УДК 681.3

МИРОШНИК М.А., доктор технических наук, профессор,
КЛИМЕНКО Л.А., кандидат технических наук, доцент (Украинский государственный университет железнодорожного транспорта),
ГЕРМАН Э.Е., кандидат технических наук, доцент (Национальный технический университет «Харьковский политехнический институт»)

Методы параллельного решения SAT-задач для реализации процедур прогнозирования трудоемкости

В статье разработана и реализована крупноблочная параллельная технология решения SAT-задач в виде MPI-программы в распределенных компьютерных системах. В данной технологии используется декомпозиция исходной SAT-задачи на множество подзадач. В работе используется процедура статистического прогнозирования трудоемкости параллельного решения SAT-задач, которая позволяет определить оптимальные прогнозируемые параметры декомпозиции. Показано, что использование параметров декомпозиции, найденных с помощью процедур прогнозирования, позволяет успешно решать SAT-задачи, кодирующие задачи обращения ряда криптографических дискретных функций.

***Ключевые слова**: распределенные компьютерные системы, SAT-задачи, SAT-решатель, параллельная технология, MPI-программа, прикладные программы, прогнозирование, кодирующие задачи, криптографическая дискретная функция, КНФ, криптоанализ.*

Постановка проблемы в общем виде и ее связь с важными научными или практическими задачами

В работе [1] был рассмотрен локальный алгоритм вычисления информации, делающий возможным решение SAT-задач и задач о раскраске графа. В дальнейшем были разработаны эффективные схемы локального алгоритма при решении конкретных задач удовлетворения ограничений, обладающих специальной структурой с использованием различных видов структурных графов.

Многие значимые в практическом отношении комбинаторные проблемы допускают эффективные сводимости к задачам поиска решений булевых уравнений вида «КНФ=1». Задачи поиска решений таких уравнений называются SAT-задачами, для их решения используются специальные программные комплексы, называемые SAT-решателями [2]. В последнее время высокими темпами развиваются параллельные SAT-решатели [3, 4]. В таких SAT-решателях используется концепция мелкозернистого параллелизма [5]. Такой подход вполне оправдывает себя на многих классах тестов, например для КНФ, кодирующих задачи верификации в микроэлектронике [6]. Применительно же к задачам обращения дискретных функций (например, к задачам криптоанализа поточных систем шифрования) высокую эффективность показала представленная в серии работ [7 - 11] крупноблочная параллельная технология решения SAT-задач в распределенных компьютерных системах (РКС). В рамках данной технологии осуществляется декомпозиция исходной SAT-задачи на семейство подзадач. Наилучшие результаты показали различные варианты декомпозиции по переменным, кодирующим вход дискретной функции, задачу обращения которой требуется решить. Для определения наилучших (по прогнозу) параметров декомпозиции используется процедура прогнозирования трудоемкости параллельного решения SAT-задач. Каждому варианту значений параметров декомпозиции соответствует случайная выборка SAT-задач. Прогноз заведомо быстро вычисляется для одной из выборок, затем он итеративно улучшается при обработке остальных выборок. Решение некоторых SAT-задач может быть прервано при превышении порогового значения.

Формулирование целей статьи (постановка задачи)

Изначально данная технология была реализована в виде прикладных программ (ПП) D-SAT [12], который функционирует под управлением инструментального комплекса DISCOMP [10]. Функциональное наполнение ПП D-SAT включает процедуры решения SAT-задач и процедуры прогнозирования трудоемкости решения SAT-задач. Дальнейшим развитием данной параллельной технологии стала ее реализация в виде MPI-программы. Подробности данной реализации рассматриваются в настоящей статье.

В MPI-программе в режиме прогнозирования все SAT-задачи по всем случайным выборкам объединяются в единый параллельный список. В результате достигается равномерная загрузка РКС, но усложняется обработка данных. Для своевременного прерывания решений SAT-задач используются неблокирующие обмены [13], что позволяют каждому процессу эффективно использовать свое рабочее время: управляющий процесс занимается отправкой заданий и обработкой решений, вычислительные процессы решают SAT-задачи.

