ЛИТЕРАТУРА

1. *Малоземов В. Н., Машарский С. М.* Основы дискретного гармонического анализа. СПб.: Лань, 2012.
2. *Залманзон Л. А.* Преобразование Фурье, Уолша, Хаара и их применение в управлении, связи и других областях. М.: Наука, 1989.
3. *Беспалов М. С.* Собственные подпространства дискретного преобразования Уолша // Проблемы передачи информации. 2010. Т. 46. № 3. С. 60–79.
4. *Моррис С.* Двойственность Понтрягина и строение локально компактных абелевых групп. М.: Мир, 1980.
5. *Трахман А. М., Трахман В. А.* Основы теории дискретных сигналов на конечных интервалах. М.: Сов. радио, 1975.
6. *Беспалов М. С., Скляренко В. А.* Дискретные функции Уолша и их приложения. Владимир: ВлГУ. 2014.
7. *Беспалов М. С.* Новая нумерация матриц Уолша // Проблемы передачи информации. 2009. Т. 45. № 4. С. 43–53.
8. *Питерсон У., Уэлдон Э.* Коды, исправляющие ошибки. М.: Мир, 1976.
9. *Беспалов М. С.* Дискретное преобразование Крестенсона // Проблемы передачи информации. 2010. Т. 46. № 4. С. 91–115.

УДК 519.7 DOI 10.17223/2226308X/13/37

ПРИМЕНЕНИЕ ИНВЕРСНЫХ ЛАЗЕЕК ДЛЯ ПОСТРОЕНИЯ АТАК ИЗ КЛАССА «УГАДЫВАЙ И ОПРЕДЕЛЯЙ» НА ХЕШ-ФУНКЦИИ СЕМЕЙСТВА MD4[1]

И. А. Грибанова, А. А. Семёнов

Приведены новые атаки из класса «угадывай и определяй» для хеш-функций вида MD4-k, $k > 39$. Описываемые атаки основаны на концепции инверсной лазейки. Для решения задач криптоанализа, ослабленных подстановками угадываемых бит, используются SAT-решатели. Задача поиска инверсной лазейки, обеспечивающей атаку с относительно малой трудоёмкостью, ставится в форме задачи минимизации специальной псевдобулевой функции. Для её решения используются три метаэвристических алгоритма: алгоритм поиска с запретами, (1+1)-FEA_β и специальный вариант генетического алгоритма. Перечисленные алгоритмы дают атаки на рассматриваемые функции с близкими оценками трудоемкости. Для функции сжатия полнораундового MD4 лучшие атаки строит генетический алгоритм.

Ключевые слова: *задача поиска прообразов криптографической хеш-функции, атаки из класса «угадывай и определяй», инверсные лазейки, SAT.*

1. О понятии инверсной лазейки

Понятие инверсной лазейки (Inverse Backdoor Set, IBS) введено в [1]. Кратко напомним его суть. Рассматривается задача обращения (поиска прообразов) произвольной функции вида

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

заданной программой (алгоритмом) M_f. Более точно, требуется по произвольному $\gamma \in$ $\in \operatorname{Range} f$ найти такой $\alpha \in \{0,1\}^n$, что $f(\alpha) = \gamma$. Подход к решению данной задачи,

[1]Работа выполнена при финансовой поддержке Российского научного фонда, проект № 16-11-10046. Грибанова И. А. поддержана стипендией Президента РФ СП-3545.2019.5.

используемый далее, относится к алгебраическому криптоанализу [2]. В соответствии с ним задачу поиска прообраза произвольного $\gamma \in \operatorname{Range} f$ можно свести к решению системы алгебраических уравнений над GF(2) либо к поиску набора, выполняющего некоторую выполнимую КНФ. Далее нам потребуется понятие шаблонной КНФ (template CNF), введённое в [3]. Шаблонная КНФ C_f строится по представлению функции f в виде схемы G_f из функциональных элементов с n входами и m выходами над произвольным полным базисом, например над $\{\wedge, \neg\}$. Для перехода от схемы G_f к C_f используются преобразования Цейтина [4].

