Двоичные диаграммы решений в параллельных алгоритмах обращения дискретных функций[1]

А.А. Семенов, А.С. Игнатьев, Д.В. Беспалов

В работе рассматривается возможность применения двоичных диаграмм решений (BDD) в задачах обращения дискретных функций на параллельных вычислительных системах. Описывается архитектура принципиально нового решателя SAT-задач. Основу данного решателя составляет базирующаяся на BDD технология уменьшения объема памяти, используемой для хранения истории поиска. В качестве тестовой рассматривается задача криптоанализа генератора ключевого потока известной системы поточного шифрования A5/1.

Введение

В настоящей работе развивается подход к решению логических (булевых) уравнений, в основе которого лежит гибридная стратегия, использующая как быстрые алгоритмы решения SAT-задач (нехронологический DPLL-вывод), так и двоичные диаграммы решений (BDD). Построенные алгоритмы тестировались на задачах обращения ряда криптографических функций. Наиболее интересные результаты касаются задачи криптоанализа генератора ключевого потока, используемого в поточном шифре A5/1. В данном контексте результаты настоящей работы улучшают результаты статьи [1].

План работы следующий. В первом пункте приведены основные понятия и вспомогательные результаты, необходимые для дальнейших построений. Во втором пункте дано описание базирующегося на двоичных диаграммах решений (BDD) подхода к поиску корней логических уравнений, кодирующих задачи обращения некоторых дискретных функций. В третьем пункте описывается гибридный подход (SAT+BDD) к обращению дискретных функций. В четвертом пункте приведены результаты численных экспериментов по логическому криптоанализу генератора A5/1, для осуществления которых была задействована параллельная вычислительная система.

1. Основные понятия

Главным объектом рассмотрения настоящей статьи являются задачи обращения тотальных дискретных функций, вычислимых за полиномиальное время. Более точно, речь идет о семействах функций вида $f_n:\{0,1\}^n \to \{0,1\}^*$, где $\{0,1\}^n$ – множество всевозможных двоичных векторов длины n, а $\{0,1\}^*$ – множество всевозможных двоичных векторов произвольной конечной длины. Предполагается, что для любого $n \in \mathrm{N}$ функция f_n всюду определена (тотальна) и существует программа для детерминированной машины Тьюринга (ДМТ), которая вычисляет все функции семейства $f = \{f_n\}_{n\in \mathrm{N}}$. Если такая программа имеет полиномиальную от n сложность, то говорим, что семейство функций f находится в классе $\Im$ (см. [2]). Задача обращения $f_n \in \Im$ заключается в том, чтобы по известному y из области значений f_n найти такое $x \in \{0,1\}^n$, что $f_n(x) = y$.

В работах [1]–[5] был развит пропозициональный подход к задачам обращения дискретных функций из класса $\Im$. В основе данного подхода лежит идея пропозиционального представления алгоритмов, восходящая к С.А. Куку (см. [6]). В соответствии с пропозициональным подходом алгоритм вычисления дискретной функции $f_n \in \Im$ представляется в виде системы логических уравнений $S(f_n)$, которая, грубо говоря, описывает все возможные варианты эволюции программы, вычисляющей f_n, на входах из $\{0,1\}^n$. После подстановки в систему $S(f_n)$ вектора y из области значений f_n имеем совместную систему

[1] Работа выполнена при поддержке гранта РФФИ № 07-01-00400-а и при поддержке гранта Президента РФ НШ-1676.2008.1.

логических уравнений $S(f_n)|_y$, решая которую находим такой $x \in \{0,1\}^n$, что $f_n(x) = y$. Для решения систем вида $S(f_n)|_y$ могут использоваться различные подходы. В [1]–[5] для этих целей применялся SAT-подход, в основе которого лежит техника приведения систем $S(f_n)|_y$ к уравнениям вида «КНФ=1». В статье [4] была предложена технология крупноблочного параллелизма, предназначенная для решения SAT-задач. В работах [1], [4], [5] данная технология использовалась для решения задач обращения некоторых криптографических функций на суперкомпьютерах.

В настоящей статье предлагается комбинированный подход к решению задач обращения дискретных функций из класса $\Im$, использующий как SAT-технологии, так и двоичные диаграммы решений (BDD). Далее кратко описываются основы теории BDD в применении к решению систем логических уравнений.

