В. В. Баев, О некоторых алгоритмах построения аннигиляторов низкой степени для булевых функций, *Дискрет. матем.*, 2006, том 18, выпуск 3, 138–151

DOI: 10.4213/dm66

Дискретная математика

том 18 выпуск 3 * 2006

УДК 519.7

О некоторых алгоритмах построения аннигиляторов низкой степени для булевых функций

© 2006 г. В. В. Баев

Алгебраический метод широко используется при анализе фильтрующих генераторов псевдослучайных последовательностей. Он основан на получении булевых уравнений низкой степени относительно битов начального состояния генератора. Задача получения таких уравнений сводится к поиску обнуляющих множителей (аннигиляторов) низкой степени для фильтрующей булевой функции. Наличие ненулевых низкостепенных аннигиляторов снижает сложность определения начального состояния генератора по его выходной последовательности.

В работе исследуется задача нахождения всех низкостепенных аннигиляторов для булевой функции, заданной в виде многочлена от нескольких переменных. Предлагаются два новых алгоритма решения этой задачи. Их сложности оцениваются сверху полиномами от количества переменных функции и от количества мономов в многочлене, который задает эту функцию. Рассмотрено также применение этих алгоритмов для реализации алгебраического метода по трем известным сценариям, в соответствии с которыми получаются уравнения низкой степени.

1. Введение

Алгебраический метод определения ключа фильтрующего генератора по его выходной последовательности состоит в сведении исходной системы булевых уравнений шифрования к более простой (в некотором смысле) и последующем решении этой системы уравнений одним из возможных способов, таких как линеаризация, частичное опробование и другие. Применимость и эффективность этих способов в частности зависит от алгебраической степени булевых уравнений, входящих в новую систему. Чем меньше эта степень, тем проще найти решение — значение ключа.

В [1] предложено несколько сценариев алгебраического метода (S1, S2, S3a, S3b, S3c, S4). Группа сценариев S3 предполагает нахождение для булевой функции f определенной ненулевой функции-множителя g такой, чтобы произведение $f \cdot g$ было равно нулю (в этом случае g называется аннигилятором функции f) либо имело низкую степень. Однако при конкретных реализациях алгебраического метода определения ключа для таких потоковых шифров, как TOYOCRYPT и LILI-128, поиск этих множителей для группы сценариев S3 был эвристическим.

В [2] описан ряд базовых свойств аннигиляторов и предложены некоторые алгоритмы, определяющие, существует или нет аннигилятор фиксированной степени для заданной

функции. Однако эти алгоритмы либо становятся неэффективными при количестве переменных $n \geqslant 20$, либо носят вероятностный характер. С помощью алгоритма 1 из [2] можно достаточно быстро найти все аннигиляторы низкой степени для функции усложнения из LILI-128, поскольку она зависит лишь от 10 переменных.

В [3] приведены верхняя и нижняя оценки минимальной степени аннигилятора для заданной функции, а также предложен алгоритм, генерирующий для функции f порождающую систему функций для линейного пространства всех аннигиляторов степени, не превосходящей некоторого фиксированного числа d. Нижние оценки сложности предложенных в [2] и [3] алгоритмов содержат в качестве множителя мощность множества тех входных наборов функции f, на которых она принимает значение 1. Для уравновешенных функций от n переменных этот множитель будет равен 2^{n-1}. Это означает, что сложности этих алгоритмов растут экспоненциально от n.

В данной работе рассматриваются два алгоритма нахождения всех множителей g для сценариев S3a, S3b и S3c. Эти алгоритмы являются достаточно эффективными (верхние оценки сложности будут получены в параграфах 3.1 и 3.3 данной статьи) для функций усложнения, алгебраическая нормальная форма которых содержит небольшое количество ненулевых коэффициентов. В частности, такие функции используются в потоковых шифрах TOYOCRYPT и LILI-128.

Данная статья имеет следующую структуру: в параграфе 2 приводятся основные обозначения, определения и утверждения, связанные с булевыми функциями. В параграфе 3 приводятся два алгоритма нахождения линейного пространства всех аннигиляторов заданной степени и оценивается сложность этих алгоритмов. Также выводится система линейных уравнений, задающая все указанные аннигиляторы. В параграфе 4 показывается, как применять эти алгоритмы для получения всех линейно независимых уравнений ограниченной сверху степени для реализации алгебраических методов по сценариям S3a, S3b и S3c.

2. Основные определения и обозначения

Введем следующие обозначения:

$\mathbb{F}_2$ — поле из двух элементов.

$V_n = \mathbb{F}_2^n$ — n-мерное векторное пространство над полем $\mathbb{F}_2$.

$\mathcal{F}_n$ — множество всех функций $\mathbb{F}_2^n \to \mathbb{F}_2$. Далее мы будем рассматривать его как кольцо и как линейное пространство над полем $\mathbb{F}_2$.

$\mathrm{wt}(x)$ — количество ненулевых компонент вектора $x \in V_n$, называемое его весом.

$|\mathcal{M}|$ — количество элементов конечного множества $\mathcal{M}$.

Определение 1. Пусть $x, y \in V_n$, $x = (x_1, \ldots, x_n)$, $y = (y_1, \ldots, y_n)$, тогда считаем

$$x \vee y = (x_1 \vee y_1, \ldots, x_n \vee y_n).$$

Определение 2. Будем говорить, что два элемента $x, y \in V_n$, $x = (x_1, \ldots, x_n)$, $y = (y_1, \ldots, y_n)$, находятся в отношении частичного порядка $\leqslant$, и писать $x \leqslant y$, если $x_i \leqslant y_i$ для любого номера i от 1 до n.

