\right)=n^{q+1}+\frac{n^2}{q\cdot\log n}-n.$$

Выбирая $q=\frac{1}{1+\epsilon}$, получим верхнюю оценку схемной сложности $(1+\epsilon)\frac{n^2}{\log n}+o\left(\frac{n^2}{\log n}\right)$ для сколь угодно малых ϵ.

Если же n – не степень двойки, то часть матрицы, относящуюся ко входам с первого по $\lfloor n/(q\cdot\log n)\rfloor\cdot q\cdot\log n$, обсчитаем вышеописанным способом, а оставшуюся подсчитаем грубой силой, потратив порядка $q\cdot\log n\cdot n$ гейтов.

2. *Нижняя оценка.* Нижнюю оценку получим подсчётом. Пусть $q=(1-\epsilon)\frac{n^2}{2\log n}$. Оценим T – число схем размера $\leqslant q$. Каждую схему можно описать, задав по два входа для каждого гейта и его тип (один из шестнадцати возможных). Таким образом, для $q>n$

$$\begin{aligned}T&\leqslant\frac{q^n\cdot(16\cdot(n+q)^2))^q}{q!}\leqslant q^n\cdot(16e)^q\cdot\frac{(n+q)^{2q}}{q^q}\\&\leqslant q^n\cdot(16e)^q\cdot\frac{(2q)^{2q}}{q^q}=q^n\cdot(64e)^q\cdot\frac{q^{2q}}{q^q}\\&=q^n\cdot(64\cdot e\cdot q)^q\leqslant(64\cdot e\cdot q)^{q+n}\leqslant(n^2)^{(1-\epsilon)\cdot\frac{n^2}{2\log n}+n}\\&=(2^{2\log n})^{(1-\epsilon)\cdot\frac{n^2}{2\log n}+n}=2^{(1-\epsilon)n^2+2n\log n}.\end{aligned}$$

Но общее число линейных булевых функций от n аргументов составляет 2^{n^2}, что превосходит $2^{(1-\epsilon)n^2+2n\log n}$. Следовательно, среди них найдется функция, чья схемная сложность превосходит $(1-\epsilon)\frac{n^2}{2\log n}$. $\square$

§4. Метод исключения гейтов для линейных функций

Метод исключения гейтов (*gate elimination*) – фактически единственный известный метод для получения нижних оценок на схемную сложность в произвольном базисе. Он использовался для получения всех ранее известных нижних оценок [5,24,32,36]. Основная идея этого метода заключается в следующем. Рассмотрим функцию f и схему C – минимальную из всех схем, вычисляющих f, – и зададим значение c для одного из входов x, получив тем самым схему, вычисляющую функцию $f\,|_{x=c}$. Теперь исходную схему C можно упростить, потому что гейты, входами для которых был x, стали либо унарными (а отрицание можно перенести в следующий гейт), либо константными (тогда и некоторые из последующих гейтов удастся исключить). Определив, сколько гейтов мы можем исключать на каждом шаге, мы можем индуктивно повторять эту операцию до тех пор, пока имеется хотя бы одна переменная, устранение которой поможет исключить должное число гейтов. Число гейтов, исключённых таким образом, является естественной оценкой для схемной сложности функции f.

Часто удается выделить случай, когда исключаемый гейт нелинеен (например, AND или OR). Тогда можно выбрать значение для его входа таким образом, чтобы гейт обратился в константу и, как следствие, позволил бы исключить дополнительные гейты на этом шаге. Однако в этой работе мы рассматриваем метод исключения гейтов для *линейных* функций и поэтому не можем предполагать наличие нелинейных гейтов в схеме. Вопрос о том, можно ли эффективнее вычислять линейные функции при помощи нелинейных гейтов, остаётся интересной открытой проблемой в теории схемной сложности.

Процесс исключения гейтов в линейном случае может быть сведен к двум простым идеям. Идея 1 проста и используется давно, тогда как идея 2 является новым результатом и будет использована при получении конструкции нового слабозащищённого линейного примитива в разделе 5. Идея 2 обобщает идею 1, но мы всё равно приведём идею 1: она проще, но при этом совмещает все ранее известные приёмы исключения гейтов, работающие в линейном случае.

Так как мы работаем с линейными функциями, результаты будет удобно формулировать в терминах матриц над $\mathbb{F}_2$; под схемной сложностью матрицы $C_\alpha(A)$ мы будем понимать схемную сложность соответствующей функции.

Обозначим через A_{-i} матрицу A без i-го столбца. Заметим, что если матрице A соответствует функция f, то матрице A_{-i} соответствует функция $f\,|_{x_i=0}$. Если в матрице A присутствует нулевой столбец A_i, то соответствующая функция не зависит от входа x_i; далее мы будем считать, что нулевые столбцы в матрицах отсутствуют; такие матрицы мы будем называть *невырожденными*.

