УДК 519.7 DOI 10.17223/2226308X/12/38

ПОИСК ЛИНЕАРИЗУЮЩИХ МНОЖЕСТВ В АЛГЕБРАИЧЕСКОМ КРИПТОАНАЛИЗЕ КАК ЗАДАЧА ПСЕВДОБУЛЕВОЙ ОПТИМИЗАЦИИ[1]

А. А. Семёнов, К. В. Антонов, И. В. Отпущенников

Вводится понятие линеаризующего множества, которое можно рассматривать как обобщение известного понятия линеаризационного множества. Линеаризующие множества используются в основе алгебраических атак, относящихся к классу «угадывай и определяй» (guess-and-determine). В таких атаках угадываются значения переменных из некоторого множества, затем эти значения подставляются в систему алгебраических уравнений, которая связывает входные и выходные данные рассматриваемого шифра. В некоторых случаях результатом такой подстановки является линейная система, которая решается эффективно. Рассматриваются алгебраические уравнения над полем GF(2). Значения переменных из линеаризующего множества (в отличие от линеаризационного) линеаризуют систему уравнений с некоторой вероятностью, которая, вообще говоря, может быть существенно меньше 1. Оценка трудоёмкости атаки на основе конкретного линеаризующего множества строится через специально определяемую псевдобулеву функцию. Минимальное значение этой функции даёт оценку трудоёмкости лучшей по эффективности атаки. Для минимизации таких функций используется метаэвристический алгоритм из класса «поиск с запретами». На данном этапе построены атаки описанного типа для ряда криптографических генераторов. В частности, для известного генератора A5/1 построена атака с оценкой трудоёмкости в 4,5 раз ниже трудоёмкости известной атаки Андерсона.

Ключевые слова: *атаки из класса «угадывай и определяй», линеаризующие множества, псевдобулева оптимизация.*

Понятие линеаризационного множества введено Г. П. Агибаловым в 2003 г. [1] в контексте проблемы построения атак на некоторые генераторы ключевого потока. Атаки, рассмотренные в [1], относятся к классу алгебраических [2]. В основе алгебраических атак лежат эффективные процедуры, сводящие задачи обращения (поиска прообразов) рассматриваемых криптографических функций к поиску решений алгебраических уравнений. Обычно такие уравнения — это уравнения над полем GF(2). Как правило, в результате такой сводимости получается система, не все уравнения которой являются линейными. Известно (см., например, [3]), что задача определения совместности даже квадратичной системы над GF(2) является NP-полной и соответственно не может быть решена в общем случае за полиномиальное время известными алгоритмами. Тем не менее может оказаться так, что угадывание значений относительно небольшого числа переменных в такой системе с последующими эффективными преобразованиями превратит эту систему в линейную. Решая все возможные такие линейные системы, в ряде случаев можно построить атаки, которые существенно эффективнее атаки методом грубой силы.

Более точно, рассмотрим всюду определённую функцию $f : \{0,1\}^n \to \{0,1\}^m$, заданную некоторым алгоритмом. Требуется по произвольному $\gamma \in \mathrm{Range}\, f$ найти такой $\alpha \in \{0,1\}^n$, что $f(\alpha) = \gamma$. Известны теоретические результаты (аналогичные теореме Кука — Левина [3]), в соответствии с которыми сформулированную задачу можно

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

эффективно свести к проблеме поиска решения совместной системы алгебраических уравнений над полем GF(2) (кратко мы опишем такую сводимость ниже). Обозначим такую систему через $E_f(\gamma)$. Пусть X — множество переменных, присутствующих в системе $E_f(\gamma)$. В соответствии с [1], множество $B \subseteq X$ называется линеаризационным (для системы $E_f(\gamma)$), если подстановка любого $\beta \in \{0,1\}^{|B|}$ в $E_f(\gamma)$ превращает эту систему в линейную. Через $\{0,1\}^{|B|}$ обозначено множество всех наборов значений переменных, входящих в B. Подстановка подразумевается в том смысле, как в [1] (или, по сути, в том же смысле, как в [4]). В [1] предъявлены примеры линеаризационных множеств для целого ряда генераторов ключевого потока (Геффе, пороговый, генератор переменного шага). Следует отметить, что в каждом из этих примеров вид множества B не зависит от $\gamma \in \operatorname{Range} f$ (как правило, такое множество образовано переменными, соответствующими одному или нескольким LFSR, входящим в состав генератора). Получаемые на основе данных множеств атаки оказываются существенно более эффективными, чем атаки методом грубой силы.

Введём понятие линеаризующего множества. Главное отличие таких множеств от линеаризационных множеств из [1] заключается в том, что «вероятность линеаризации» рассматриваемой системы наборами значений переменных, образующих такое множество, вообще говоря, меньше 1.