Определение 3. Для любых двух элементов $x, y \in V_n$ введем отрезок между ними

$$[x; y] = \{z \in V_n \mid x \leqslant z \leqslant y\}.$$

Замечание 1. Если $x \not\leqslant y$, то $[x; y] = \varnothing$. Если же $x \leqslant y$, то $[x; y]$ представляет собой вершины куба размерности $\mathrm{wt}(y) - \mathrm{wt}(x)$ и содержит $2^{\mathrm{wt}(y)-\mathrm{wt}(x)}$ элементов.

Операции сложения и вычитания во всех группах (в поле $\mathbb{F}_2$, в векторных пространствах V_n и $\mathcal{F}_n$, в кольце целых чисел) будем обозначать $+$ и $-$, соответственно, а сумму — знаком $\sum$. Хотя для векторных пространств над полем $\mathbb{F}_2$ сложение и вычитание представляют собой одну и ту же операцию, в случае $x \leqslant y$ для двух векторов $x, y \in V_n$ нагляднее писать $y - x$, вместо $y + x$. Недоразумений, связанных с одинаковым обозначением различных операций, не возникнет, так как мы будем складывать и вычитать только элементы из одной группы.

Определение 4. Линейное отображение $\mu: \mathcal{F}_n \to \mathcal{F}_n$, задаваемое формулой

$$\mu(f)(u) = \sum_{x \in V_n : x \leqslant u} f(x),$$

называется преобразованием Мебиуса.

Для удобства образ преобразования Мебиуса функции f будем еще обозначать

$$\mu f = \mu(f).$$

Определение 5. Функцию $g \in \mathcal{F}_n$ будем называть аннигилятором функции $f \in \mathcal{F}_n$, если $f \cdot g = 0$.

Предложение 1 ([4]). *Для каждой функции $f \in \mathcal{F}_n$ верно, что $\mu(\mu(f)) = f$.*

Пусть $x, \alpha \in V_n$, $x = (x_1, \ldots, x_n)$, $\alpha = (\alpha_1, \ldots, \alpha_n)$. Введем обозначение

$$x^\alpha = \prod_{i=1}^{n} x_i^{\alpha_i},$$

где

$$x_i^{\alpha_i} = \begin{cases} x_i, & \alpha_i = 1, \\ 1, & \alpha_i = 0. \end{cases}$$

Индексом монома x^α относительно переменных $x_1, \ldots, x_n$ будем называть вектор $\alpha \in V_n$.

Предложение 2 ([4]). *Значения функции μf являются коэффициентами алгебраической нормальной формы (АНФ) функции f:*

$$f(x_1, \ldots, x_n) = \sum_{\alpha \in V_n} \mu f(\alpha) \cdot x^\alpha = \sum_{(\alpha_1, \ldots, \alpha_n) \in V_n} \mu f(\alpha_1, \ldots, \alpha_n) \cdot x_1^{\alpha_1} \cdots x_n^{\alpha_n}.$$

3. Задача нахождения всех аннигиляторов низкой степени

В этой части будут подробно описаны два метода получения линейного пространства всех аннигиляторов степени, не превосходящей d, для функции $f \in \mathcal{F}_n$. Будем предполагать, что эта функция задана списком индексов всех мономов, входящих с ненулевым коэффициентом в ее АНФ. Это представление в силу предложения 2 эквивалентно перечислению всех элементов множества

$$\mathcal{M}_f = \{u \in V_n \mid \mu f(u) = 1\}.$$

Форма представления функции f будет нам важна для оценки сложности предлагаемых алгоритмов. Используя множество $\mathcal{M}_f$, можно выразить f следующим образом:

$$f(x) = \sum_{u \in \mathcal{M}_f} x^u. \tag{1}$$

3.1. Прямой метод

Мы хотим найти все такие функции $g \in \mathcal{F}_n \setminus \{0\}$, степени не более некоторого фиксированного натурального числа d, что $f \cdot g = 0$. Вместе с нулевой функцией они образуют линейное подпространство в $\mathcal{F}_n$:

$$\mathrm{Ann}_d(f) = \{g \in \mathcal{F}_n \mid \deg g \leqslant d,\ f \cdot g = 0\}.$$

Каждая функция $g \in \mathrm{Ann}_d(f)$ имеет вид

$$g(x) = \sum_{v \in V_n : \mathrm{wt}(v) \leqslant d} z_v \cdot x^v, \tag{2}$$

где $z_v \in \mathbb{F}_2$ — неизвестные коэффициенты, относительно которых мы будем составлять систему линейных уравнений.

Условие принадлежности g подпространству $\mathrm{Ann}_d(f)$ равносильно тому, что для любого $x \in V_n$

$$0 = f(x) \cdot g(x) = \sum_{u \in \mathcal{M}_f} x^u \cdot \sum_{v \in V_n : \mathrm{wt}(v) \leqslant d} z_v \cdot x^v = \sum_{u \in \mathcal{M}_f} \sum_{v \in V_n : \mathrm{wt}(v) \leqslant d} z_v \cdot x^{u \vee v}.$$

Далее, группируя слагаемые по общему множителю-моному x^w, мы получим равенство вида

$$\sum_{w \in \mathcal{M}} \left(\sum_{v \in \mathcal{N}_w} z_v \right) \cdot x^w = 0. \tag{3}$$

Точные описания множества $\mathcal{M}$ и семейства множеств $\mathcal{N}_w$, $w \in \mathcal{M}$, будут приведены позднее в параграфе 3.2.

Поскольку равенство (3) выполняется для каждого $x \in V_n$, коэффициент при каждом мономе x^w равен нулю:

$$\sum_{v \in \mathcal{N}_w} z_v = 0, \qquad \forall w \in \mathcal{M}. \tag{4}$$

Таким образом, функция g вида (2) является аннигилятором функции $f \in \mathcal{F}_n$ тогда и только тогда, когда коэффициенты z_v, $v \in V_n$: $\mathrm{wt}(v) \leqslant d$, удовлетворяют системе линейных однородных уравнений (4). Эту систему уравнений можно решить методом Гаусса, выразив часть переменных через оставшиеся (свободные). Свободные переменные могут принимать независимые друг от друга значения, задавая все линейное пространство решений $\mathrm{Ann}_d(f)$.

