ПРЕПРИНТЫ ПОМИ РАН

Учредитель: Санкт-Петербургское отделение Математического института им. В. А. Стеклова Российской академии наук

Свидетельство о регистрации средства массовой информации: ЭЛ №ФС 77-33560 от 16 октября 2008 г. Выдано Федеральной службой по надзору в сфере связи и массовых коммуникаций

Контактные данные: 191023, г. Санкт-Петербург, наб. реки Фонтанки, дом 27

телефоны:(812)312-40-58; (812) 571-57-54

e-mail: admin@pdmi.ras.ru

http://www. pdmi.ras.ru /preprint/

Заведующая информационно-издательским сектором Симонова В.Н

ПРЕПРИНТ ПОМИ 16/2008

НАДЕЖНАЯ В СЛАБОМ СМЫСЛЕ ФУНКЦИЯ С СЕКРЕТОМ*

Э. А. Гирш **С. И. Николенко**

Санкт-Петербургское отделение
Математического института им. В. А. Стеклова
Российской Академии наук

http://logic.pdmi.ras.ru/{~hirsch,~sergey}

12 сентября 2008 г.

Аннотация В 1992 году А. Хильтген [Hil92] построил первые конструкции доказуемо надежных криптографических примитивов, а именно *функций, односторонних в слабом смысле*. Такие функции доказуемо сложнее обратить, чем вычислить, но сложность при этом (рассматриваемая как схемная сложность над схемами с произвольными бинарными гейтами) увеличивается лишь в константное число раз (в работах Хильтгена эта константа приближается к 2).
В классической криптографии односторонние функции являются базовыми примитивами для протоколов согласования ключа и цифровой подписи, в то время как криптосистемы с открытым ключом конструируются на базе функций с секретом. Мы продолжаем исследования Хильтгена и строим *надежную в слабом смысле функцию с секретом*; в нашей конструкции противник гарантированно потратит больше времени на обращение функции, чем честные участники протокола (также в константное число раз).

Ключевые слова: функции, односторонние в слабом смысле, схемная сложность, нижние оценки, функции с секретом.

*При частичной поддержке фонда Династия, грантов РФФИ 06-01-00502-а и 08-01-00640-а, гранта Президента РФ подержки ведущих научных школ НШ 4392.2008.1, программы фундаментальных исследований РАН.

ПРЕПРИНТЫ
Санкт-Петербургского отделения
Математического института им. В. А. Стеклова
Российской Академии наук

PREPRINTS
of the St.Petersburg Department of Steklov Institute of Mathematics

---

1 Введение

В течение последних 30 лет было разработано большое количество различных криптографических протоколов с открытым ключом, однако до сих пор не существует *ни единого* доказательства их надежности: ни безусловного доказательства (что неизбежно влекло бы $\mathrm{P} \neq \mathrm{NP}$), ни доказательства, основанного на классических (не зависящих от конкретной задачи) предположениях структурной сложности. В то время как полные примитивы для односторонних функций [Lev87] и криптосистем с открытым ключом известны [HKN+05] (см. также [GHP08]), они никак не связаны с основными предположениями классической структурной теории сложности. Более того, асимптотическая природа полиномиальных сведе́ний не оставляет надежды доказать, что та или иная конкретная схема надежна на той или иной конкретной длине ключа. В общем, представляется, что криптографии еще очень далеко до каких-либо *доказуемо* надежных конструкций с открытым ключом.

Если мы не можем доказать сверхполиномиальную разницу между сложностью честных участников протокола и взломщика, можем ли мы доказать *хоть какую-нибудь* разницу? В 1992 году Ален Хильтген [Hil92] сумел построить функцию, обратить которую *вдвое* сложнее, чем вычислить. Его пример представляет собой линейную функцию над $GF(2)$ с матрицей, у которой мало ненулевых элементов, но зато в обратной к ней ненулевых элементов много. Разница в сложностях получается из несложного наблюдения Ламаньи и Сэвиджа [LS73,Sav76]: каждый бит выхода нетривиально зависит от многих битов входа, и все эти биты соответствуют разным функциям, следовательно, не так-то просто вычислить все эти функции сразу. В качестве модели вычислений здесь рассматривается самая общая модель: количество гейтов в булевской схеме, которая может использовать произвольные бинарные булевские гейты. Отметим, что от такой модели трудно ожидать более сильных оценок, потому что известные нижние оценки схемной сложности над полным бинарным базисом линейны от количества входов [Blu84,Weg87].

В настоящей работе мы представляем конструкцию другого криптографического примитива, надежного в слабом смысле: надежную в слабом смысле функцию с секретом. Чтобы получить этот результат, нужно доказать нижнюю оценку на схемную сложность определенной функции; мы используем метод исключения гейтов, при помощи которого в восьмидесятые годы прошлого века были доказаны все вышеупомянутые нижние оценки.

Наша конструкция состоит из двух функций, слитых в единую конструкцию. В первой функции задача взломщика («Чарли») сложнее, чем задача кодирующего («Боба»), а во второй функции задача взломщика сложнее, чем задача декодирующего («Алисы»). Мы докажем, что если часть входа подать первой функции, а часть — второй, то задача Чарли станет сложнее, чем задачи и Алисы, и Боба. Формально говоря, сложность обращения (декодирования) без использования секретной информации будет по меньшей мере в $\frac{25}{22}$ раза больше, чем сложности честного кодирования, декодирования и генерации ключей.

В разделе 2 мы даем базовые определения. В разделе 3 проводится некоторая подготовительная работа — мы устанавливаем комбинаторные свойства матриц, соответствующих кандидатам на звание сложных функций. Раздел 4 содержит обзор базового метода исключения гейтов, которым мы пользуемся для доказательства нижних оценок схемной сложности, а также применяет этот метод в контексте надежности в слабом смысле. Раздел 5 приводит минимальный пример конструкции, для которого уже заметна разница между взломщиком и честными участниками. Наконец, разделы 6 и 7 представляют конструкцию надежной в слабом смысле функции с секретом в общем виде, раздел 8 содержит доказательство более сильных гарантий надежности против более слабых противников, а раздел 9 обсуждает потенциальные направления дальнейших исследований.

2 Определения

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. Нашей вычислительной моделью являются булевские схемы с произвольными бинарными гейтами. Булевская схема — это ациклический направленный граф, каждая вершина которого либо не имеет ни одного входящего ребра (такие вершины называются *входами схемы*, или *переменными*), или два входящих ребра (такие вершины называются *гейтами*[1]) Каждый гейт помечен бинарной булевской функцией, то есть любой из шестнадцати функций из $\mathbb{B}_{2,1}$. Некоторые гейты помечены как *выходы* (по техническим причинам мы также допускаем, что выходом может быть *отрицание* вычисляемого в гейте значения; таким образом мы можем избежать лишних гейтов–отрицаний). Схема с n входами и m выходами естественным образом вычисляет функцию из $\mathbb{B}_{n,m}$.

Размер схемы мы будем обозначать как Size. *Схемная сложность* (или просто *сложность*) функции f, обозначаемая через $C(f)$, — это наименьшее количество гейтов в схеме, вычисляющей f, то есть

$$C(f) = \min_{c:\forall x\, c(x)=f(x)} \mathrm{Size}(c).$$

Такая схема называется *оптимальной схемой* для функции f. Можно без потери общности предполагать, что каждый гейт схемы нетривиально зависит от обоих входов, то есть что в схеме нет функций–констант и унарных функций Id и NOT, потому что такие гейты легко исключить из схемы, не увеличивая количество гейтов в ней.

А. Хильтген ввёл для каждой *обратимой* (инъективной) функции $f_n \in \mathbb{B}_{n,m}$ понятие *меры сложности в слабом смысле* (measure of feeble one-wayness)

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

(напомним, что $C(f)$ — размер минимальной схемы из бинарных булевских функций, реализующей f). Результаты Хильтгена заключались в том, что он нашёл последовательности функций $\{f_n\}_{n=1}^{\infty}$ с нетривиальными константами

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

которые Хильтген называет *порядком необратимости* (order of one-wayness). Его результаты мы подробнее обсудим в разделе 4.

Нам потребуется ввести весьма детальное определение семейства функций с секретом, потому что в контексте сложности в слабом смысле важна каждая константа. Поскольку определение 1 ничего не утверждает о собственно сложности и необратимости, мы назвали эту конструкцию *кандидатом в функции с секретом*.

Определение 1 Семейство кандидатов в функции с секретом — *это тройка*

$$\mathcal{C} = \{\mathrm{Key}_n, \mathrm{Eval}_n, \mathrm{Inv}_n\},$$

где:

- Key_n — *это семейство сэмплирующих схем* $\mathrm{Key}_n : \mathbb{B}^n \to \mathbb{B}^{\mathrm{pi}(n)} \times \mathbb{B}^{\mathrm{ti}(n)}$,
- Eval_n — *это семейство вычисляющих функцию схем* $\mathrm{Eval}_n : \mathbb{B}^{\mathrm{pi}(n)} \times \mathbb{B}^{m(n)} \to \mathbb{B}^{c(n)}$, *а*
- Inv_n — *это семейство обращающих функцию схем* $\mathrm{Inv}_n : \mathbb{B}^{\mathrm{ti}(n)} \times \mathbb{B}^{c(n)} \to \mathbb{B}^{m(n)}$,

[1] В русскоязычной литературе встречается термин «вентиль», а лет сорок назад он был общепринят. Но мы не рискнем его употреблять.

