Будем говорить, что *раундовая функция g сохраняет разбиение* $\mathbf{W}$ *справа*, если $W^{g_k} \in \mathbf{W}$ для всех $(W,k) \in \mathbf{W} \times X$.

Получены ограничения на группы $(X,*)$, $G = \langle b, X^* \rangle$, а также на блоки $W_0, \ldots, W_{r-1}$, вытекающие из условия сохранения раундовой функцией нетривиального разбиения $\mathbf{W}$. Доказано, что $\mathbf{W}$ — система импримитивности группы G. Кроме того, показано, что $(W_0,*)$ — подгруппа группы $(X,*)$, причём W_j — j-й правый смежный класс группы $(X,*)$ по $(W_0,*)$ для $j = 0, \ldots, r-1$.

Из импримитивности группы G следуют включения

$$b \in \mathrm{IG}_{\mathbf{W}}, \quad \langle g_k | k \in X \rangle \leqslant \mathrm{IG}_{\mathbf{W}},$$

где $\mathrm{IG}_{\mathbf{W}}$ — максимальная подгруппа группы $S(X)$, сохраняющая разбиение $\mathbf{W}$.

Если группа $G = \langle b, X^* \rangle$ импримитивна с системой импримитивности $\mathbf{W}$, то существует естественный гомоморфизм $\varphi_{\mathbf{W}} : G \to S(\{0, \ldots, r-1\})$, $1 = \varphi_{\mathbf{W}}(e)$. Доказано, что если $(W_0,*)$ нормальна в $(X,*)$ ($(W_0,*) \lhd (X,*)$), $\mathbf{W}$ — множество всех смежных классов группы $(X,*)$ по $(W_0,*)$, то условие $*_{\mathbf{W}}$-марковости алгоритма $\mathrm{C}_l(*,b)$ эквивалентно существованию у него гомоморфизма, задаваемого отображением $\varphi_{\mathbf{W}}$.

Для криптографических приложений представляют интерес группы порядка 2^m. В [3, теорема 12.5.1] описаны все неабелевы группы порядка 2^m, обладающие циклической подгруппой индекса два. Таких групп всего четыре, включая группу диэдра и обобщённую группу кватернионов. Для всех четырёх групп описаны $*_{\mathbf{W}}$-марковские подстановки из $S(X)$ относительно разбиения $\mathbf{W} = \{W_0, \ldots, W_{r-1}\}$, блоками которого являются все правые смежные классы группы $(X,*)$ по подгруппе $(W_0,*)$, но $(W_0,*) \not\lhd (X,*)$.

ЛИТЕРАТУРА

1. *Lai X., Massey J. L., and Murphy S.* Markov ciphers and differential cryptanalysis // EUROCRYPT 1991. LNCS. 1991. V. 547. P. 17–38.
2. *Погорелов Б. А., Пудовкина М. А.* Разбиения на биграммах и марковость алгоритмов блочного шифрования // Математические вопросы криптографии. 2017. Т. 8. № 1. С. 107–142.
3. *Холл М.* Теория групп. М.: ИЛ, 1962. 468 с.

УДК 519.7 DOI 10.17223/2226308X/11/26

АТАКИ ИЗ КЛАССА «УГАДЫВАЙ И ОПРЕДЕЛЯЙ» И АВТОМАТИЧЕСКИЕ СПОСОБЫ ИХ ПОСТРОЕНИЯ[1]

А. А. Семёнов

Представлен краткий обзор подходов к построению криптографических атак, относящихся к классу «угадывай и определяй». Основной акцент сделан на относительно недавних работах, в которых описаны автоматические способы построения таких атак с ипользованием алгоритмов решения проблемы булевой выполнимости (SAT). С этой целью задачи построения атак из рассматриваемого класса ставятся как задачи оптимизации на булевом гиперкубе специальных оценочных функций. Для решения последних используются метаэвристические алгоритмы, широко применяемые в дискретной оптимизации. В упомянутых работах введены два типа оценочных функций, которые можно рассматривать как конкретиза-

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

ции понятий «UNSAT-иммунность» и «SAT-иммунность», неформально введённых Н. Куртуа в 2012 г. Приведены примеры построения атак указанного типа для ряда блочных и поточных алгоритмов шифрования.