Пусть U — множество всех булевых переменных, присутствующих в C_f; $X = \{x_1, \ldots, x_n\}$ — переменные, которые приписаны входу схемы G_f. Используем понятие подстановки произвольного значения $\lambda \in \{0,1\}$ произвольной переменной $u \in U$ в формулу C_f. Это понятие дается стандартным образом — например, в соответствии с [5]: то есть каждое вхождение переменной u в C_f заменяется на λ, после чего выполняются все возможные элементарные преобразования. В результате таких преобразований ряд не означенных ранее переменных могут принять конкретные значения. Будем говорить про такие значения, что они индуцированы соответствующей подстановкой.

Пусть $\alpha \in \{0,1\}^n$ — произвольный набор значений переменных из X. Как показано в [3], подстановка α в C_f индуцирует набор значений всех переменных из $U \setminus X$, в том числе и набор значений $\gamma = (\gamma_1, \ldots, \gamma_m)$ переменных из $Y = \{y_1, \ldots, y_m\}$, которые приписаны выходу схемы G_f. При этом имеет место $f(\alpha) = \gamma$.

Рассмотрим произвольное $B \subseteq U \setminus Y$. Зададим на $\{0,1\}^n$ равномерное распределение и свяжем с выбранным случайно набором $\alpha = (\alpha_1, \ldots, \alpha_n) \in \{0,1\}^n$ индуцированные подстановкой $x_1 = \alpha_1, \ldots, x_n = \alpha_n$ наборы значений переменных из B и из Y, которые обозначим β_α и γ_α соответственно. Обозначим через $C_f[\beta_\alpha/B, \gamma_\alpha/Y]$ КНФ, которая получена из C_f в результате подстановки в неё наборов β_α, γ_α. Зафиксируем некоторое число $t > 0$ и рассмотрим произвольный детерминированный алгоритм A решения SAT. Рассмотрим следующую величину:

$$\rho_B(t) = \frac{\#\{\alpha \in \{0,1\}^n : A(C_f[\beta_\alpha/B, \gamma_\alpha/Y]) \leqslant t\}}{2^n}. \tag{2}$$

В числителе (2) стоит число таких $\alpha \in \{0,1\}^n$, что время нахождения алгоритмом A набора, выполняющего $C_f[\beta_\alpha/B, \gamma_\alpha/Y]$, не превосходит t; в знаменателе — общее число различных α. Таким образом, $\rho_B(t)$ — вероятность следующего события: случайное α индуцирует такие β_α, γ_α, что SAT в отношении КНФ $C_f[\beta_\alpha/B, \gamma_\alpha/Y]$ решается алгоритмом A за время $\leqslant t$. Множество B называется инверсной лазейкой с параметрами $(t, \rho_B(t), s)$, где $s = |B|$. В [1] показано, как на основе инверсной лазейки с данными параметрами построить атаку из класса «угадывай и определяй» на криптографическую функцию вида (1), трудоёмкость которой равна

$$T = 2^s \cdot t \left\lceil \frac{3}{\rho_B(t)} \right\rceil.$$

Далее в [1] предлагается рассматривать задачу поиска инверсной лазейки B с относительно малой трудоёмкостью как задачу минимизации специальной функции

$$\Phi : \{0,1\}^n \to \mathbb{R}. \tag{3}$$

Напомним, что функции вида (3) называются псевдобулевыми [6]. Множество B ищется среди всевозможных подмножеств множества X. Функция (3) определяется следующим образом. Предполагается, что произвольный $\delta \in \{0,1\}^n$ задаёт конкретное B:

единицы в δ соответствуют тем и только тем переменным из X, которые входят в B. По произвольному $\delta \in \{0,1\}^n$ строится множество B, после чего генерируется случайная выборка $\alpha^1, \ldots, \alpha^N$, $\alpha^j \in \{0,1\}^n$, $j \in \{1, \ldots, N\}$. Затем наблюдаются N значений случайной величины ξ: для каждого $j \in \{1, \ldots, N\}$ данная величина принимает значение $\xi^j = 1$, если алгоритм A решает SAT для КНФ $C_f[\beta_{\alpha^j}/B, \gamma_{\alpha^j}/Y]$ за время $\leqslant t$, в противном случае $\xi^j = 0$. В роли оценки $\rho_B(t)$ используется величина $\dfrac{1}{N}\sum\limits_{j=1}^{N}\xi^j$. Соответствующее значение функции (3) определяется как

$$\Phi(\delta) = 2^{\mathrm{wt}(\delta)} \cdot t \cdot 3N / \sum_{j=1}^{N} \xi^j, \tag{4}$$

где $\mathrm{wt}(\delta)$ — вес Хэмминга вектора δ.

