УДК 519.7

ТЕХНОЛОГИЯ КРУПНОБЛОЧНОГО ПАРАЛЛЕЛИЗМА В SAT-ЗАДАЧАХ[1]

О.С. Заикин, А.А. Семенов

Институт динамики систем и теории управления СО РАН, г. Иркутск

Предложен новый подход к решению SAT-задач, основанный на концепции крупноблочного параллелизма, свойственного многочисленным задачам большой размерности. В рамках данного подхода строится декомпозиция исходной конъюнктивной нормальной формы (КНФ) на семейство КНФ с последующим решением SAT-задачи для каждой КНФ полученного семейства на отдельном вычислительном узле кластера. Планирование оптимального по трудоемкости вычисления осуществляется через решение задачи оптимизации специальной прогнозной функции. Эффективность подхода подтверждена на примере задач криптоанализа суммирующего и порогового генераторов.

ВВЕДЕНИЕ

В последнее время отмечен реальный прогресс в построении программных решателей в задачах поиска на булевых структурах большой размерности (далее используется термин «логический поиск»). Наиболее известные задачи логического поиска — это так называемые SAT-задачи (boolean satisfiability problem — или просто SAT), прототипом которых служит задача о выполнимости произвольной конъюнктивной нормальной формы (КНФ). К SAT-задачам сводятся многие практически важные проблемы: задачи синтеза и верификации дискретных управляющих систем, задачи теоретического программирования, а также многочисленные криптографические задачи.

Высокая вычислительная сложность SAT-задач делает их привлекательными объектами с позиций теории параллельных вычислений. Далее мы описываем подход к решению SAT-задач, использующий концепцию крупноблочного параллелизма. Особо отметим, что одной из первых работ по крупноблочному распараллеливанию вычислительных задач большой размерности была статья [1].

[1] Работа выполнена при поддержке РФФИ, гранты № 07-01-00400-а и НШ-9508.2006.1 (государственная поддержка ведущих научных школ).

В основе предлагаемой в настоящей работе технологии лежит декомпозиция исходной SAT-задачи на семейство SAT-задач, в результате которой область поиска разбивается на непересекающиеся подобласти. Проблема выбора оптимальных параметров декомпозиции решается с помощью процедуры статистического прогнозирования, представляющей собой последовательность статистических экспериментов, результатом каждого из которых служит некоторая случайная выборка КНФ. На множестве таких выборок определяется специальная прогнозная функция. Прогноз минимального времени параллельного решения SAT-задачи строится на основе знания глобального минимума прогнозной функции на рассматриваемом множестве случайных выборок КНФ.

1. ОСНОВНЫЕ ОПРЕДЕЛЕНИЯ

Переменные, принимающие значения в множестве $\{0, 1\}$, называются логическими или булевыми. Пусть $X = \{x_1, ..., x_n\}$ — множество булевых переменных, термы x_i и $\bar{x}_i$, $i \in \{1, ..., n\}$, называются литералами над X (через $\bar{x}$ обозначается логическое отрицание x). Литералы x и $\bar{x}$ называются контрарными. Дизъюнктом над X называется дизъюнкция различных литералов, среди которых нет контрарных. Конъюнктивной нормальной

формой над X называется конъюнкция различных дизъюнктов над X. Если A — произвольная формула, реализующая некоторую функцию алгебры логики над X [2], то запись «$A|_{(\alpha_1, ..., \alpha_n)} = \beta$, $\beta \in \{0, 1\}$», означает, что A принимает значение β при подстановке $x_1 = \alpha_1$, ..., $x_n = \alpha_n$, $\alpha_i \in \{0, 1\}$, $i \in \{1, ..., n\}$.

Пусть C — КНФ над множеством булевых переменных $X = \{x_1, ..., x_n\}$, двоичный вектор $(\alpha_1, ..., \alpha_n)$ называется набором, выполняющим C, если $C|_{(\alpha_1, ..., \alpha_n)} = 1$. Конъюнктивная нормальная форма над X, для которой существует выполняющий набор, называется выполнимой, в противном случае — невыполнимой. К SAT-задачам относится задача распознавания выполнимости произвольной КНФ, а также задача поиска выполняющего набора произвольной выполнимой КНФ. Данные задачи являются вычислительно трудными — задача распознавания выполнимости произвольной КНФ NP-полна [3].

Для решения SAT-задач применяются специализированные программы, называемые SAT-решателями. Характерные особенности архитектуры современных эффективных SAT-решателей приведены в обзоре [4].

Пусть $L(x_1, ..., x_n)$ — произвольная формула, реализующая некоторую функцию алгебры логики от булевых переменных $x_1, ..., x_n$. Выражения вида $L(x_1, ..., x_n) = 0$, $L(x_1, ..., x_n) = 1$ называются логическими уравнениями. Решить логическое уравнение $L(x_1, ..., x_n) = \beta$, $\beta \in \{0, 1\}$, означает найти такой набор $(\alpha_1, ..., \alpha_n)$, $\alpha_i \in \{0, 1\}$, $i \in \{1, ..., n\}$,что $L(x_1, ..., x_n)|_{(\alpha_1, ..., \alpha_n)} = \beta$. Если такого набора не существует, то говорят, что логическое уравнение не имеет решений.