Ключевые слова: *атаки из класса «угадывай и определяй», проблема булевой выполнимости, SAT.*

Метод «угадывай и определяй» (guess-and-determine) — это общая техника криптоанализа, применимая к обширному множеству криптографических функций. На сегодняшний день трудно перечислить все опубликованные атаки, относящиеся к данному классу (соответствующие работы исчисляются, по-видимому, сотнями).

В данном обзоре рассмотрены нетривиальные атаки из класса «угадывай и определяй», которые удаётся строить при помощи алгоритмов решения проблемы булевой выполнимости в отношении блочных и поточных шифров. Во всех случаях рассматриваются атаки, использующие «Known plaintext scenario», то есть предполагается, что криптоаналитику известны произвольные открытые тексты и соответствующие криптограммы. Требуется вычислить неизвестный секретный ключ. В отношении поточных шифров в таких случаях мы говорим о задаче восстановления секретного ключа по известному фрагменту ключевого потока. Данную задачу можно рассматривать как задачу обращения (всюду определённой) дискретной функции вида

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

то есть функции, которая преобразует двоичные слова длины n в двоичные слова длины m. Относительно функции (1) будем полагать, что она задаётся некоторым известным алгоритмом. По произвольному $y \in \operatorname{Range} f$ требуется найти произвольный такой $z \in \{0,1\}^n$, что $f(z) = y$.

В случае блочных шифров рассматриваются (всюду определённые) функции вида

$$F : \{0,1\}^l \times \{0,1\}^n \to \{0,1\}^m, \tag{2}$$

где $\{0,1\}^l$ — множество открытых текстов; $\{0,1\}^n$ — множество секретных ключей; $\operatorname{Range} F$ — множество криптограмм ($\operatorname{Range} F \subseteq \{0,1\}^m$). В отношении функций вида (2) рассматривается следующая задача: при известных $y \in \operatorname{Range} F$ и $x \in \{0,1\}^l$, таких, что $y = F(x,z)$ для некоторого $z \in \{0,1\}^n$, найти этот z.

Для криптоанализа интерес представляют ситуации, когда число n фиксировано, а числа l и m могут варьироваться — в таких случаях говорят о поиске секретного ключа на основе анализа соответствующего количества шифртекста (и открытого текста). При фиксации l и m, как правило, исходят из требования решить задачу нахождения именно того ключа, который использовался для построения y. В подавляющем большинстве случаев для решения данной задачи требуется анализ шифртекста, сопоставимого по длине с искомым ключом. Везде далее мы придерживаемся этого правила, хоть оно (в целом) и является эмпирическим и не избавляет от возможности найти эквивалентный, а иногда и ложный ключ (можно предъявить соответствующие примеры).

Далее будем полагать, что функции вида (1) и (2) задаются детерминированными алгоритмами. Примем здесь без доказательства тот факт, что по этим алгоритмам могут быть эффективно построены схемы из функциональных элементов над базисом $\{\wedge, \neg\}$, реализующие эти функции; будем обозначать их через $S(f)$ и $S(F)$ соответственно. У схемы $S(f)$ имеется n входов и m выходов, у $S(F)$ — $l + n$ входов и

m выходов. Поясним необходимые понятия на задаче обращения функции вида (1) (на функции вида (2) они распространяются естественным образом).

Итак, рассматриваем схему $S(f)$. Припишем каждому узлу схемы $S(f)$ булеву переменную, полученное множество переменных обозначим через V. Преобразования Цейтина [1] эффективно (за линейное от размера $S(f)$ время) сопоставляют схеме $S(f)$ конъюнктивную нормальную форму (КНФ) над множеством переменных V. Обозначим полученную КНФ через $C(f)$. Выделим в V множество $Z = \{z_1, \ldots, z_n\}$, образованное переменными, приписанными входам схемы $S(f)$. Через $Y = \{y_1, \ldots, y_m\}$ обозначим множество переменных, приписанных выходам $S(f)$. Пусть $v \in V$ — произвольная булева переменная. Далее используется обозначение v^σ, введенное в [2]:

$$v^\sigma = \begin{cases} \overline{v}, \ \sigma = 0, \\ v, \ \sigma = 1. \end{cases}$$

Рассмотрим произвольный набор $z = (\alpha_1, \ldots, \alpha_n)$, $\alpha_i \in \{0, 1\}$, $i = \{1, \ldots, n\}$. Построим КНФ

$$C(z, f) = z_1^{\alpha_1} \wedge \cdots \wedge z_n^{\alpha_n} \wedge C(f). \tag{3}$$