Выделим основные шаги приведенного выше алгоритма и оценим сложность каждого из них. При этом мы будем использовать асимптотические оценки вида $O(\varphi(n, d, |\mathcal{M}_f|))$ при $d \to \infty$, $n \geqslant d$, $|\mathcal{M}_f| \to \infty$.

Считаем, что функция f задана множеством $\mathcal{M}_f$ единичных коэффициентов своей алгебраической нормальной формы.

Алгоритм 1. Вход: $n \in \mathbb{N}$, $d \in \{1, \ldots, n\}$, список элементов множества $\mathcal{M}_f \subset V_n$.

Выход: Линейное пространство $\mathrm{Ann}_d(f)$, заданное в виде параметрического семейства многочленов от n булевых переменных степени, не превосходящей d.

1. Представляем функции f и g в виде сумм (1) и (2) соответственно.
2. Раскрываем скобки в произведении $f \cdot g$ и, группируя слагаемые $z_v \cdot x^w$ путем сортировки по w, получаем уравнение (3).
3. Составляем систему линейных однородных уравнений (4).
4. Находим общее решение системы (4) в параметрическом виде и подаем его на выход алгоритма.

Остаток данной части будет посвящен описанию конкретной реализации шагов алгоритма и оценке их сложности. При желании читатель может пропустить этот рутинный участок, приняв лишь к сведению, что битовая сложность алгоритма 1 оценивается как $O(|\mathcal{M}_f|(S_n^d)^3)$, где

$$S_n^d = \sum_{i=0}^{d} \binom{n}{i},$$

а $\binom{n}{i}$ — биномиальные коэффициенты.

Шаг 1. Представление (1) задается списком элементов множества $\mathcal{M}_f$, являющегося частью входных данных алгоритма. Представление (2) эквивалентно заданию списка из S_n^d переменных z_v, $v \in V_n$: $\mathrm{wt}(v) \leqslant d$. Генерация такого списка потребует $O(nS_n^d)$ битовых операций.

Шаг 2. Для раскрытия скобок в произведении двух сумм, содержащих $|\mathcal{M}_f|$ и S_n^d слагаемых соответственно, где длина каждого слагаемого есть $O(n)$, нужно $O(n|\mathcal{M}_f|S_n^d)$ битовых операций. Сортировать слагаемые $z_v \cdot x^w$ по ключу w можно методом сортировки слиянием ([5]). Количество этих пар равно $|\mathcal{M}_f|S_n^d$, поэтому при их сортировке будет выполнено $O(|\mathcal{M}_f|S_n^d \log(|\mathcal{M}_f|S_n^d))$ операций сравнения, присваивания, сложения и вычитания векторов из V_n и целых чисел. Модуль используемых целых чисел не превосходит $|\mathcal{M}_f|S_n^d \leqslant |V_n||V_n| = 2^{2n}$. Длина двоичного представления таких чисел меньше либо равна $1 + \log_2(2^{2n}) = 2n + 1$. Таким образом, каждая из указанных операций использует $O(n)$ битовых операций. Битовая сложность сортировки будет $O(n|\mathcal{M}_f|S_n^d \log(|\mathcal{M}_f|S_n^d))$.

Шаг 3. Будем генерировать матрицу линейной системы (4) построчно, проходя по отсортированному списку слагаемых суммы (3), полученному на предыдущем шаге. Все слагаемые с мономом x^w идут подряд, поэтому изменение индекса монома w при переходе к следующему слагаемому означает смену текущей строки матрицы. При этом новая строка длины S_n^d инициализируется нулями, на что потребуется $O(S_n^d)$ операций. При обработке очередного слагаемого $z_v \cdot x^w$ нужно найти номер переменной z_v, пройдя по списку этих переменных, сгенерированному на шаге 1, и прибавить единицу к элементу с этим номером в текущей строке. Это потребует $O(nS_n^d)$ операций, что больше сложности затрат на начальное заполнение строки нулями. Всего слагаемых — $|\mathcal{M}_f|S_n^d$, то есть сложность всего шага есть $|\mathcal{M}_f|S_n^d O(nS_n^d) = O(n|\mathcal{M}_f|(S_n^d)^2)$.

Шаг 4. Решение системы линейных однородных уравнений над полем $\mathbb{F}_2$ с матрицей размера $m_1 \times m_2$ методом Гаусса потребует $O(m_1 m_2 \min(m_1, m_2))$ битовых операций. В нашем случае $m_1 \leqslant |\mathcal{M}_f|S_n^d$, $m_2 = S_n^d$. Таким образом, битовая сложность решения системы (4) методом Гаусса будет $O(|\mathcal{M}_f|(S_n^d)^3)$.

Складывая сложности всех четырех шагов и принимая во внимание неравенства

$$\log_2(|\mathcal{M}_f|S_n^d) \leqslant \log_2(|V_n||V_n|) = 2n, \qquad n \leqslant S_n^d,$$

получаем, что битовую сложность алгоритма 1 можно оценить как $O(|\mathcal{M}_f|(S_n^d)^3)$.

3.2. Другой вывод системы уравнений, задающей аннигиляторы низкой степени

В этой части мы рассмотрим другой метод получения линейного пространства $\mathrm{Ann}_d(f)$. Он будет основан на выводе явного вида линейной системы однородных уравнений относительно коэффициентов АНФ аннигилирующей функции. Конечно же, эта система будет равносильна полученной ранее системе (4). Основным итогом этой части станет теорема 1′.