Для понимания сути вводимого понятия удобно задать функцию f схемой S_f из функциональных элементов базиса $\{\wedge, \neg\}$. Припишем входным полюсам схемы переменные, образующие множество $X = \{x_1, \ldots, x_n\}$. Внутренним узлам схемы, которые соответствуют функциональным элементам, припишем переменные, образующие множество $V = \{v_1, \ldots, v_N\}$, $V \cap X = \varnothing$. В множестве V выделим подмножество $Y = \{y_1, \ldots, y_m\}$, переменные которого приписаны выходам S_f. Отметим, что любому $\alpha \in \{0,1\}^{|X|}$, поданному на вход S_f, однозначно соответствует набор значений всех переменных из V, получаемый в результате последовательного вычисления значений функций, соответствующих внутренним узлам S_f.

Построим по S_f систему алгебраических уравнений над полем $\mathrm{GF}(2) = \langle \{0,1\}, \oplus, \wedge \rangle$ по следующим правилам. Пусть g — произвольный внутренний узел схемы S_f и $v(g) \in$ $\in V$ — приписанная ему переменная. Если g — это И-узел, то он имеет в графе, представляющем S_f, двух прямых предшественников, пусть им приписаны переменные u и w. Если g — это НЕ-узел, то g имеет единственного предшественника, пусть u — приписанная ему переменная. Сопоставим каждому g уравнение над GF(2):

1) если g — И-узел, то ему сопоставляется уравнение $u \wedge w \oplus v(g) = 0$;
2) если g — НЕ-узел, то ему сопоставляется уравнение $u \oplus v(g) = 1$.

Объединим уравнения по всем внутренним узлам схемы S_f в общую систему, которую обозначим через E_f.

В систему E_f будем подставлять значения некоторых переменных из множества $U = X \cup V$. Подстановку определим в соответствии с [4], т. е. для произвольной переменной $x \in U$ и константы $\lambda \in \{0,1\}$ скажем, что происходит подстановка $x = \lambda$ в систему E_f, если все вхождения переменной в E_f заменены вхождением константы λ. После подстановки к соответствующим уравнениям применяются преобразования, называемые элементарными. Например, результат подстановки $w = 1$ и соответствующего элементарного преобразования в отношении уравнения $u \wedge w \oplus v = 0$ — линейное уравнение $u \oplus v = 0$. Иногда результатом подстановки и последующих преобразований могут стать значения некоторых переменных. Например, подстановка $v = 1$ и последующие преобразования в отношении $u \wedge w \oplus v = 0$ дают уравнение $u \wedge w = 1$, откуда

следует $u = w = 1$. Назовём эти значения индуцированными подстановкой $v = 1$. Индуцированные значения также могут быть подставлены в систему. Подстановка значений для упорядоченного множества переменных определяется индуктивно: подставляется значение первой переменной и все индуцированные данной подстановкой значения, затем значение второй переменной и т. д.

Рассмотрим произвольный $\gamma \in \operatorname{Range} f$ как набор значений переменных из Y и подставим его в E_f. Обозначим полученную систему через $E_f(\gamma)$. Можно показать, что система $E_f(\gamma)$ совместна, и если найдено некоторое её решение, то из него можно эффективно выделить такой $\alpha \in \{0,1\}^n$, что $f(\alpha) = \gamma$.

Зададим на $\{0,1\}^n$ равномерное распределение и выберем в соответствии с ним $\alpha \in \{0,1\}^n$. Пусть $\gamma_\alpha = f(\alpha)$ и B — произвольное подмножество в множестве $U \setminus Y$. Как было сказано выше, подав α на вход схеме S_f, мы эффективно (в общем случае за линейное от числа узлов в S_f время) вычислим значения всех переменных из V, в том числе значения переменных, входящих в B. Обозначим соответствующий набор через β_α. Теперь подставим в E_f наборы γ_α и β_α. Результат этой подстановки и последующих элементарных преобразований обозначим через $E_f(\gamma_\alpha, \beta_\alpha)$. Действуя по аналогии с [5], введём случайную величину ξ, которая принимает значение 1, если система $E_f(\gamma_\alpha, \beta_\alpha)$ — линейная система над GF(2). В противном случае полагаем, что $\xi = 0$. Пусть p_B — доля таких векторов $\alpha \in \{0,1\}^n$, для которых $\xi = 1$. Таким образом, p_B — это некоторая числовая характеристика множества B. Очевидно, что $p_B = M[\xi]$. Тогда для конкретного B по схеме, которая аналогична предложенной в [5], можно оценить p_B при помощи метода Монте-Карло [6].

