9. *Glover F. and Laguna M.* Tabu Search. Norwell: Kluwer Academic Publishers, 1997. 401 p.
10. *Doerr B., Le H., Makhmara R., et al.* Fast genetic algorithms // Proc. GECCO'17. 2017. P. 777–784.
11. *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.
12. *Muhlenbein H.* How genetic algorithms really work: Mutation and hill climbing // Proc. PPSN-II. 1992. P. 15–26.
13. *Wegener I.* Theoretical aspects of evolutionary algorithms // ICALP 2001. LNCS. 2001. V. 2076. P. 64–78.
14. *Droste S., Jansen T., and Wegener I.* On the analysis of the (1+1) evolutionary algorithm // Theor. Comput. Sci. 2002. V. 276 (1–2). P. 51–81.
15. *Luke S.* Essentials of Metaheuristics. Second Edition. 2015. 261 p. `https://cs.gmu.edu/~sean/book/metaheuristics/Essentials.pdf`.
16. *Rivest R. L.* The MD4 message digest algorithm // CRYPTO'90. LNCS. 1990. V. 537. P. 303–311.
17. *Dobbertin H.* The first two rounds of MD4 are not one-way // FSE 1998. LNCS. 1998. V. 1372. P. 284–292.
18. *De D., Kumarasubramanian A., and Venkatesan R.* Inversion attacks on secure hash functions using SAT Solvers // FSE 2007. LNCS. 2007. V. 4501. P. 377–382.
19. *Gribanova I. and Semenov A.* Using automatic generation of relaxation constraints to improve the preimage attack on 39-step MD4 // Proc. 41st Intern. Convention MIPRO 2018. Opatija, 2018. P. 1174–1179.
20. *Грибанова И. А., Семёнов А. А.* Об аргументации отсутствия свойств случайного оракула у некоторых криптографических хеш-функций // Прикладная дискретная математика. Приложение. 2019. № 12. С. 95–98.
21. *Gribanova I. A. and Semenov A. A.* Parallel guess-and-determine preimage attack with realistic complexity estimation for MD4-40 cryptographic hash function // Труды XIII Междунар. конф. «Параллельные вычислительные технологии», Калининград, 02–04 апреля 2019. С. 8–18.
22. Иркутский суперкомпьютерный центр СО РАН. `http://hpc.icc.ru`.

УДК 519.7 DOI 10.17223/2226308X/13/38

ПРИМЕНЕНИЕ SAT-РЕШАТЕЛЕЙ ДЛЯ ПОСТРОЕНИЯ БУЛЕВЫХ ФУНКЦИЙ С ЗАДАННЫМИ КРИПТОГРАФИЧЕСКИМИ СВОЙСТВАМИ[1]

А. Е. Доронин, К. В. Калгин

Представлен подход к решению некоторых криптографических задач, основанный на их сведении к классической задаче о выполнимости и последующем использовании SAT-решателей. Построены формулы, определяющие условия взаимной однозначности и дифференциальной равномерности векторной булевой функции.

Ключевые слова: *SAT-решатели, криптография, булевы функции.*

[1]Работа выполнена при поддержке РФФИ (проект № 18-07-01394) и Лаборатории криптографии JetBrains Research.

В настоящее время SAT-решатели используются для решения криптографических задач разного типа. Например, проведён криптоанализ асимметричной криптосистемы RSA [1], в результате которого удалось факторизовать числа до 417 бит; выполнен криптоанализ шифра Trivium и его модификаций [2]. В [3] представлена гомоморфная криптосистема с открытым ключом, основанная на SAT-задаче. С помощью SAT-решателей успешно проверяется обратимость векторных булевых функций [4].

В данной работе предлагается использование SAT-решателей в задачах построения криптографических булевых функций и проверки эквивалентности двух булевых функций. Для получения набора булевых формул использованы следующие понятия и свойства.

Определение 1. Векторная булева функция $F : \mathbb{Z}_2^n \to \mathbb{Z}_2^n$ называется *взаимно однозначной*, если она инъективна и сюръективна, то есть одновременно выполняются следующие условия:

1) $\forall x' \in \mathbb{Z}_2^n \; \forall x'' \in \mathbb{Z}_2^n \; (x' \neq x'' \to F(x') \neq F(x''))$;
2) $\forall y \in \mathbb{Z}_2^n \; \exists x \in \mathbb{Z}_2^n \; (F(x) = y)$.