Рассмотрим преобразование Мебиуса от произведения $f \cdot g$

$$\begin{aligned}
\mu(f\cdot g)(\alpha) &= \sum_{x\in V_n: x\leqslant\alpha} f(x)\cdot g(x) = \sum_{x\in V_n: x\leqslant\alpha} \mu(\mu f)(x)\cdot\mu(\mu g)(x)\\
&= \sum_{x\in V_n: x\leqslant\alpha}\left(\sum_{y\in V_n: y\leqslant x}\mu f(y)\right)\cdot\left(\sum_{z\in V_n: z\leqslant x}\mu g(z)\right)\\
&= \sum_{x\in V_n: x\leqslant\alpha}\ \sum_{y,z\in V_n: y\leqslant x,\ z\leqslant x}\mu f(y)\cdot\mu g(z)\\
&= \sum_{x,y,z\in V_n: x\leqslant\alpha,\ y\leqslant x, z\leqslant x}\mu f(y)\cdot\mu g(z) = \sum_{y,z\in V_n}\ \sum_{x\in V_n: x\leqslant\alpha,\ y\leqslant x, z\leqslant x}\mu f(y)\cdot\mu g(z)\\
&= \sum_{y,z\in V_n: y\leqslant\alpha, z\leqslant\alpha}\ \sum_{x\in V_n: x\leqslant\alpha,\ y\leqslant x, z\leqslant x}\mu f(y)\cdot\mu g(z)\\
&= \sum_{y,z\in V_n: y\leqslant\alpha,\ z\leqslant\alpha}\ \sum_{x\in V_n: y\vee z\leqslant x\leqslant\alpha}\mu f(y)\cdot\mu g(z)\\
&= \sum_{y,z\in V_n: y\leqslant\alpha,\ z\leqslant\alpha}\ \sum_{x\in[y\vee z;\alpha]}\mu f(y)\cdot\mu g(z).
\end{aligned}$$

Из замечания 1 следует, что при $y \vee z \neq \alpha$ внутренняя сумма по x содержит четное число слагаемых, то есть равна нулю. Используя это, продолжаем равенство:

$$\sum_{y,z\in V_n: y\leqslant\alpha,\ z\leqslant\alpha} \sum_{x\in[y\vee z;\alpha]} \mu f(y)\cdot\mu g(z) = \sum_{y,z\in V_n: y\leqslant\alpha,\ z\leqslant\alpha,\ y\vee z=\alpha} \mu f(y)\cdot\mu g(z). \qquad (5)$$

Чтобы разделить сумму по y и z на повторную вида $\sum_z \sum_y$, нужно найти множество значений, которое может принимать y при фиксированных α и z, $z \leqslant \alpha$, чтобы выполнялись условия

$$y \leqslant \alpha, \qquad y \vee z = \alpha. \qquad (6)$$

Строгим рассуждениям предпошлем наглядную схему

$$\alpha = (\overbrace{1,\dots,1}^{\mathrm{wt}(\alpha)},1,\dots,1,0,\dots,0)$$
$$z = (\overbrace{1,\dots,1}^{\mathrm{wt}(z)},0,\dots,0,0,\dots,0)$$
$$y = (*,\dots,*,1,\dots,1,0,\dots,0).$$

Условия (6) равносильны тому, что каждая i-я компонента вектора y удовлетворяет условиям $y_i \leqslant \alpha_i$ и $y_i \vee z_i = \alpha_i$. В силу того, что $z \leqslant \alpha$, пара (z_i, α_i) может принимать три значения:

(1) $z_i = 0$, $\alpha_i = 0$: $y_i \leqslant 0$ и $y_i \vee 0 = 0$ равносильно тому, что $y_i = 0$.

(2) $z_i = 0$, $\alpha_i = 1$: $y_i \leqslant 1$ и $y_i \vee 0 = 1$ равносильно тому, что $y_i = 1$.

(3) $z_i = 1$, $\alpha_i = 1$: $y_i \leqslant 1$ и $y_i \vee 1 = 1$ равносильно тому, что y_i может принимать любое из двух значений.

Эти три случая можно объединить в одно условие: $\alpha_i - z_i \leqslant y_i \leqslant \alpha_i$. Поскольку эти неравенства должны выполняться для всех i от 1 до n, то искомое множество значений вектора y задается так: $\alpha - z \leqslant y \leqslant \alpha$. На основании проведенных рассуждений мы теперь можем преобразовать сумму по y и z из правой части равенства (5):

$$\sum_{y,z\in V_n: y\leqslant\alpha,\ z\leqslant\alpha,\ y\vee z=\alpha} \mu f(y)\cdot\mu g(z) = \sum_{z\in V_n: z\leqslant\alpha} \sum_{y\in[\alpha-z;\alpha]} \mu f(y)\cdot\mu g(z).$$

Таким образом,

$$\mu(f\cdot g)(\alpha) = \sum_{z\in V_n: z\leqslant\alpha} \left(\sum_{y\in[\alpha-z;\alpha]} \mu f(y) \right) \cdot \mu g(z). \qquad (7)$$

Замечание 2. Сумма $\sum_{y\in[\alpha-z;\alpha]} \mu f(y) = 1$ тогда и только тогда, когда $|[\alpha - z;\alpha] \cap \mathcal{M}_f|$ нечетно.

Вспомним, что мы хотим решить относительно функции g следующее уравнение с условием:

$$f \cdot g = 0, \qquad \deg g \leqslant d.$$

Замечание 3. В силу предложения 1 линейное отображение μ обратимо, поэтому $f \cdot g = 0$ тогда и только тогда, когда $\mu(f \cdot g)(\alpha) = 0$ для каждого $\alpha \in V_n$.