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

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

где $\mathrm{Key}_{n,1}(s)$ и $\mathrm{Key}_{n,2}(s)$ — первые $\mathrm{pi}(n)$ битов («публичная информация», public information) и последние $\mathrm{ti}(n)$ битов («секрет», trapdoor information) выхода схемы $\mathrm{Key}_n(s)$, соответственно.

Сейчас мы менее формально объясним смысл определения 1. Число n — это параметр надёжности, длина начального числа генератора случайных чисел. Длину входа функции 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 1 ничего не говорится о надёжности, а только вводятся обозначения для размерностей и устанавливается корректность обращения. Во всех конструкциях, встречающихся в настоящей работе, $m(n) = c(n)$ и $\mathrm{pi}(n) = \mathrm{ti}(n)$, но мы сочли разумным дать максимально общее определение.

Чтобы понять, насколько функция надёжна, нужно ввести понятие взлома функции. Неформально говоря, противник должен обратить функцию, не зная секрета. Мы вводим успешный взлом как обращение с вероятностью более 50% (позже мы рассмотрим и противников, имеющих меньшие вероятности успеха).

Через $C_{1/2}(f)$ мы будем обозначать минимальный размер схемы, которая вычисляет функцию $f \in \mathbb{B}_{n,m}$ на более чем половине её входов (длины n). Очевидно, $C_{1/2}(f) \le C(f)$ для любой функции f.

Определение 2 *Схема N* успешно обращает *семейство кандидатов в функции с секретом f на входах длины n, если для равномерного распределения U, взятого по начальным числам генератора $s \in \mathbb{B}^n$ и входам $m \in \mathbb{B}^{m(n)}$,*

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

Замечание 1 *На самом деле мы в дальнейшем докажем более сильный результат; мы докажем, что ни одна схема не сможет успешно обратить построенное нами семейство функций* ни для какого начального числа генератора *s, то есть для каждого начального числа s любой противник выдаёт неверный ответ в не менее чем 50% случаев.*

Чтобы семейство функций с секретом было надёжным, в контексте надёжности в слабом смысле достаточно, чтобы обращающие схемы были хоть немного (в константное число раз) больше, чем схемы, участвующие в его вычислении.

Определение 3 *Семейство кандидатов в функции с секретом f имеет* порядок надёжности *k, если для каждой последовательности схем $\{N_n\}_{n=1}^{\infty}$, которые успешно обращают f на каждой длине входа n,*

$$\liminf_{i \to \infty} \min\left\{\frac{\mathrm{Size}(N_n)}{C(\mathrm{Key}_n)}, \frac{\mathrm{Size}(N_n)}{C(\mathrm{Eval}_n)}, \frac{\mathrm{Size}(N_n)}{C(\mathrm{Inv}_n)}\right\} \ge k.$$

Замечание 2 *Как отмечалось в замечании 1, на самом деле мы докажем более сильный результат, а именно*

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

где $f_{\mathrm{pi}(n)+c(n)} \in \mathbb{B}_{\mathrm{pi}(n)+c(n),m(n)}$ отображает $\left(\mathrm{Key}_{n,1}(s), \mathrm{Eval}_n(\mathrm{Key}_{n,1}(s), m)\right)$ в m.

Замечание 3 *Можно было бы рассмотреть порождение ключей как отдельный процесс и исключить его сложность из определения порядка надёжности. Однако мы доказываем наши результаты для этого определения, и они от этого становятся только сильнее.*

Замечание 4 *Следует явно отметить, что мы говорим исключительно об* единовременной *надёжности. Противник может использовать меньшее количество 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).*

В следующих разделах мы разработаем конструкцию функции с секретом, надёжной в слабом смысле, т.е. семейства кандидатов в функцию с секретом, у которого будет нетривиальный порядок надёжности.

3 Матрицы сложных функций

Все наши конструкции основаны на линейной функции $f : \mathbb{B}^n \to \mathbb{B}^n$, для которой А. Хильтген доказал, что она является односторонней в слабом смысле с порядком надёжности 3/2 [Hil92]. Мы ограничимся случаем, когда $n \equiv 0 \pmod 4$, по причинам, которые будут изложены чуть ниже. Заметим, что определение 3 без проблем выдерживает это ограничение: для $n \not\equiv 0 \pmod 4$ можно просто рассмотреть схему, длина входа которой представляет собой наименьшее делящееся на 4 число, превосходящее n.

В дальнейшем мы обозначим матрицу функции Хильтгена f через A. В каждой строке A стоят по две единицы (на диагонали и над ней), кроме последней строки, в которой единиц целых три: в первом столбце, в столбце $\lceil \frac{n}{2} \rceil$ и в последнем столбце. Вот матрицы A и A^{-1}:

$$A\boldsymbol{x} = \begin{pmatrix} 1\,1\,0\ldots0\ldots0\,0\,0 \\ 0\,1\,1\ldots0\ldots0\,0\,0 \\ 0\,0\,1\ldots0\ldots0\,0\,0 \\ \vdots\,\vdots\,\vdots\quad\vdots\quad\vdots\,\vdots\,\vdots \\ 0\,0\,0\ldots0\ldots1\,1\,0 \\ 0\,0\,0\ldots0\ldots0\,1\,1 \\ 1\,0\,0\ldots1\ldots0\,0\,1 \end{pmatrix}\boldsymbol{x}, \quad A^{-1}\boldsymbol{x} = \left(\begin{array}{c|c} 0\,0\ldots0 & 1\,1\ldots1\,1 \\ 1\,0\ldots0 & 1\,1\ldots1\,1 \\ 1\,1\ldots0 & 1\,1\ldots1\,1 \\ \vdots\,\vdots\quad\vdots & \vdots\,\vdots\quad\vdots\,\vdots \\ 1\,1\ldots0 & 1\,1\ldots1\,1 \\ \hline 1\,1\ldots1 & 1\,1\ldots1\,1 \\ 1\,1\ldots1 & 0\,1\ldots1\,1 \\ 1\,1\ldots1 & 0\,0\ldots1\,1 \\ \vdots\,\vdots\quad\vdots & \vdots\,\vdots\quad\vdots\,\vdots \\ 1\,1\ldots1 & 0\,0\ldots0\,1 \end{array}\right)\boldsymbol{x}.$$

Заметим, что в каждой строке A^{-1} есть по меньшей мере $\lceil \frac{n}{2} \rceil$ ненулевых элементов (единиц), а $\lceil \frac{n}{2} \rceil$-я строка и вовсе нулей не содержит.

Кроме матрицы A^{-1}, нам также потребуется её квадрат A^{-2}, который выглядит как

$$A^{-2}\boldsymbol{x} = \left(\begin{array}{cccccc|cccccc}
1&1&1&1&\dots&1\,1\,1 & 1&0&1&\dots&1\,0\,1\\
1&1&1&1&\dots&1\,1\,1 & 0&1&0&\dots&0\,1\,0\\
0&1&1&1&\dots&1\,1\,1 & 1&0&1&\dots&1\,0\,1\\
1&0&1&1&\dots&1\,1\,1 & 0&1&0&\dots&0\,1\,0\\
0&1&0&1&\dots&1\,1\,1 & 1&0&1&\dots&1\,0\,1\\
\vdots&\vdots&\vdots&\vdots&&\vdots\,\vdots\,\vdots & \vdots&\vdots&\vdots&&\vdots\,\vdots\\
1&0&1&0&\dots&1\,1\,1 & 0&1&0&\dots&0\,1\,0\\
0&1&0&1&\dots&0\,1\,1 & 1&0&1&\dots&1\,0\,1\\
\hline
1&0&1&0&\dots&1\,0\,1 & 0&1&0&\dots&0\,1\,0\\
0&1&0&1&\dots&0\,1\,0 & 1&0&1&\dots&1\,0\,1\\
1&0&1&0&\dots&1\,0\,1 & 1&1&0&\dots&0\,1\,0\\
0&1&0&1&\dots&0\,1\,0 & 1&1&1&\dots&1\,0\,1\\
\vdots&\vdots&\vdots&\vdots&&\vdots\,\vdots\,\vdots & \vdots&\vdots&\vdots&&\vdots\,\vdots\\
0&1&0&1&\dots&1\,0\,1 & 1&1&1&\dots&1\,0\,1\\
1&0&1&0&\dots&0\,1\,0 & 1&1&1&\dots&1\,1\,0
\end{array}\right)\boldsymbol{x},$$

если n делится на 4 (именно поэтому мы ограничиваемся этим случаем). Заметим, что у этой матрицы каждая строка содержит по меньшей мере $\frac{n}{2}$ единиц; более того, её треугольные блоки, состоящие из единиц, совпадают с треугольными блоками матрицы A^{-1}, состоящими из нулей, а остальное пространство покрыто нулями и единицами в шахматном порядке.