2. Двоичные диаграммы решений

Двоичные диаграммы решений (Binary Decision Diagram, BDD) являются удобным инструментом представления и оперирования булевыми функциями. Формально BDD определяется как направленный ациклический граф, две вершины которого имеют метки «0» и «1». Данные вершины называются терминальными и соответствуют значениям представляемой булевой функции. Выходная степень терминальных вершин равна 0. Все остальные вершины имеют выходную степень 2. Одна вершина имеет входную степень равную 0, эта вершина называется корневой, а остальные вершины имеют входную степень ≥ 1. BDD ориентируется от корня к терминальным вершинам. Ребра (дуги) BDD помечаются двумя возможными способами: пунктиром (нижние или low-ребра) либо сплошной линией (верхние или high-ребра). Любая вершина, кроме терминальных, обязательно имеет двух детей – один соединен с ней low-ребром, другой high-ребром. Более подробное изложение теории BDD можно найти, например, в книге [7].

Применяя к произвольной тотальной булевой функции разложение Шеннона (см. [7]), можно получить представление данной функции в виде BDD. Переменным булевой функции при этом сопоставляются вершины BDD (одной и той же переменной могут соответствовать несколько различных вершин BDD). Пусть BDD $B(f)$ представляет булеву функцию f над множеством булевых переменных $X = \{x_1,...,x_n\}$. Пусть π– произвольный путь в $B(f)$ из корневой вершины в любую из терминальных. Прохождению данного пути соответствует некоторое последовательное означивание переменных из X (возможно не всех). Следовательно, π задает некоторый (вообще говоря, частичный) порядок на множестве X. Если прохождение каждого пути в BDD из корня в любую из терминальных вершин подчинено некоторому фиксированному полному порядку на X, то такая BDD называется упорядоченной (ordered) или кратко OBDD. Упорядоченная BDD называется сокращенной или редуцированной (ROBDD), если она не содержит повторяющихся фрагментов. На более формальном уровне это означает, что если в ROBDD вершины u и v соответствуют переменной x, причем low-ребенок u совпадает с low-ребенком v, и high-ребенок u совпадает с high-ребенком v, то вершины u и v совпадают. Кроме этого в ROBDD ни для какой вершины low-ребенок не может совпадать с high-ребенком. Фундаментальный результат, полученный Р.Е. Брайантом в [8], состоит в том, что произвольная всюду определенная булева функция имеет единственное ROBDD-представление (с точностью до изоморфизма соответствующих графов).

Пусть $X = \{x_1,...,x_n\}$– множество булевых переменных. Рассмотрим произвольную систему логических уравнений над X следующего вида:

$$\begin{cases} U_1(x_1,...,x_n) = 1 \\ .... \\ U_m(x_1,...,x_n) = 1 \end{cases} \qquad (1)$$

Данная система эквивалентна одному логическому уравнению

$$U_1(x_1,...,x_n) \cdot ... \cdot U_m(x_1,...,x_n) = 1.$$

Введем в рассмотрение булеву функцию $U:\{0,1\}^n \to \{0,1\}$, которую определим следующим образом:

$$U(\alpha) = \left(U_1(x_1,\ldots,x_n)\cdot\ldots\cdot U_m(x_1,\ldots,x_n)\right)|_{\alpha},\ \alpha\in\{0,1\}^n .$$

Назовем данную функцию характеристической функцией системы (1). Обозначим через $B(U)$ ROBDD-представление функции U. Если система (1) совместна, то произвольный путь π в ROBDD $B(U)$ из корня в терминальную вершину «1» определяет некоторое семейство наборов значений истинности переменных из X, являющихся решениями (1).

Основные алгоритмы «манипулирования» булевыми функциями при помощи BDD были описаны в работе [8]. Построение ROBDD B_3 булевой функции $f_3 = f_1 \circ f_2$ на основе ROBDD B_1, B_2, представляющих функции f_1 и f_2 (предполагается, что все функции заданы над множеством булевых переменных X, а «$\circ$»– произвольная бинарная логическая связка), можно осуществить при помощи алгоритма APPLY (см. [8]). Порядок переменных в B_1, B_2 при этом должен быть одинаков (как следствие тот же порядок будет иметь и ROBDD B_3). Сложность процедуры APPLY построения B_3 в таком случае ограничена сверху величиной $O(|B_1|\cdot|B_2|)$ (здесь и далее через $|B|$ обозначается число вершин в ROBDD B). Несложно понять, что найти решение системы (1) можно, построив при помощи алгоритма APLLY ROBDD-представление булевой функции U, например, по следующей схеме:

$$B(U) = \mathrm{APPLY}\left(B_1\cdot \mathrm{APPLY}\left(B_2\cdot\ldots\cdot \mathrm{APPLY}\left(B_{m-1}\cdot B_m\right)\right)\right),$$

