Секция 7

ВЫЧИСЛИТЕЛЬНЫЕ МЕТОДЫ В ДИСКРЕТНОЙ МАТЕМАТИКЕ

УДК 519.7 DOI 10.17223/2226308X/13/34

ПРИМЕНЕНИЕ SAT-ОРАКУЛОВ ДЛЯ ГЕНЕРАЦИИ ДОПОЛНИТЕЛЬНЫХ ЛИНЕЙНЫХ ОГРАНИЧЕНИЙ В ЗАДАЧАХ КРИПТОАНАЛИЗА НЕКОТОРЫХ ЛЕГКОВЕСНЫХ ШИФРОВ[1]

К. В. Антонов, А. А. Семёнов

Описывается новая техника, применимая к задачам алгебраического криптоанализа. В рамках предлагаемой техники строятся линейные уравнения над полем из двух элементов, которыми дополняется система алгебраических уравнений, представляющая криптоанализ рассматриваемого шифра. Для генерации новых линейных уравнений используется SAT-решатель. Показано, что применение этой техники позволяет повысить эффективность атак из класса «угадывай и определяй», основанных на понятии линеаризующего множества. Эффективность предложенной техники подтверждается вычислительными экспериментами, проведёнными для ряда ослабленных по числу шагов инициализации версий известного поточного шифра Trivium.

Ключевые слова: *линеаризующие множества, атаки из класса «угадывай и определяй», квадратичные системы над GF(2), псевдобулева оптимизация, Trivium.*

Настоящую работу можно рассматривать как прямое продолжение [1]. Приведём краткое описание используемых обозначений, понятий и вспомогательных результатов из [1]. Будем рассматривать задачу обращения (поиска прообразов) всюду определённой дискретной функции

$$f : \{0,1\}^n \to \{0,1\}^m, \tag{1}$$

которая задана некоторой программой (алгоритмом) M_f. Иными словами, зная текст программы M_f и произвольный $\gamma \in \operatorname{Range} f$, требуется найти такой $\alpha \in \{0,1\}^n$, что $f(\alpha) = \gamma$. Известно, что по M_f можно эффективно построить схему из функциональных элементов над базисом $\{\wedge, \neg\}$ (конъюнкция, отрицание), которая задаёт функцию f. Такого рода схемы в символьной верификации называются И-НЕ-графами (And-Inverter Graph или AIG [2]). На практике для построения И-НЕ-графа по конкретному алгоритму M_f можно задействовать специализированные программные средства. Мы использовали для этих целей программный комплекс Transalg [3, 4].

Пусть G_f — И-НЕ-граф, который задаёт функцию (1). В G_f выделены n вершин, не имеющие предшественников (соответствуют аргументу функции f), эти вершины называются входными. Всем остальным вершинам G_f приписаны функциональные элементы из базиса $\{\wedge, \neg\}$, эти вершины называются внутренними вершинами или узлами. Среди внутренних вершин выделены m вершин, не имеющих потомков, эти вер-

[1]Работа выполнена при финансовой поддержке Российского научного фонда, проект № 16-11-10046.

шины соответствуют значению функции f и называются выходными. С входными вершинами G_f связываются булевы переменные, образующие множество $X = \{x_1, \ldots, x_n\}$. С каждой внутренней вершиной g графа G_f связывается булева переменная v, называемая вспомогательной, множество всех вспомогательных переменных обозначается через V. В множестве V выделяется подмножество $Y = \{y_1, \ldots, y_m\}$, образованное переменными, которые приписаны выходным вершинам G_f.

По графу G_f, используя преобразования Цейтина [5], можно построить КНФ C_f, которая называется шаблонной (template CNF [6]). Также по графу G_f можно построить систему алгебраических уравнений над GF(2), каждое уравнение которой имеет степень не выше 2. Опишем кратко соответствующую процедуру. Рассматриваем множества переменных X и V. Каждой вершине g с приписанной ей переменной $v \in V$ сопоставим алгебраическое уравнение над полем GF(2). Если g — И-узел, то g имеет двух прямых предшественников в графе G_f. Предположим, что им приписаны переменные u и w. Если g — НЕ-узел, то он имеет в G_f единственного прямого предшественника, которому приписана переменная u. Произвольному g сопоставим уравнение над GF(2) по следующим правилам. Если g — И-узел, то имеем уравнение

$$u \wedge w \oplus v = 0, \tag{2}$$

если g — НЕ-узел, то имеем уравнение

$$u \oplus v = 1. \tag{3}$$

Определение 1. Пусть E_f — система, образованная уравнениями вида (2) или (3) по всем узлам графа G_f. Назовём E_f шаблонной системой уравнений над GF(2) для функции (1).