Нас также интересует матрица $\mathfrak{A}$ размером $n \times 2n$, состоящая из A^{-2} и A^{-1}, расположенных друг за другом:

$$\mathfrak{A} = \left(A^{-2}\, A^{-1}\right).$$

Нам потребуются следующие свойства A^{-1} и $\mathfrak{A}$.

Лемма 1 *Пусть* $n \equiv 0 \pmod 4$. *Тогда*

1. *Все столбцы* $\mathfrak{A}$ *(и, следовательно,* A^{-1}*) различны.*
2. *В каждой строке матрицы* A^{-1} *(соответственно,* $\mathfrak{A}$*) есть по меньшей мере* $\frac{n}{2}$ *(соответственно,* $\frac{5n}{4}$*) ненулевых элементов.*
3. *После удаления всех, кроме двух (соответственно, всех, кроме пяти) столбцов матрицы* A^{-1} *(соответственно,* $\mathfrak{A}$*) в ней останется по крайней мере одна строка с двумя ненулевыми элементами.*

Доказательство. Первое утверждение очевидно.

Строка матрицы A^{-1} с номером i содержит $\frac{n}{2} + i$ ненулевых элементов для $i \le \frac{n}{2}$ и $\frac{n}{2} + n - i$ ненулевых элементов для $i \ge \frac{n}{2}$. Таким образом, второе утверждение выполняется для матрицы A^{-1}. В то же самое время, i-я строка A^{-2} содержит по меньшей мере $\frac{3n}{4} - \frac{i}{2}$ ненулевых элементов для $i \le \frac{n}{2}$ и по меньшей мере $\frac{n}{2} + \frac{1}{2}(i - \frac{n}{2} - 1)$ ненулевых элементов для $i \ge \frac{n}{2}$. Поэтому i-я строка A^{-2} содержит по меньшей мере

$$\frac{n}{2} + i + \frac{3n}{4} - \frac{i}{2} = \frac{5n}{4} + \frac{i}{2}$$

ненулевых элементов для $i \le \frac{n}{2}$ и по меньшей мере

$$\frac{n}{2} + n - i + \frac{n}{2} + \frac{1}{2}(i - \frac{n}{2} - 1) = \frac{7n}{4} - \frac{1}{2}(i-1) \ge \frac{5n}{4}$$

ненулевых элементов для $i \geq \frac{n}{2}$.

Докажем теперь третий пункт леммы. Поскольку в A^{-1} есть строка, целиком состоящая из единиц, придётся удалить все столбцы этой матрицы, кроме одного, чтобы в строке осталась только одна единица. То же самое верно для левой части матрицы A^{-2} (обратите внимание на её первую строку), а также для правой части матрицы A^{-2}, за исключением её последнего столбца (обратите внимание на её последнюю строку). □

4 Исключение гейтов

В этом разделе мы сначала кратко напомним методы доказательства оценок, использованные в работах Хильтгена, а затем введём метод исключения гейтов как основной метод доказательства нижних оценок в этой работе. Хильтген доказал все свои оценки при помощи следующего весьма простого наблюдения, впервые сделанного Ламаньей и Сэвиджем.

Предложение 1 ([LS73,Sav76]; [Hil92, Теоремы 3 и 4]) *1. Предположим, что функция $f : \mathbb{B}^n \to \mathbb{B}$ нетривиально зависит от каждой из n своих переменных, то есть 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 $C(f) \geq n - 1$.

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

Доказательство. 1. Рассмотрим минимальную схему размера s, вычисляющую f. Поскольку f зависит (здесь и далее мы будем говорить «зависит», имея в виду «нетривиально зависит») от всех n своих переменных, у каждого из входов схемы должно быть по крайней мере одно исходящее ребро. Поскольку схема минимальна, у каждого из оставшихся гейтов в ней, кроме, возможно, выхода, тоже должно быть по крайней мере одно исходящее ребро. Поэтому в схеме должно быть по крайней мере $s + n - 1$ рёбер. С другой стороны, в схеме из s бинарных гейтов не может быть более чем $2s$ рёбер. Поэтому $2s \geq s + n - 1$.

2. Рассмотрим схему, вычисляющую f. Отметим, что в ней есть по меньшей мере $c - 1$ гейтов, не вычисляющих никакую функцию схемной сложности c или более (в качестве таких гейтов можно рассмотреть первые $c - 1$ гейтов в каком-либо топологическом порядке). Но для вычисления любой функции–компоненты $f^{(i)}$ требуется добавить по меньшей мере ещё один гейт, причём потребуется добавлять по одному гейту для каждой компоненты, ведь один новый гейт добавляет только одну новую функцию. Так и получается необходимая оценка в $c+m-1$ гейтов. □

Оценки Хильтгена доказаны следующим образом: сначала доказывается оценка на сложность вычисления одного бита выхода (например, поскольку в каждой строке A есть по крайней мере $\frac{n}{2}$ единиц, то минимальная сложность каждой компоненты $A\boldsymbol{y}$ составляет $\frac{n}{2} - 1$), а из них получаются нижние оценки на сложность вычисления всей функции, то есть обращения A^{-1} (например, сложность вычисления $A\boldsymbol{y}$ составляет $\frac{n}{2} + n - 2 = \frac{3n}{2} - 2$).

Кроме того, в криптографии обычно желательно доказать не только оценки в худшем случае, но также тот факт, что противник неспособен обратить функцию на значительной доле её входов. Первым шагом к доказательству этого утверждения может стать следующий факт.

Предложение 2 *Для каждой из матриц A, A^2, $\mathfrak{A}$ любая схема, использующая менее чем минимально необходимое для точного вычисления соответствующей функции количество гейтов, обращает функцию на не более чем $\frac{1}{2}$ всех входов.*

В работах Хильтгена этот факт доказывался при помощи следующего очень простого наблюдения (которое там даже явно не сформулировано).

Лемма 2 *Рассмотрим функцию $f = \bigoplus_{i=1}^{n} x_i$. Для каждой функции g, которая нетривиально зависит только от m из этих переменных при $m < n$,*

$$\Pr_{x_1,\ldots,x_n}[f(x_1,\ldots,x_n) = g(x_{i_1},\ldots,x_{i_m})] = \frac{1}{2}.$$

Доказательство. Поскольку $m < n$, существует такой номер $j \in 1..n$, что g не зависит от x_j. Это значит, что для каждого набора значений всех остальных переменных для одного из значений x_j результат вычисления g совпадает с результатом f, а для другого значения не совпадает. Это значит, что f отличается от g в точности на $\frac{1}{2}$ своих входов. □

Такого рассуждения вполне достаточно для необратимости в слабом смысле по Хильтгену для квадратной матрицы A^{-1}: сначала мы применяем первую часть предложения 1 и получаем, что сложность каждого выхода по меньшей мере $\frac{n}{2} - 1$, а затем вторая часть предложения 1 даёт нам желаемую оценку в $\frac{3n}{2} - 1$ гейтов. Более того, если у схемы гейтов меньше, чем требуется, один из её выходов неизбежно должен будет зависеть от меньшего числа входных переменных, чем нужно. А это, по лемме 2, немедленно даёт нам долю ошибки в $\frac{1}{2}$.

Однако в этой работе мы пользуемся также и прямоугольными матрицами, и оказывается, что аналогичное простое рассуждение уже не так замечательно работает в этом случае. Поэтому нам придётся использовать другой способ доказательства нижних оценок, а именно *исключение гейтов*, которое до сих пор было использовано во всех доказательствах нижних оценок на размеры схем произвольного вида [Weg87].

Основная идея этого метода — индуктивное рассуждение следующего вида. Рассмотрим функцию f и подставим некоторое значение c в некоторую переменную x, получив таким образом схему, вычисляющую функцию $f|_{x=c}$. Эту схему можно упростить, потому что гейты, в которые входила эта переменная, теперь либо стали унарными (а отрицание можно удалить из схемы, перенеся его роль на следующие гейты), либо и вовсе превратились в константы (и тогда можно будет удалять и их потомков). Важный случай здесь тот, когда гейт нелинеен, например гейт, вычисляющий конъюнкцию AND или дизъюнкцию OR. В этом случае всегда возможно выбрать такое значение для входа, что гейт станет константным. Этот процесс можно продолжать индуктивно, пока можно выбирать подходящую переменную, входящую в достаточное количество гейтов. Очевидно, количество исключённых в течение этого процесса гейтов представляет собой нижнюю оценку на схемную сложность f.

В нашей работе исключение гейтов использовано в одной из своих простейших форм. Вот главная лемма, которую мы позже применим к интересующим нас матрицам.