где через B_i, $i\in\{1,\ldots,m\}$, обозначены ROBDD-представления булевых функций $U_i:\{0,1\}^n\to\{0,1\}$, фигурирующих в левых частях системы (1) (в качестве бинарной связки используется конъюнкция).

Определение 1.

Пусть система (1) совместна. Пусть $B(U)$ – *ROBDD-представление характеристической функции данной системы и* π – *произвольный путь из корня* $B(U)$ *в терминальную вершину «1». Данный путь определяет некоторое множество решений (1), обозначаемое через* $\mathrm{A}(\pi)$, $|\mathrm{A}(\pi)|\geq 1$. *Про любое* $\alpha\in \mathrm{A}(\pi)$ *говорим также, что* π *содержит* α.

3. Комбинированный подход (SAT+ROBDD) в задачах обращения дискретных функций

В данном пункте описывается подход к решению систем логических уравнений, кодирующих задачи обращения функций из класса $\Im$, составными частями которого являются как SAT-технологии, так и двоичные диаграммы решений.

3.1 Особенности DPLL-вывода в задачах обращения дискретных функций. Обсуждаемые в данном подразделе особенности имеют отношение к SAT-подходу в обращении дискретных функций. Предположим, что рассматривается задача обращения функции $f_n:\{0,1\}^n\to\{0,1\}$, $f_n\in\Im$, в точке $y\in range\ f_n$ и $C(V)=1$ – логическое уравнение вида «КНФ=1», кодирующее данную задачу. Здесь $C(Y)$ – КНФ над множеством булевых переменных $V=\{x_1,\ldots,x_n,y_1,\ldots,y_{p(n)}\}$, $p(\cdot)$ – некоторый полином, переменные множества $X=\{x_1,\ldots,x_n\}$ символизируют переменные входа функции f_n. Переменные из $Y=V\setminus X$ – это переменные, появляющиеся в результате осуществления преобразований Цейтина (см. [3], [1]) при переходе от неоднородного формата исходной системы логических уравнений к формату «КНФ=1». Далее переменные из множества Y называем цейтиновскими.

Пусть $C=C(X)$ – произвольная КНФ над множеством булевых переменных $X=\{x_1,\ldots,x_k\}$. Рассмотрим некоторое множество $X'=\{x'_1,\ldots,x'_r\}$: $X'\subseteq X$. Пусть $(\alpha_1,\ldots,\alpha_r)$ – произвольный вектор, образованный значениями истинности переменных из X'. Подставим в C значение $x'_1=\alpha_1$. Данная подстановка заключается в вычеркивании из C некоторых литералов и дизъюнктов. При этом отслеживаются возможности срабатывания правила единичного дизъюнкта с последующими подстановками в C соответствующих индуцированных значений. Если в результате не выведен конфликт или выполняющий C набор, то в КНФ $C|_{x'_1=\alpha_1}$

осуществляется подстановка значения $x'_2 = \alpha_2$. И так далее. Описанная процедура определяет последовательную подстановку в C вектора $(\alpha_1,...,\alpha_r)$ относительно порядка $x'_1 \prec ... \prec x'_r$. Возможны три исхода такой подстановки. Ситуация 1 состоит в том, что подстановка порождает конфликт. Ситуация 2 состоит в том, что подстановка приводит к выводу набора, выполняющего C. Наконец, результатом подстановки может быть КНФ C', каждый дизъюнкт которой содержит более одного литерала.

Определение 2.

Если результатом последовательной подстановки $x'_1 = \alpha_1,...,x'_r = \alpha_r$ в C является либо ситуация 1, либо ситуация 2, то говорим, что подстановка индуцирует в C детерминированный DPLL-вывод соответственно конфликта или выполняющего набора.

Определение 3.