Определение 2. Векторная булева функция $F : \mathbb{Z}_2^n \to \mathbb{Z}_2^n$ является *дифференциально δ-равномерной*, если для любого ненулевого $a \in \mathbb{Z}_2^n$ и произвольного $b \in \mathbb{Z}_2^n$ уравнение $F(x) \oplus F(x \oplus a) = b$ имеет не более δ решений.

Условия, фигурирующие в определениях, представляются в виде КНФ и подаются на вход SAT-решателя. В результате его работы происходит означивание переменных таким образом, чтобы формулы были истинными, а следовательно, условия выполнялись.

Векторные булевы функции были закодированы в двух представлениях:

1) В *разреженном* представлении используется 2^{2n} переменных $f_{x,y}$, из которых 2^n равны 1, остальные равны 0: $f_{x,y} = 1 \iff F(x) = y$, где $x, y \in \mathbb{Z}_2^n$.
2) В *плотном* представлении используется $n2^n$ переменных $fb_{x,k}$: $fb_{x,k} = 1 \iff F_k(x) = 1$, где $F(x) = (F_0(x), F_2(x), \ldots, F_{n-1}(x))$, $F_k : \mathbb{Z}_2^n \to \mathbb{Z}_2$, $k = 0, \ldots, n-1$; $x \in \mathbb{Z}_2^n$.

Для записи условий на переменные $f_{x,y}$ и $fb_{x,k}$ понадобятся следующие вспомогательные переменные:

— $fbq_{x,y,k} = 1 \iff fb_{x,k} \neq fb_{y,k}$, где $k = 0, \ldots, n-1$; $x, y \in \mathbb{Z}_2^n$;
— $d_{x,a,b} = 1 \iff F(x) \oplus F(x \oplus a) = b$, где $x, a, b \in \mathbb{Z}_2^n$;
— $de_{x,y,a} = 1 \iff F(x) \oplus F(x \oplus a) = F(y) \oplus F(y \oplus a)$, где $x, y, a \in \mathbb{Z}_2^n$;
— $dbq_{x,y,a,k} = 1 \iff fbq_{x,x\oplus a,k} = fbq_{y,y\oplus a,k}$, где $k = 0, \ldots, n-1$, $x, y, a \in \mathbb{Z}_2^n$.

В КНФ эти зависимости записываются следующим образом:

$$\mathbf{SoP^D}(fb, fbq) = \bigwedge_{x,y,k} (fbq_{x,y,k} \vee fb_{x,k} \vee \overline{fb_{y,k}}) \wedge (fbq_{x,y,k} \vee \overline{fb_{x,k}} \vee fb_{y,k}) \wedge$$
$$\wedge (\overline{fbq_{x,y,k}} \vee fb_{x,k} \vee fb_{y,k}) \wedge (\overline{fbq_{x,y,k}} \vee \overline{fb_{x,k}} \vee \overline{fb_{y,k}});$$
$$\mathbf{SpDen}(f, fb) = \bigwedge_{x,y,k} (\overline{f_{x,y}} \vee fb_{x,k}) \wedge (f_{x,y} \vee \overline{fb_{x,0}^{y_0}} \vee \ldots \vee \overline{fb_{x,n-1}^{y_{n-1}}});$$
$$\mathbf{Der^S}(f, d) = \bigwedge_{b,a,z,x} (f_{x,z} \vee f_{x\oplus a,z\oplus b} \vee \overline{d_{x,a,b}}) \wedge (f_{x,z} \vee \overline{f_{x\oplus a,z\oplus b}} \vee \overline{d_{x,a,b}}) \wedge$$
$$\wedge (\overline{f_{x,z}} \vee f_{x\oplus a,z\oplus b} \vee \overline{d_{x,a,b}}) \wedge (\overline{f_{x,z}} \vee \overline{f_{x\oplus a,z\oplus b}} \vee d_{x,a,b});$$