Лемма 3 *Пусть $t, u \geq 1$. Предположим, что $\chi : \mathbb{B}^{v(n)} \to \mathbb{B}^n$ — это линейная функция с матрицей X над полем $GF(2)$. Предположим также, что:*

- *все столбцы X различны;*
- *в каждой строке X есть по меньшей мере u ненулевых элементов;*
- *после удаления любых t столбцов X в матрице всё ещё останется строка, содержащая по крайней мере два ненулевых элемента.*

Тогда $C(\chi) \geq u + t$ *и, более того,* $C_{1/2}(\chi) \geq u + t$.

Доказательство. Рассмотрим схему, вычисляющую χ на более чем $\frac{1}{2}$ её входов. Зафиксируем какой-нибудь топологический порядок на её входах. Мы будем обозначать функцию, которую схема вычисляет *на самом деле*, через h (h не обязательно линейна, но должна совпадать с χ на большинстве входов).

Рассмотрим самый верхний гейт g по зафиксированному порядку. Поскольку g — самый верхний, входящие в него рёбра должны идти из входов схемы. Обозначим их через x и y. По лемме 2, ни один из этих входов не может сам по себе быть выходом схемы, ведь даже после удаления первых $u - 1$ столбцов из матрицы X каждая строка будет содержать по меньшей мере две единицы, то есть каждый выход будет зависеть по меньшей мере от двух переменных.

Возможны следующие случаи.

1. Одна из переменных, входящих в g, например x, входит также в какой-нибудь другой гейт. В этом случае, подставив любое значение вместо x, мы сможем исключить два гейта.
2. g — гейт, вычисляющий нелинейную функцию, и ни x, ни y не подаются на вход другим гейтам. Если у g есть потомки, то, подставив в качестве x подходящую константу, мы сможем исключить и сам гейт g, и его потомков, то есть по меньшей мере два гейта.
   Если же у g нет потомков, это значит, что g является выходом; рассмотрим этот выход h_i. Если χ_i зависит от какой-либо другой переменной z, мы достигаем вероятности ошибки в 50%, меняя значения z. Для этого рассмотрим два различных значения переменной z (0 и 1), и зафиксируем значения всех остальных переменных. Для одного из значений z функция h будет непременно давать неверный ответ, потому что
   $$h_i(\ldots, z = 0, x = a, y = b, \ldots) = h_i(\ldots, z = 1, x = a, y = b, \ldots)$$
   (h_i не зависит от z), но
   $$\chi_i(\ldots, z = 0, x = a, y = b, \ldots) \neq \chi_i(\ldots, z = 1, x = a, y = b, \ldots),$$
   потому что $\chi_i = z \oplus \ldots$. А если $\chi_i = x$, $\chi_i = y$, $\chi_i = x \oplus y$ или $\chi_i = x \oplus y \oplus 1$, мы можем заменить g на линейный гейт (а то и его отсутствие), который будет точно вычислять χ_i; при такой замене в схеме больше ничего не изменится, поскольку у g нет детей. Следовательно, мы попадаем в условия случая 3.
3. g вычисляет линейную функцию, и ни x, ни y не подаются на вход другим гейтам. В этом случае h не зависит от x или y по отдельности, а зависит только от $x \oplus y$. Для схемы, которая вычисляла бы χ точно, это значило бы, что в матрице A есть два совпадающих столбца, что приводит к противоречию.
   Но наша схема всего лишь пытается приблизить χ. Давайте покажем, что в этом случае h отличается от χ по крайней мере на половине своих входов. Во-первых, заметим, что поскольку χ зависит от x и y по отдельности, существует выход χ_i, который зависит только от одной из этих переменных, например x (других возможностей быть не может, поскольку χ линейна). Тогда $\chi_i = x \oplus \bigoplus_{z \in Z} z$, где $y \notin Z$. Возможны два случая: либо выход h_i функции h зависит от $x \oplus y$, либо он не зависит ни от x, ни от y.
   В первом случае рассмотрим два различных значения переменной y (0 и 1), и зафиксируем значения всех остальных переменных. Для одного из значений y функция h будет непременно давать неверный ответ, потому что
   $$h_i(\ldots, y = 0, x = a, \ldots) \neq h_i(\ldots, y = 1, x = a)$$

(когда мы меняем значение y, мы тем самым меняем $x \oplus y$), но

$$\chi_i(\ldots, y = 0, x = a, \ldots) = \chi_i(\ldots, y = 1, x = a),$$

потому что χ_i вовсе не зависит от y. Во втором случае рассуждение аналогично, но менять нужно значение x: h не изменит значение выхода, а χ изменит.

Таким образом, пока все строки A содержат больше одного ненулевого элемента, мы можем исключать по меньшей мере два гейта после подстановки одной переменной; будем называть такие переменные «хорошими». Всего это даст нам по крайней мере $2(u-1)$ гейтов. Заметим, что подстановка переменной эквивалентна удалению столбца из матрицы.

Что произойдёт, когда мы удалим все «хорошие» переменные? Мы продолжаем тот же процесс исключения гейтов[2]. Этот процесс можно продолжать до тех пор, пока в матрице присутствует хотя бы одна строка с двумя ненулевыми элементами: если хотя бы одна такая строка есть, значит, схема должна быть нетривиальной, и, значит, верхний гейт существует. Правда, теперь возможен ещё и случай (и в точном вычислении функции χ, и в её приближении), когда одна из входных переменных сразу же подаётся на выход всей функции, и мы, таким образом, сможем исключить за один шаг только один гейт. Однако исключение одного гейта после подстановки переменной нам гарантировано в любом случае. Таким образом, мы суммарно исключим по меньшей мере

$$2(u-1) + ((t+1) - (u-1)) = u + t$$

гейтов. □

В своих конструкциях мы будем использовать и блочно-диагональные матрицы. Конечно, «интуитивно ясно», что совместное вычисление двух функций, у которых ни входы, ни выходы не пересекаются, должно быть не проще, чем их вычисление по отдельности, а нижняя оценка должна получаться как сумма соответствующих нижних оценок. Но, как ни странно, этот факт в общем случае неверен, как показано, например, в [Weg87, Section 10.2]. Поэтому мы покажем только частный случай, для интересующих нас линейных функций. Для ясности изложения мы сначала докажем результат для двух блоков, а затем распространим его на произвольное их количество.

Лемма 4 *Пусть матрица X_1 размером $n \times v_1(n)$ и матрица X_2 размером $w(n) \times v_2(n)$ удовлетворяют условиям леммы 3 с параметрами $u_1(n)$ (соответственно, $u_2(n)$) и $t_1(n)$ (соответственно, $t_2(n)$). (Отметим, что они могут быть прямоугольными и иметь разный размер.)*

Обозначим через $\zeta(\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}) = (X_1\boldsymbol{x}^{(1)}, X_2\boldsymbol{x}^{(2)})$ функцию, определённую блочно-диагональной матрицей

$$\begin{pmatrix} X_1 & 0 \\ 0 & X_2 \end{pmatrix}.$$

Тогда

$$C(\zeta) \geq u_1(n) + t_1(n) + u_2(n) + t_2(n) - 1$$

и, более того,

$$C_{1/2}(\zeta) \geq u_1(n) + t_1(n) + u_2(n) + t_2(n) - 1.$$

Доказательство. Доказательство почти повторяет доказательство леммы 3. Заметим, что когда мы подставляем переменную из $x^{(1)}$, она ничего не меняет в матрице X_2, и наоборот. Таким образом,

[2] В доказательстве этой леммы его можно было бы и не продолжать, а сразу получить результат из утверждения 1, но мы позже используем это доказательство ещё раз в лемме 4.

мы подставляем «хорошие» переменные (которые позволяют исключить два гейта) до тех пор, пока они есть, а затем подставляем «плохие» переменные (которые исключают только один гейт) *отдельно для каждой матрицы*. Если одна из матриц тривиализуется, то есть в ней не остаётся строк с двумя ненулевыми элементами (это может произойти после того, как мы подставим в неё $u_i(n) - 1$ «хороших», а затем $t_i(n) - u_i(n) + 2$ других переменных), мы просто подставим все остальные переменные, соответствующие этой части блочно-диагональной матрицы (не исключая ни одного гейта — матрица своё уже отработала) и забудем о ней.

Однако может случиться так, что один из входов в верхний гейт происходит из $x^{(1)}$, а другой — из $x^{(2)}$. Это ничего не меняет, когда обе переменные «хорошие» или обе «плохие». Однако если «хороша» только одна из них, ситуация немного меняется, и нам придётся перепроверить случаи 1–3 из доказательства леммы 3.