Формальной вычислительной моделью, рассматриваемой далее, служит машина Тьюринга с входным алфавитом $\Sigma = \{0, 1\}$. Через $\{0, 1\}^n$, $n \in N$, обозначается множество всех двоичных последовательностей длины n. Также используется обозначение $\{0, 1\}^* = \bigcup\limits_{n \in N} \{0, 1\}^n$. Дискретной функцией называется произвольная (вообще говоря, частичная) функция вида f_n: $\{0, 1\}^n \to \{0, 1\}^*$, $n \in N$. Через $dom\, f_n \subseteq \{0, 1\}^n$ обозначается область определения функции f_n, а через $range\, f_n \subset \{0, 1\}^*$ ее область значений. Дискретную функцию f_n назовем всюду определенной или кратко «тотальной», если $dom\, f_n = \{0, 1\}^n$. Пусть M — программа машины Тьюринга (МТ-программа), которая останавливается на произвольном слове из $\{0, 1\}^*$, причем в заключительном состоянии на ленте записано некоторое двоичное слово. Очевидно, что данная программа вычисляет семейство тотальных дискретных функций, которое обозначим через $f^M = \{f_n^M\}_{n \in N}$. Под алгоритмически вычислимыми семействами тотальных дискретных функций понимаются именно семейства вида f^M. Далее индекс M в обозначении таких семейств опускается. Вычислительная сложность МТ-программы, останавливающейся на произвольном двоичном входе, определяется стандартным образом [5].

2. КРУПНОБЛОЧНЫЙ ПАРАЛЛЕЛИЗМ В SAT-ЗАДАЧАХ

Высокая вычислительная сложность SAT-задач придает принципиальное значение проблеме использования для их решения параллельных вычислительных технологий. В данном контексте SAT-задачи могут исследоваться по двум направлениям: построение SAT-решателей с параллельной вычислительной архитектурой; общая концепция «распараллеливания по данным», когда область поиска разбивается на непересекающиеся подобласти, каждая из которых обрабатывается SAT-решателем на отдельном вычислительном узле. Первое направление представляется технически труднореализуемым, поскольку SAT-решатели представляют собой довольно сложные программы в плане многообразия используемых в них технологий. В настоящей работе развивается второе направление. Результатом является описанная ниже параллельная технология решения SAT-задач на вычислительных кластерах. Ключевой момент этой технологии состоит в моделировании параметров параллельного вычисления, оптимального по временным затратам.

Опишем используемую далее общую схему крупноблочного распараллеливания SAT-задач. Рассматривается произвольная КНФ C над множеством булевых переменных $X = \{x_1, ..., x_n\}$, состоящая из m дизъюнктов. Выбираем в множестве X некоторое подмножество

$$X^d = \{x_{i_1}, ..., x_{i_d}\}, \{i_1, ..., i_d\} \subseteq \{1, ..., n\}, d \in N.$$

Через $Y_1, ..., Y_k$, $k = 2^d$, обозначим различные двоичные векторы длины d, каждый из которых рассматривается как набор значений булевых переменных из множества X^d; используем обозначение $Y^d = \{Y_1, ..., Y_k\}$. Каждому вектору $Y_j \in Y^d$, $j \in \{1, ..., k\}$, поставим в соответствие КНФ $C_j = C|_{Y_j}$, полученную подстановкой в C значений перемен-

ных x_{i_v} $v \in \{1, ..., d\}$, из вектора Y_j. В результате имеем семейство КНФ $\Delta_d(C) = \{C_1, ..., C_k\}$, которое называем семейством, порожденным из C множеством X^d. Дополнительно полагаем, что $X^0 = \varnothing$ и семейство, порожденное из C множеством X^0, состоит из единственной КНФ C.

Пусть семейство КНФ $\Delta_d(C) = \{C_1, ..., C_k\}$ порождено из КНФ C множеством X^d ($k = 2^d$). Несложно видеть, что всякому набору, выполняющему исходную КНФ C, соответствует набор, выполняющий некоторую КНФ из семейства $\Delta_d(C)$. Наоборот, произвольному набору, выполняющему некоторую КНФ из $\Delta_d(C)$, соответствует единственный набор, выполняющий КНФ C. Следовательно, C выполнима тогда и только тогда, когда выполнима хотя бы одна КНФ семейства $\Delta_d(C)$. Таким образом, решение исходной SAT-задачи для КНФ C сводится к решению k SAT-задач для КНФ $C_1, ..., C_k$.

Пусть имеется кластер, состоящий из r процессоров, где $r \in N$. Возможны следующие два случая.