Замечание 4. Неравенство $\deg g \leqslant d$ равносильно условию

$$\forall z \in V_n : \mathrm{wt}(z) > d \qquad \mu g(z) = 0,$$

которое позволяет в правой части равенства (7) брать сумму не по всем $z \leqslant \alpha$, а только по тем из них, вес которых не превосходит d.

Принимая во внимание замечания 2, 3 и 4, а также равенство (7), выпишем систему уравнений относительно переменных $\mu g(z)$, являющихся коэффициентами АНФ функции $g \in \mathrm{Ann}_d(f)$:

$$\sum_{z \in V_n : z \leqslant \alpha,\ \mathrm{wt}(z) \leqslant d,\ |[\alpha - z; \alpha] \cap \mathcal{M}_f| \equiv 1 \pmod 2} \mu g(z) = 0, \qquad \forall \alpha \in V_n. \tag{8}$$

Итак, мы доказали следующее утверждение.

Теорема 1. *Пусть* $d \in \{1, \ldots, n\}$, $f, g \in \mathcal{F}_n$, $\deg g \leqslant d$, $\mathcal{M}_f = \{\alpha \in V_n | \mu f(\alpha) = 1\}$. *Тогда* $f \cdot g = 0$ *если и только если выполняется система* (8).

Могут найтись такие векторы $\alpha \in V_n$, для которых соответствующее уравнение в системе (8) имеет вид $0 = 0$ из-за нулевого количества слагаемых в левой части. Сейчас мы опишем дополнение к этому множеству векторов в пространстве V_n. А именно, рассмотрим множество

$$C_d(\mathcal{M}_f) = \{\alpha \in V_n \mid \exists y \in \mathcal{M}_f : y \leqslant \alpha,\ \mathrm{wt}(\alpha - y) \leqslant d\}$$

и докажем, что оно является множеством всех $\alpha \in V_n$, для которых соответствующее уравнение в системе (8) не вырождается в $0 = 0$, а имеет хотя бы одно неизвестное слагаемое $\mu g(z)$ в левой части.

Пусть α принадлежит $C_d(\mathcal{M}_f)$, тогда множество

$$D_d(\alpha, \mathcal{M}_f) = \{y \in \mathcal{M}_f \mid y \leqslant \alpha,\ \mathrm{wt}(\alpha - y) \leqslant d\}$$

не пусто и, естественно, конечно. Выберем в нем элемент y^* с максимальным весом, то есть такой, что

$$\mathrm{wt}(y^*) = \max_{y \in D_d(\alpha, \mathcal{M}_f)} \mathrm{wt}(y).$$

Докажем, что

$$[y^*; \alpha] \cap \mathcal{M}_f = \{y^*\}. \tag{9}$$

Действительно, y^* принадлежит $\mathcal{M}_f$, поскольку $y^* \in D_d(\alpha, \mathcal{M}_f) \subset \mathcal{M}_f$. Каждый вектор $y \in [y^*; \alpha] \setminus \{y^*\}$ удовлетворяет условиям $y \leqslant \alpha$ и $\mathrm{wt}(\alpha - y) \leqslant d$, а также $\mathrm{wt}(y) > \mathrm{wt}(y^*)$. Если бы такой y принадлежал $\mathcal{M}_f$, то он бы принадлежал и $D_d(\alpha, \mathcal{M}_f)$, что противоречит выбору элемента y^*. Равенство (9) доказано.

Положим $z^* = \alpha - y^*$. Поскольку $y^* \in D_d(\alpha, \mathcal{M}_f)$, то $y^* \leqslant \alpha$ и $\mathrm{wt}(\alpha - y^*) \leqslant d$, а следовательно

$$z^* \leqslant \alpha, \qquad \mathrm{wt}(z^*) \leqslant d. \tag{10}$$

Из (9), (10) и определения z^* следует, что для выбранного $\alpha \in C_d(\mathcal{M}_f)$ в соответствующем ему уравнении системы (8) будет присутствовать, по крайней мере, одно неизвестное слагаемое $\mu g(z^*)$.

Обратно, пусть уравнение системы (8) для некоторого $\alpha \in V_n$ не вырождается в $0 = 0$. Это значит, что хотя бы для одного $z \in V_n : z \leqslant \alpha, \mathrm{wt}(z) \leqslant d$ куб $[\alpha - z; \alpha]$ имеет непустое пересечение с $\mathcal{M}_f$, что равносильно условию

$$\mathcal{M}_f \cap \bigcup_{z \in V_n : z \leqslant \alpha,\ \mathrm{wt}(z) \leqslant d} [\alpha - z; \alpha] \neq \varnothing. \tag{11}$$

Преобразуем объединение по z из этого неравенства:

$$\bigcup_{z \in V_n : z \leqslant \alpha,\ \mathrm{wt}(z) \leqslant d} [\alpha - z; \alpha] = \bigcup_{z \in V_n : z \leqslant \alpha,\ \mathrm{wt}(z) \leqslant d} \{\alpha - z\} = \bigcup_{y \in V_n : y \leqslant \alpha,\ \mathrm{wt}(\alpha - y) \leqslant d} \{y\}$$
$$= \{y \in V_n \mid y \leqslant \alpha,\ \mathrm{wt}(\alpha - y) \leqslant d\}.$$

Таким образом, (11) равносильно условию

$$\mathcal{M}_f \cap \{y \in V_n \mid y \leqslant \alpha, \mathrm{wt}(\alpha - y) \leqslant d\} \neq \varnothing,$$

то есть α принадлежит $C_d(\mathcal{M}_f)$ по определению этого множества.

Итак, нами доказано, что система (8) равносильна следующей системе, каждое уравнение которой содержит хотя бы одну переменную $\mu g(z)$:

$$\sum_{z \in V_n : z \leqslant \alpha,\ \mathrm{wt}(z) \leqslant d,\ |[\alpha - z; \alpha] \cap \mathcal{M}_f| \equiv 1 \pmod 2} \mu g(z) = 0 \qquad \forall \alpha \in C_d(\mathcal{M}_f). \tag{12}$$

Таким образом, теорему 1 можно переформулировать для системы (12) в следующем виде.