Пусть $X = \{x_1,...,x_k\}$ – множество булевых переменных и $X' \subseteq X$. Проекцией вектора $\alpha = (\alpha_1,...,\alpha_k)$ значений переменных из X на X' называется вектор, образованный теми компонентами α, которые являются значениями переменных из X'. Проекцию α на X' обозначаем через $\alpha_{X'}$.

Определение 4.

Ядром DPLL-вывода для КНФ $C = C(X)$, называется такое множество $X^{\ker}(C) \subseteq X$ с заданным на нем порядком τ, что:

1. *для любого вектора $\alpha = (\alpha_1,...,\alpha_k)$, выполняющего C, подстановка в C вектора $\alpha_{X^{\ker}(C)}$ относительно τ индуцирует детерминированный DPLL-вывод α;*
2. *для любого вектора $\beta = (\beta_1,...,\beta_k) : C|_{\beta} = 0$, подстановка в C вектора $\beta_{X^{\ker}(C)}$ относительно τ индуцирует детерминированный DPLL-вывод конфликта.*

Ядро $X^{\ker}(C) : X^{\ker}(C) = X$, называется тривиальным. Ядро наименьшей мощности называется минимальным и обозначается через $X_*^{\ker}(C)$.

Теорема 1.

Пусть $C(V) = 1$, $C(V)$ – КНФ над $V = X \cup Y$, – логическое уравнение, кодирующее задачу обращения функции f_n в произвольной точке $y \in range\, f_n$, X_τ – множество переменных входа f_n с заданным на нем произвольным порядком τ. Тогда $V_*^{\ker}(C(V)) \subseteq X_\tau$.

Доказательство данной теоремы мы здесь не приводим ввиду ограничений на объем текста. Обозначим лишь основную идею. При переходе от исходной системы к уравнению вида «КНФ=1» происходит ввод цейтиновских переменных. Произвольная цейтиновская переменная – это булева переменная, интерпретирующая выход некоторой булевой функций, аргументами которой могут быть как переменные входа функции f_n, так и цейтиновские переменные, введенные на более ранних этапах. Подстановка произвольного вектора значений переменных входа f_n в КНФ $C(V)$ порождает вывод по правилу единичного дизъюнкта значений всех последующих функционально с ними связанных цейтиновских переменных. Однако при этом могут возникнуть конфликты с подставленными значениями переменных, символизирующих биты выхода функции f_n. Все это укладывается в правила DPLL-вывода (BCP-стратегию). Тем самым множество переменных входа f_n образует ядро DPLL-вывода КНФ $C(V)$. Очень важно, что порядок подстановки значений переменных из X может быть произвольным. Данный факт означает, что для обращения f_n можно применять произвольный GRASP-подобный алгоритм, использующий нехронологический бэктрекинг, CL-процедуру и рестарты (см. [9]).

Итак, теорема 1 позволяет сделать вывод о том, что при рассмотрении задачи обращения дискретной функции как SAT-задачи имеется очень важная дополнительная информация о том, какие переменные являются, грубо говоря, «истинно информативными». Это переменные входа рассматриваемой функции. От них функционально зависят все остальные переменные, поэтому любая подстановка в КНФ значений переменных из X порождает либо бесконфликтный вывод значений всех остальных переменных по правилу единичного дизъюнкта, либо вывод конфликта. Следовательно, на КНФ, кодирующих задачи обращения дискретных функций, имеет место полная стратегия логического вывода, в соответствии с которой выбор переменных уровней решения осуществляется только из множества X (переменных входа функции f_n). Тем самым при осуществлении данной стратегии можно гарантировать, что порождаемые конфликтные дизъюнкты будут содержать только литералы над множеством X.

При программной реализации описанной стратегии требуется учитывать ряд особенностей. В частности, требуется изменить процедуру анализа конфликтов и построения конфликтных дизъюнктов. В большинстве высокоскоростных SAT-решателей, таких как minisat (см. [10]) или zchaff (см. [11]) используется процедура анализа конфликта, называемая «First UIP-схемой» (кратко «FUIP-схема»). Аббревиатура «FUIP» (First Unique Implication Point) обозначает ближайшую к конфликту на графе вывода точку, доминирующую над обеими конфликтными вершинами (см. [12], [13]). При решении SAT-задачи, кодирующей обращение функции $f_n \in \Im$, вместо FUIP-схемы можно использовать гораздо более простой механизм, состоящий в следующем. Каждой точке на графе вывода (Implication Graph, [9]) ставится в соответствие двоичный вектор длины $|X|$. Единицы в данном векторе означают наличие среди предков рассматриваемой вершины соответствующих переменных из X, нули – отсутствие. Описанный прием позволяет в момент конфликта определить значения переменных из X, ответственные за конфликт. Таким образом, для построения конфликтного дизъюнкта не требуется обратный ход по графу вывода.