Заметим, что E_f образована уравнениями над GF(2) степени не выше 2. Стандартным образом [7, 8] определим для произвольной переменной $x \in U$, $U = X \cup V$, подстановку её значения $x = \lambda \in \{0, 1\}$ в систему E_f. Иногда в результате подстановки $x = \lambda$ вид некоторого уравнения может упроститься таким образом, что станет известным значение некоторой переменной $x' \in X \setminus \{x\}$. В таких случаях будем говорить, что соответствующее значение переменной x' индуцировано подстановкой $x = \lambda$. Например, подстановка $u = 1$ в (3) индуцирует значение 0 переменной v.

Аналогичным образом определяется произвольная подстановка вида $x = \lambda$ в шаблонную КНФ C_f. В [6] показано, что подстановка в C_f набора γ значений переменных из Y, $\gamma \in \operatorname{Range} f$, даёт выполнимую КНФ $C_f(\gamma)$, из выполняющего набора которой эффективно извлекается такое $\alpha \in \{0, 1\}^n$, что $f(\alpha) = \gamma$. Рассуждая по аналогии, можно показать, что подстановка $\gamma \in \operatorname{Range} f$ в E_f даёт совместную систему уравнений $E_f(\gamma)$ над GF(2) и из произвольного решения $E_f(\gamma)$ можно эффективно извлечь такое $\alpha \in \{0, 1\}^n$, что $f(\alpha) = \gamma$.

Понятие линеаризующего множества сформулировано в [1]. Оно обобщает понятие линеаризационного множества, введённого в [8]. Неформально говоря, линеаризующее множество линеаризует систему вида $E_f(\gamma)$ с некоторой вероятностью, которая может быть существенно меньше 1, но давать при этом атаку с относительно малой трудоёмкостью. Более точно, линеаризующее множество определяется на базе конструкции, с использованием которой в [9] предложены новые атаки из класса «угадывай и определяй». В соответствии с этой конструкцией с произвольным α, которое выбирается из $\{0, 1\}^n$ согласно равномерному распределению, и произвольным $B \subseteq X$ связывается

набор значений переменных β_α (получается в результате выбора соответствующих B компонент из α), а также $\gamma_\alpha \in \operatorname{Range} f$, такой что $f(\alpha) = \gamma_\alpha$. Вероятность линеаризации p_B — это доля таких $\alpha \in \{0,1\}^n$, что подстановка пары β_α, γ_α в систему E_f превращает её в линейную.

В [1] задача поиска линеаризующего множества с относительно малой трудоёмкостью соответствующей атаки ставится как проблема оптимизации специальной псевдобулевой функции [10], значения которой вычисляются в результате вероятностного эксперимента. В [1] для этой цели используется алгоритм, основанный на концепции tabu search [11], а в [12] — один вариант генетического алгоритма, описанный в [13]. Как итог, для задачи криптоанализа генератора A5/1 найдены линеаризующие множества, дающие атаки, трудоёмкость которых существенно меньше трудоёмкости известной атаки Р. Андерсона.

В настоящей работе описана техника, которая позволяет дополнять системы вида $E_f(\gamma)$ новыми линейными уравнениями, что в ряде случаев позволяет построить существенно более эффективные (в смысле трудоёмкости соответствующих атак) линеаризующие множества.

Итак, рассматривается задача обращения функции вида (1). Предположим, что функция f представлена в виде И-НЕ-графа G_f и по G_f построены шаблонная КНФ C_f и шаблонная система уравнений E_f над GF(2). Таким образом, и в C_f, и в E_f фигурируют переменные, образующие множество $U = X \cup V$.