Будем применять к КНФ (3) правило единичной дизъюнкции (Unit Propagation rule, UP) [3]. Итеративное применение данного правила порождает формулы вида v^σ для переменных $v \in V \setminus Z$. Можно показать, что результатом не более чем M-кратного применения правила UP к КНФ (3) будет множество формул $\Psi_z = \{v^\sigma\}_{v \in V}$ (по всем $v \in V$), где M — общее число вхождений переменных из V в (3). Рассмотрим множество $\{y_1^{\beta_1}, \ldots, y_m^{\beta_m}\} \subset \Psi_z$. В силу сделанных предположений оказывается, что $f(z) = y$, где $y = (\beta_1, \ldots, \beta_m)$.

Определение 1. Будем говорить, что произвольное означивание переменных из множества Z в КНФ $C(f)$ выводит значения всех переменных, входящих в $V \setminus Z$, по правилу единичной дизъюнкции.

Пусть $y = (\beta_1, \ldots, \beta_m)$ — набор значений переменных из Y, такой, что $y \in \operatorname{Range} f$. Рассмотрим КНФ

$$C(f, y) = C(f) \wedge y_1^{\beta_1} \wedge \ldots \wedge y_m^{\beta_m}. \tag{4}$$

Используя идеи С. А. Кука, изложенные им при доказательстве базовой теоремы теории NP-полноты [4], можно показать, что при всех перечисленных условиях КНФ (4) выполнима и из любого выполняющего её набора можно эффективно извлечь такой $z \in \{0, 1\}^n$, что $f(z) = y$.

Пусть $z = (\alpha_1, \ldots, \alpha_n)$ — произвольный набор значений переменных из Z. Применим итеративно правило UP к КНФ

$$z_1^{\alpha_1} \wedge \ldots \wedge z_n^{\alpha_n} \wedge C(f, y). \tag{5}$$

Обозначим полученное в результате множество формул вида v^σ через $\Psi_{(z,y)}$. В силу сказанного выше, при $f(z) \neq y$ множество $\Psi_{(z,y)}$ содержит формулы $y_r^{\beta_r}$ и $y_r^{\overline{\beta_r}}$ для некоторого $r \in \{1, \ldots, m\}$. Будем говорить, что в этом случае множество $\Psi_{(z,y)}$ противоречиво. Если $f(z) = y$, то множество $\Psi_{(z,y)}$ не является противоречивым. В соответствующих случаях КНФ вида (5) также будем называть противоречивой или непротиворечивой.

По аналогии с КНФ вида $C(f)$ по схеме $S(f)$ может быть эффективно построена система квадратичных уравнений над полем GF(2). Для данной системы также

можно определить аналог процедуры означивания переменных ключа, понятие эффективного вывода из системы значений всех входящих в неё переменных (в том числе переменных, кодирующих выход рассматриваемой функции), а также понятие противоречивости (непротиворечивости). Естественным образом определяется и система уравнений, являющаяся аналогом КНФ вида (4).

Везде далее под функцией Ф понимаются функции вида (1) или (2). Пусть $E(\Phi)$ — КНФ вида (4) либо аналогичная ей (в указанном смысле) система алгебраических уравнений (в целях единообразия КНФ будем рассматривать как систему булевых уравнений). Для многих стойких шифров (таких, например, как полнораундовый AES) поиск решения системы $E(\Phi)$ — крайне сложная задача, которая на сегодняшний день не под силу ни одному из известных алгоритмов. Однако в множестве переменных, встречающихся в $E(\Phi)$, всегда можно указать такое подмножество, означивание которых делает систему $E(\Phi)$ существенно более простой. Простейший пример — переменные, соответствующие секретному ключу: если означить их все (например, угадав каким-либо образом соответствующие значения), то полученная система решится тривиальным образом. Можно перебирать все возможные варианты значений ключа и проверять получаемые системы уравнений на противоречивость. Случай непротиворечивой системы соответствует нахождению верного ключа. Конечно, описанная атака является предельно неэффективной, поскольку представляет по своей сути атаку методом грубой силы.

Для целого ряда шифров удаётся найти множество $B \subset V$, обладающее следующими свойствами:

1) $|B| = s$, $s < k$, где k — длина секретного ключа;
2) $2^s \cdot T_A \ll 2^k \cdot T_0$.