3.2 Гибридная стратегия (SAT+ROBDD) в задачах обращения дискретных функций. Основная идея описываемой стратегии состоит в представлении базы конфликтных дизъюнктов в виде ROBDD. Данная идея продиктована восприятием ROBDD как наиболее компактной формы представления булевых функций в классе графов специального вида.

Теорема 2.

Рассматриваем задачу обращения функции $f_n \in \Im$ *в точке* $y \in range\, f_n$*, кодируемую уравнением* $C(V)=1$. *Пусть* X *– множество переменных входа функции* f_n. *Обозначим через* $D_1(X),\ldots,D_k(X)$ *конфликтные дизъюнкты, выведенные из* $C(V)$ *в ходе применения DPLL-вывода с выбором переменных только из* X. *Обозначим через* B_k *ROBDD булевой функции* $D_1(X)\cdot\ldots\cdot D_k(X)$. *Тогда существует такой путь* π *из корня* B_k *в терминальную вершину «1», что* $\alpha \in \mathrm{A}(\pi)$, *где* $\alpha \in \{0,1\}^n$ *– решение рассматриваемой задачи обращения.*

Доказательство (кратко).

Докажем данную теорему от противного. Пусть $\alpha=(\alpha_1,\ldots,\alpha_n)$ – произвольное решение задачи обращения функции f_n в точке $y \in range\, f_n$, кодируемой уравнением $C(V)=1$. Предположим, что вопреки утверждению теоремы ни один путь из корня B в терминальную вершину «1» не содержит α в смысле определения 1. Рассмотрим следующую КНФ

$$C(D)=D_1(X)\cdot\ldots\cdot D_k(X).$$

Поскольку $\bigcup_{j=1}^{k} X_j \subseteq V^{\ker}(C) \subseteq X$, то сделанное предположение означает, что $C(D)|_{\alpha}=0$ (через X_j, $j\in\{1,\ldots,m\}$, здесь обозначено множество переменных из X, фигурирующих в $D_j(X)$). С другой стороны, в силу природы алгоритма DPLL, дополненного CL-процедурой (см. [9]), каждый дизъюнкт $D_j(X), i\in\{1,\ldots,m\}$, является логическим следствием КНФ C. Поэтому логическим следствием C является и КНФ $C(D)$. Однако подстановка α в C в силу теоремы 1 индуцирует детерминированный DPLL-вывод набора $\beta=(\beta_1,\ldots,\beta_{q(n)})$, выполняющего C (β – набор значений истинности переменных множества V). При этом из сделанного предположения следует, что $C(D)|_{\beta}=0$ (поскольку $C(D)|_{\alpha}=0$). Данный факт противоречит тому, что $C(D)$ является логическим следствием C. Полученное противоречие означает справедливость утверждения теоремы. Теорема 2 доказана.

Отметим, что теорема 2 может рассматриваться как обоснование «стратегии абдукции» в задаче обращения функций из $\Im$ в том смысле, что начиная с некоторого шага, выбор значений переменных уровней решения в DPLL-выводе над $C(V)$ определяется структурой (ROBDD B_k), которая является логическим следствием из $C(V)$.

4. Использование нового подхода (SAT+ROBDD) в логическом криптоанализе

В статье [1] был описан параллельный логический криптоанализ генератора ключевого потока шифра A5/1. В настоящей работе приводятся результаты параллельного логического криптоанализа данного шифра с использованием описанной гибридной стратегии (SAT+ROBDD).