- $k \leqslant r$ — число КНФ в семействе, порожденном X^d, не превосходит числа процессоров кластера. В этом случае для каждой КНФ из семейства $\Delta_d(C)$ SAT-задача решается на отдельном процессоре кластера.
- $k > r$ — число КНФ больше числа процессоров кластера. В данном случае каждому вектору Y_j, $j \in \{1, ..., k\}$, ставится в соответствие натуральное число N_j, двоичное представление которого вектор Y_j. Данное число назовем натуральным индексом КНФ C_j. Семейство КНФ $\Delta_d(C)$ упорядочивается по возрастанию их натуральных индексов. Произвольную КНФ из $\Delta_d(C)$ назовем связанной, если в рассматриваемый момент времени SAT-задача для нее либо уже решена, либо решается на некотором процессоре кластера. Остальные КНФ называем свободными. Выбираются первые r КНФ $C_1, ..., C_r$ из семейства $\Delta_d(C)$. Для каждой КНФ $C_1, ..., C_r$ решается SAT-задача на отдельном процессоре кластера. Как только освобождается некоторый из r процессоров кластера, на нем запускается процедура решения SAT-задачи для первой (в смысле введенного выше порядка) свободной КНФ семейства $\Delta_d(C)$. Данный процесс продолжается до тех пор, пока не будет найден выполняющий набор некоторой КНФ из $\Delta_d(C)$ либо пока не будет доказана невыполнимость всех КНФ из $\Delta_d(C)$. В силу изложенного выше, описанная процедура решает SAT-задачу для произвольной КНФ C корректно (т. е. находит набор, выполняющий C, либо доказывает ее невыполнимость).

3. ПЛАНИРОВАНИЕ ВЫЧИСЛИТЕЛЬНОГО ЭКСПЕРИМЕНТА. ОПТИМИЗАЦИЯ ПРОГНОЗНОЙ ФУНКЦИИ ВРЕМЕНИ РЕШЕНИЯ

Случай, когда число КНФ в семействе, порожденном множеством X^d, не превосходит числа процессоров, тривиален. Как правило, для относительно маломощных кластеров такое распараллеливание SAT-задачи мало что дает, поскольку получаемые КНФ остаются очень сложными для SAT-решателей.

Гораздо больший интерес представляет ситуация, когда $k > r$. Подстановка значительного числа переменных может дать КНФ, которые существенно проще для SAT-решателя, чем исходная, однако общее число таких КНФ ($k = 2^d$) может потребовать для полного решения задачи нереализуемого перебора.

Выход из данной ситуации состоит в прогнозировании параметров оптимального (в смысле вычислительной трудоемкости) разбиения области поиска. Процедура прогнозирования представляет собой последовательность случайных тестов, результатом которых служит специальная прогнозная функция. Термином «прогнозная функция» мы обозначаем семейство функций, параметрически зависящих от выбранного SAT-решателя. Аргументами данных функций являются случайные выборки КНФ. Прогнозная функция является частично определенной на заданном множестве выборок и каждой «точке» своей области определения ставит в соответствие некоторое положительное рациональное число. Знание глобального минимума данной функции на ее области определения позволяет получить представление (прогноз) относительно минимального общего объема вычислений, требуемого для решения распараллеленной исходной задачи.

Введем в рассмотрение натуральное число R_0, предназначенное для разделения ситуаций — когда есть необходимость формирования случайной выборки и когда такой необходимости нет. Например, за R_0 можно принять число, близкое к числу процессоров в кластере. Если при некотором разбиении области поиска мощность семейства $\Delta_d(C)$ слишком велика, то представление о времени соответствующего параллельного вычисления можно составить на основе знания среднего времени решения SAT-задач для серии КНФ, выбранных случайным образом из $\Delta_d(C)$. Через q_d обозначаем объем такой выборки.

Каждому значению параметра $d \in \{0, 1, ..., n\}$ такому, что $2^d > R_0$ ставится в соответствие множест-

во векторов $\{Y_{j_1}, ..., Y_{j_{qd}}\}$, выбираемых из Y^d в соответствии с равномерным распределением, а также множество (выборка) КНФ $\Theta_d = \{C_{j_1} = C|_{Y_{j_1}}, ..., C_{j_{q_d}} = C|_{Y_{j_{q_d}}}\}$. Каждому значению параметра $d \in \{0, 1, ..., n\}$ такому, что $2^d \leqslant R_0$, ставится в соответствие множество Y^d и множество КНФ $\Theta_d = \{C_1 = C|_{Y_1}, ..., C_{2^d} = C|_{Y_{2^d}}\}$ (в данном случае $\Theta_d = \Delta_d(C)$). Множество выборок $\{\Theta_d\}_{d \in \{0, 1, ..., n\}}$ обозначим через Θ.

Фиксируем некоторый SAT-решатель S. Обозначим через $t(C')$ время работы (число битовых операций) SAT-решателя S на произвольном входе C'. Введем в рассмотрение функцию $\tau: \Theta \to N$,

$$\tau_S(\Theta_d) = \sum_{C' \in \Theta_d} t(C').$$

Значением данной функции при каждом фиксированном $d \in \{0, 1, ..., n\}$ является суммарное время (число битовых операций) работы SAT-решателя S по всем КНФ из Θ_d.

Следует учитывать, что при некоторых значениях параметра d (например, при $d = 0$) КНФ из Θ_d могут оказаться очень сложными для SAT-решателя, и в этом случае время подсчета соответствующего значения прогнозной функции может превысить разумные границы. Для учета данного факта вводится в рассмотрение специальная функция $g(C) = p(m \cdot n)$, здесь m — число дизъюнктов в КНФ C, а $p(\cdot)$— некоторый полином, степень которого больше 1.