Здесь T_0 — время опробования одного набора значений переменных, соответствующих секретному ключу (иными словами, $2^k \cdot T_0$ — это время атаки методом грубой силы); T_A — некоторая оценка времени работы алгоритма A, который решает систему, полученную из $E(\Phi)$ в результате подстановки набора значений переменных из множества B. Если справедливы свойства п. 1–2, то говорят, что имеет место атака типа «угадывай и определяй» (guess-and-determine) на основе множества угадываемых бит (guessed bits) B.

Как было сказано выше, атаки данного типа образуют один из наиболее многочисленных классов. Выбор алгоритма A очень часто зависит от особенностей рассматриваемой криптографической функции. Например, для широко известного алгоритма A5/1 с длиной ключа 64 бита Р. Андерсеном ещё в 1994 г. описана атака из класса «угадывай и определяй», в которой используется множество B, состоящее из 53 переменных [5]. Алгоритм A в этом случае простейший — фактически, подстановка значений переменных из B в систему уравнений, описывающую генерацию ключевого потока; результирующая система становится тривиально разрешимой. В дальнейшем атака Андерсона была реализована как с применением специально спроектированной для этой цели ПЛИС-архитектуры [6], так и в распределённой среде, использующей общедоступные GPU [7].

Известно довольно много примеров атак типа «угадывай и определяй», в которых алгоритм A — это алгоритм решения систем линейных алгебраических уравнений (СЛАУ). Один из наиболее известных примеров данного типа — атака на генератор A5/1, описанная Й. Голичем в 1997 г. [8]. В [9] приведены примеры аналогичных по смыслу атак для целого ряда генераторов ключевого потока. В [9, 10] ставятся

и решаются задачи поиска *линеаризационных множеств*, являющихся множествами типа B в предположении, что A — алгоритм решения СЛАУ.

Алгоритм A, однако, не обязан быть полиномиальным. Он должен быть эффективен на достаточно широком классе частных случаев массовой задачи, для решения которой предназначен. Один из наиболее известных примеров такого рода дают алгоритмы решения систем уравнений второй степени над полем GF(2) и базирующиеся на них техники криптоанализа [11]. Ещё один хороший пример — алгоритмы решения проблемы булевой выполнимости (SAT), основанные на концепции CDCL (Conflict-Driven Clause Learning) [12]. Потенциал CDCL применительно к задачам криптоанализа отмечен в ряде работ середины 2000-х годов. В частности, в [13] предъявлен основанный на использовании SAT-подхода эффективный алгоритм поиска коллизий полнораундовой хеш-функции MD4. В [14] с использованием SAT-подхода решены задачи обращения неполнораундовых вариантов MD4 до 39 шагов включительно. В [15] представлена основанная на SAT-подходе атака типа «угадывай и определяй» на алгоритм A5/1, которая была реализована в распределённой вычислительной среде. Эти результаты впервые опубликованы на русском языке в [16] и явились (наряду с [6]) первыми примерами выполненного за реальное время криптоанализа неослабленного A5/1. Ряд атак из класса «угадывай и определяй», использующих алгоритмы решения SAT, описан в книге Г. Барда по алгебраическому криптоанализу [17].

В большинстве случаев известные атаки типа «угадывай и определяй» являются результатом скрупулёзного анализа особенностей рассматриваемого шифра. В работах [18, 19] описан метод автоматического поиска декомпозиционных представлений SAT-задач, а также продемонстрировано применение данного метода к криптоанализу некоторых генераторов ключевого потока. Уже после выхода работ [18, 19] стало понятно, что описанные в этих работах алгоритмы можно использовать для автоматического построения атак типа «угадывай и определяй» применительно к широкому классу шифров. Основная идея, использованная в [18, 19], заключается в рассмотрении задачи построения хорошей атаки в форме проблемы оптимизации специальной оценочной функции. Данная оценочная функция является конкретизацией неформально введённого Н. Куртуа понятия «UNSAT-иммунность» [20, 21].

В [22] описан новый класс атак типа «угадывай и определяй», основанных на понятии «инверсной лазейки» (Inverse Backdoor Set, IBS). IBS можно рассматривать как специальный случай «строгих лазеек» (Strong Backdoor Set), введённых в [23]. В [22] предложена автоматическая процедура поиска инверсных лазеек, использующая метаэвристическую оптимизацию специальной оценочной функции. Саму эту оценочную функцию можно рассматривать как далёкое развитие неформально введённого Н. Куртуа понятия «SAT-иммунность» [20, 21]. В [22] с использованием IBS построены атаки типа «угадывай и определяй», оценки трудоёмкости которых для ряда шифров существенно ниже всех известных близких по смыслу оценок.