Описание генератора ключевого потока A5/1 взято из статьи [14]. В основе A5/1 находятся три регистра сдвига с линейной обратной связью (РСЛОС, см., например [15]), задаваемые следующими полиномами обратной связи: РСЛОС 1: $X^{19}+X^{18}+X^{17}+X^{14}+1$; РСЛОС 2: $X^{22}+X^{21}+1$; РСЛОС 3: $X^{23}+X^{22}+X^{21}+X^{8}+1$. В A5/1 используется принцип асинхронного сдвига регистров в зависимости от значений специальной функции

$$\chi_i\left(b_s^1, b_s^2, b_s^3\right)=\begin{cases}1, & b_s^i = \mathrm{majority}\left(b_s^1, b_s^2, b_s^3\right)\\ 0, & b_s^i \neq \mathrm{majority}\left(b_s^1, b_s^2, b_s^3\right)\end{cases}$$

здесь через b_s^i обозначен т.н. «серединный бит» i-того регистра, $i \in \{1,2,3\}$. Схема работы генератора A5/1 приведена на рисунке 1.

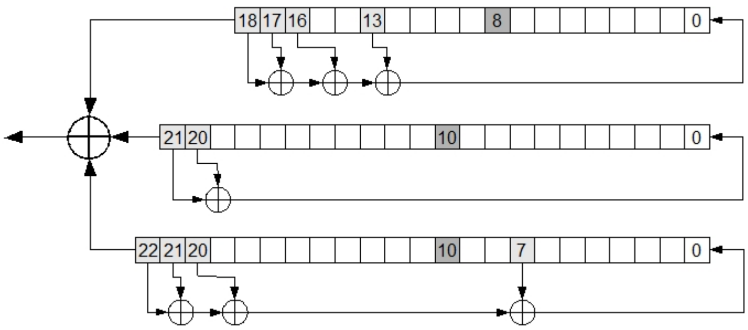

Рис. 1 Схема работы генератора ключевого потока шифра A5/1

Серединные биты каждого из регистров выделены более темной заливкой. Функция «majority»– это функция большинства, определяемая следующим образом:

$$\mathrm{majority}(x,y,z) = x \cdot y \vee x \cdot z \vee y \cdot z$$

Сдвиг i-того регистра осуществляется лишь тогда, когда значение функции $\chi_i(\cdot)$ равно 1.

В контексте сказанного задачу вычисления начального заполнения регистров 1–3 по известным m битам ключевого потока, порожденного данным генератором, можно рассматривать как задачу обращения дискретной функции $f_{A5/1}:\{0,1\}^{64} \to \{0,1\}^m$. В соответствии с пропозициональным подходом сводим данную задачу к задаче поиска решения уравнения вида «КНФ=1», то есть к некоторой SAT-задаче.

В статье [1] была описана процедура параллельного криптоанализа генератора A5/1 в рамках общей технологии крупноблочного параллельного решения SAT-задач, развитой в [4], [5]. Основная идея процедуры криптоанализа, описанной в [1], заключалась в использовании для распараллеливания декомпозиционного множества (см. [4]), схематично изображенного на рисунке 2 темной заливкой.

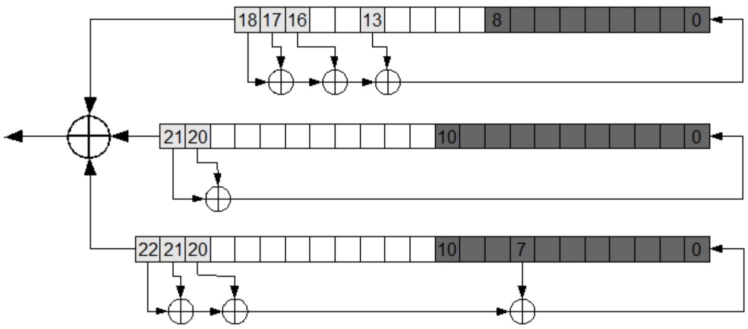

Рис. 2 Схема построения декомпозиционного множества из 31 переменной (X')

В соответствии с данной схемой декомпозиционное множество, по которому осуществляется распараллеливание (см. [4]), образовано следующими переменными входа функции $f_{A5/1}$:

$$X' = \{x_1,...,x_9, x_{20},...,x_{30}, x_{42},...,x_{52}\}$$

(всего 31 переменная). Прогноз криптоанализа A5/1 на кластере Т-60 (см. [16]) при описанной декомпозиционной схеме с применением пакета Distributed-SAT (см. [17]) составил 19-21 часов работы данного кластера (при анализе 150 битов ключевого потока). Интересно, что улучшить данный прогноз за счет уменьшения декомпозиционного множества с использованием техники прогнозных функций (см. [4]) не удалось.