В данной работе впервые приведены результаты параллельного логического криптоанализа суммирующего генератора на основе четырех регистров сдвига с линейной обратной связью. Также приведены улучшенные результаты логического криптоанализа ряда других генераторов.

Крупноблочная параллельная технология решения SAT-задач

Далее приведено краткое описание крупноблочной параллельной технологии решения SAT-задач, представленной в работах [7 - 9].

Под распределенной компьютерной системой (РКС) понимается совокупность вычислительных единиц, объединенных коммуникационной сетью. В качестве вычислительной единицы РКС выступает программно-аппаратный ресурс, требуемый для решения некоторой вычислительной задачи. В качестве вычислительной единицы далее рассматривается одно ядро процессора, часть общей оперативной памяти и памяти жесткого диска, а также системное программное обеспечение.

Рассматривается произвольная КНФ C над множеством булевых переменных $X=\{x_1,...,x_n\}$. В множестве X выбирается некоторое подмножество $\{x_{j1},...,x_{jd}\}$, $\{x_1,...,x_d\}\subseteq\{1,...,n\}$, $d\in\{1,...,n\}$. Множество $X'=\{x_{i1},...,x_{id}\}$ называется *декомпозиционным множеством*, а d - *размерностью декомпозиционного множества*. Дополнительно полагается, что при $d=0$ декомпозиционное множество пусто. Декомпозиционному множеству X':$|X'|=d$, $d>0$ ставится в соответствие множество $Y(X')=\{Y_i,...,Y_k\}$, состоящее из $k=2^d$ различных двоичных векторов длины d, каждый из которых является набором значений переменных из множества X'. *Декомпозиционным семейством*, порожденным из КНФ C множеством X', называется множество $\Delta_x(C)$ КНФ, полученных подстановками в C векторов Y_j, $j\in\{1,...,k\}$: $\Delta_x(C)=\{C|_{y1},\ldots,C_k=C|_{yk}\}$, $\Delta_\varnothing(C)=\{C\}$. КНФ, полученная подстановкой в C вектора Yj, обозначается через $C_j=C|_{yj}$.

Пусть $\Delta_x(C)=\{C_1,\ldots,C_k\}$ - декомпозиционное семейство КНФ, порожденное из КНФ C некоторым декомпозиционным множеством X' мощности d. Всякому набору, выполняющему исходную КНФ C, соответствует набор, выполняющий некоторую КНФ из семейства $\Delta_x(C)$.

Наоборот, произвольному набору, выполняющему некоторую КНФ из $\Delta_x(C)$, соответствует единственный набор, выполняющий КНФ C. Следовательно, исходная КНФ C выполнима тогда и только тогда, когда выполнима хотя бы одна КНФ семейства $\Delta_x(C)$. Таким образом, решение исходной SAT-задачи для КНФ C сводится к решению, вообще говоря, $k=2^d$ SAT-задач для КНФ $C_i,...,C_k$ соответственно. Если исходная КНФ C выполнима, то по набору, выполняющему некоторую КНФ семейства $\Delta_x(C)$, можно эффективно перейти к набору, выполняющему исходную КНФ C.

Пусть имеется РВС, состоящая из $r\in N$ вычислительных единиц. Возможны следующие два случая:

1) $k\le r$ - число КНФ в семействе $\Delta_x(C)$ не превосходит числа вычислительных единиц РКС. В этом случае для каждой КНФ из семейства $\Delta_x(C)$ SAT-задача решается на отдельной вычислительной единице РКС;

2) $k>r$ - число КНФ в семействе $\Delta_x(C)$ больше числа вычислительных единиц РКС.

Крупноблочное распараллеливание SAT-задач для случая $k\le r$ рассматривалось ранее. Для случая $k>r$ предлагается следующая процедура.