Определение 1. В контексте рассмотренной задачи назовём множество B линеаризующим множеством с вероятностью линеаризации p_B.

Для произвольного $B \subseteq U \setminus Y$ и вероятности линеаризации p_B можно описать следующую стратегию поиска прообразов функции f. Полагаем, что α выбран из $\{0,1\}^n$ в соответствии с равномерным распределением. Известен алгоритм вычисления f и $\gamma_\alpha = f(\alpha)$. Требуется найти α. Пусть известно некоторое множество B с вероятностью линеаризации p_B. Для каждого $\beta \in \{0,1\}^{|B|}$ будем рассматривать систему $E_f(\gamma_\alpha, \beta)$. Если $E_f(\gamma_\alpha, \beta)$ не является линейной системой, то не делаем в её отношении никаких действий и переходим к следующему β. Вероятность события «$E_f(\gamma_\alpha, \beta_\alpha)$ — линейная» равна p_B. Если $E_f(\gamma_\alpha, \beta_\alpha)$ — линейная, то, решив её, найдём α. Таким образом, для конкретного γ_α и конкретного B, перебрав всё множество $\{0,1\}^{|B|}$, мы линеаризуем систему с помощью вектора $\beta_\alpha \in \{0,1\}^{|B|}$ (и соответственно находим α) с вероятностью p_B. Если для конкретного γ_α описанная стратегия не даёт результата (то есть при переборе всех векторов из $\{0,1\}^{|B|}$ вектор β_α не линеаризует соответствующую систему), то рассматриваем следующий выход функции f (полагаем, что он также построен по случайно выбранному из $\{0,1\}^n$ входу). Вероятность того, что для случайно и независимо выбранных входов $\alpha_1, \ldots, \alpha_r$ при помощи описанной стратегии удастся обратить хотя бы один $\gamma_{\alpha_1}, \ldots, \gamma_{\alpha_r}$ $\left(\gamma_{\alpha_j} = f\left(\alpha_j\right), j \in \{1, \ldots, r\}\right)$ есть $P_B(r) = 1 - (1 - p_B)^r$ и стремится к 1 с ростом r. Если мы хотим, чтобы $P_B(r) > 95\,\%$, то, как нетрудно видеть, при малых p_B ($p_B < 0{,}2$) r должно быть не меньше $3/p_B$ (аналогичная ситуация разобрана в [5]). Будем считать единицей трудоёмкости операцию подстановки β в систему $E_f(\gamma_\alpha)$ и проверки получаемой системы на линейность. Тогда трудоёмкость описанной атаки при условии, что $P_B(r) > 95\,\%$, составит $\approx 2^{|B|} \cdot 3/p_B$ таких единиц.

Отметим, что линеаризационные множества, примеры которых приведены в [1] в отношении генераторов Геффе, порогового и переменного шага (Alternating Step Generator), — это линеаризующие множества с $p_B = 1$.

Нетрудно понять, что, например, $B = X$ — это тривиальный пример линеаризующего множества с $p_B = 1$. Соответственно нетривиальные такие множества, вероятность p_B для которых может быть существенно меньше 1, можно искать как подмножества X. С этой целью, действуя по аналогии с [5], введём специальную функцию $\Phi(B)$, оценивающую трудоёмкость атаки описанного типа и использующую множество B. На вход она получает булев вектор, единицы в котором означают присутствие переменных из X в множестве B. Значение функции — это оценка величины $2^{|B|} \cdot 3/p_B$, для построения которой вероятность p_B оценивается методом Монте-Карло. Функция $\Phi(B)$ — это «псевдобулева» функция [7]. Множество B, на котором значение $\Phi(B)$ минимально, даст атаку описанного типа с лучшей трудоёмкостью. Для минимизации псевдобулевых функций, которые не заданы аналитически (как в нашем случае), используются различные метаэвристические алгоритмы. Мы использовали для этих целей метаэвристику, известную как «поиск с запретами» (Tabu Search) [8]. Соответствующий алгоритм реализован в форме многопоточного приложения и запускался на одном рабочем узле кластера «Академик В. М. Матросов» Иркутского суперкомпьютерного центра [9] (36 ядер процессора Intel Xeon E5-2695). Для построения систем вида E_f использовались возможности программы Transalg [10, 11] (в частности, схема S_f строилась в виде И-НЕ-графа).