Отметим, что во всех численных экспериментах, описанных в [1], в качестве решателя SAT-задач использовался незначительно модифицированный решатель minisat (см. [18]). В настоящей работе мы приводим результаты прогноза параллельного криптоанализа A5/1 на кластере Т-60 с применением нового решателя, в котором помимо DPLL-вывода используется ROBDD-модуль.

Разработанная схема логического криптоанализа A5/1 включает в себя процедуру препроцессинга, для осуществления которой используется новый гибридный решатель (SAT+ROBDD). В этом смысле описываемая атака близка к атакам, использующим идею «пространственно-временного компромисса» (см. [14]).

На этапе препроцессинга решатель накапливает базу конфликтных дизъюнктов $\{D_1,...,D_s\}$. Причем в соответствии с результатами пункта 3 $D_j, j \in \{1,...,s\}$ – дизъюнкты над множеством переменных входа функции $f_{A5/1}$. Далее строится ROBDD-представление булевой функции $\delta = D_1 \cdot ... \cdot D_s$ (для этой цели в решатель был встроен программный модуль, описанный в [19]). Полученную ROBDD обозначаем через B. Данной ROBDD за полиномиальное в общем случае от числа вершин в B время можно сопоставить логическое уравнение вида $F(x_1,...,x_n) = 1$, где $F(x_1,...,x_n)$ – формула, задающая δ как сложную функцию в виде скобочной формы. При помощи преобразований Цейтина (см. [20]) от уравнения $F(x_1,...,x_n) = 1$ делается переход к уравнению вида $C(V) = 1$, где $C(V)$ – КНФ над множеством булевых переменных $V = \{x_1,...,x_n, v_1,...,v_{p(n)}\}$, $p(\cdot)$ – некоторый полином. Множества решений уравнений $F(x_1,...,x_n) = 1$ и $C(V) = 1$ равномощны и от любого решения $C(V) = 1$ можно эффективно перейти к соответствующему решению уравнения $F(x_1,...,x_n) = 1$ (в терминологии работы [21] данные уравнения обозначаются как консервативно изоморфные). Рассматриваем КНФ $C' = C \cdot C(V)$, где C – исходная КНФ, кодирующая задачу обращения функции f_n (в нашем случае функции $f_{A5/1}$) в некоторой точке из области значений. Можно показать, что множества уравнения $C = 1$ и $C' = 1$ также являются консервативно изоморфными.

Обратим особое внимание на КНФ C'. В данной КНФ содержится информация о конфликтах, описанных дизъюнктами $D_1,...,D_s$. С другой стороны, объем памяти, занимаемой КНФ $C(V)$, может быть существенно (на некоторых примерах в миллионы раз) меньше объема, который занимает КНФ $D_1 \cdot ... \cdot D_s$. Переход от $D_1 \cdot ... \cdot D_s$ к $C(V)$ можно рассматривать как вклад в решение проблемы неполноты вывода (см. [5]), являющейся следствием использования эвристических процедур чистки баз конфликтных дизъюнктов в «стандартных» SAT-решателях.

В проведенной серии экспериментов препроцессинг с использованием нового гибридного (SAT+ROBDD) решателя накапливал информацию в виде конфликтных дизъюнктов в течении примерно суток работы вычислительного узла со следующими характеристиками: одно ядро процессора Intel Xeon Quad-Core E5345 2.33 GHz, 4GB RAM. Для полученной в результате КНФ C' задача поиска выполняющего набора решалась в соответствии со схемой крупноблочного параллелизма, в целом аналогичной [1]. Прогноз времени параллельного решения данной SAT-задачи на кластере Т-60 составил 16-18 часов (в отличие от 19-21 часов без препроцессинга).

Заключение

Описанная в работе технология может оказаться полезной в решении сложных задач на вычислительных кластерах, полный доступ к которым представляется маловероятным. Так, задачу получения полного доступа к ресурсам Т-60 на сутки можно считать практически неосуществимой. Однако после продолжительного препроцессинга, для выполнения которого достаточно обычного ПК, получаемая задача может потребовать существенно меньших ресурсов кластера.

Предложенный в работе подход предполагается применять при решении в распределенных вычислительных средах систем логических (булевых) уравнений большой размерности. Такого рода задачи возникают в логическом программирования, верификации дискретных автоматов, а также в криптоанализе различных систем шифрования (особый интерес в этом плане представляют проблемы поиска коллизий для криптографических хэш-функций).