Идея 1. *Допустим, что в течение n шагов мы можем удалять хотя бы по одному гейту. Тогда $C(f) \geqslant n$.*

Теорема 3. *Зафиксируем вещественное число $\alpha \in [0,1]$. Пусть $\mathcal{P} = \{P_n\}_{n=1}^{\infty}$ – набор предикатов, определённых на матрицах над $\mathbb{F}_2$, обладающих следующими свойствами:*

- *если выполняется $P_1(A)$, то $C_\alpha(A) \geqslant 1$;*
- *если выполняется $P_n(A)$, то и $P_m(A)$ выполняется для любого $1 \leqslant m \leqslant n$;*
- *если выполняется $P_n(A)$, то для любого i выполняется $P_{n-1}(A_{-i})$.*

Тогда для любой матрицы A, содержащей хотя бы $n+1$ столбец, если выполняется $P_n(A)$, то $C_\alpha(A) \geqslant n$.

Доказательство. Доказательство ведётся по индукции по индексу i. Из первого свойства $\mathcal{P}$ сразу же получаем базу индукции, а с помощью остальных получим шаг индукции. Рассмотрим схему C, реализующую A. Из монотонности $\mathcal{P}$ и базы индукции можно заключить, что схема нетривиальна, а значит, в ней найдётся хотя бы один гейт. Рассмотрим переменную x_i, входящую в первый гейт (здесь и далее под словами "первый гейт" мы будем понимать один из тех гейтов, входы которых соединены только со входами схемы, но не с выходами других гейтов, т.е. гейт, первый в некотором топологическом порядке). Заметим, что если C вычисляет f на доле α входов, то для некоторой константы $c \in \{0,1\}$ $C\,|_{x_i=c}$ вычисляет $f\,|_{x_i=c}$ на доле α входов. Если мы подставим значение c вместо соответствующей переменной, мы получим схему $C\,|_{x_i=c}$, размер которой не более $\mathrm{size}(C)-1$ гейтов и которая вычисляет A_{-i} хотя бы на доле α входов. $\square$

Несмотря на простоту этой теоремы, до недавнего времени это был, по сути, единственный доступный инструмент для оценки схемной сложности. Обычно для оценки сложности использовалось ещё более простое утверждение, известное с середины 1970 гг.

Предложение 1 ([22,28]; [15, Теоремы 3 и 4]; [18, Предложение 1])**.**

(1) *Пусть* $f : \mathbb{B}^n \to \mathbb{B}$ *нетривиально зависит от каждой из* n *переменных, т.е. для любого* i *найдутся такие значения* $a_1, \ldots, a_{i-1}, a_{i+1}, \ldots, a_n \in \mathbb{B}$, *что* $f(a_1, \ldots, a_{i-1}, 0, a_{i+1}, \ldots, a_n) \neq f(a_1, \ldots, a_{i-1}, 1, a_{i+1}, \ldots, a_n)$. *Тогда* $C(f) \geqslant n - 1$.

(2) *Пусть* $f = (f^{(1)}, \ldots, f^{(m)}) : \mathbb{B}^n \to \mathbb{B}^m$, *где* $f^{(k)}$ *– k-я компонента* f. *Если* m *компонент* $f^{(i)}$ *попарно различны, и для каждой из них* $C(f^{(i)}) \geqslant c \geqslant 1$, *то* $C(f) \geqslant c + m - 1$.

Доказательство предложения 1 приводится, например, в [18]. Первая часть этого утверждения, конечно, тривиальна для любых функций, но всё же заметим для полноты картины, что для линейных функций она следует из теоремы 3 для

$P_n(A) =$ "A содержит строку, в которой хотя бы $n + 1$ единица".

Идея 2. *Предположим, что на протяжении* n *шагов элиминации гейтов в схеме остаётся хотя бы один вход, с исходящей степенью по крайней мере* 1 *и, более того, в* m *из этих случаев у этого входа исходящая степень по крайней мере* 2 *(т.е. он соединён с двумя гейтами, а не с гейтом и выходом). Тогда* $C(f) \geqslant n + m$.

Теорема 4. *Назовём ненулевой элемент матрицы* уникальным, *если это единственный ненулевой элемент в своей строке. Пусть* $\mathcal{P} = \{P_n\}_{n=1}^{\infty}$ *– набор предикатов, определённых на матрицах над* $\mathbb{F}_2$ *и удовлетворяющих следующим свойствам:*

- *если выполняется* $P_1(A)$, *то* $C(A) \geqslant 1$;
- *если выполняется* $P_n(A)$, *то и* $P_m(A)$ *выполняется для любого* $1 \leqslant m \leqslant n$;
- *если выполняется* $P_n(A)$, *то, для любого индекса* i, *если* i*-й столбец матрицы* A *содержит хотя бы один уникальный элемент, то выполняется* $P_{n-1}(A_{-i})$, *иначе – выполняется* $P_{n-2}(A_{-i})$.

Тогда для любой матрицы A, *которая содержит хотя бы* $n+1$ *столбец и все столбцы которой различны, если для некоторого* n *выполняется* $P_n(A)$, *то* $C(A) \geqslant n$ *и, более того,* $C_{\frac{3}{4}}(A) \geqslant n$.