Рассмотрим произвольный И-узел g в графе G_f. Пусть узлу g приписана переменная v, а прямым предшественникам g — переменные u и w. С узлом g связана булева функция $\varphi_g : \{0,1\}^3 \to \{0,1\}$, заданная формулой $u \wedge w \equiv v$. При построении C_f формула $u \wedge w \equiv v$ приводится к КНФ по таблице T_g (табл. 1).

Таблица 1

Табличное задание функции φ_g

u	w	v	φ_g
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Обозначим через S_g множество решений связанного с g уравнения $u \wedge w \oplus v = 0$. Очевидно, что S_g образовано всеми теми наборами значений переменных u, w, v, которым в таблице T_g соответствует $\varphi_g = 1$. С другой стороны, в соответствии с преобразованиями Цейтина при построении C_f в эту КНФ войдут дизъюнкции вида $u^{\neg\sigma_1} \vee w^{\neg\sigma_2} \vee v^{\neg\sigma_3}$ по всем таким наборам $(\sigma_1\sigma_2\sigma_3)$ из T_g, на которых $\varphi_g = 0$.

В основе приведённого далее результата лежит следующее наблюдение: оказывается, для целого ряда криптографических функций вида (1) для существенной доли И-узлов g в G_f в множестве S_g существуют такие наборы $(\sigma_1\sigma_2\sigma_3)$, что КНФ $u^{\sigma_1} \wedge w^{\sigma_2} \wedge v^{\sigma_3} \wedge C_f$ невыполнима. С использованием рассуждений из [6] можно показать, что данная ситуация соответствует тому факту, что никакой вход $\alpha \in \{0,1\}^n$ не может индуцировать для переменных из множества $\{u, w, v\}$ значение $(\sigma_1\sigma_2\sigma_3)$. Применительно к системе вида $E_f(\gamma)$ для произвольного $\gamma \in \operatorname{Range} f$ это означает,

что вектор $(\sigma_1\sigma_2\sigma_3)$ может быть заведомо исключён из возможных решений данной системы. Может показаться удивительным, но, как правило, на доказательство невыполнимости КНФ вида $u^{\sigma_1} \wedge w^{\sigma_2} \wedge v^{\sigma_3} \wedge C_f$ у современного SAT-решателя уходят доли секунды. Таким образом, можно говорить, что такой SAT-решатель выполняет роль оракула, эффективно отсеивающего некоторые наборы из S_g. Основной результат настоящей работы состоит в следующем.

Теорема 1. Пусть g — И-узел в И-НЕ-графе G_f, представляющем произвольную функцию f вида (1); $X_g = \{u, w, v\}$ — множество переменных, связанных с g; S_g — множество решений уравнения $u \wedge w \oplus v = 0$. Предположим, что для некоторого $\sigma = (\sigma_1\sigma_2\sigma_3)$, $\sigma \in S_g$, SAT-оракул доказал невыполнимость КНФ $u^{\sigma_1} \wedge w^{\sigma_2} \wedge v^{\sigma_3} \wedge C_f$. Тогда для любого такого σ имеет место

$$S_g \setminus \{\sigma\} = S_g \cap S_{L(X_g)},$$

где $S_{L(X_g)}$ — множество решений некоторого линейного уравнения $L(X_g)$ над F(2).

Доказательство данной теоремы получается в результате разбора всех возможных случаев исключения σ из S_g. Множество S_g приведено в табл. 2.

Таблица 2

Множество S_g для И-узла g

u	w	v	φ_g
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	1

Важное следствие теоремы 1 состоит в том, что если для И-узла g графа G_f SAT-оракул доказал невыполнимость КНФ $u^{\sigma_1} \wedge w^{\sigma_2} \wedge v^{\sigma_3} \wedge C_f$ для некоторого набора $\sigma = (\sigma_1\sigma_2\sigma_3)$, $\sigma \in S_g$, то для любого $\gamma \in \mathrm{Range}\, f$ можно добавить к системе $E_f(\gamma)$ некоторое линейное уравнение и получить эквивалентную систему. Тем самым использование шаблонной КНФ C_f и SAT-оракула позволяет эффективно добавлять в системы вида $E_f(\gamma)$ новые линейные уравнения.