Заметим, что ξ — случайная величина Бернулли, $\mathrm{M}[\xi] = \rho_B(t)$, $\mathrm{D}[\xi] = \rho_B(t)(1-\rho_B(t))$. Учитывая это и используя неравенство Чебышёва [7], можно показать, что для любого $\varepsilon > 0$ имеет место

$$\mathsf{P}\left[\left|\rho_B(t) - \frac{1}{N}\sum_{j=1}^{N}\xi^j\right| \leqslant \varepsilon\right] \geqslant 1 - \frac{1}{4 \cdot \varepsilon^2 \cdot N},$$

то есть $\dfrac{1}{N}\sum\limits_{j=1}^{N}\xi^j$ позволяет оценить $\rho_B(t)$ с любой наперёд заданной точностью за счёт увеличения объёма выборки N.

2. Алгоритмы поиска инверсных лазеек

Как следует из сказанного выше, имеет смысл искать инверсные лазейки с как можно меньшим значением функции (4). Поскольку функция (4) не задана аналитически, мы можем использовать для её минимизации только эвристические алгоритмы. В настоящей работе использованы следующие алгоритмы: алгоритм из [8], относящийся к классу алгоритмов поиска с запретами [9]; т.н. «быстрый эволюционный алгоритм» (1+1)–FEA_β [10], а также специальный вариант генетического алгоритма [11]. Дадим краткое описание этих алгоритмов.

Алгоритм поиска с запретами (далее — TS от Tabu Search) из [8] — это вариант локального поиска, который хранит все пройденные точки $\{0,1\}^n$ в специальных списках и запрещает повторно вычислять значения функции (4) в точке, в которой эта функция уже вычислялась. Многократное вычисление значений (4) в одних и тех же точках приводит к замедлению поиска, поскольку каждое такое вычисление требует существенного времени. К тому же, как показано в [9], такая стратегия позволяет алгоритму выходить из точек локального минимума (полнота при отсутствии ограничений по памяти).

Алгоритм (1+1)–FEA_β, описанный в [10], представляет собой усложнённый вариант известного эволюционного алгоритма (1+1)–EA [12]. Идея, лежащая в основе (1+1)–FEA_β, состоит в том, чтобы использовать переменную вероятность мутации. В классическом (1+1)–EA вероятность мутации, то есть изменения произвольного бита в рассматриваемом слове $\alpha \in \{0,1\}^n$ на противоположный, постоянна и равна $1/n$. Если α — исходное слово из $\{0,1\}^n$, а α' — результат случайной мутации α в соответствии с (1+1)–EA, то математическое ожидание случайной величины $\mathrm{H}(\alpha, \alpha')$ (расстояния Хэмминга между α и α') есть $\mathrm{M}[\mathrm{H}(\alpha, \alpha')] = 1$. Это свойство очень важно [13], поскольку оно означает, что в среднем данный алгоритм ведёт себя похожим на стандартный локальный поиск образом и соответственно имеет возможность приспосабли-

ваться под «ландшафт» рассматриваемой функции. С другой стороны, этот алгоритм с ненулевой вероятностью переходит в произвольную точку гиперкуба $\{0,1\}^n$. Однако (1+1)–EA имеет крайне плохую верхнюю оценку сложности в смысле меры, введённой в [14], — конкретно, данная оценка имеет вид n^n и, таким образом, (1+1)–EA существенно менее эффективен (в данном смысле), чем простой случайный поиск. В алгоритме (1+1)–FEA_β вероятность мутации зависит от поведения специальным образом определённой случайной величины. В зависимости от значений параметра β алгоритм (1+1)–FEA_β может демонстрировать различные сочетания основных свойств. Наиболее интересным с практических позиций является значение $\beta = 3$, так как в этом случае верхняя оценка сложности (1+1)–FEA_3 в смысле [14] есть $\Theta(n^3 \cdot 2^n)$, притом что $\mathrm{M}[\mathrm{H}(\alpha, \alpha')] \approx \dfrac{\zeta(2)}{\zeta(3)} \approx 1{,}3685$ (здесь ζ — дзета-функция Римана).