Доказательство. Доказательство ведётся индукцией по n. Для $n = 1$ утверждение очевидно, т.к. если $C(A) = 1$, то и $C_{\frac{3}{4}}(A) = 1$: выход единственного бинарного гейта g в оптимальной схеме (напомним, что все гейты нетривиальные, т.к. унарные и константные гейты в наших определениях можно удалить из схемы без потери общности) не более чем на трёх четвертях входов может совпадать с любой из входных переменных (а если вспомнить, что мы рассматриваем линейные функции, то не более чем на половине, т.е. даже $C_{\frac{1}{2}}(A) = 1$).

Рассмотрим первый гейт g в оптимальной схеме, вычисляющей A на более чем α входов для некоторого $\alpha \geqslant \frac{3}{4}$. Поскольку g – первый, его входы являются входами схемы; обозначим их за x_i и x_j. Возможны три случая.

1. Один из входов g, например x_i, соединён непосредственно с выходом y_k. Тогда, подставив в x_i константу, мы сможем исключить один гейт. В этом случае y_k соответствует строке с единственным ненулевым элементом, то есть i-й столбец содержит уникальный элемент, и выполняется $P_{n-1}(A_{-i})$. Теперь, воспользовавшись предположением индукции, мы получаем, что $C_\alpha(A_{-i}) \geqslant n - 1$, а значит, исходная оценка верна (мы предполагаем по определению, что каждый гейт вычисляет как функцию, так и её отрицание, так что возможное инвертирование выходов – не проблема).

2. Один из входов g, например x_i, соединён с ещё одним гейтом. Тогда, установив x_i в константу, мы сможем исключить два гейта. По свойствам P_n выполняется $P_{n-2}(A_{-i})$, а значит, мы можем воспользоваться индукционным предположением, показывающим, что $C_\alpha(A_{-i}) \geqslant n - 2$.

3. Ни x_i, ни x_j не входят в другой гейт или в выход. В этом случае A – функция не от x_i и x_j, а только от $g(x_i, x_j)$; покажем, что для схемы, вычисляющей A более чем на $\frac{3}{4}$ входов, это невозможно. A зависит от x_i и x_j по отдельности, потому что все её столбцы различны; в частности, для одной из этих переменных, например x_i, найдется выход y_k, зависящий от x_i, но не от x_j: $y_k = x_i \oplus \bigoplus\limits_{x \in X} x$, где $x_j \notin X$. С другой стороны, поскольку каждый гейт в оптимальной схеме, нетривиально зависящий от x_i, нетривиально зависит и от x_j, найдутся такие значения a и b, что $g(0, a) = g(1, b)$. Таким образом, для каждого означивания остальных переменных либо на входе с $(x_i = 0, x_j = a)$, либо

на входе с $(x_i = 1, x_j = b)$ схема ошибётся, что гарантирует ошибку хотя бы на $\frac{1}{4}$ от всех входов. □

Следствие 1. *Зафиксируем вещественное число $\alpha \in [0,1]$. Пусть $\mathcal{R} = \{R_n\}_{n=0}^{\infty}$ и $\mathcal{Q} = \{Q_m\}_{m=0}^{\infty}$ – два набора предикатов, определённых на матрицах над $\mathbb{F}_2$ и удовлетворяющих следующим свойствам:*

- *$R_0(A)$ и $Q_0(A)$ тождественно истинны;*
- *если выполняется $R_1(A)$, то $C(A) \geqslant 1$;*
- *если выполняется $R_n(A)$, то и $R_k(A)$ выполняется для любого $0 \leqslant k \leqslant n$;*
- *если выполняется $R_n(A)$, то для любого индекса i выполняется $R_{n-1}(A_{-i})$;*
- *если выполняется $Q_1(A)$, то $C(A) \geqslant 1$;*
- *если выполняется $Q_m(A)$, то и $Q_k(A)$ выполняется для любого $0 \leqslant k \leqslant m$;*
- *если выполняется $Q_m(A)$, то для любого индекса i выполняется $Q_{m-1}(A_{-i})$;*
- *если выполняется $Q_m(A)$ и матрица A_{-i} содержит больше нулевых строк, чем A (т.е. удаление i-го столбца приводит к удалению последнего ненулевого элемента из как минимум одной строки), то выполняется $Q_m(A_{-i})$.*

В таком случае для любой матрицы A, содержащей хотя бы $n+1$ различный столбец, если для некоторых n и m, $n \geqslant m$, выполняются $R_n(A)$ и $Q_m(A)$, то $C(A) \geqslant n+m$ и, более того, $C_{\frac{3}{4}}(A) \geqslant n+m$.