$$\mathbf{SoPEq^D}(fbq, dbq) = \bigwedge_{a,x,y,k} (dbq_{x,y,a,k} \vee fbq_{x,x\oplus a,k} \vee fbq_{y,y\oplus a,k}) \wedge$$
$$\wedge (dbq_{x,y,a,k} \vee \overline{fbq_{x,x\oplus a,k}} \vee \overline{fbq_{y,y\oplus a,k}}) \wedge (\overline{dbq_{x,y,a,k}} \vee fbq_{x,x\oplus a,k} \vee \overline{fbq_{y,y\oplus a,k}}) \wedge$$
$$\wedge (\overline{dbq_{x,y,a,k}} \vee \overline{fbq_{x,x\oplus a,k}} \vee fbq_{y,y\oplus a,k});$$
$$k = 0, \ldots, n-1;\ x, y, z, a, b \in \mathbb{Z}_2^n.$$

Свойства из определений 1 и 2 можно записать следующими формулами.

Теорема 1. Переменные $f_{x,y}$ задают функцию $F : \mathbb{Z}_2^n \to \mathbb{Z}_2^n$ тогда и только тогда, когда следующая формула является истинной:

$$\mathbf{F^S}(f) = \bigwedge_{x\in\mathbb{Z}_2^n} \Big(\bigwedge_{\substack{y',y''\in\mathbb{Z}_2^n \\ y'<y''}} \overline{f_{x,y'}} \vee \overline{f_{x,y''}} \Big) \wedge \bigwedge_{x\in\mathbb{Z}_2^n} \Big(\bigvee_{y\in\mathbb{Z}_2^n} f_{x,y} \Big). \tag{1}$$

Формула (1) состоит из $2^{3n-1} - 2^{2n-1}$ дизъюнкций длины 2 и 2^n дизъюнкций длины 2^n.

Теорема 2. Переменные $f_{x,y}$ задают взаимно однозначную векторную булеву функцию $F : \mathbb{Z}_2^n \to \mathbb{Z}_2^n$ тогда и только тогда, когда следующая формула является истинной:

$$\mathbf{P^S}(f) = \bigwedge_{y\in\mathbb{Z}_2^n} \Big(\bigwedge_{\substack{x',x''\in\mathbb{Z}_2^n \\ x'<x''}} \overline{f_{x',y}} \vee \overline{f_{x'',y}} \Big) \wedge \mathbf{F^S}(f). \tag{2}$$

Формула (2) состоит из $2^{3n} - 2^{2n}$ дизъюнкций длины 2 и 2^n дизъюнкций длины 2^n.

Теорема 3. Переменные $fb_{x,k}$ и $fbq_{x,y,k}$ задают взаимно однозначную векторную булеву функцию $F : \mathbb{Z}_2^n \to \mathbb{Z}_2^n$ тогда и только тогда, когда следующая формула является истинной:

$$\mathbf{P^D_{sum}}(fb, fbq) = \bigwedge_{x,y\in\mathbb{Z}_2^n} \bigvee_k fbq_{x,y,k} \wedge \mathbf{SoP^D}(fb, fbq). \tag{3}$$

В формуле (3) содержится $n(2^{2n} - 2^n)$ дизъюнкций длины 3 и $2^{2n} - 2^n$ дизъюнкций длины n.

Теорема 4. Переменные $f_{x,y}$ и $fb_{x,k}$ задают взаимно однозначную векторную булеву функцию $F : \mathbb{Z}_2^n \to \mathbb{Z}_2^n$ тогда и только тогда, когда следующая формула является истинной:

$$\mathbf{P^D_{sparse}}(f, fb) = \bigwedge_x \bigvee_{y\neq x} f_{x,y} \wedge \mathbf{SpDen}(f, fb). \tag{4}$$

В формуле (4) содержится по $n(2^{2n} - 2^n)$ дизъюнкций длины 2 и n и 2^n дизъюнкций длины 2^n.