Допустим, что в соответствии с перечисленными правилами построено семейство выборок $\{\Theta_d\}_{d \in \{0, 1, ..., n\}}$ (при фиксированном R_0). Прогнозную функцию определим следующим образом:

$$T(\Theta_d) = \begin{cases} \dfrac{2^d}{q_d}\tau_S(\Theta_d),\ 2^d > R_0,\ \tau_S(\Theta_d) < g(C), \\ \tau_S(\Theta_d),\ 2^d \leq R_0,\ \tau_S(\Theta_d) < g(C), \\ \infty,\ \tau_S(\Theta_d) \geq g(C). \end{cases} \quad (1)$$

Запись «$T(\Theta_d) = \infty$» означает, что функция не определена на выборке Θ_d. Рациональное число $T(\Theta_d)$ является прогнозом общего объема битовых операций, требуемого для решения исходной SAT-задачи при декомпозиции КНФ C на семейство КНФ, порожденное множеством X^d. Тем самым задача прогнозного планирования оптимального по трудоемкости параллельного вычисления сводится к задаче минимизации функции T на множестве $domT \subseteq \Theta$. Знание глобального минимума функции T на $domT$ дает представление о возможности параллельного решения SAT-задачи для КНФ C «за разумное время».

Далее для упрощения рассуждений рассматриваем прогнозные функции описанного типа, в которых объем случайной выборки (когда она осуществляется) есть постоянное число q. Относительно данных прогнозных функций справедлив следующий результат.

Теорема 1. *Пусть C — произвольная КНФ над множеством булевых переменных мощности n, состоящая из m дизъюнктов, и пусть $\{\Theta_d\}_{d \in \{0, 1, ..., n\}}$ — произвольное семейство выборок КНФ, сформированное согласно перечисленным выше правилам. Для прогнозной функции $T: \Theta \to Q$ вида* (1)*, в которой объем произвольной случайной выборки есть постоянное число q, справедливы следующие свойства:*

1) $domT \neq \varnothing$;

2) глобальный минимум T на $domT$ находится в общем случае за время, ограниченное полиномом от $m \cdot n$. ♦

Доказательство см. в Приложении.

4. SAT-ПОДХОД В ЗАДАЧАХ ОБРАЩЕНИЯ ДИСКРЕТНЫХ ФУНКЦИЙ

Постановка задач обращения дискретных функций как SAT-задач приведена в работах [4, 6, 7]. Здесь мы кратко остановимся лишь на основных понятиях и результатах.

Определение (см., например, работу [8]). Класс $\Im$ образован всеми такими алгоритмически вычислимыми семействами тотальных дискретных функций, которые могут быть вычислены МТ-программами с полиномиальной от n вычислительной сложностью. Проблема обращения дискретной функции f_n семейства $f \in \Im$ ставится следующим образом: дано двоичное слово $y \in range\ f_n$, требуется найти такое слово $x \in \{0, 1\}^n$, что $f_n(x) = y$. ♦

Формулы алгебры логики, рассматриваемые как слова над конечными алфавитами, при помощи взаимно-однозначных кодирований могут быть преобразованы в двоичные слова [5]. При этом под объемом логического уравнения понимается длина двоичного слова, кодирующего данное уравнение в некоторой фиксированной кодировке. Далее рассматриваются только «разумные» (в терминологии работы [5]) кодировки.

Следующее утверждение в своей основе аналогично теореме Кука [3], поэтому его доказательство здесь не приводится (достаточно полное доказательство теоремы Кука имеется, например, в работе [5]).

Теорема 2. *Рассмотрим произвольное семейство дискретных функций* $f = \{f_n\}_{n \in N}, f \in \Im$. *Проблема обращения функций данного семейства за полиномиальное от n время преобразуется в проблему поиска решений логических уравнений вида* $L(x_1, ..., x_{p(n)}) = 1$, *где* $p(\cdot)$ *— некоторый полином, а формула* $L(x_1, ..., x_{p(n)})$ *есть КНФ над* $X = \{i_1, ..., x_{p(n)}\}$. *Объем двоичных кодировок получаемого при этом семейства КНФ ограничен полиномом от n.*♦

Эффективный (полиномиальный по сложности) переход от достаточно произвольных логических уравнений к формату «КНФ = 1» оказывается возможным благодаря вводу дополнительных логических переменных. В работе [9] описана схема приведения произвольных систем логических уравнений к данному формату, позволяющая гарантировать, что вычисляемый набор $x' \in \{0, 1\}^n$ действительно является прообразом $y \in range\ f_n$.

Все сказанное означает, что задача обращения произвольной функции из класса $\Im$ может быть решена следующим образом. Алгоритм вычисления произвольной функции f_n семейства $f = \{f_n\}_{n \in N}$, $f \in \Im$, преобразуется в логическое уравнение вида $L(x_1, ..., x_{p(n)}) = 1$, в левой части которого находится КНФ над множеством булевых переменных $X = \{i_1, ..., x_{p(n)}\}$. Затем в это уравнение подставляется известный вектор $y \in range\ f_n$. Полученная в результате КНФ $C|_y$ является выполнимой, а из выполняющего ее набора можно выделить компоненты вектора $x \in \{0, 1\}^n$ такого, что $f_n(x) = y$. Тем самым задача обращения функции f_n в точке $y \in range\ f_n$ оказывается сведенной к задаче поиска выполняющего набора КНФ $C|_y$.