Теорема 1′. *Функция $g \in \mathcal{F}_n : \deg g \leqslant d$ является аннигилятором функции $f \in \mathcal{F}_n$, если и только если коэффициенты ее АНФ $\mu g(z)$, $z \in V_n : \mathrm{wt}(z) \leqslant d$, удовлетворяют системе* (12).

Данная теорема, в частности, показывает, что системы уравнений (4) и (12) равносильны. Более того, нетрудно видеть, что множество $\mathcal{M}$, появившееся в уравнении (3), равно $C_d(\mathcal{M}_f)$, а для каждого $w \in \mathcal{M}$

$$\mathcal{N}_w = \{z \in V_n : z \leqslant w,\ \mathrm{wt}(z) \leqslant d,\ |[w - z; w] \cap \mathcal{M}_f| \equiv 1 \pmod 2\}.$$

3.3. Второй алгоритм получения всех аннигиляторов низкой степени

На основании теоремы 1′ мы можем использовать следующий алгоритм получения всех аннигиляторов степени, не превосходящей d, для заданной функции $f \in \mathcal{F}_n$. Этот алгоритм реализует то же соответствие входных и выходных данных, что и алгоритм 1.

Алгоритм 2. Вход: $n \in \mathbb{N}$, $d \in \{1, \ldots, n\}$, список элементов множества $\mathcal{M}_f \subset V_n$.

Выход: Линейное пространство $\mathrm{Ann}_d(f)$, заданное в виде параметрического семейства многочленов от n булевых переменных степени, не превосходящей d.

1. Получаем список всех элементов множества $C_d(\mathcal{M}_f)$.

2. Составляем матрицу системы линейных однородных уравнений (12).

3. Находим общее решение системы (12) в параметрическом виде и подаем его на выход алгоритма.

Оценим сложность каждого из шагов алгоритма.

Шаг 1. Выразим множество $C_d(\mathcal{M}_f)$ по-другому:

$$C_d(\mathcal{M}_f) = \bigcup_{y \in \mathcal{M}_f} \{\alpha \in V_n \mid y \leqslant \alpha,\ \mathrm{wt}(\alpha - y) \leqslant d\}, \tag{13}$$

для того, чтобы дать одну из верхних оценок его мощности:

$$\begin{aligned} |C_d(\mathcal{M}_f)| &\leqslant \sum_{y \in \mathcal{M}_f} |\{\alpha \in V_n \mid y \leqslant \alpha,\ \mathrm{wt}(\alpha - y) \leqslant d\}| \\ &= \sum_{y \in \mathcal{M}_f} \sum_{i=0}^{d} \binom{n - \mathrm{wt}(y)}{i} \leqslant |\mathcal{M}_f| S_n^d. \end{aligned} \tag{14}$$

Имея список всех элементов множества $\mathcal{M}_f$, будем генерировать конечную последовательность элементов множеств из правой части представления (13). При этом некоторые элементы будут получаться повторно. Согласно (14), получение этой последовательности потребует $O(n|\mathcal{M}_f|S_n^d)$ битовых операций. Множитель n появляется из-за того, что каждый элемент последовательности представляет собой n-мерный вектор из V_n. Чтобы избавиться от повторений, сначала мы отсортируем полученный набор, а затем, проверяя на равенство лишь соседние элементы, выкинем повторяющиеся. Если использовать идеи из шага 2 алгоритма 1, то для этого понадобится $O(n|\mathcal{M}_f|S_n^d \log(|\mathcal{M}_f|S_n^d))$ битовых операций.

Таким образом, сложность шага 1 есть $O(n|\mathcal{M}_f|S_n^d \log(|\mathcal{M}_f|S_n^d))$.

Шаг 2. Система (12) имеет матрицу размера $|C_d(\mathcal{M}_f)| \times S_n^d$. Количество элементов в этой матрице, согласно (14), не превосходит $|\mathcal{M}_f|(S_n^d)^2$. Пусть ее строки параметризуются вектором $\alpha \in C_d(\mathcal{M}_f)$, а столбцы (переменные) — вектором $z \in V_n$: $\mathrm{wt}(z) \leqslant d$. Вычисление элемента матрицы в позиции (α, z) производится путем подсчета мощности множества $[\alpha - z; \alpha] \cap \mathcal{M}_f$. Для каждого $x \in \mathcal{M}_f$ проверяется условие $\alpha - z \leqslant x \leqslant \alpha$. Для этого понадобится $O(n|\mathcal{M}_f|)$ битовых операций. Таким образом, на составление матрицы системы потребуется $O(n|\mathcal{M}_f|^2(S_n^d)^2)$ битовых операций.

Шаг 3. Аналогично шагу 4 алгоритма 1, битовая сложность решения системы (12) методом Гаусса составит $O(|\mathcal{M}_f|(S_n^d)^3)$.

Сложность всего алгоритма будет $O(n|\mathcal{M}_f|^2(S_n^d)^2 + |\mathcal{M}_f|(S_n^d)^3)$. Как мы видим, асимптотические верхние оценки алгоритмов 1 и 2 отличаются слагаемым $n|\mathcal{M}_f|^2(S_n^d)^2$, содержащим множитель $|\mathcal{M}_f|^2$. Тем не менее, вид системы (12) будет полезен при дальнейших исследованиях, связанных с аннигиляторами низкой степени.

4. Сценарии алгебраических методов

Как мы уже говорили, суть алгебраического метода для потоковых шифров, основанных на регистре сдвига с линейной обратной связью, заключается в нахождении и решении уравнений небольшой степени, которым удовлетворяет ключ.