Процедура 1. Каждому вектору Y_j, $y\in\{1,...,k\}$ ставится в соответствие натуральное число N_j, двоичным представлением которого является вектор Y_j. Данное число назовем натуральным индексом КНФ C_j. Семейство КНФ $\Delta_x(C)$ упорядочивается некоторым образом. Произвольная КНФ из $\Delta_x(C)$ называется *связанной*, если в рассматриваемый момент времени SAT-задача для нее либо уже решена, либо решается на некоторой вычислительной единице РКС. Остальные КНФ называются *свободными*. Выбираются первые r КНФ $C_i,...,C_r$ из семейства $\Delta_x(C)$. Для каждой из выбранных КНФ $C_i,...,C_r$ решается SAT-задача на отдельной вычислительной единице РКС. Как только освобождается некоторая из r вычислительных единиц РКС, на ней запускается процедура решения SAT-задачи для первой (в смысле введенного выше порядка) свободной КНФ семейства $\Delta_x(C)$. Данный процесс продолжается до тех пор, пока не будет найден выполняющий набор некоторой КНФ из $\Delta_x(C)$, либо пока не будет доказана невыполнимость всех КНФ из $\Delta_x(C)$. Описанная процедура решает SAT-задачу для произвольной КНФ C корректно.

Пусть выбрано некоторое декомпозиционное множество X'. Представляет интерес построение такого $\tilde{X}\subset X'$, использование которого в качестве декомпозиционного множества делает декомпозицию более эффективной, чем на основе X'. Данная проблема весьма нетривиальна. Если мощность $\tilde{X}$ мала, то SAT-задачи, получаемые при декомпозиции КНФ, как правило, весьма сложны. Если мощность $\tilde{X}$ велика, то велика и мощность декомпозиционного семейства $\tilde{A}(C)$, и в этом случае простота SAT-задач КНФ данного семейства мало что дает. Для решения данной задачи предлагается следующая процедура статистического прогнозирования [7].

Процедура 2. Используется натуральное число R_0, от значения которого зависит: имеется необходимость формирования случайной выборки или нет. Например, за R_0 можно принять число вычислительных единиц в РКС. Если при некотором $\tilde{X}\subseteq X'$, $|X|$=d мощность семейства $\tilde{A}(C)$ 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 $\tilde{A}(C)$. Через q_d обозначаем объем такой выборки.

Через Y^d обозначается множество, образованное всеми различными векторами значений переменных из $\tilde{X}$:$|\tilde{X}|$=d. Каждому значению параметра $d\in\{0,1,\ldots,|\tilde{X}|\}$ такому, что $2^d > R_0$, ставится в соответствие множество векторов $\{Y_{j1},\ldots, Y_{jqd}\}$, выбираемых из $Y(X')$ в соответствии с равномерным распределением, а также выборка КНФ $\Theta_d=\{C_{j1}=C|_{Y1},\ldots,C_{j1}=C|_{Y2}\}$. Каждому значению параметра $d\in\{0,1,\ldots, |\tilde{X}|\}$ такому, что $2^d\le R_0$, ставится в соответствие множество $Y(X')$ и множество КНФ $\Theta_d=\Delta_x(C)$. Множество выборок $\{\Theta_d\}$ $d\in\{0,1,\ldots, |X\sim|\}$ обозначается через Θ. Фиксируется SAT-решатель S. Обозначим через $t(C')$ время работы SAT- решателя S на произвольном входе C'. Вводится в рассмотрение функция τ_s:$\Theta\to N$, τ_s:$(\Theta_d)=\sum t(C)$ значением которой при каждом фиксированном $d\in\{0,1,\ldots, |\tilde{X}|\}$ является суммарное время работы SAT-решателя S по всем КНФ из Θ_d. При некоторых значениях параметра d (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 $\Theta=\{\Theta d\}$ $d\in\{0,1,\ldots, |X\sim|\}$ (при фиксированном Ro). Прогнозная функция определяется следующим образом:

$$T(\Theta_d)=\begin{cases}\dfrac{2^d}{q_d}\cdot\tau_S(\Theta_d),\ 2^d>R_0,\ \tau_S(\Theta_d)<g(C)\\ \tau_S(\Theta_d),\ 2^d\le R_0,\ \tau_S(\Theta_d)<g(C);\\ \infty,\ \tau_S(\Theta_d)\ge g(C).\end{cases}$$