Успех той или иной технологии всегда должен подтверждаться адекватными тестами. Под адекватностью тестов здесь понимается их аргументированно высокая сложность. Аргументация сложности может состоять в том, что решение тестового примера за приемлемое время означает решение некоторой сложной в вычислительном отношении задачи, например, задачи криптоанализа. Подход к задачам криптоанализа симметричных шифров как к SAT-задачам был развит в работах [6, 10] и получил название «логический криптоанализ». В данной работе мы используем описанную выше параллельную SAT-технологию для решения задач криптоанализа ряда генераторов двоичных последовательностей, «последовательный подход» в отношении которых оказался неэффективным.

Генератор двоичной последовательности (генератор) — это семейство дискретных функций $g = \{g_n\}_{n \in N}$ вида g_n: $\{0, 1\}^n \to \{0, 1\}^*$, $n \in N$, $g \in \Im$.

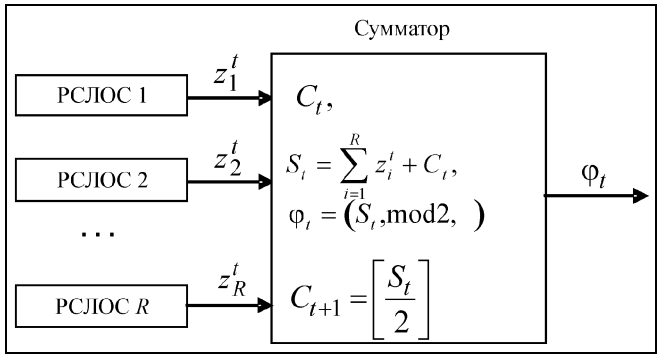

Рис. 1. Схема работы суммирующего генератора

Двоичная последовательность, подаваемая на вход генератора, называется инициализирующей. Последовательность, получаемая на выходе генератора, называется выходной последовательностью или ключевым потоком. Алгоритм вычисления функций g_n называется порождающим алгоритмом генератора. Под криптоанализом генератора понимается поиск инициализирующей последовательности $x = (x_1, ..., x_n)$ по известному фрагменту y выходной последовательности.

Конъюнктивные нормальные формы, кодирующие алгоритмы генераторов, на практике получаются с помощью специального программного комплекса, в который входят LC-язык и LC-транслятор [11]. Порождающий алгоритм генератора записывается на LC-языке, после чего LC-транслятор переводит его в КНФ. Реализованный на обычном персональном компьютере SAT-подход к криптоанализу оправдал себя применительно к генераторам Геффе и Вольфрама [7] и оказался неэффективным в отношении суммирующего и порогового генераторов.

Суммирующий генератор (рис. 1) впервые был описан в работе Р. Рюппеля [12] (см. также [13]). Он состоит из $R \geqslant 2$ регистров сдвига с линейной обратной связью (РСЛОС) [14] и специального суммирующего регистра, называемого далее сумматором. Сумматор представляет собой одномерный массив ячеек памяти размерности $[\log R]$. В каждый момент времени t, $t \in \{1, 2, ...\}$, в ячейках сумматора записана последовательность бит, являющаяся двоичным представлением некоторого натурального числа C_t. Начальное заполнение сумматора служит частью секретного ключа. В каждый такт синхронно снимаемые с выходов РСЛОС биты подаются на вход сумматора, где целочисленно складываются с числом C_t. Полученное натуральное число обозначим через S_t. Выходом генератора в момент t является младший бит двоичного представления числа S_t, т. е. остаток от деления

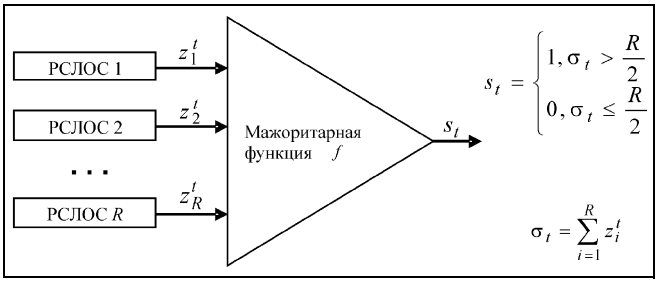

Рис. 2. Схема работы порогового генератора

числа S_t на 2. Двоичное представление частного от деления числа S_t на 2 образует новое заполнение ячеек сумматора (число C_{t+1}). Для суммирующего генератора на основе двух РСЛОС (сумматор состоит из одной ячейки) известны корреляционные атаки [13]. Для $R \geqslant 3$ данный генератор считается довольно стойким, поскольку шифрующее преобразование обладает высокими показателями корреляционной иммунности и линейной сложности.

Параллельный логический криптоанализ был применен к суммирующему генератору на основе следующих трех РСЛОС: (19, $x^{19} + x^5 + x + 1$), (22, $x^{22} + x + 1$) и (23, $x^{23} + x^{15} + x^2 + x + 1$). Длина инициализирующей последовательности 64 бита. Анализировался фрагмент ключевого потока длиной 180 бит. При постановке задачи криптоанализа данного генератора как SAT-задачи были использованы его канонические уравнения в классе полиномов над GF(2), приведенные в работе [15].