В [1] введены следующие три сценария алгебраического метода для функции f высокой степени. В соответствии с этими сценариями строятся уравнения низкой степени.

S3a: Найдется функция $g \in \mathcal{F}_n$ низкой степени такая, что произведение $h = f \cdot g$ не равно тождественно нулю и также имеет низкую степень.

S3b: Найдется функция $g \in \mathcal{F}_n$ низкой степени такая, что $f \cdot g = 0$.

S3c: Найдется функция $h \in \mathcal{F}_n$ низкой степени такая, что для некоторой функции $g \in \mathcal{F}_n$ будет выполнено $f \cdot g = h$.

В [2] были установлены некоторые связи между этими сценариями. Сейчас мы покажем, как применять любой из двух алгоритмов, приведенных в параграфе 3, для получения уравнений низкой степени по этим трем сценариям. Мы будем предполагать, что нам известна функция f, линейное преобразование регистра $L\colon V_n \to V_n$ и выходная битовая последовательность фильтрующего генератора. Пусть $b_t \in \mathbb{F}_2$ — бит выходной последовательности, а $s_t \in V_n$ — состояние регистра в момент времени t. Далее мы получим системы уравнений относительно состояний s_t, для некоторых t, для которых у нас есть значение b_t (для каких именно t, будет указываться отдельно для каждого сценария). Подставив во все полученные уравнения выражение $s_t = L^t(k)$, мы получим систему булевых уравнений той же степени относительно битов ключа k. Эту систему можно решать методом линеаризации ([1]), а также некоторыми другими. Далее под Основным алгоритмом мы будем подразумевать либо алгоритм 1, либо алгоритм 2, неважно, какой именно, так как оба этих алгоритма получают одинаковый результат.

4.1. Сценарий S3b

Если $b_t = 1$, то из уравнения $f(s_t) = b_t$ следует

$$g(s_t) = 0 \tag{15}$$

для функции g из сценария S3b. Вспомним, что Основной алгоритм выдает линейное пространство функций g для этого сценария. Вид уравнений (15) относительно состояния s_t показывает, что из всего пространства функций, выданного Основным алгоритмом, достаточно взять только линейно независимые $g_1, \ldots, g_m$ и для них составить уравнения (15). Строго говоря, имеет место следующее утверждение из линейной алгебры.

Предложение 3. *Пусть W — некоторое линейное подпространство функций в $\mathcal{F}_n$, $g_1, \ldots, g_m$ — базис в W. Тогда равносильны следующие две системы уравнений относительно неизвестного вектора $s \in V_n$:*

$$g(s) = 0 \ \forall g \in W \iff g_i(s) = 0 \ \forall i \in \{1, \ldots, m\}.$$

Таким образом, для сценария S3b мы получим m линейно независимых уравнений степени, не превосходящей d, относительно битов состояния s_t:

$$g_1(s_t) = 0$$
$$\dots$$
$$g_m(s_t) = 0.$$

Если Основной алгоритм выдал единственную нулевую функцию, то данный сценарий не применим. Это же замечание относится и к остальным двум сценариям.

4.2. Сценарий S3a

Опишем, как модифицировать алгоритм 2, чтобы он выдавал пары функций для сценария S3a. Алгоритм 1 может быть модифицирован аналогичным образом.

Вывод системы (12) мы начали с того, что приравняли все коэффициенты АНФ произведения $f \cdot g$ к нулю. Условие $\deg(f \cdot g) \leqslant d'$ равносильно тому, что $\mu(f \cdot g)(\alpha) = 0$ для всех $\alpha \in V_n$ веса, большего d'. Иными словами, чтобы получить систему уравнений относительно коэффициентов АНФ функции g для сценария S3a, нужно из системы (12) выбросить уравнения для всех $\alpha \in C_d(\mathcal{M}_f)$, вес которых не превосходит d'.

Введем обозначение

$$C_{d,d'}(\mathcal{M}_f) = \{\alpha \in C_d(\mathcal{M}_f) \mid \mathrm{wt}(\alpha) > d'\}.$$

Тогда система уравнений

$$\sum_{z \in V_n : z \leqslant \alpha,\ \mathrm{wt}(z) \leqslant d,\ |[\alpha - z; \alpha] \cap \mathcal{M}_f| \equiv 1 \pmod 2} \mu g(z) = 0 \qquad \forall \alpha \in C_{d,d'}(\mathcal{M}_f) \tag{16}$$

задает все функции g, $\deg g \leqslant d$, для сценария S3a с ограничением $\deg h \leqslant d'$, а также все аннигиляторы степени, не превосходящей d, в случаях, когда $f \cdot g = h = 0$.

Замечание 5. Для функций g и h из этого сценария соответствие $\psi: g \mapsto h$, $\psi(g) = f \cdot g$, линейно, то есть линейному пространству G функций g, полученных при решении системы (16), соответствует линейное пространство функций h, равное образу пространства G при линейном отображении ψ.

Следствием уравнения $f(s_t) = b_t$ для сценария S3a будет уравнение $h(s_t) = b_t \cdot g(s_t)$. В случае $b_t = 0$ получим, что

$$h(s_t) = 0. \tag{17}$$

Пусть функции $g_1, \dots, g_m$ образуют базис в пространстве решений системы (16), тогда в силу замечания 5 функции $h_i = \psi(g_i)$, $i \in \{1, \dots, m\}$, порождают линейное пространство функций h сценария S3a. Принимая во внимание предложение 3, для того, чтобы учесть все уравнения вида (17), достаточно выбрать из $h_1, \dots, h_m$ базис $h_{i_1}, \dots, h_{i_M}$, и мы получим линейно независимые уравнения степени, не превосходящей d':

$$h_{i_1}(s_t) = 0,$$
$$\dots$$
$$h_{i_M}(s_t) = 0. \tag{18}$$

Теперь рассмотрим случай $b_t = 1$:

$$g(s_t) + h(s_t) = 0.$$

Аналогично предыдущему случаю, выберем максимальный линейно независимый набор из порождающего $g_i + \psi(g_i)$, $i \in \{1, \ldots, m\}$, и составим систему уравнений степени, не большей $\max(d, d')$, аналогичную системе (18).