Ещё один алгоритм, использованный для минимизации (4), — это специальный случай генетического алгоритма, который описан в [11] (далее — GA от Genetic Algorithm). В данном алгоритме по набору векторов $P = \{\lambda^1, \ldots, \lambda^M\}$, $\lambda^i \in \{0,1\}^n$, $i \in \{1, \ldots, M\}$, строится новый набор $\tilde{P} = \{\tilde{\lambda}^1, \ldots, \tilde{\lambda}^M\}$ в соответствии с несколькими базовыми концепциями теории генетических алгоритмов. Начальный набор из M векторов строится либо случайным образом, либо как результат работы других алгоритмов, например (1+1)–EA. Часть наборов в $\tilde{P}$ состоит из лучших по значению целевой функции элементов P. Другая часть наборов в $\tilde{P}$ есть результат стандартных (1+1)–EA мутаций над несколькими наборами, случайно выбранными из P. Наконец, оставшиеся наборы из $\tilde{P}$ получаются в результате операции двухточечного кроссовера [15] над наборами, случайно выбираемыми из P. Для каждого элемента $\tilde{P}$ вычисляется значение функции (4).

3. Атаки на основе инверсных лазеек на функции вида MD4-k, $k > 39$

Везде далее под MD4-k понимается функция сжатия, задаваемая первыми k шагами известного алгоритма хеширования MD4 [16]. В основе предлагаемых атак лежит идея дополнения уравнений криптоанализа функций вида MD4-k ослабляющими ограничениями. Данная идея высказана Г. Доббертином в [17] и адаптирована к использованию SAT-решателей в [18]. В [19] описан алгоритм, позволяющий генерировать ослабляющие ограничения «типа Доббертина» автоматически. С использованием данного алгоритма построена рекордная по трудоёмкости атака на функцию MD4-39. В дальнейшем при помощи подхода из [19] были построены новые атаки для функций вида MD4-k до $k = 48$ включительно. В частности, атака такого типа на полнораундовую функцию сжатия MD4, представленная в [20], показывает, что данная функция не обладает свойствами случайного оракула. Атаки, описанные в [19 – 21], эксплуатируют общую идею перехода от задачи обращения функции MD4-k к задаче обращения вспомогательных функций вида

$$g^{\lambda}_{\text{MD4-}k} : \{0,1\}^d \to \{0,1\}^{128}, \tag{5}$$

таких, что $d \ll 512$. Через λ в (5) обозначен булев вектор, задающий ослабляющие ограничения «типа Доббертина». Генерируя при помощи алгоритма из [19] различные λ, можно строить нетривиальные атаки на функции вида MD4-k.

В рамках настоящей работы мы использовали описанные в п. 2 метаэвристические алгоритмы для поиска инверсных лазеек в задачах обращения следующих функций:

$$\begin{gathered} g^{\lambda_1}_{\text{MD4-40}} : \{0,1\}^{128} \to \{0,1\}^{128}, \quad g^{\lambda_1}_{\text{MD4-48}} : \{0,1\}^{128} \to \{0,1\}^{128}, \\ g^{\lambda_3}_{\text{MD4-48}} : \{0,1\}^{96} \to \{0,1\}^{128}, \quad g^{\lambda_5}_{\text{MD4-48}} : \{0,1\}^{64} \to \{0,1\}^{128}. \end{gathered} \tag{6}$$

Более подробную информацию о векторах λ_1, λ_3 и λ_5 и перечисленных функциях можно найти в [20]. Заметим, что если $\chi \in \{0,1\}^{128}$ — значение любой из функций (6) и x — прообраз χ в смысле этой функции, то от x можно эффективно перейти к MD4-прообразу χ.