Полученные на данном этапе вычислительные результаты можно оценивать как предварительные. Так, для ASG-192 (генератора переменного шага с ключом длиной 192 бит) в автоматическом режиме (как результат минимизации функции $\Phi(B)$) найдено линеаризующее множество, состоящее из переменных управляющего регистра (т. е. фактически линеаризационное множество из [1]). Для ASG-96 лучшая оценка трудоёмкости достигнута для множества с $p_B \approx 99\,\%$. Интересный результат получен для известного генератора A5/1. Одна из первых атак на данный генератор, описанная в [12] (атака Андерсона), фактически использовала линеаризационное множество, состоящее из 53 бит (в A5/1 используется ключ длиной 64 бита). То есть «множество Андерсона» из [12] — это линеаризующее множество с $p_B = 1$. С использованием процедуры минимизации функции вида $\Phi(B)$ для A5/1 было найдено линеаризующее множество, состоящее из 47 переменных с $p_B \approx 0{,}071$. Оценка трудоёмкости атаки на основе этого множества оказалась примерно в 4,5 раз ниже, чем трудоёмкость атаки Андерсона.

ЛИТЕРАТУРА

1. *Агибалов Г. П.* Логические уравнения в криптоанализе генераторов ключевого потока // Вестник Томского госуниверситета. Приложение. 2003. № 6. С. 31–41.
2. *Bard G.* Algebraic Cryptanalysis. Springer Publishing Company, Inc., 2009.
3. *Goldreich O.* Computational Complexity: A Conceptual Perspective. Cambridge: Cambridge University Press, 2008.
4. *Чень Ч., Ли Р.* Математическая логика и автоматическое доказательство теорем. М.: Наука, 1983.
5. *Semenov A., Zaikin O., Otpuschennikov I., et al.* On cryptographic attacks using backdoors for SAT // Thirty-Second AAAI Conf., 2018. P. 6641–6648. `https://arxiv.org/abs/1803.04646`.

6. *Metropolis N. and Ulam S.* The Monte Carlo method // J. Amer. Statistical Association. 1949. V. 44. No. 247. P. 335–341.
7. *Boros E. and Hammer P.* Pseudo-Boolean optimization // Discr. Appl. Math. 2002. V. 123, Iss. 1–3. P. 155–225.
8. *Glover F. and Laguna M.* Tabu Search. Norwell: Kluwer Academic Publishers, 1997.
9. ЦКП Иркутский суперкомпьютерный центр СО РАН. `http://hpc.icc.ru`
10. *Отпущенников И. В., Семенов А. А.* Технология трансляции комбинаторных проблем в булевы уравнения // Прикладная дискретная математика. 2011. № 1(11). С. 96–115.
11. *Otpuschennikov I., Semenov A., Gribanova I., et al.* Encoding cryptographic functions to SAT using TRANSALG system // Frontiers in Artificial Intelligence and Applications. 2016. V. 285. P. 1594–1595.
12. *Anderson R.* A5 (Was: Hacking digital phones). Newsgroup Communication. 1994. `http://yarchive.net/phone/gsmcipher.html`.

УДК 519.719.2 DOI 10.17223/2226308X/12/39

ОБ АППАРАТНОЙ РЕАЛИЗАЦИИ ОДНОГО КЛАССА БАЙТОВЫХ ПОДСТАНОВОК

Д. Б. Фомин, Д. И. Трифонов

Рассмотрены вопросы реализации на ПЛИС и СБИС одного класса подстановок и проведено сравнение с реализациями произвольных байтовых отображений. Изучен способ реализации произвольной подстановки. Показано, что любая подстановка на множестве V_8 может быть реализована с использованием 40 LUT (812 GE). Для одного класса подстановок на множестве V_8, обладающего высокими криптографическими свойствами, показана возможность реализации с использованием 19 LUT (147 GE).

Ключевые слова: *S-Box, подстановка, ПЛИС, СБИС.*

Согласно критерию Шеннона [1], каждая криптографически безопасная функция должна представлять собой композицию функций, реализующих свойство перемешивания и рассеивания. Наиболее широко распространённый способ обеспечить свойство перемешивания — использование нелинейных преобразований, в частности подстановок. Подстановки являются неотъемлемой частью большого класса криптографических функций, таких, как поточные и блочные шифры, хэш-функции. К подстановкам предъявляются требования, позволяющие гарантировать невозможность применимости известных методов криптографического анализа, таких, как линейный, алгебраический и разностный.

Помимо криптографических требований, также предъявляются требования и к реализации подстановок, что порождает подходы к построению подстановок больших размерностей с использованием преобразований меньших размерностей. Это позволяет добиться возможности:

— программной реализации с большими таблицами замен;
— программной реализации с меньшим количеством битовых преобразований (bitslice-реализации [2]);
— использования подстановок для низкоресурсной реализации на ПЛИС и СБИС;
— эффективного аппаратного маскирования [3, 4].

Известно большое количество способов построения подстановок с использованием преобразований меньшей размерности: на основе сети Фейстеля [5 – 7], с использовани-