4.3. Сценарий S3c

Предложение 4. *Условие сценария* S3c *выполняется тогда и только тогда, когда найдется функция $h \in \mathcal{F}_n$ низкой степени (та же, что и в самом сценарии) такая, что $(f + 1)h = 0$.*

Доказательство. Пусть выполняется условие сценария S3c. Умножим обе части равенства $f \cdot g = h$ на f. Получим $f \cdot g = f \cdot h$. Из этих двух равенств вытекает, что $f \cdot h = h \iff (f + 1) \cdot h = 0$.

Обратно, $(f + 1) \cdot h = 0 \iff f \cdot h = h$, то есть в условии сценария S3c можно взять g, равное h.

Таким образом, мы свели сценарий S3c для функции f к сценарию S3b, но уже для функции $f + 1$. Сложность Основного алгоритма существенно не изменится при замене f на $f + 1$, поскольку $|\mathcal{M}_f|$ и $|\mathcal{M}_{f+1}|$ отличаются на единицу.

5. Заключение

В этой статье были рассмотрены два новых алгоритма нахождения всех аннигиляторов низкой степени для булевой функции f, заданной в виде алгебраической нормальной формы. Сложности предлагавшихся ранее детерминированных (не вероятностных) алгоритмов [2, 3] зависят линейно от количества наборов входных переменных, на которых функция f принимает значение 1. Поэтому для уравновешенных функций эти алгоритмы имеют сложность, экспоненциально зависящую от количества переменных n. Как мы увидели в параграфах 3.1 и 3.3, сложности новых алгоритмов ограничены сверху полиномом от n и от количества мономов, входящих в АНФ функции f с ненулевыми коэффициентами. Таким образом, при анализе фильтрующего генератора, который использует функцию усложнения f, заданную в виде суммы малого числа мономов (такие функции используются в потоковых шифрах TOYOCRYPT и LILI-128), мы можем достаточно быстро выяснить, применимы ли алгебраические методы по каждому из сценариев группы S3. Если выбранный сценарий возможен (существуют соответствующие функции низкой степени), то каждый из двух алгоритмов (или их модификации) можно использовать для получения всех линейно независимых вариантов. Как было показано в параграфе 4, наличие всех линейно независимых функций из сценариев позволяет использовать данный алгебраический метод для решения задачи нахождения ключа в полную силу в смысле предложения 3 данной статьи.

Функции усложнения, используемые в шифрах TOYOCRYPT и LILI-128, зависят от 128 и 10 переменных и представлены в виде суммы 67 и 46 мономов соответственно. При этом в TOYOCRYPT лишь 3 монома имеют степень, большую 2. Это позволит использовать предложенные в данной статье алгоритмы для проверки реализуемости сценариев S3a,

S3b, S3c и для построения соответствующих систем булевых уравнений низкой степени достаточно быстро и систематично, а не эвристически, как это делалось раньше.

Автор выражает большую признательность и благодарность Логачеву О. А. за помощь в оформлении данной статьи и полезные обсуждения.

Список литературы

1. Courtois N., Meier W., Algebraic attacks on stream ciphers with linear feedback. *Lecture Notes Computer Sci.* (2003) **2656**, 345–359.
2. Meier W., Pasalic E., Carlet C., Algebraic attacks and decomposition of Boolean functions. *Lecture Notes Computer Sci.* (2004) **3027**, 474–491.
3. Armknecht F., On the existence of low-degree equations for algebraic attacks. `http://eprint.iacr.org/2004/185`
4. Мак-Вильямс Ф., Слоэн Н., *Теория кодов, исправляющих ошибки.* Связь, Москва, 1979.
5. Кнут Д., *Искусство программирования для ЭВМ, т. 3.: Сортировка и поиск.* Мир, Москва, 1978.
6. Armknecht F.,: Improving fast algebraic attacks. *Lecture Notes Computer Sci.* (2004) **3017**, 65–82.
7. Courtois N., Fast algebraic attacks on stream ciphers with linear feedback. *Lecture Notes Computer Sci.* (2003) **2729**, 177–194.
8. Courtois N., Higher order correlation attacks, XL algorithm and cryptanalysis of Toyocrypt. *Lecture Notes Computer Sci.* (2002) **2587**, 182–199.
9. Courtois N., Klimov A., Patarin J., Shamir A., Efficient algorithms for solving overdefined systems of multivariate polynomial equations. *Lecture Notes Computer Sci.* (2000) **1807**, 392–407.
10. Hawkes P., Rose G., Rewriting variables: the complexity of fast algebraic attacks on stream ciphers. *Lecture Notes Computer Sci.* (2004) **3152**, 390–406.
11. Menezes A., van Oorschot P., Vanstone S., *Handbook of applied cryptography.* CRC Press, Boca Raton, FL, 1996.
12. Mihaljevic M., Imai H., Cryptanalysis of Toyocrypt-HS1 stream cipher. *IEICE Trans. Fund.* (2002) **E85-A**, 66–73.
13. Saarinen M.-J., A time-memory tradeoff attack against LILI-128. *Lecture Notes Computer Sci.* (2002) **2365**, 231–236.
14. Simpson L., Dawson E., Golic J., Millan W., LILI keystream generator. *Lecture Notes Computer Sci.* (2000) **2012**, 248–261.

Статья поступила 15.06.2005.