Первые два случая проходят гладко: мы подставляем значение в ту из переменных, которая позволяет удалить два гейта. Случай 3 всё ещё не может произойти в схеме, вычисляющей функцию ζ точно, потому что у нас всё ещё невозможны одинаковые столбцы, да и в приближении доказательство ничем не отличается от леммы 3. □

Лемма 5 *Пусть линейная функция ζ определяется блочно-диагональной матрицей*

$$\zeta\left(\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \ldots, \boldsymbol{x}^{(m)}\right) = \begin{pmatrix} X_1 & 0 & \ldots & 0 \\ 0 & X_2 & \ldots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \ldots & X_m \end{pmatrix} \begin{pmatrix} \boldsymbol{x}^{(1)} \\ \boldsymbol{x}^{(2)} \\ \vdots \\ \boldsymbol{x}^{(m)} \end{pmatrix},$$

и матрицы X_j удовлетворяют условиям леммы 3 с параметрами $u_j(n)$ и $t_j(n)$, соответственно (матрицы могут быть прямоугольными и иметь разный размер). Тогда

$$C(\zeta) \geq \sum_{j=1}^{m} (u_j(n) + t_j(n)) - m + 1$$

и, более того,

$$C_{1/2}(\zeta) \geq \sum_{j=1}^{m} (u_j(n) + t_j(n)) - m + 1.$$

Доказательство. Все рассуждения об исключении гейтов касаются ровно двух входных переменных. Эти переменные могут происходить из не более чем двух различных блоков матрицы функции ζ. Следовательно, доказательство в данном случае полностью идентично доказательству леммы 4. □

Сформулируем теперь прямые следствия этих лемм и комбинаторные леммы о конкретных интересующих нас матрицах.

Лемма 6 *Пусть $n, n' \equiv 0 \pmod 4$,*

$$\alpha(\boldsymbol{x}) = A^{-1}\boldsymbol{x}, \qquad \alpha_2(\boldsymbol{x}) = \begin{pmatrix} A^{-1} & A^{-2} \end{pmatrix} \boldsymbol{x}, \qquad \alpha_*(\boldsymbol{x}) = \begin{pmatrix} A^{-1} & A^{-2} & 0 \\ 0 & 0 & A_*^{-1} \end{pmatrix} \boldsymbol{x},$$

где A_*^{-1} обозначает матрицу с той же структурой, что и A^{-1}, но с размерностью n' вместо n. Тогда

$$C(\alpha) \geq \frac{3n}{2} - 2, \qquad C(\alpha_2) \geq \frac{13n}{4} - 5, \qquad C(\alpha_*) \geq \frac{3n'}{2} + \frac{13n}{4} - 7,$$

и, более того,

$$C_{1/2}(\alpha) \geq \frac{3n}{2} - 2, \qquad C_{1/2}(\alpha_2) \geq \frac{13n}{4} - 5, \qquad C_{1/2}(\alpha_*) \geq \frac{3n'}{2} + \frac{13n}{4} - 7,$$

Доказательство. Следует из леммы 3 и леммы 4 подстановкой доказанных в лемме 1 оценок на $u(n)$ и $t(n)$ (в частности, $t(n) = n - 2$ для матрицы A^{-1}, а для матрицы $\mathfrak{A}$ $t(n) = 2n - 5$). □

С другой стороны, верна следующая верхняя оценка.

Лемма 7 *Следующие утверждения верны.*

1. *Существует схема размера* $\frac{3n}{2} - 1$, *вычисляющая линейную функцию* $\phi : \mathbb{B}^n \to \mathbb{B}^n$ *с матрицей* A^{-1}.
2. *Существует схема размера* $\frac{7n}{2}$, *вычисляющая линейную функцию* $\phi : \mathbb{B}^{2n} \to \mathbb{B}^n$ *с матрицей* $\begin{pmatrix} A^{-1} & A \end{pmatrix}$.
3. *Существует схема размера* $\frac{5n}{2} - 1$, *вычисляющая линейную функцию* $\phi : \mathbb{B}^{2n} \to \mathbb{B}^n$ *с матрицей* $\begin{pmatrix} A^{-1} & A^{-1} \end{pmatrix}$.

Доказательство. 1. Сначала построим большую сумму $\bigoplus_{i=1}^{n/2} x_i$ (на это потребуется $\frac{n}{2} - 1$ гейтов). Затем, последовательно добавляя входы x_i, $i = \frac{n}{2}..n$, мы сможем вычислить выходы y_i, $i = \frac{n}{2}..n$, а в результате получится сумма всех входов $\bigoplus_{i=1}^{n} x_i$ (на это уйдёт ещё $\frac{n}{2}$ гейтов). Наконец, первые $\frac{n}{2}$ выходов получатся последовательным «вычитанием» первых $\frac{n}{2}$ входов из суммы всех входов (ещё $\frac{n}{2}$ гейтов).

2. Мы только что выяснили, что для того чтобы реализовать левую часть этой матрицы, нужны $\frac{3n}{2} - 1$ гейтов. Затем мы просто прибавим к каждому выходу по отдельности по два бита, которые нужно добавить из правой части (в последней строке — три бита); это займёт ещё $2n + 1$ гейтов.

3. Заметим, что в данном случае

$$\phi(a, b) = \begin{pmatrix} A^{-1} & A^{-1} \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = A^{-1}(a \oplus b)$$

для любых $a, b \in \mathbb{B}^n$. Поэтому мы сначала сложим $a \oplus b$ (что займёт n гейтов), а затем реализуем A^{-1} (что займёт ещё $\frac{3n}{2} - 1$ гейтов).

□

5 Простейший пример

Обычно очень полезно привести простой и наглядный пример основной конструкции, прежде чем начинать подробно описывать доказательство в общем случае. В этом разделе мы покажем пример конструкции функции с секретом, надёжной в слабом смысле.

Как мы уже упоминали, все конструкции здесь основаны на линейной функции $f : \mathbb{B}^n \to \mathbb{B}^n$, разработанной А. Хильтгеном [Hil92]. Наименьшее n, для которого f действительно труднее обратить, чем вычислить, равно 7, но поскольку нам нужно, чтобы n делилось на 4, мы в этом примере будем рассматривать $n = 8$ (кроме того, в терминах раздела 7 мы используем $\alpha = 1$). При

этом

$$A = \begin{pmatrix} 1&1&0&0&0&0&0&0\\ 0&1&1&0&0&0&0&0\\ 0&0&1&1&0&0&0&0\\ 0&0&0&1&1&0&0&0\\ 0&0&0&0&1&1&0&0\\ 0&0&0&0&0&1&1&0\\ 0&0&0&0&0&0&1&1\\ 1&0&0&1&0&0&0&1 \end{pmatrix}, \quad A^{-1} = \begin{pmatrix} 0&0&0&1&1&1&1&1\\ 1&0&0&1&1&1&1&1\\ 1&1&0&1&1&1&1&1\\ 1&1&1&1&1&1&1&1\\ 1&1&1&0&1&1&1&1\\ 1&1&1&0&0&1&1&1\\ 1&1&1&0&0&0&1&1\\ 1&1&1&0&0&0&0&1 \end{pmatrix}, \quad A^{-2} = \begin{pmatrix} 1&1&1&1&0&1&0&1\\ 1&1&1&0&1&0&1&0\\ 0&1&1&1&0&1&0&1\\ 1&0&1&0&1&0&1&0\\ 0&1&0&1&0&1&0&1\\ 1&0&1&1&1&0&1&0\\ 0&1&0&1&1&1&0&1\\ 1&0&1&1&1&1&1&0 \end{pmatrix}.$$

В данном случае $C(f) = 9$, $C(f^{-1}) = 11$, $C(f^{-2}) = 11$.

Наш кандидат в функции с секретом будет состоять из следующих компонентов.

$$\mathrm{Key}_8 = \left(\begin{array}{c} 1&0&0&0&0&0&0&0\\ 0&1&0&0&0&0&0&0\\ 0&0&1&0&0&0&0&0\\ 0&0&0&1&0&0&0&0\\ 0&0&0&0&1&0&0&0\\ 0&0&0&0&0&1&0&0\\ 0&0&0&0&0&0&1&0\\ 0&0&0&0&0&0&0&1\\ \hline 0&0&0&1&1&1&1&1\\ 1&0&0&1&1&1&1&1\\ 1&1&0&1&1&1&1&1\\ 1&1&1&1&1&1&1&1\\ 1&1&1&0&1&1&1&1\\ 1&1&1&0&0&1&1&1\\ 1&1&1&0&0&0&1&1\\ 1&1&1&0&0&0&0&1 \end{array}\right), \quad \mathrm{Eval}_8 = \left(\begin{array}{c|c|c} 0&0&0&1&1&1&1&1 & 1&1&0&0&0&0&0&0 & 0&0&0&0&0&0&0&0\\ 1&0&0&1&1&1&1&1 & 0&1&1&0&0&0&0&0 & 0&0&0&0&0&0&0&0\\ 1&1&0&1&1&1&1&1 & 0&0&1&1&0&0&0&0 & 0&0&0&0&0&0&0&0\\ 1&1&1&1&1&1&1&1 & 0&0&0&1&1&0&0&0 & 0&0&0&0&0&0&0&0\\ 1&1&1&0&1&1&1&1 & 0&0&0&0&1&1&0&0 & 0&0&0&0&0&0&0&0\\ 1&1&1&0&0&1&1&1 & 0&0&0&0&0&1&1&0 & 0&0&0&0&0&0&0&0\\ 1&1&1&0&0&0&1&1 & 0&0&0&0&0&0&1&1 & 0&0&0&0&0&0&0&0\\ 1&1&1&0&0&0&0&1 & 1&0&0&1&0&0&0&1 & 0&0&0&0&0&0&0&0\\ \hline 0&0&0&0&0&0&0&0 & 0&0&0&0&0&0&0&0 & 1&1&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0 & 0&0&0&0&0&0&0&0 & 0&1&1&0&0&0&0&0\\ 0&0&0&0&0&0&0&0 & 0&0&0&0&0&0&0&0 & 0&0&1&1&0&0&0&0\\ 0&0&0&0&0&0&0&0 & 0&0&0&0&0&0&0&0 & 0&0&0&1&1&0&0&0\\ 0&0&0&0&0&0&0&0 & 0&0&0&0&0&0&0&0 & 0&0&0&0&1&1&0&0\\ 0&0&0&0&0&0&0&0 & 0&0&0&0&0&0&0&0 & 0&0&0&0&0&1&1&0\\ 0&0&0&0&0&0&0&0 & 0&0&0&0&0&0&0&0 & 0&0&0&0&0&0&1&1\\ 0&0&0&0&0&0&0&0 & 0&0&0&0&0&0&0&0 & 1&0&0&1&0&0&0&1 \end{array}\right),$$