Теорема 5. Переменные $f_{x,y}$ и $d_{x,a,b}$ задают APN-функцию $F : \mathbb{Z}_2^n \to \mathbb{Z}_2^n$ тогда и только тогда, когда выполняются условия теоремы 1 и следующая формула является истинной:

$$\mathbf{APN^S}(f, d) = \mathbf{Der^S}(f, d) \wedge \bigwedge_{\substack{b\neq 0, a\neq 0, \\ x,y\neq x}} (\overline{d_{x,a,b}} \vee \overline{d_{y,a,b}}). \tag{5}$$

В формуле (5) содержится порядка 2^{4n} дизъюнкций длины 3 и 2.

Теорема 6. Переменные $f_{x,y}$ и $d_{x,a,b}$ задают APN-функцию $F : \mathbb{Z}_2^n \to \mathbb{Z}_2^n$ тогда и только тогда, когда следующая формула является истинной:

$$\mathbf{APN^D}(fb, fbq, dbq) = \mathbf{SoPEq^D}(fbq, dbq) \wedge \mathbf{SoP^D}(fb, fbq) \wedge \bigwedge_{a,x,y} \bigvee_k \overline{dbq_{x,y,a,k}}. \tag{6}$$

В формуле (6) содержится порядка 2^{3n} дизъюнкций длины 3 и n.

На основе полученных формул генерируется входной файл для SAT-решателя. Формулы можно также использовать для тестирования работы новых SAT-решателей, созданных для криптографических задач.

ЛИТЕРАТУРА

1. *Огородников Ю. Ю.* Комбинированная атака на алгоритм RSA с использованием SAT-подхода // Динамика систем, механизмов и машин. Омск: ОмГТУ, 2016. С. 276–284.
2. *Заикин О. С., Отпущенников И. В., Семёнов А. А.* Оценки стойкости шифров семейства Trivium к криптоанализу на основе алгоритмов решения проблемы булевой выполнимости // Прикладная дискретная математика. Приложение. 2016. № 9. С. 46–48.
3. *Schmittner S. E.* A SAT-based Public Key Cryptography Scheme. IACR Cryptol. ePrint Arch. 2015. `https://eprint.iacr.org/2015/771.pdf`.
4. *Wille R., Lye A., and Niemann P.* Checking reversibility of Boolean functions // LNCS. 2016. V. 9720. P. 322–337.

УДК 519.688 DOI 10.17223/2226308X/13/39

О ВЫЧИСЛЕНИИ СИСТЕМЫ ПЕРЕПИСЫВАЮЩИХ ПРАВИЛ В КОНЕЧНОЙ ГРУППЕ

А. А. Кузнецов

Представлен алгоритм, определяющий переписывающую систему конечной группы, заданной фиксированным порождающим множеством. Необходимым условием эффективной реализации алгоритма является наличие быстрой процедуры умножения элементов в группе. Такой групповой операцией может быть композиция подстановок, умножение матриц, вычисление полиномов Холла и т. д. Алгоритм был применён для исследования переписывающих систем в конечных двупорождённых группах периода 5.

Ключевые слова: *система переписывающих правил, группа Бернсайда.*

Решение некоторых задач теории кодирования и криптографии сводится к исследованию подходящих графов Кэли, например открытая проблема эффективного восстановления вершин в графе Хэмминга [1].

Поиск кратчайших путей в графах Кэли является труднорешаемой проблемой, поэтому исследователям приходится идти на различные уловки и приёмы, чтобы получить решение за приемлемое время. Например, в [2] сначала определяют автоматическую структуру группы, которая порождает соответствующий граф Кэли. Автоматическая структура группы состоит из конечных автоматов специального вида [3]. Для их вычисления требуется определить множество соотношений в группе, используя известный алгоритм Кнута — Бендикса [4].

Зачастую алгоритм Кнута — Бендикса работает недопустимо долго, например в конечных группах, заданных коммутаторными соотношениями. В этом случае разворачивание коммутаторных соотношений приводит к очень длинным словам, что катастрофически замедляет работу алгоритма.

Настоящая работа представляет собой попытку устранить указанный недостаток. Остановимся подробнее на основных определениях.

Пусть $G = \langle X \rangle$ — конечная группа, порождённая упорядоченным множеством $X = \{x_1 \prec x_2 \prec \ldots \prec x_m\}$, которое также называют алфавитом. Множество всех