Доказательство. Немедленно следует из теоремы 4 для $P_n(A) = \exists k \in [0..n] R_k(A) \wedge Q_{n-k}(A)$. □

Следствие 2 ([18, Лемма 5])**.** *Пусть $t, v \geqslant 1$, а χ – линейная функция над $\mathbb{F}_2$ с матрицей A. Предположим, что все столбцы A различны, каждая строка A содержит хотя бы v ненулевых элементов, и после удаления любых t столбцов A матрица будет содержать хотя бы одну строку, содержащую хотя бы два ненулевых элемента. Тогда $C(\chi) \geqslant v+t$ и, более того, $C_{\frac{3}{4}}(\chi) \geqslant v+t$.*

Доказательство. Возьмём $R_n(A)$ = "После удаления любых n столбцов в A по-прежнему остаётся хотя бы одна ненулевая строка", $Q_0(A)$ = "true" и $Q_m(A)$ = "Каждая строка A содержит хотя бы $m+1$ единицу" для $m > 0$. Тогда $R_{t+1}(A)$ и $Q_{u-1}(A)$ выполняются, и $\mathcal{R}$ и $\mathcal{Q}$

удовлетворяют условиям следствия 1, с помощью которого можно получить требуемые оценки. Заметим, что в этом случае Q_m для $m > 0$ не может быть выполнен для матрицы, в которой строка содержит только одну единицу; поэтому в методе исключения гейтов для первых $u-1$ шагов, а затем в течение $t-u+2$ шагов, будет исключаться один гейт. □

Также из теоремы 4 можно вывести новое следствие, которое будет важно для новой слабозащищённой конструкции.

Следствие 3. *Пусть $t \geqslant v \geqslant 2$. Предположим, что A – матрица размера $v \times t$, причём все её столбцы различны, и каждый столбец A содержит хотя бы два ненулевых элемента (хотя бы две единицы). Тогда $C(A) \geqslant 2t - v$ и, более того, $C_{\frac{3}{4}}(A) \geqslant 2t - v$.*

Доказательство. Возьмем $P_n(A)$ ="удвоенное число ненулевых столбцов A за вычетом числа ненулевых строк A больше либо равно n". Тогда будет выполнено $P_{2t-v}(A)$, а $\mathcal{P}$ удовлетворяет условиям теоремы 4. □

В заключение этого раздела расширим доказанные выше результаты на блочно-диагональные матрицы. В общем случае мы не можем доказать, что прямая сумма двух функций будет иметь схемную сложность, равную сумме их сложностей; контрпримеры, известные как "mass production", можно найти в [36]. Однако для линейных функций и метода исключения гейтов с использованием теорем 3 и 4 подобное доказательство возможно. Следующая теорема обобщает лемму 6 из [18].

Теорема 5. *Предположим, что линейная функция χ задана блочно-диагональной матрицей*

$$\begin{pmatrix} A_1 & 0 & \cdots & 0 \\ 0 & A_2 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & A_k \end{pmatrix},$$

каждая A_j удовлетворяет условиям теоремы 4 с предикатами $\mathcal{P}^j = \{P_n^j\}_{n=1}^{\infty}$, и для каждого j найдётся такое n_j, что $P_{n_j}^j(A_j)$ выполняется. Тогда $C(\chi) \geqslant \sum\limits_{j=1}^{k} n_j$.

Доказательство. Применим теорему 4 с предикатом, составленным из исходных предикатов:

$$P_n = \bigvee_{i_1+\ldots+i_k=n} P^1_{i_1} \wedge P^2_{i_2} \wedge \ldots \wedge P^k_{i_k}.$$

Теперь можно непосредственно проверить, что $\mathcal{P} = \{P_n\}_{n=1}^{\infty}$ удовлетворяет условиям теоремы 4 (так как удаление одного столбца задействует не более одного блока), и для блочно-диагональной матрицы выполнен предикат $P_{n_1+\ldots+n_k}$. □

§5. Функции с секретом, доказуемо надёжные в слабом смысле

В нашей конструкции мы воспользуемся идеями из [18]: сначала мы построим слабозащищённую функцию-кандидата, для взлома которой будет необходимо время большее, чем для дешифровки, однако шифрование будет занимать еще большее время. Затем мы добавим к этой конструкции блок из слабозащищённой односторонней функции и в результате не слишком сильно увеличим время, необходимое для шифрования, но существенно затрудним как процесс дешифровки, так и процесс взлома. Подобная конструкция подробно описана в [18].

Введём несколько обозначений. Через U_n мы будем обозначать верхнетреугольную матрицу размера $n \times n$, обратную к двухдиагональной:

$$U_n = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ 0 & 1 & \cdots & 1 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 1 \end{pmatrix}, \qquad U_n^{-1} = \begin{pmatrix} 1 & 1 & 0 & \cdots & 0 \\ 0 & 1 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{pmatrix};$$

заметим, что U_n^2 – верхнетреугольная матрица, в которой ноли и единицы чередуются в шахматном порядке:

$$U_n^2 = \begin{pmatrix} 1 & 0 & 1 & \cdots & (n-1) \bmod 2 & n \bmod 2 \\ 0 & 1 & 0 & \cdots & n \bmod 2 & (n-1) \bmod 2 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & 1 \end{pmatrix}$$

В дальнейшем мы часто будем использовать матрицы, составленные из матриц меньшего размера как из блоков; например, $\begin{pmatrix} U_n & U_n \end{pmatrix}$ – матрица размера $n \times 2n$, состоящая из двух верхнетреугольных блоков.

Лемма 1.