Пороговый генератор (рис. 2) был предложен Дж.О. Брюером в 1984 г. [16], (см. также работу [13]). В каждый момент времени t, $t \in \{1, 2, ...\}$, биты, снимаемые синхронно с РСЛОС $R \geqslant 3$, подаются на вход мажоритарной функции f, значением которой является бит выходной последовательности, имеющий номер t. Мажоритарная функция выдает бит 1, если среди битов, поступивших ей на вход, большинство составляют единицы, и выдает бит 0 в противном случае. При $R = 3$ пороговый генератор представляет собой усиленный вариант генератора Геффе [7, 13]. Полиномиальное представление функции f при $R = 5$ [15] имеет следующий вид:

$$f(x_1, x_2, x_3, x_4, x_5) =$$
$$= x_1x_2x_3 \oplus x_1x_2x_4 \oplus x_1x_2x_5 \oplus x_1x_3x_4 \oplus x_1x_3x_5 \oplus x_1x_4x_5 \oplus$$
$$\oplus x_2x_3x_4 \oplus x_2x_3x_5 \oplus x_2x_4x_5 \oplus x_3x_4x_5 \oplus x_1x_2x_3x_4 \oplus$$
$$\oplus x_1x_2x_3x_5 \oplus x_1x_2x_4x_5 \oplus x_1x_3x_4x_5 \oplus x_2x_3x_4x_5.$$

Параллельный логический криптоанализ был применен к пороговому генератору на основе следующих пяти РСЛОС: (11, $x^{11} + x^{10} + x^8 + x^3 + 1$), (13, $x^{13} + x^9 + x^8 + x^2 + 1$), (15, $x^{15} + x^{14} + x^{12} + x^2 + 1$), (16, $x^{16} + x^{14} + x^8 + x^3 + 1$) и (17, $x^{17} + x^{15} + x^{13} + x^{12} + x^{11} + x^{10} + 1$). Длина инициализирующей последовательности 72 бита. Анализировался фрагмент ключевого потока длиной 150 бит. Результаты успешного криптоанализа данного генератора в открытой литературе нами не обнаружены.

5. ВЫЧИСЛИТЕЛЬНЫЙ ЭКСПЕРИМЕНТ

Описанная параллельная SAT-технология была использована для криптоанализа порогового и суммирующего генераторов с перечисленными выше параметрами. Была выбрана простейшая стратегия формирования множества X^d — в него включались переменные x_1, ..., x_d, соответствующие первым d неизвестным компонентам инициализирующей последовательности.

Для проведения вычислительного эксперимента была организована распределенная вычислительная среда (РВС) на основе вычислительного кластера МВС-1000/16. На управляющем узле кластера располагался работающий под ОС Linux сервер РВС с установленными модулями декомпозиции, прогнозирования и анализа результатов. На вычислительных узлах кластера SAT-задачи решались SAT-решателем minisat [17]. Архитектура РВС для решения SAT-задач подробно описана в работе [18].

На рис. 3 и 4 приведены графики оптимизации прогнозных функций для суммирующего и порогового генераторов. Заштрихованные столбцы означают, что для соответствующих значений пара-

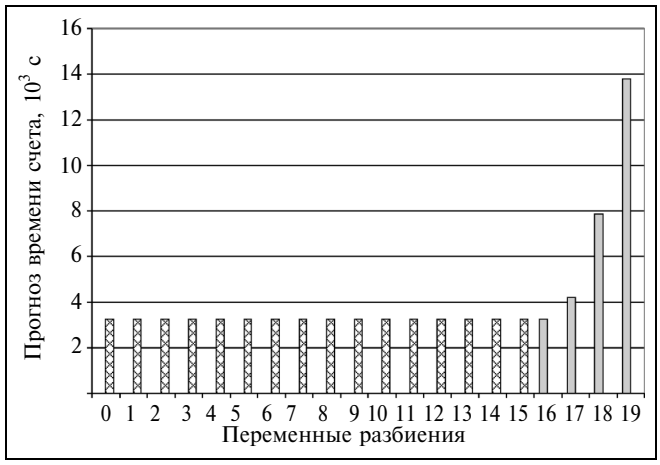

Рис. 3. Пример оптимизации прогнозной функции для суммирующего генератора с инициализирующей последовательностью 64 бита и выходом 180 бит

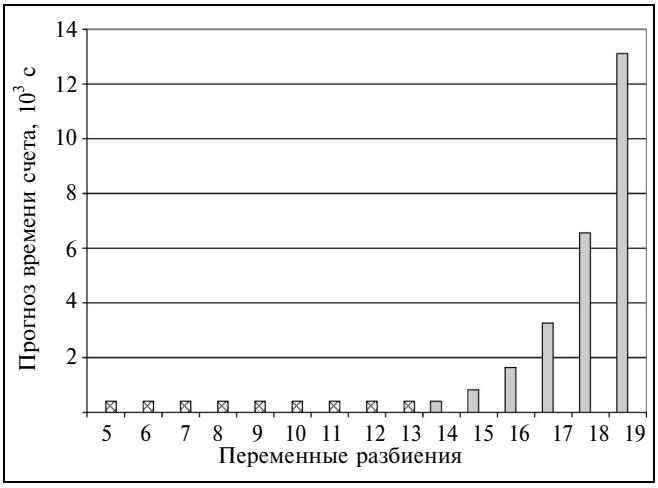

Рис. 4. Пример оптимизации прогнозной функции для порогового генератора с инициализирующей последовательностью 72 бита и выходом 150 бит

метра d вычисление прогнозной функции было прервано, так как превышалось текущее пороговое значение (см. § 3).