Алгоритм, проверяющий для каждого И-узла g в G_f выполнимость КНФ $u^{\sigma_1} \wedge \wedge w^{\sigma_2} \wedge v^{\sigma_3} \wedge C_f$ по всем $\sigma = (\sigma_1\sigma_2\sigma_3)$, $\sigma \in S_g$, реализован в виде программы на C++. В вычислительных экспериментах описанная техника тестировалась на задачах криптоанализа ослабленных по числу шагов инициализации вариантов известного шифра Trivium [14]. Отметим, что Trivium является одним из победителей конкурса eSTREAM, и для данного шифра не известно на сегодняшний день убедительных атак, позволяющих найти секретный ключ существенно быстрее, чем полным перебором. Особенность Trivium заключается в том, что перед генерацией ключевого потока в этом шифре выполняется стадия инициализации, в ходе которой секретный ключ длиной 80 бит смешивается с несекретной 80-битной инициализирующей последовательностью. Изначально в стандарте Trivium предусмотрено 1152 шага инициализации. Однако даже при существенно меньшем числе шагов инициализации получаемые варианты Trivium оказываются стойкими ко всем известным видам криптоанализа. По-видимому, лучшими известными атаками на ослабленные по числу шагов инициализации варианты Trivium являются т. н. «кубические атаки», описанные в [15]. Следует отметить, что атаки из [15] весьма специфичны по ряду моментов. В частности,

предполагается, что противник ищет ключ, который использовался многократно совместно с различными инициализирующими векторами. В атаках, построенных нами, мы исходим из более реалистичного сценария — предполагается, что различные ключи могут использоваться совместно с некоторым фиксированным инициализирующим вектором.

Более конкретно, рассмотрены варианты Trivium с числом шагов инициализации $N = 160, 192, 288, 384$. Для каждого случая решается задача обращения функции

$$f_{(\mathrm{Tr},N)} : \{0,1\}^{80} \to \{0,1\}^{300},$$

которая соответствует алгоритму Trivium с числом шагов инициализации N и известным инициализирующим вектором (во всех экспериментах использовался один и тот же инициализирующий вектор).

Для каждого из полученных шифров мы рассматривали задачу поиска линеаризующего множества с минимальной трудоёмкостью в двух вариантах. В первом варианте использован подход [1, 12]: мы искали множество, линеаризующее систему квадратичных уравнений над GF(2), построенную по И-НЕ-графу $G_{f_{(\mathrm{Tr},N)}}$. Во втором варианте к такой системе добавляются дополнительные линейные уравнения, сгенерированные при помощи SAT-оракула в соответствии с описанной выше техникой.

Задача поиска эффективного линеаризующего множества ставится как задача минимизации псевдобулевой функции, описанной в [12]. Для её решения используется генетический алгоритм [13]. Вычислительные эксперименты проводились на кластере «Академик В. М. Матросов» Иркутского суперкомпьютерного центра СО РАН [16]. Результаты экспериментов в виде оценок трудоёмкости соответствующих атак приведены в табл. 3.

Таблица 3

Результаты сравнения двух подходов

N	Метод	$\lvert B\rvert$	p_B	Сложность атаки (число решённых систем уравнений)
160	Алгоритм из [12]	44	0,774	6,82e+13
	Алгоритм с SAT-оракулом	39	0,350	4,71e+12
192	Алгоритм из [12]	58	0,431	2,01e+18
	Алгоритм с SAT-оракулом	48	0,219	3,86e+15
288	Алгоритм из [12]	73	0,506	5,60e+22
	Алгоритм с SAT-оракулом	66	0,457	4,84e+20
384	Алгоритм из [12]	78	0,954	9,50e+23
	Алгоритм с SAT-оракулом	74	0,093	6,12e+23

Комментарии к табл. 3. В первом столбце приведено число шагов инициализации в рассматриваемой версии шифра Trivium. Во втором столбце указаны алгоритмы: мы сравниваем метод, описанный в [12], с методом, представленным в настоящей работе (с использованием SAT-оракула). В последующих столбцах приведены мощность линеаризующего множества, вероятность линеаризации и оценка числа систем линейных уравнений, которые необходимо решить для нахождения 80-битного секретного ключа.