(1) $C_{\frac{3}{4}}(U_n) = n - 1$.
(2) $C_{\frac{3}{4}}(U_n^2) = n - 2$.
(3) $C_{\frac{3}{4}}(U_n^{-1}) = n - 1$.
(4) $C_{\frac{3}{4}}(\begin{pmatrix} U_n & U_n \end{pmatrix}) = 2n - 1$.
(5) $3n - 6 \leqslant C_{\frac{3}{4}}(\begin{pmatrix} U_n^2 & U_n \end{pmatrix}) \leqslant C(\begin{pmatrix} U_n^2 & U_n \end{pmatrix}) \leqslant 3n - 3$.
(6) $3n - 4 \leqslant C_{\frac{3}{4}}(\begin{pmatrix} U_n & U_n^{-1} \end{pmatrix}) \leqslant C(\begin{pmatrix} U_n & U_n^{-1} \end{pmatrix}) \leqslant 3n - 2$.

Доказательство. Нижние оценки из пунктов 1–3 очевидны: в матрицах нет одинаковых строк, и ни один вход, кроме одного (двух для пункта 2) не присоединён непосредственно к выходу. Нижняя оценка для пункта 4 получается простым подсчётом: первая строка матрицы содержит $2n$ ненулевых элементов, так что для того чтобы её вычислить, потребуется хотя бы $2n - 1$ гейт. Нижняя оценка для пункта 5 (соответственно, 6) следует из следствия 3: матрица $\begin{pmatrix} U_n^2 & U_n \end{pmatrix}$ (соответственно, $\begin{pmatrix} U_n & U_n^{-1} \end{pmatrix}$) удовлетворяет предположению следствия 3 целиком, за исключением трёх (соответственно, двух) столбцов, и мы можем воспользоваться следствием 3 для $t = 2n - 3$ (соответственно, $t = 2n - 2$) и $u = n$.

Для доказательства верхних оценок мы приведём явные конструкции. Для вычисления матрицы из пункта 1 заметим, что каждая строка отличается от предыдущей только в одной позиции, так что мы можем вычислять входы по формуле: $out_i = out_{i+1} \oplus in_i$. Более того, $out_n = in_n$, и для его вычисления не нужно гейтов. Та же идея работает и для пункта 2, но в этом случае out_n и out_{n-1} вычисляются непосредственно, а $out_i = out_{i-2} \oplus in_i$. Чтобы вычислить матрицу из пункта 3, непосредственно вычислим каждую строку. Для вычисления 4 воспользуемся идеей из [18]. Заметим, что $\begin{pmatrix} U_n & U_n \end{pmatrix} \cdot \begin{pmatrix} a \\ b \end{pmatrix} = U_n \cdot a \oplus U_n \cdot b = U_n \cdot (a \oplus b)$. Используем n гейтов для вычисления $a \oplus b$, а затем получим результат с помощью $n - 1$ гейтов. Для вычисления 5 и 6 заметим, что $\begin{pmatrix} A & B \end{pmatrix} \cdot \begin{pmatrix} a \\ b \end{pmatrix} = A \cdot a \oplus B \cdot b$. Таким образом, мы разделили процесс вычисления на две части, которые могут быть независимо вычислены с использованием предыдущих алгоритмов, после чего за n гейтов можно вычислить окончательный результат как XOR промежуточных. □

Для первой конструкции положим длины публичного ключа pi, секретного ключа ti, сообщения m и закодированного сообщения c одинаковыми и равными n. Положим

$$ti = U_n \cdot pi, \quad c = \begin{pmatrix} U_n^{-1} & U_n \end{pmatrix} \cdot \begin{pmatrix} m \\ pi \end{pmatrix}.$$

В этом случае взломщику придётся вычислить матрицу

$$\begin{pmatrix} U_n & U_n \end{pmatrix} \cdot \begin{pmatrix} c \\ ti \end{pmatrix} = \begin{pmatrix} U_n & U_n^2 \end{pmatrix} \cdot \begin{pmatrix} c \\ pi \end{pmatrix}.$$

Получается, что обращение этой функции без секрета (взлом) труднее, чем с секретом (честное обращение), но сложность прямого вычисления функции примерно равна сложности взлома, поэтому назвать данную функцию доказуемо надёжной в слабом смысле нельзя.

Для того чтобы справиться с этой проблемой, рассмотрим слабо одностороннюю линейную функцию A и организуем протокол следующим образом (через I_n здесь обозначена единичная матрица):

$$\begin{aligned} \mathrm{Seed}_n &= \begin{pmatrix} U_n \\ I_n \end{pmatrix} \cdot s = \begin{pmatrix} t_i \\ p_i \end{pmatrix}, \\ \mathrm{Eval}_n &= \begin{pmatrix} U_n^{-1} & U_n & 0 \\ 0 & 0 & A \end{pmatrix} \cdot \begin{pmatrix} m_1 \\ pi \\ m_2 \end{pmatrix} = \begin{pmatrix} c_1 \\ c_2 \end{pmatrix}, \\ \mathrm{Inv}_n &= \begin{pmatrix} U_n & U_n & 0 \\ 0 & 0 & A^{-1} \end{pmatrix} \cdot \begin{pmatrix} c_1 \\ ti \\ c_2 \end{pmatrix} = \begin{pmatrix} m_1 \\ m_2 \end{pmatrix}. \end{aligned}$$