Литература

1. *Семенов А.А., Заикин О.С., Беспалов Д.В., Буров П.С., Хмельнов А.Е.* Решение задач обращения дискретных функций на многопроцессорных вычислительных системах // Труды Четвертой Международной конференции «Параллельные вычисления и задачи управления» РАСО’2008 (Москва 26-29 октября 2008) – С. 152-176.
2. *Семенов А.А., Заикин О.С., Беспалов Д.В., Ушаков А.А.* SAT-подход в криптоанализе некоторых систем поточного шифрования // Вычислительные технологии. 2008. Т. 13. №6. С. 134-150.
3. *Семенов А.А.* Логико-эвристический подход в криптоанализе генераторов двоичных последовательностей // Труды международной научной конференции ПАВТ’07. Челябинск, ЮУрГУ, 2007. – Т. 1. С. 170-180.
4. *Заикин О.С., Семенов А.А.* Технология крупноблочного параллелизма в SAT-задачах // Проблемы управления. 2008. №1. С. 43–50.
5. *Семенов А.А., Заикин О.С.* Неполные алгоритмы в крупноблочном параллелизме комбинаторных задач // Вычислительные методы и программирование. 2008. Т. 9. №1. С. 112–122.
6. *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. Вып. 12, с. 5-15, – М., «Мир», 1975.]
7. *Meinel Ch., Theobald T.* Algorithms and Data Structures in VLSI-Design: OBDD-Foundations and Applications. Berlin: Springer-Verlag, 1998. 267 p.
8. *Bryant R.E.* Graph-Based Algorithms for Boolean Function Manipulation // IEEE Transactions on Computers. 1986. V. 35. № 8. P. 677–691.
9. *Marqeus-Silva J. P., Sakallah K.A.* GRASP: A search algorithm for propositional satisfiability // IEEE Transactions on Computers. 1999. V. 48. № 5. P. 506–521.
10. http://www.cs.chalmers.se/Cs/Research/FormalMethods/MiniSat/MiniSat.html
11. http://www.ee.princeton.edu/~chaff/zchaff/zchaff.2004.11.15.zip.txt
12. *Zhang L., Madigan C., Moskewicz M., Malik S.* Efficient conflict driven learning in a Boolean satisfiability solver // In Proc. of the International Conference on Computer Aided Design (ICCAD), San Jose, California. 2001. P. 279–285.
13. *Семенов А.А., Отпущенников И.В.* Об алгоритмах обращения дискретных функций из одного класса // Прикладные алгоритмы в дискретном анализе. Серия: Дискретный анализ и информатика, Вып. 2. 2008. Иркутск: Изд-во ИГУ. С. 127-156.
14. *Biryukov A., Shamir A., Wagner D.* Real Time Cryptanalysis of A5/1 on a PC // Lecture Notes in Computer Science. Springer-Verlag. 2001. V. 1978. P. 1-18.
15. *Menezes A., Van Oorschot P., Vanstone S.* Handbook of Applied Cryptography. – CRC Press, 1996. – 657 p.
16. Научно-исследовательский вычислительный центр МГУ имени М.В.Ломоносова [http://parallel.ru/cluster/]
17. *Заикин О.С.* Пакет прикладных программ Distributed-SAT: Свидетельство об официальной регистрации программы для ЭВМ № 2008610423. – М.: Федеральная служба по интеллектуальной собственности, патентам и товарным знакам, 2008.
18. *MiniSat* [http://minisat.se/MiniSat.html]

19. *Семенов А.А., Игнатьев А.С.* Логические уравнения и двоичные диаграммы решений // Прикладные алгоритмы в дискретном анализе. Серия: Дискретный анализ и информатика, Вып. 2. 2008. Иркутск: Изд-во ИГУ. С. 99-126.
20. *Семенов А.А.* Трансляция алгоритмов вычисления дискретных функций в выражения пропозициональной логики // Прикладные алгоритмы в дискретном анализе. Серия: Дискретный анализ и информатика, Вып. 2. 2008. Иркутск: Изд-во ИГУ. С. 70-98.
21. *Семенов А.А.* Консервативные преобразования систем логических уравнений // Вестник ТГУ. Приложение.–2007.– №23.– С. 52-59.