В таблице приведены результаты успешного логического криптоанализа суммирующего и порогового генераторов с параметрами декомпозиции, найденными на этапе прогнозирования и оптимизации прогнозной функции.

ЗАКЛЮЧЕНИЕ

Представлена новая технология решения SAT-задач, базирующаяся на концепции крупноблочного параллелизма. В основе технологии лежит процедура прогнозирования оптимального по трудоемкости вычисления с помощью специальной прогнозной функции. Для класса прогнозных функций, используемого в работе, приведен алгоритм решения соответствующей оптимизационной задачи полиномиальной сложности. Практическая ценность технологии подтверждена аргументированно сложными тестами — технология успешно апробирована на задачах криптоанализа некоторых систем поточного шифрования (суммирующего и порогового генераторов). Описанный подход может быть применен в решении многочисленных комбинаторных проблем, допускающих эффективную сводимость к SAT-задачам. К таковым можно отнести проблемы верификации дискретных управляющих систем, верификацию в программных логиках (model checking), а также различные задачи с криптографическим подтекстом (криптоанализ, верификация протоколов аутентификации и многое другое).

Авторы выражают благодарность И.А. Сидорову и А.Г. Феоктистову за помощь в программной реализации и проведении вычислительного эксперимента.

ПРИЛОЖЕНИЕ

Доказательство теоремы 1.

Найдем значение прогнозной функции при $d = n$. Поскольку $X^n = X$, то для любого вектора $Y \in \{0, 1\}^n$ SAT-задача применительно к КНФ $C|_Y$ заключается в подстановке компонент вектора Y в C. Данная процедура требует $O(n \cdot m)$ битовых операций. Но тогда вычисление величины $\frac{1}{q}\sum_{C' \in \Theta_n} t(C')$ требует в общем случае времени $O(n \cdot m)$ (q — не зависящая от входа константа). Умножение полученного числа на 2^n потребует еще $O(n)$ битовых операций. Таким образом, общая сложность вычисления $\tau_S(n)$ и $T(\Theta_n)$ ограничена величиной $O(n \cdot m)$. Данный факт говорит о том, что функция T определена хотя бы при $d = n$.

Далее мы описываем алгоритм итеративного улучшения значений функции T на множестве $domT$, являющийся алгоритмом типа «динамического программирования».

Через i будем обозначать номер итерации алгоритма. Результатом начальной ($i = 0$) итерации алгоритма являются значения $\tau_S(\Theta_n)$ и $T(\Theta_n)$. Основная идея алгоритма состоит в том, что целесообразность продолжения вычислений на итерации с номером i, $i \in \{1, 2, ..., n\}$, определяется на основе «пороговых» значений функций τ_S, T, найденных на предыдущих итерациях. Например, если обозначить через τ_*^0 значение $\tau_S(\Theta_n)$ и пе-

Результаты криптоанализа суммирующего и порогового генераторов (время по нескольким тестам)

Вид генератора	Время решения на однопроцессорном персональном компьютере	Прогноз времени решения на МВС 1000/16 с найденными параметрами декомпозиции	Реальное время решения на МВС 1000/16 с найденными параметрами декомпозиции
Суммирующий (инициализирующая последовательность 64 бита, выход 180 бит)	21...23 ч	Около 1 ч	1...1,5 ч
Пороговый (инициализирующая последовательность 72 бита, выход 150 бит)	Более 28 сут	10...30 мин	1...1,5 ч

рейти к следующей итерации, то неравенство $\tau_S(\Theta_{n-1}) > 2 \cdot \tau_*^0$ при $2^{n-1} > R_0$ означает, что $T(\Theta_{n-1}) > T(\Theta_n)$. Очевидно, что в этом случае глобальный минимум T на $domT$ заведомо не может достигаться при $d = n - 1$. Тем самым справедливость неравенства $\tau_S(\Theta_{n-1}) \geqslant 2 \cdot \tau_*^0$ означает, что соответствующую итерацию можно прервать и перейти к следующей, оставив текущее пороговое значение функции T неизменным. Таким образом, на каждой итерации с номером $i \in \{1, ..., n\}$ происходит сравнение величины $\tau_S(\Theta_{n-i})$ с некоторым текущим пороговым значением τ_*^{i-1}, подсчитанным на предыдущей итерации. По аналогии с T вводим следующую функцию:

$$T_*^0 = T(\Theta_n),$$

$$T_*^i = \begin{cases} \dfrac{2^{n-i}}{q}\tau_*^i,\ 2^{(n-i)} > R_0,\ \tau_*^i < g(C), \\ \tau_*^i,\ 2^{(n-i)} \leq R_0,\ \tau_*^i < g(C),\ i \in \{1, ..., n\}, \\ \infty,\ \tau_*^i \geq g(C). \end{cases}$$

Числа T_*^i, $i \in \{0, 1, ..., n\}$, определяются из следующих рекуррентных соотношений:

$$\tau_*^0 = \tau_S(\Theta_n),$$

$$\tau_*^i = \begin{cases} \min\{\tau_S(n-i),\ 2\tau_*^{i-1},\ g(C)\},\ 2^{(n-i)} > R_0; \\ \min\{\tau_S(n-i),\ T_*^{i-1},\ g(C)\},\ 2^{(n-i)} \leq R_0, \\ i \in \{1, ..., n\}. \end{cases}$$

Величина T_*^i представляет собой пороговое значение прогнозной функции при фиксированном $i \in \{0, 1, ..., n\}$.