Теперь взломщику придётся вычислить

$$\mathrm{Adv}_n = \begin{pmatrix} U_n & U_n^2 & 0 \\ 0 & 0 & A^{-1} \end{pmatrix} \cdot \begin{pmatrix} c_1 \\ pi \\ c_2 \end{pmatrix} = \begin{pmatrix} m_1 \\ m_2 \end{pmatrix}.$$

В качестве слабоодносторонней функции A возьмём одну из линейных функций со сложностью $2 - \epsilon$, построенных Хильтгеном, для некоторого $\epsilon > 0$ [15]; матрицу этой функции возьмём порядка λn, где λ будет выбрана ниже. Для такой матрицы $C_{\frac{3}{4}}(A) = \lambda n + o(n)$, и $C_{\frac{3}{4}}(A^{-1}) = (2-\epsilon)\lambda n + o(n)$. Теперь лемма 1 и теорема 5 дают следующие оценки на сложность:

$$\begin{aligned} C_{\frac{3}{4}}(\mathrm{Seed}_n) &= n - 1, \\ C_{\frac{3}{4}}(\mathrm{Eval}_n) &= 3n + \lambda n + o(n) &&= (3+\lambda)n + o(n), \\ C_{\frac{3}{4}}(\mathrm{Inv}_n) &= 2n + (2-\epsilon)\lambda n + o(n) &&= (2 + (2-\epsilon)\lambda)n + o(n), \\ C_{\frac{3}{4}}(\mathrm{Adv}_n) &= 3n + (2-\epsilon)\lambda n + o(n) &&= (3 + (2-\epsilon)\lambda)n + o(n). \end{aligned}$$

Порядок надёжности этого протокола равен

$$\lim_{n\to\infty}\left(\min\left(\frac{C_{3/4}(\mathrm{Adv}_n)}{C(\mathrm{Eval}_n)},\frac{C_{3/4}(\mathrm{Adv}_n)}{C(\mathrm{Inv}_n)},\frac{C_{3/4}(\mathrm{Adv}_n)}{C(\mathrm{Seed}_n)}\right)\right)$$
$$=\min\left(\frac{3+(2-\epsilon)\lambda}{3+\lambda},\frac{3+(2-\epsilon)\lambda}{2+(2-\epsilon)\lambda}\right).$$

Это выражение достигает максимума при $\lambda=\frac{1}{1-\epsilon}$, и этот максимум равен $\frac{5-4\epsilon}{4-\epsilon}$, что стремится к $\frac{5}{4}$ при $\epsilon\to 0$. Таким образом, мы доказали Теорему 1.

§6. Заключение

В этой работе мы подробно исследовали схемную сложность линейных булевых функций. Мы доказали два достаточно общих утверждения, которые выражают метод исключения гейтов для линейных функций в общем случае, и получили из них несколько важных следствий, которые удалось применить для построения нового семейства линейных функций с секретом, доказуемо надёжных в слабом смысле. Порядок надёжности построенного семейства превосходит известный ранее [18]. Конечно, в настоящее время криптографические конструкции, доказуемо надёжные в слабом смысле, вряд ли могут найти какое-либо практическое применение, однако они представляются важными с теоретической точки зрения. К сожалению, результаты этой статьи и работ [18, 43] – это передний край математически доказуемых результатов о криптографической надёжности; чтобы получить существенно более сильные результаты (с нелинейной разницей между честными участниками и взломщиком), нужно доказывать нелинейные нижние оценки на схемную сложность булевых функций в общем базисе, что является фундаментальной открытой проблемой теории сложности. С другой стороны, в разделе 3 мы видели, что эти оценки – далеко не предел возможного даже для линейных булевых функций, не говоря уже о нелинейных.

Можно выделить два направления для дальнейших исследований в этом направлении. Во-первых, можно продолжать исследовать примитивы, доказуемо надёжные в слабом смысле. Скорее всего, порядки надёжности, полученные в наших работах, можно улучшить; возможно, удастся сформулировать определения новых примитивов (протоколов согласования ключа, доказательств с неразглашением и т.п.),

доказуемо надёжных в слабом смысле, и предъявить соответствующие конструкции. Такие исследования могут расширить сферу применимости методов надёжности в слабом смысле, но настоящий прорыв на этом направлении вряд ли можно ожидать. Становится ясно, что криптографические интересы требуют дальнейших исследований и продвижений в области схемной сложности в общем случае. С 1980-х гг. в схемной сложности не было получено новых прорывных результатов; легко доказать неконструктивные нижние оценки, сравнив число схем и функций, но конструктивные оценки получить не удаётся: лучшая известная нижняя оценка – это оценка $3n - o(n)$, доказанная в 1984 г. [5]. Ни один из известных методов не позволяет надеяться на нелинейные нижние оценки схемной сложности, и для таких оценок потребуется разработка принципиально новых методов. Важность такого прорыва трудно переоценить; доказуемо надёжные криптографические конструкции – это лишь одно потенциальное применение для нелинейных нижних оценок схемной сложности.

Благодарности. Мы благодарны Э. А. Гиршу и О. Меланич за плодотворные обсуждения результатов и доказательств, приведённых в этой работе, а также анонимным рецензентам, чьи комментарии позволили существенно улучшить текст статьи.