$$\mathrm{Inv}_8 = \left(\begin{array}{c|c|c} 0&0&0&1&1&1&1&1 & 0&0&0&1&1&1&1&1 & 0&0&0&0&0&0&0&0\\ 1&0&0&1&1&1&1&1 & 1&0&0&1&1&1&1&1 & 0&0&0&0&0&0&0&0\\ 1&1&0&1&1&1&1&1 & 1&1&0&1&1&1&1&1 & 0&0&0&0&0&0&0&0\\ 1&1&1&1&1&1&1&1 & 1&1&1&1&1&1&1&1 & 0&0&0&0&0&0&0&0\\ 1&1&1&0&1&1&1&1 & 1&1&1&0&1&1&1&1 & 0&0&0&0&0&0&0&0\\ 1&1&1&0&0&1&1&1 & 1&1&1&0&0&1&1&1 & 0&0&0&0&0&0&0&0\\ 1&1&1&0&0&0&1&1 & 1&1&1&0&0&0&1&1 & 0&0&0&0&0&0&0&0\\ 1&1&1&0&0&0&0&1 & 1&1&1&0&0&0&0&1 & 0&0&0&0&0&0&0&0\\ \hline 0&0&0&0&0&0&0&0 & 0&0&0&0&0&0&0&0 & 0&0&0&1&1&1&1&1\\ 0&0&0&0&0&0&0&0 & 0&0&0&0&0&0&0&0 & 1&0&0&1&1&1&1&1\\ 0&0&0&0&0&0&0&0 & 0&0&0&0&0&0&0&0 & 1&1&0&1&1&1&1&1\\ 0&0&0&0&0&0&0&0 & 0&0&0&0&0&0&0&0 & 1&1&1&1&1&1&1&1\\ 0&0&0&0&0&0&0&0 & 0&0&0&0&0&0&0&0 & 1&1&1&0&1&1&1&1\\ 0&0&0&0&0&0&0&0 & 0&0&0&0&0&0&0&0 & 1&1&1&0&0&1&1&1\\ 0&0&0&0&0&0&0&0 & 0&0&0&0&0&0&0&0 & 1&1&1&0&0&0&1&1\\ 0&0&0&0&0&0&0&0 & 0&0&0&0&0&0&0&0 & 1&1&1&0&0&0&0&1 \end{array}\right).$$

Легко проверить, что, действительно,

$$\mathrm{Inv}_8 \begin{pmatrix} \mathrm{Key}_{8,2}(r) \\ \mathrm{Eval}_8(\mathrm{Key}_{8,1}(r), m) \end{pmatrix} = m$$

для каждого входа $m \in \mathbb{B}^8$ и любой битовой строки (начального значения) $r \in \mathbb{B}^8$, где $\mathrm{Key}_{8,1}(r)$ и $\mathrm{Key}_{8,2}(r)$ — первая и вторая половина (первые и вторые восемь бит) $\mathrm{Key}_8(r)$, соответственно. Лемма 7 позволяет оценить сложности этих функций:

$$C(\mathrm{Key}_8) = 11,$$
$$C(\mathrm{Eval}_8) = 29,$$
$$C(\mathrm{Inv}_8) = 30.$$

Противнику, чтобы обратить эту функцию с секретом, придётся вычислить следующую функцию:

$$\mathrm{Adv}_8 = \left(\begin{array}{cccc|cccc|cccc}
1&1&1&1&0&1&0&1 & 0&0&0&1&1&1&1&1 & 0&0&0&0&0&0&0&0\\
1&1&1&0&1&0&1&0 & 1&0&0&1&1&1&1&1 & 0&0&0&0&0&0&0&0\\
0&1&1&1&0&1&0&1 & 1&1&0&1&1&1&1&1 & 0&0&0&0&0&0&0&0\\
1&0&1&0&1&0&1&0 & 1&1&1&1&1&1&1&1 & 0&0&0&0&0&0&0&0\\
0&1&0&1&0&1&0&1 & 1&1&1&0&1&1&1&1 & 0&0&0&0&0&0&0&0\\
1&0&1&1&1&0&1&0 & 1&1&1&0&0&1&1&1 & 0&0&0&0&0&0&0&0\\
0&1&0&1&1&1&0&1 & 1&1&1&0&0&0&1&1 & 0&0&0&0&0&0&0&0\\
1&0&1&1&1&1&1&0 & 1&1&1&0&0&0&0&1 & 0&0&0&0&0&0&0&0\\
\hline
0&0&0&0&0&0&0&0 & 0&0&0&0&0&0&0&0 & 0&0&0&1&1&1&1&1\\
0&0&0&0&0&0&0&0 & 0&0&0&0&0&0&0&0 & 1&0&0&1&1&1&1&1\\
0&0&0&0&0&0&0&0 & 0&0&0&0&0&0&0&0 & 1&1&0&1&1&1&1&1\\
0&0&0&0&0&0&0&0 & 0&0&0&0&0&0&0&0 & 1&1&1&1&1&1&1&1\\
0&0&0&0&0&0&0&0 & 0&0&0&0&0&0&0&0 & 1&1&1&0&1&1&1&1\\
0&0&0&0&0&0&0&0 & 0&0&0&0&0&0&0&0 & 1&1&1&0&0&1&1&1\\
0&0&0&0&0&0&0&0 & 0&0&0&0&0&0&0&0 & 1&1&1&0&0&0&1&1\\
0&0&0&0&0&0&0&0 & 0&0&0&0&0&0&0&0 & 1&1&1&0&0&0&0&1
\end{array}\right).$$

По лемме 6,

$$C(\mathrm{Adv}_8) \geq 31.$$

Таким образом, мы получили функцию с секретом, надёжную в слабом смысле, для которой обращение без знания секретного ключа занимает по меньшей мере 31 гейт, в то время как вычисление самой функции и её обращение с секретом можно реализовать за 29 и 30 гейтов соответственно. Это значит, что этот кандидат в функции с секретом имеет, так сказать, порядок надёжности $\frac{31}{30}$ («так сказать», потому что определение 3 рассматривает только пределы при стремящемся к бесконечности n). В дальнейшем мы обобщим эту конструкцию и докажем порядок надёжности $\frac{25}{22}$.

6 Две конструкции

Уже почти всё готово для того, чтобы представить конструкции наших функций с секретом, доказуемо надёжных в слабом смысле. В этом разделе мы рассмотрим две различные конструкции кандидатов в функции с секретом. Ни одна из них не работает; но в следующем разделе мы их совместим, и уже комбинация будет иметь нетривиальный порядок надёжности.