Результатом работы алгоритма является число $d_* = n - i_*$. Здесь i_* — такой наименьший номер, что, во-первых, T_*^{i*} определено, а во вторых, для всех $j > i_*$ либо $T_*^j \geqslant T_*^{i*}$, либо $T_*^j = \infty$. Несложно видеть, что при выполнении данных условий $(\Theta_{d*}, T(\Theta_{d*}))$ — «точка» глобального минимума функции T на множестве $domT$. Для подсчета τ_*^i, $i \in \{0, 1, ..., n\}$, достаточно отслеживать суммарное время работы SAT-решателя S на соответствующей выборке. При каждом $i \in \{0, 1, ..., n\}$ данное время не должно превосходить некоторой полиномиальной границы (иначе функция не определена), таким образом, сложность процедуры поиска глобального минимума T на $domT$ в общем случае ограничена полиномом от $m \cdot n$. Теорема доказана.

ЛИТЕРАТУРА

1. *Фаддеев Д.К., Фаддеева В.Н.* Параллельные вычисления линейной алгебры // Кибернетика. — 1977. — № 6. — С. 28—40.
2. *Яблонский С.В.* Введение в дискретную математику. — М., Наука, 1986. — 384 с.
3. *Cook S.A.* The complexity of theorem-proving procedures // Proc. 3rd Ann. ACM Symp. on Theory of Computing, Association for Computing Machinery, Ohio, 1971. — P. 151—159. [Перевод: Кук С.А. Сложность процедур вывода теорем // Кибернетический сборник: Новая серия. — 1975. — Вып. 12, С. 5—15].
4. *Семенов А.А., Беспалов Д.В.* Технологии решения многомерных задач логического поиска // Вестник Томского гос. ун-та / Приложение. — 2005. — № 14. — С. 61—73.
5. *Гэри М., Джонсон Д.* Вычислительные машины и трудноразрешаемые задачи. — М.: Мир, 1982. — 416 с.
6. *Семенов А.А., Буранов Е.В.* Погружение задачи криптоанализа симметричных шифров в пропозициональную логику // Вычислительные технологии (совместный выпуск с журн. «Региональный вестник Востока»). — 2003. — Т. 8. — С. 118—126.
7. *Семенов А.А., Ушаков А.А., Буранов Е.В.* Эвристический поиск в криптоанализе генераторов двоичных последовательностей // Вестник Томского гос. ун-та. — Приложение. — 2006. — № 17. — С. 127—133.
8. *Семенов А.А.* О сложности обращения дискретных функций из одного класса// Дискретный анализ и исследование операций. — 2004. — Т. 11. — № 4. — С. 44—55.
9. *Семенов А.А.* Логико-эвристический подход в криптоанализе генераторов двоичных последовательностей // Труды Междунар. науч. конф. ПАВТ'07, Челябинск, ЮУрГУ, 2007. — Т. 1. — С. 170—180.
10. *Massacci F., Marraro L.* Logical Cryptoanalysis as a SAT Problem: the Encoding of the Data Encryption Standard // Preprint. Dipartimento di Imformatica e Sistemistica, Universita di Roma «La Sapienza», 1999.
11. *Буранов Е.В.* Программная трансляция процедур логического криптоанализа симметричных шифров // Вестник Томского гос. ун-та / Приложение. — 2004. — № 9 (1). — С. 60—65.
12. *Rueppel R.A.* Correlation immunity and the summation combiner / in Lecture Notes in Computer Science 218; Advances in Cryptology: Proc. Crypto'85, H. C. Williams Ed., Santa Barbara, CA, Aug. 18—22, 1985, p. 260—272. — Berlin: Springer-Verlag, 1986.
13. *Поточные* шифры. Результаты зарубежной открытой криптологии. — М.: Мир, 1997. — 389 с.
14. *Menezes A., Van Oorschot P., Vanstone S.* Handbook of Applied Cryptography. — CRC Press, 1996. — 657 с.
15. *Агибалов Г.П.* Логические уравнения в криптоанализе генераторов ключевого потока // Вестник Томского гос. ун-та // Приложение. — 2003. — № 6.— С. 31—41.
16. *Bruer J.O.* On pseudo random sequences as crypto generators // Proc. Int. Zurich Seminar on Digital Communication. — Switzerland, 1984.
17. *MiniSat:* ⟨http://www.cs.chalmers.se/Cs/Research/FormalMethods/MiniSat/MiniSat.html⟩.
18. *Заикин О.С., Сидоров И.А.* Технология крупноблочного распараллеливания в криптоанализе некоторых генераторов двоичных последовательностей // Труды Междунар. науч. конф. ПАВТ'07, Челябинск, ЮУрГУ, 2007. — Т. 1. — С. 158—169.

☎ *(3952) 51-15-08,*
e-mail: oleg.zaikin@icc.ru, biclop@rambler.ru

Статья представлена к публикации членом редколлегии С.Н. Васильевым. ❑