ЛИТЕРАТУРА

1. *Семёнов А. А., Антонов К. В., Отпущенников И. В.* Поиск линеаризующих множеств в алгебраическом криптоанализе как задача псевдобулевой оптимизации // Прикладная дискретная математика. Приложение. 2019. № 12. С. 130–134.

2. *Biere A.* Bounded Model Checking // Handbook of Satisfiability. Amsterdam: IOS Press, 2009. P. 457–481.
3. *Отпущенников И. В., Семёнов А. А.* Технология трансляции комбинаторных проблем в булевы уравнения // Прикладная дискретная математика. 2011. № 1 (11). С. 96–115.
4. *Otpuschennikov I., Semenov A., Gribanova I., et al.* Encoding cryptographic functions to SAT using TRANSALG system // Proc. 22nd European Conf. ECAI 2016. Frontiers in Artificial Intelligence and Applications. 2016. V. 285. P. 1594–1595.
5. *Цейтин Г. С.* О сложности вывода в исчислении высказываний // Записки научных семинаров ЛОМИ АН СССР. 1968. Т. 8. С. 234–259.
6. *Semenov A., Otpuschennikov I., Gribanova I., et al.* Translation of algorithmic descriptions of discrete functions to SAT with application to cryptanalysis problems // Log. Methods Comput. Sci. 2020. V. 16. Iss. 1. P. 29:1–29:42.
7. *Чень Ч., Ли Р.* Математическая логика и автоматическое доказательство теорем. М.: Наука, 1983.
8. *Агибалов Г. П.* Логические уравнения в криптоанализе генераторов ключевого потока // Вестник Томского госуниверситета. Приложение. 2003. № 6. С. 31–41.
9. *Semenov A., Zaikin O., Otpuschennikov I., et al.* On cryptographic attacks using backdoors for SAT // Proc. 32nd AAAI Conf. 2018. P. 6641–6648.
10. *Boros E. and Hammer P.* Pseudo-Boolean optimization // Discr. Appl. Math. 2002. V. 123. Iss. 1–3. P. 155–225.
11. *Glover F. and Laguna M.* Tabu Search. Norwell: Kluwer Academic Publishers, 1997.
12. *Антонов К. В., Семёнов А. А.* Применение метаэвристических алгоритмов псевдобулевой оптимизации к поиску линеаризующих множеств в криптоанализе криптографических генераторов // Материалы 6-й Междунар. школы-семинара «Синтаксис и семантика логических систем». Иркутск: ИГУ, 2019. С. 13–18.
13. *Pavlenko A., Semenov A., and Ulyantsev V.* Evolutionary computation techniques for constructing SAT-based attacks in algebraic cryptanalysis // LNCS. 2019. V. 11454. P. 237–253.
14. *De Canniere C.* Trivium: A stream cipher construction inspired by block cipher design principles // LNCS. 2006. V. 4176. P. 171–186.
15. *Dinur I. and Shamir A.* Cube attacks on tweakable black box polynomials // LNCS. 2009. V. 5479. P. 278–299.
16. ЦКП Иркутский суперкомпьютерный центр СО РАН. `http://hpc.icc.ru`.

УДК 519.7 DOI 10.17223/2226308X/13/35

О ДИФФЕРЕНЦИАЛАХ ДЛЯ МОДИФИКАЦИИ ШИФРА SIMON НА ОСНОВЕ СХЕМЫ ЛАЯ — МЕССИ[1]

А. А. Белоусова, Н. Н. Токарева

Рассматриваются блочный итеративный шифр Simon 32/64, основанный на сети Фейстеля, и его модификации на основе схемы Лая — Месси. Получены оценки вероятностей дифференциалов 12 раундов исходного шифра и его модификаций.

Ключевые слова: *схема Лая — Месси, сеть Фейстеля, дифференциальный криптоанализ.*

[1]Работа выполнена в рамках государственного задания Института математики им. С. Л. Соболева СО РАН (проект № 0314-2019-0017) при поддержке РФФИ (проект № 18-07-01394) и Лаборатории криптографии JetBrains Research.