ЛИТЕРАТУРА

1. *Цейтин Г. С.* О сложности вывода в исчислении высказываний // Записки научных семинаров ЛОМИ АН СССР. 1968. Т. 8. С. 234–259.
2. *Яблонский С. В.* Введение в дискретную математику. М.: Наука, 1986.
3. *Dowling W. and Gallier J.* Linear-time algorithms for testing the satisfiability of propositional horn formulae // J. Logic Programming. 1984. V. 7. P. 267–284.
4. *Cook S. A.* The complexity of theorem-proving procedures // Third Ann. ACM Symp. on Theory of Computing. 1971. P. 151–159.

5. *Anderson R.* A5 (Was: Hacking Digital Phones). Newsgroup Communication. 1994. `http://yarchive.net/phone/gsmcipher.html`
6. *Gendrullis T., Novotny M., and Rupp A.* A real-world attack breaking A5/1 within hours // LNCS. 2008. V. 5154. P. 266–282.
7. *Bulavintsev V., Semenov A., Zaikin O., and Kochemazov S.* A bitslice implementation of Anderson's attack on A5/1 // Open Eng. 2018. V. 8. P. 7–16.
8. *Golic J. Dj.* Cryptanalysis of alleged A5 stream cipher // LNCS. 1997. V. 1233. P. 239–255.
9. *Агибалов Г. П.* Логические уравнения в криптоанализе генераторов ключевого потока // Вестник Томского государственного университета. Приложение. 2003. № 6. С. 31–41.
10. *Тимошевская Н. Е.* Задачи о кратчайшем линеаризационном множестве // Вестник Томского государственного университета. Приложение. 2005. № 4. С. 79–83.
11. *Courtois N. and Pieprzyk J.* Cryptanalysis of block ciphers with overdefined systems of equations // LNCS. 2003. V. 2501. P. 267–287.
12. *Marques-Silva J., Lynce I., and Malik S.* CDCL Solvers. Handbook of Satisfiability. IOS Press, 2009. P. 131–153.
13. *Mironov I. and Zhang L.* Applications of SAT solvers to cryptanalysis of hash functions // LNCS. 2006. V. 4121. P. 102–115.
14. *De D., Kumarasubramanian A., and Venkatesan R.* Inversion attacks on secure hash functions using SAT solvers // LNCS. 2007. V. 4501. P. 377–382.
15. *Semenov A., Zaikin O., Bespalov D., and Posypkin M.* Parallel logical cryptanalysis of the generator A5/1 in BNB-Grid system // LNCS. 2011. V. 6873. P. 473–483.
16. *Посыпкин М. А., Заикин О. С., Беспалов Д. В., Семенов А. А.* Решение задач криптоанализа поточных шифров в распределенных вычислительных средах // Труды ИСА РАН. 2009. № 46. С. 119–137.
17. *Bard G.* Algebraic Cryptanalysis. Springer, 2009.
18. *Semenov A. and Zaikin O.* Using Monte Carlo method for searching partitionings of hard variants of Boolean satisfiability problem // LNCS. 2015. V. 9251. P. 222–230.
19. *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. No. 1. P. 1–16.
20. *Courtois N. T., Gawinecki J. A., and Song G.* Contradiction immunity and guess-then-determine attacks on GOST // Tatra Mountains Mathematical Publications. 2012. V. 53. No. 1. P. 2–13.
21. *Courtois N. T.* Algebraic Complexity Reduction and Cryptanalysis of GOST. Preprint. 2010–2013. `https://eprint.iacr.org/2011/626.pdf`
22. *Semenov A., Zaikin O., Otpuschennikov I., et al.* On cryptographic attacks using backdoors for SAT // The Thirty-Second AAAI Conf. on Artificial Intelligence, AAAI'2018. New Orleans, Louisiana, USA, 2018. P. 6641–6648.
23. *Williams R., Gomes C. P., and Selman B.* Backdoors to typical case complexity // Proc. of IJCAI. Acapulco, Mexico, 2003. P. 1173–1178.