В первой конструкции обращение с секретом быстрее, чем обращение без секрета, но, к сожалению, прямое вычисление функции ещё сложнее. В терминах определения 1 мы представляем кандидата в функции с секретом с одинаковыми длинами начальных чисел генератора, публичной информации, секрета, входа и выхода:

$$c(n) = m(n) = \mathrm{pi}(n) = \mathrm{ti}(n) = n.$$

Сэмплер производит пару из публичной информации и секрета (pi, ti), где ti — это само начальное число генератора, а $pi = A(ti)$ (таким образом, на генератор нужно $\frac{3n}{2} + O(1)$ гейтов). В этой конструкции вычисление функции производит код c для сообщения m следующим образом:

$$\mathrm{Eval}(pi, m) = A^{-1}(pi) \oplus A(m).$$

Вот матрица вычисления функции:

$$\left(\begin{array}{ccc|ccc} 0\,0\ldots0 & 1\,1\ldots1 & 1\,1 & 0\ldots0 & 0\,0\,0\ldots0 & 0\,0 \\ 1\,0\ldots0 & 1\,1\ldots1 & 1\,1 & 0\ldots0 & 0\,1\,1\ldots0 & 0\,0 \\ 1\,1\ldots0 & 1\,1\ldots1 & 1\,1 & 0\ldots0 & 0\,0\,1\ldots0 & 0\,0 \\ \vdots\;\vdots & \vdots\;\vdots\;\vdots & \vdots\;\vdots & \vdots\;\vdots\;\vdots & \vdots\;\vdots\;\vdots & \vdots\;\vdots\;\vdots \\ 1\,1\ldots1 & 1\,1\ldots1 & 1\,1 & 0\ldots0 & 0\,0\,0\ldots1 & 1\,0 \\ 1\,1\ldots1 & 0\,1\ldots1 & 1\,1 & 0\ldots0 & 0\,0\,0\ldots0 & 1\,1 \\ 1\,1\ldots1 & 0\,0\ldots1 & 1\,1 & 0\ldots0 & 0\,0\,0\ldots0 & 0\,1 \\ \vdots\;\vdots & \vdots\;\vdots\;\vdots & \vdots\;\vdots & \vdots\;\vdots\;\vdots & \vdots\;\vdots\;\vdots & \vdots\;\vdots\;\vdots \\ 1\,1\ldots1 & 0\,0\ldots1 & 1\,1 & 0\ldots0 & 0\,0\,0\ldots0 & 1\,1 \\ 1\,1\ldots1 & 0\,0\ldots1 & 1\,1 & 0\ldots0 & 0\,0\,0\ldots0 & 1\,1 \\ 1\,1\ldots1 & 0\,0\ldots0 & 1\,1 & 0\ldots0 & 0\,0\,0\ldots0 & 0\,1 \end{array}\right.\left.\begin{array}{c} 1\,1\,0\ldots0\,0\,0\ldots0\,0\,0 \\ 0\,1\,1\ldots0\,0\,0\ldots0\,0\,0 \\ 0\,0\,1\ldots0\,0\,0\ldots0\,0\,0 \\ \vdots\;\vdots\;\vdots\;\vdots \\ 0\,0\,0\ldots1\,1\,0\ldots0\,0\,0 \\ 0\,0\,0\ldots0\,1\,1\ldots0\,0\,0 \\ 0\,0\,0\ldots0\,0\,1\ldots0\,0\,0 \\ \vdots\;\vdots\;\vdots\;\vdots \\ 0\,0\,0\ldots0\,0\,0\ldots1\,1\,0 \\ 0\,0\,0\ldots0\,0\,0\ldots0\,1\,1 \\ 1\,0\,0\ldots1\,0\,0\ldots0\,0\,1 \end{array}\right) .$$

Верхнюю оценку на схемную сложность мы уже доказали в лемме 7; можно вычислить эту функцию схемой размера $\frac{7n}{2}$.

Обращение с секретом происходит следующим образом:

$$\mathrm{Inv}(ti, c) = A^{-1}(A^{-1}(pi) \oplus c) = A^{-1}(ti \oplus c).$$

Благодаря линейности (отметим, что оценки из предыдущих параграфов здесь неприменимы, потому что в матрице обращения полно одинаковых столбцов) эту схему можно реализовать за $\frac{5n}{2} - 1$ гейтов: сначала за n гейтов можно вычислить $ti \oplus c$, а затем применить A ещё за $\frac{3n}{2} - 1$ гейтов (см. лемму 7).

Наконец, противнику придётся обращать функцию более сложным способом, он ведь не знает ti:

$$m = A^{-1}(A^{-1}(pi) \oplus c) = \mathfrak{A}\begin{pmatrix} pi \\ c \end{pmatrix}.$$

По лемме 6, схемная сложность этой функции не меньше $\frac{13n}{4} - 5$ гейтов, и каждый противник с менее чем $\frac{13n}{4} - 5$ гейтами в своём распоряжении не сможет её вычислить более чем на 50% входов.

В этой конструкции вычисление функции оказывается сложнее, чем обращение без секрета. Чтобы это исправить, рассмотрим другую конструкцию, также семейство кандидатов в функции с секретом. Но теперь $c(n) = m(n) = n$ и $\mathrm{pi}(n) = \mathrm{ti}(n) = 0$, то есть ровным счётом никакой информации, ни секретной, ни публичной, здесь нет. Эта конструкция — просто односторонняя функция Хильтгена. Формально функции определяются следующим образом:

$$\begin{aligned} \mathrm{Eval}_n(m) &= A(m), \\ \mathrm{Inv}_n(c) &= A^{-1}(c), \\ \mathrm{Adv}_n(c) &= A^{-1}(c). \end{aligned}$$

Конечно, это вовсе не функции с секретом, ведь обращение реализуется вообще безо всякого секрета. Для сообщения m длины $|m| = n$ вычисляющая функцию схема состоит из $n + 1$ гейтов, а обращение, по лемме 7, требует схем размером не меньше $\frac{3n}{2} - 1$ гейтов каждая. Поэтому во второй конструкции вычислять функцию легко, а обращать трудно, как для противника, так и для честного участника протокола.

7 Семейство функций с секретом, надёжных в слабом смысле

В предыдущем разделе мы построили два семейства кандидатов в функции с секретом. В одном из них вычислять функцию было легко, а обращать сложно, а в другом обращение с секретом было простым, а вычисление и обращение без секрета — сложными.

Теперь объединим эти две функции. В результате и обращение с секретом, и вычисление функции окажутся проще, чем обращение без секрета.

Разделим вход на две части: первую часть m_1 длины n подвергнем нашей первой (менее тривиальной) конструкции, а ко второй части m_2 длины αn применим вторую конструкцию. Мы выберем α позже, так, чтобы максимизировать относительную сложность для противника.

Теперь каждый участник описывается блочно-диагональной матрицей:

$$\mathrm{Eval}_n(pi, m) = \begin{pmatrix} A^{-1} & A & 0 \\ 0 & 0 & A_* \end{pmatrix} \begin{pmatrix} pi \\ m_1 \\ m_2 \end{pmatrix} = \begin{pmatrix} c_1 \\ c_2 \end{pmatrix};$$

$$\mathrm{Inv}_n(ti, c) = \begin{pmatrix} A^{-1} & A^{-1} & 0 \\ 0 & 0 & A_*^{-1} \end{pmatrix} \begin{pmatrix} ti \\ c_1 \\ c_2 \end{pmatrix} = \begin{pmatrix} m_1 \\ m_2 \end{pmatrix},$$

$$\mathrm{Adv}_n(pi, m) = \begin{pmatrix} A^{-2} & A^{-1} & 0 \\ 0 & 0 & A_*^{-1} \end{pmatrix} \begin{pmatrix} pi \\ c_1 \\ c_2 \end{pmatrix} = \begin{pmatrix} m_1 \\ m_2 \end{pmatrix},$$

где A' обозначает матрицу с той же структурой, что и A, но для размерности αn вместо n. Тогда в терминах определения 1 мы получаем семейство кандидатов в функции с секретом, где входы и выходы функции длиннее начального числа генератора, а также публичной информации и секрета:

$$\begin{aligned} \mathrm{pi}(n) &= \mathrm{ti}(n) = n, \\ c(n) &= m(n) = (1+\alpha)n. \end{aligned}$$

Лемма 7 даёт верхние оценки сложности вычисления функции и обращения с секретом:

$$\begin{aligned} C(\mathrm{Eval}_n) &\le \frac{7n}{2} + \alpha n + 1, \\ C(\mathrm{Inv}_n) &\le \frac{5n}{2} + \frac{3\alpha n}{2} - 2. \end{aligned}$$

А лемма 6 даёт нижнюю оценку сложности обращения функции без секрета:

$$C_{1/2}(\mathrm{Adv}_n) \ge \frac{13n}{4} + \frac{3\alpha n}{2} - 7.$$

Таким образом, чтобы получить семейство надёжных в слабом смысле функций с секретом, нам достаточно выбрать α так, чтобы

$$\begin{aligned} \frac{13}{4} + \frac{3\alpha}{2} &> \frac{7}{2} + \alpha, \\ \frac{13}{4} + \frac{3\alpha}{2} &> \frac{5}{2} + \frac{3\alpha}{2}. \end{aligned}$$

Второе неравенство тривиально, а из первого получается, что $\alpha > \frac{1}{2}$.

Мы бы хотели максимизировать порядок надёжности в слабом смысле (см. определение 3); поскольку генерация секрета в этой конструкции всегда строго быстрее вычисления и обращения с секретом, мы на самом деле максимизируем

$$\min\left\{\lim_{n\to\infty}\frac{C_{1/2}(\mathrm{Adv}_n)}{C(\mathrm{Inv}_n)},\lim_{n\to\infty}\frac{C_{1/2}(\mathrm{Adv}_n)}{C(\mathrm{Eval}_n)}\right\}=\min\left\{\frac{\frac{13}{4}+\frac{3\alpha}{2}}{\frac{5}{2}+\frac{3\alpha}{2}},\frac{\frac{13}{4}+\frac{3\alpha}{2}}{\frac{7}{2}+\alpha}\right\}.$$

Это выражение достигает максимума при $\alpha=2$, и порядок надёжности при этом достигает $\frac{25}{22}$. В итоге мы доказали следующую теорему.