Литература

1. M. Ajtai, Σ_1^1-*formulae on finite structures.* — Annals Pure Appl. Logic. **24** (1983), 1–48.
2. M. Ajtai, C. Dwork, *A public-key cryptosystem with worst-case/average-case equivalence.* — In: Proceedings of the 29th Annual ACM Symposium on Theory of Computing (1997), pp. 284–293.
3. E. Allender, *Circuit complexity before the dawn of the new millennium.* — In: Proceedings of the 16th Conference on Foundations of Software Technology and Theoretical Computer Science (1996), pp. 1–18.
4. N. Alon, M. Karchmer, A. Wigderson, *Linear circuits over* GF(2). — SIAM J. Comput. **19**, No. 6 (1990), 1064–1067.
5. N. Blum, *A boolean function requiring* $3n$ *network size.* — Theor. Computer Sci. **28** (1984), 337–345.
6. R. B. Boppana, M. Sipser, *The complexity of finite functions.* — Handbook Theor. Computer Sci., Volume A: Algorithms and Complexity, J. van Leeuwen (Ed.) The MIT Press/Elsevier, 1990.
7. A. Davydow, S. I. Nikolenko, *Gate elimination for linear functions and new feebly secure constructions.* — In: Proceedings of the 6th Computer Science Symposium in Russia, Lect. Notes Computer Sci. **6651** (2011), pp. 148–161.

8. W. Diffie, M. Hellman, *New directions in cryptography.* — IEEE Transact. Inform. Theory **IT-22** (1976), 644–654.
9. C. Dwork, *Positive applications of lattices to cryptography.* — In: Proceedings of the 22nd International Symposium on Mathematical Foundations of Computer Science, Lect. Notes Computer Sci. **1295** (1997), pp. 44–51.
10. M. Furst, J. Saxe, M. Sipser, *Parity, circuits, and the polynomial-time hierarchy.* — Math. Systems Theory **17** (1984), 13–27.
11. O. Goldreich, *Foundations of Cryptography. Basic Tools.* Cambridge University Press, 2001.
12. D. Grigoriev, E. A. Hirsch, K. Pervyshev, *A Complete public-key cryptosystem.* — Groups, Complexity, and Cryptology **1** (2009), 1–12.
13. D. Harnik, J. Kilian, M. Naor, O. Reingold, A. Rosen, *On robust combiners for oblivious transfers and other primitives.* — Proc. EuroCrypt T05, Lect. Notes Computer Sci. **3494** (2005), pp. 96–113.
14. J. Håstad, *Computational Limitations for Small Depth Circuits.* MIT Press, Cambridge, MA, 1987.
15. A. P. Hiltgen, *Constructions of feebly-one-way families of permutations.* — In: Proc. of AsiaCrypt '92 (1992), pp. 422–434.
16. A. P. Hiltgen, *Cryptographically relevant contributions to combinatorial complexity theory.* — ETH Series Inform. Proc. (J. L. Massey (Ed.)), Konstanz, Hartung-Gorre Vol. 3, 1994.
17. A. P. Hiltgen, *Towards a better understanding of one-wayness: facing linear permutations.* — In: Proceedings of EuroCrypt '98, Lect. Notes Computer Sci. **1233** (1998), pp. 319–333.
18. E. A. Hirsch, S. I. Nikolenko, *A feebly secure trapdoor function.* — In: Proceedings of the 4th Computer Science Symposium in Russia, Lect. Notes Computer Sci. **5675** (2009), pp. 129–142.
19. N. Koblitz, *The uneasy relationship between mathematics and cryptography.* — Notices Amer. Math. Soc. **54** (2007), 972–979.
20. N. Koblitz, A. Menezes, *Another look at "Provable security."* II. — In: Proceedings Progr. Cryptology – Indocrypt (2006), Lect. Notes Computer Sci. **4329** (2006), pp. 148–175.
21. N. Koblitz, A. Menezes, *Another look at "Provable security."* — J. Cryptology **20**, No. 1 (2007), 3–37.
22. E. A. Lamagna, J. E. Savage, *On the logical complexity of symmetric switching functions in monotone and complete bases.* — Tech. Rep. Rhode Island: Brown Univ. (July 1973).
23. L. A. Levin, *One-way functions and pseudorandom generators.* — Combinatorica **7**, No. 4 (1987), 357–363.
24. W. J. Paul, *A* $2,5n$ *lower bound on the combinational complexity of Boolean functions.* — SIAM J. Comput. **6** (1977), 427–443.
25. O. Regev, *On lattices, learning with errors, random linear codes, and cryptography.* — In: Proceedings of the 37th Annual ACM Symposium on Theory of Computing (2005), pp. 84–93.