Результаты построения инверсных лазеек описанными алгоритмами и оценки трудоёмкости соответствующих атак из класса «угадывай и определяй» в применении к функциям (6) приведены в таблице.

Алгоритм	$g^{\lambda_1}_{\text{MD4-40}}$	$g^{\lambda_1}_{\text{MD4-48}}$	$g^{\lambda_3}_{\text{MD4-48}}$	$g^{\lambda_5}_{\text{MD4-48}}$
TS 1 ($t = 100$)	$\langle 20\rangle$ 4,4 e+10	$\langle 100\rangle$ 4,5 e+34	$\langle 66\rangle$ 1,2 e+24	$\langle 28\rangle$ 7,8 e+12
TS 2 ($t = 200$)	$\langle 15\rangle$ 2,4 e+9	$\langle 98\rangle$ 2,7 e+34	$\langle 63\rangle$ 7,2 e+23	–
(1+1)–FEA_3 1 ($t = 100$)	$\langle 23\rangle$ 2,8 e+11	$\langle 104\rangle$ 2,1 e+35	$\langle 65\rangle$ 6,7 e+23	$\langle 28\rangle$ 1,1 e+13
(1+1)–FEA_3 2 ($t = 200$)	$\langle 20\rangle$ 9,1 e+10	$\langle 100\rangle$ 1,1 e+35	$\langle 64\rangle$ 5,7 e+23	–
GA 1 ($t = 100$)	$\langle 22\rangle$ 1,7 e+11	$\langle 100\rangle$ 2,0 e+34	$\langle 63\rangle$ 3,6 e+23	$\langle 27\rangle$ 3,5 e+12
GA 2 ($t = 200$)	$\langle 21\rangle$ 1,5 e+11	–	–	–

К о м м е н т а р и и к т а б л и ц е. В первом столбце приведено название алгоритма. В ряде случаев процесс поиска лазейки разбивался на два этапа: на первом этапе использовалось значение $t = 100$ с (см. (4)), затем с лучшей найденной точки запускался этот же алгоритм с параметром $t = 200$ с. В последующих столбцах приведено число переменных в соответствующих лазейках и значения функции (4) для этих лазеек. Каждое такое значение даёт оценку времени выполнения атаки из класса «угадывай и определяй» в секундах для соответствующей функции вида (6) на одном ядре используемого процессора (в нашем случае — на одном ядре AMD Opteron 6276). По результатам экспериментов можно сделать вывод о том, что все сравниваемые алгоритмы дают атаки с близкими оценками трудоёмкости. Для полнораундовой версии функции сжатия MD4 атаки с наименьшей трудоёмкостью строит генетический алгоритм с $M = 10$.

Все вычислительные эксперименты по поиску инверсных лазеек для функций вида (6) проводились на вычислительном кластере «Академик В. М. Матросов» Иркутского суперкомпьютерного центра [22].

ЛИТЕРАТУРА

1. *Semenov A., Zaikin O., Otpuschennikov I., et al.* On cryptographic attacks using backdoors for SAT // Proc. 32nd AAAI Conf. 2018. P. 6641–6648.
2. *Bard G.* Algebraic Cryptanalysis. Springer Publishing Company, Inc., 2009. 356 p.
3. *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.
4. *Цейтин Г. С.* О сложности вывода в исчислении высказываний // Записки научных семинаров ЛОМИ АН СССР. 1968. Т. 8. С. 234–259.
5. *Чень Ч., Ли Р.* Математическая логика и автоматическое доказательство теорем. М.: Наука, 1983. 360 с.
6. *Boros E. and Hammer P. L.* Pseudo-Boolean optimization // Discrete Appl. Math. 2002. V. 123 (1-3). P. 155–225.
7. *Феллер У.* Введение в теорию вероятностей и ее приложения. Т. 1. М.: Мир, 1964. 500 с.
8. *Semenov A. and Zaikin O.* Algorithm for finding partitionings of hard variants of Boolean satisfiability problem with application to inversion of some cryptographic functions // SpringerPlus. 2016. V. 5 (1). P. 1–16.

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.