Теорема 1 *Существует семейство надёжных в слабом смысле функций с секретом с длиной начального числа генератора* $\mathrm{pi}(n)=\mathrm{ti}(n)=n$*, длинами входа и выхода функций* $c(n)=m(n)=3n$ *и порядком надёжности в слабом смысле* $\frac{25}{22}$*.*

8 Функция с секретом с экспоненциальной гарантией надёжности

Мы построили конструкцию семейства линейных надёжных в слабом смысле функций с секретом, которые гарантировали, что любая схема с числом гейтов менее требуемого не сможет обратить эти функции на более чем 50% их входов.

Сейчас мы используем эту конструкцию, чтобы создать систему с суперполиномиальными гарантиями надёжности (а именно $2^{-c\sqrt{n}+o(\sqrt{n})}$). Обозначим через h функцию, которую противник должен был вычислять в предыдущем разделе, а её матрицу — через X.

Рассмотрим линейную функцию H, определённую следующей блочно-диагональной матрицей:

$$H\left(\boldsymbol{x}^{(1)},\boldsymbol{x}^{(2)},\ldots,\boldsymbol{x}^{(m)}\right)=\begin{pmatrix}X & 0 & \ldots & 0\\ 0 & X & \ldots & 0\\ \vdots & \vdots & & \vdots\\ 0 & 0 & \ldots & X\end{pmatrix}\begin{pmatrix}\boldsymbol{x}^{(1)}\\ \boldsymbol{x}^{(2)}\\ \vdots\\ \boldsymbol{x}^{(m)}\end{pmatrix}.$$

По лемме 5, схемная сложность H составляет по меньшей мере $m(C(h)-1)$. Размерности матрицы X составляют $(1+\alpha)n\times(2+\alpha)n$; введём для удобства обозначение $n'=(1+\alpha)n$.

Лемма 8 *Зафиксируем многочлен* p*. Если схема вычисляет функцию* H *на более чем* $\frac{1}{p(m)}$ *доле её входов, и для каждого блока* H*:*

- *все столбцы* X *различны;*
- *каждая строка* X *содержит не менее* u *ненулевых элементов;*
- *после удаления из* X *любых* t *столбцов оставшаяся матрица всё ещё содержит по крайней мере одну строку с не менее чем двумя ненулевыми элементами;*

то сложность этой схемы составляет не менее $(u+t)(m-\log p(m))$*.*

Доказательство. Прежде всего вспомним, что H состоит из m отдельных блоков с непересекающимися множествами переменных X_i; обозначим $h_i=H\mid_{X_i}$. Поскольку X_i не пересекаются, ошибки в вычислении функций h_i независимы: если схема C вычисляет функцию h_i на доле β_i её входов, а функцию h_j — на доле β_j её входов, она не может вычислить H более чем на доле $\beta_i\beta_j$ её входов. Следовательно, в матрице H есть не более $\log p(m)$ блоков, где схема C может себе позволить ошибиться на целой половине входов. На протяжении этого доказательства мы будем называть их «ужасными» блоками.

После этого замечания мы снова начинаем то же самое доказательство методом исключения гейтов, к которому мы уже несколько раз прибегали. Рассмотрим верхний гейт в некотором топологическом порядке и две переменных, которые в него входят. В доказательстве леммы 4 мы называли переменные «хорошими» и «плохими» в зависимости от того, входят ли они в блок, в котором все «хорошие» переменные уже закончились. На этот раз мы делаем то же самое, но заранее называем все переменные в «ужасных» блоках «плохими».

Как и в предыдущих доказательствах, если в верхний гейт входит хотя бы одна «плохая» переменная, мы присвоим значение именно ей и удалим на этом шаге из схемы всего один гейт. А когда в верхний гейт входят две «хороших» переменных, мы всегда имеем возможность удалить как минимум два гейта из схемы. Давайте немного подробнее рассмотрим эту ситуацию.

Перед нами могут возникнуть всё те же три основных случая, которые мы рассматривали в доказательстве леммы 3. Однако на этот раз мы не можем подставить в точности то значение, которое хотим подставить. На этот раз нам придётся всё время оставаться в той половине входов, на которой схема ошибается на менее чем половине оставшихся входов (поскольку исходная схема ошибается не более чем на $\frac{1}{2}$ от всех входов, такая подсхема должна существовать). Однако мы можем объединить третий и первый случаи без потери общности: формула может зависеть исключительно от $x \wedge y$ не в большей степени, чем от $x \oplus y$; оба случая одинаковыми рассуждениями приводят к вероятности ошибки не менее $\frac{1}{2}$.

Остаток доказательства полностью повторяет лемму 3. Мы ведём доказательство по индукции, исключая по два гейта, если в верхний гейт входят две «хороших» переменных, и по одному, если есть среди них «плохая». Таким образом, общая сложность не меньше, чем количество «хороших» переменных, умноженное на два, плюс количество оставшихся «плохих» переменных. «Ужасные» блоки приходится полностью выбрасывать: в конце концов, их сложность и на самом деле может быть равна нулю. Таким образом, мы получаем нижнюю оценку схемной сложности $(t+u)(m - \log p(m))$. □

Заметим также, что копирование исходных матриц в большую блочно-диагональную конструкцию не меняет параметры (в том числе порядок обратимости) семейства функций с секретом, надёжных в слабом смысле. Таким образом, мы получаем следующую теорему.

Теорема 2 *Существует семейство надёжных в слабом смысле функций с секретом*

$$\mathcal{C} = \{\mathrm{Key}_n, \mathrm{Eval}_n, \mathrm{Inv}_n\}$$

с длиной начального числа генератора $\mathrm{pi}(n) = \mathrm{ti}(n) = n$*, длинами входа и выхода функций* $c(n) = m(n) = 3n$*, схемными сложностями*

$$C(\mathrm{Inv}_n) \le \frac{11n}{2} + O(1), \quad C(\mathrm{Eval}_n) \le \frac{11n}{2} + O(1), \quad C(\mathrm{Key}_n) = n + 1$$

и порядком надёжности в слабом смысле $\frac{25}{22}$.

Более того, ни один противник, располагающий менее чем $\frac{25}{4}n - \frac{5}{2}\delta\sqrt{n}$ *гейтами, не способен обратить эти функции с секретом на более чем* $2^{-\delta\sqrt{n}+o(\sqrt{n})}$ *доле их входов для любой константы* $\delta > 0$.

9 Заключение

В настоящей работе мы представили новую конструкцию криптографического примитива, надежного в слабом смысле. Это первая известная конструкция доказуемо надежной функции с секретом, хотя «надежность» здесь следует понимать в очень ограниченном смысле определения 3. Вот список возможных направлений для дальнейшей работы.

1. Разработать более естественную конструкцию. И вторая из наших конструкций (без ключа), и блочно–диагональное их объединение антиинтуитивны.
2. Существенно улучшить порядок надежности. Представляется, что некоторое улучшение константы $\frac{25}{22}$ можно получить достаточно просто (например, вместо A' можно использовать другую матрицу Хильтгена, порядок надежности которой стремится к 2 вместо $\frac{3}{2}$). Однако очевидно, что метод исключения гейтов не позволит доказать более чем линейных оценок. Поэтому оптимизация константы (пока константа эта меньше 2) не представляется стоящим делом. Это общее и очень существенное препятствие в теории схемной сложности, и нам кажется, что должен произойти серьезный прорыв, прежде чем в использованной нами модели вычислений можно будет доказать более чем линейные нижние оценки.
3. Разработать конструкции других надежных в слабом смысле криптографических примитивов.

Список литературы

[Blu84] Norbert Blum. A boolean function requiring $3n$ network size. *Theoretical Computer Science*, 28:337–345, 1984.

[GHP08] Dima Grigoriev, Edward A. Hirsch, and Konstantin Pervyshev. A complete public-key cryptosystem. *Groups, Complexity, Cryptology*, 1(1), 2008.

[Hil92] Alain P. Hiltgen. Constructions of freebly-one-way families of permutations. In *Proc. of AsiaCrypt '92*, pages 422–434, 1992.

[HKN+05] D. Harnik, J. Kilian, M. Naor, O. Reingold, and A. Rosen. On robust combiners for oblivious transfers and other primitives. In *Proc. of EuroCrypt'05, LNCS*, volume 3494, pages 96–113, 2005.

[Lev87] Leonid A. Levin. One-way functions and pseudorandom generators. *Combinatorica*, 7(4):357–363, 1987.

[LS73] E. A. Lamagna and J. E. Savage. On the logical complexity of symmetric switching functions in monotone and complete bases. Technical report, Brown University, Rhode Island, jul 1973.

[Sav76] John E. Savage. *The Complexity of Computing.* Wiley, New York, 1976.

[Weg87] Ingo Wegener. *The Complexity of Boolean Functions.* B. G. Teubner, and John Wiley & Sons, 1987.