26. O. Regev, *Lattice-based cryptography.* — In: Proceedings of the 26th Annual International Cryptology Conference (CRYPTO'06), Lect. Notes Computer Sci. **4117** (2006), pp. 131–141.
27. R. L. Rivest, A. Shamir, L. Adleman, *A method for obtaining digital signatures and public-key cryptosystems.* — Commun. ACM **21**, No. 2 (1978), 120–126.
28. J. E. Savage, *The Complexity of Computing.* New York, Wiley, 1976.
29. C. E. Shannon, *Communication theory of secrecy systems.* — Bell System Technical J. **28**, No. 4 (1949), 656–717.
30. R. Smolensky, *Algebraic methods in the theory of lower bounds for Boolean circuit complexity.* — In: Proceedings of the 19th Annual ACM Symposium on Theory of Computing (1987), pp. 77–82.
31. L. Stockmeyer, *The complexity of decision problems in automata theory and logic.* Ph.D.Thesis Massachusetts Institute Technology (1974).
32. L. Stockmeyer, *On the combinational complexity of certain symmetric Boolean functions.* — Math. Systems Theory **10** (1977), 323–326.
33. L. Stockmeyer, *Classifying the computational complexity of problems.* — J. Symbolic Logic **52** (1987), 1–43.
34. G. S. Vernam, *Cipher printing telegraph systems for secret wire and radio telegraphic communications.* — J. IEEE **55** (1926), 109–115.
35. H. Vollmer, *Introduction to Circuit Complexity: a Uniform Approach.* Springer Verlag, 1999.
36. I. Wegener, *The Complexity of Boolean Functions.* B. G. Teubner and John Wiley & Sons, 1987.
37. R. Williams, *Non-uniform ACC circuit lower bounds.* — In: Proceedings of the 26nd Annual IEEE Conference on Computational Complexity (2011), pp. 115–125.
38. A. C.-C. Yao, *On ACC and threshold circuits.* — In: Proceedings of the 31st Annual IEEE Symposium on the Foundations of Computer Science (1990), pp. 619–627.
39. А. Кожевников, С. И. Николенко, *О полных односторонних функциях.* — Проблемы передачи информации **45**, No. 2 (2009), 101–118.
40. Л. А. Левин, *Односторонние функции.* — Проблемы передачи информации **39**, No. 1 (2003), 92–103.
41. О. Б. Лупанов, *Об одном подходе к синтезу управляющих систем – принципе локального кодирования.* — Проблемы кибернетики **14** (1965), 31–110.
42. А. А. Марков, *О минимальных контактно-вентильных двухполюсниках для монотонных симметрических функций.* — Проблемы кибернетики **8** (1962), 117–121.
43. О. Меланич, *Нелинейные надежные в слабом смысле криптографические примитивы.* Препринт ПОМИ **12** (2009).
44. Э. И. Нечипорук, *Об одной булевской функции.* — Докл. Акад. Наук СССР **169**, No. 4 (1966), 765–766.
45. Р. Г. Нигматуллин, *Сложность булевых функций.* Наука, М., 1991.
46. А. А. Разборов, *Нижние оценки монотонной сложности логического перманента.* — Матем. заметки **37**, No. 6 (1985), 887–900.
47. А. А. Разборов, *Нижние оценки размера схем ограниченной глубины в полном базисе, содержащем функцию логического сложения.* — Матем. заметки **41**, No. 4 (1987), 598–608.

48. А. А. Разборов, *Нижние оценки сложности реализации симметрических булевых функций контактно-вентильными схемами.* — Матем. заметки **48**, No. 6 (1990), 79–90.
49. Б. А. Субботовская, *О реализации линейных функций формулами в базисе* $\vee$, $\&$, $\neg$. — Докл. Акад. Наук СССР **136**, No. 3 (1961), 553–555.
50. Б. А. Субботовская, *О сравнении базисов при реализации функций алгебры логики формулами.* — Докл. Акад. Наук СССР **149**, No. 4 (1963), 784–787.
51. В. М. Храпченко, *О сложности реализации линейной функции в классе* π*-схем.* — Матем. заметки **9**, No. 1 (1971), 36–40.
52. Л. А. Шоломов, *О реализации недоопределённых булевых функций схемами из функциональных элементов.* — Проблемы кибернетики **21** (1969), 215–226.
53. С. В. Яблонский, *О классах функций алгебры логики, допускающих простую схемную реализацию.* — Успехи мат. наук **12**, No. 6 (1957), 189–196.

Davydow A. P., Nikolenko S. I. Circuit complexity of linear functions: gate elimination and feeble security.

In this work, we consider provably secure cryptographic constructions in the context of circuit complexity. Based on the ideas of provably secure trapdoor functions previously developed in (Hirsch, Nikolenko, 2009; Melanich, 2009), we present a new linear construction of a provably secure trapdoor function with order of security 5/4. Besides, we present an in-depth general study of the gate elimination method for the case of linear functions. We also give a non-constructive proof of nonlinear lower bounds on the circuit complexity of linear Boolean functions and upper bounds on circuit implementations of linear Boolean functions, obtaining specific constants.

Санкт-Петербургское отделение
Математического института
им. В.А. Стеклова РАН, Фонтанка 27,
Санкт-Петербург 191023, Россия
E-mail: sergey@logic.pdmi.ras.ru

СПбАУ НОЦНТ РАН
ул. Хлопина, д. 8, корпус 3,
Санкт-Петербург 194021, Россия
E-mail: adavydow@gmail.com

Поступило 31 января 2012 г.