Запись «$T(\Theta_d)=\infty$» означает, что функция не определена на выборке Θ_d. Рациональное число $T(\Theta_d)$ является прогнозом времени, 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 $domT\subseteq\Theta$. Идея оптимизации функции T состоит в том, что значение $T(\Theta_{d|X'|})$ вычисляется заведомо эффективно. Затем значение T итеративно улучшается при обработке остальных выборок из $domT$. Если время обработки выборки превышает пороговое значение, обработка данной выборки прерывается. Результатом работы описанной процедуры является наилучшее значение $d_*\in domT$ мощности декомпозиционного множества X', а также соответствующее прогнозное время $T(\Theta_d)$.

Эффективность процедуры 1 существенным образом зависит от структуры декомпозиционного множества. Выбор декомпозиционного множества - это отдельная нетривиальная проблема. Некоторые общие стратегии построения декомпозиционных множеств с ориентацией на задачи криптоанализа генераторов ключевого потока были рассмотрены в [10].

Описание MPI-программы PD-SAT

Приведенная ранее технология была реализована в виде MPI-программы PD-SAT, которую можно также назвать параллельным SAT-решателем. PD-SAT может функционировать в режиме решения SAT-задачи и в режиме прогнозирования трудоемкости решения SAT-задачи. Следует особо отметить принципиальные различия данных, обрабатываемых в указанных режимах. Если в режиме решения заданиями являются списки SAT-задач, то в режиме прогнозирования заданиями являются конкретные SAT-задачи.

В режиме решения SAT-задачи декомпозиционное семейство разбивается на непересекающиеся подсемейства КНФ. Каждое такое подсемейство

образует задание, которое обрабатывается на фиксированной вычислительной единице РКС. Под решением задания понимается решение SAT-задач для всех КНФ из соответствующего подсемейства: ответ на задание «UNSAT», если все КНФ из подсемейства невыполнимы; ответ «SAT», если хотя бы одна КНФ из подсемейства выполнима.

Режим прогнозирования реализует процедуру статистического прогнозирования трудоемкости решения SAT-задачи. В данном режиме заданиями являются SAT-задачи для КНФ, образующих обрабатываемую случайную выборку. Тем самым каждой такой выборке сопоставляется *выборка заданий*.

Все сказанное позволяет выделить следующие классы заданий:

- *свободные задания* - задания, процесс решения которых на текущий момент не был запущен;
- *связанные незавершенные задания* - задания, которые решаются на текущий момент;
- *связанные завершенные задания* - задания, которые на текущий момент уже решены.

Реализация режима решения SAT-задачи

Данный режим основан на процедуре 1, приведенной выше. Вычисления разделены на три этапа.

Этап 1. PD-SAT запущен на n процессах: процесс номер 1 управляющий, процессы с номерами 2,...,n - вычислительные. Управляющий процесс по входным данным формирует список заданий, вычислительные процессы при этом простаивают. Число заданий D равно ближайшей справа степени двойки от числа $(n-1)\cdot C$. Здесь C - константа, влияющая на загрузку вычислительных процессов. Данная константа определяется эмпирически; в компьютерных экспериментах использовалась $C=4$.

Пусть, например, дана РКС, состоящая из четырех вычислительных единиц. В MPI- программе один управляющий процесс и три вычислительных. Пусть $C=2$. В соответствии со сказанным выше управляющий процесс сгенерирует $D=8$ заданий.

Этап 2. С управляющего процесса отсылаются первые $n-1$ свободных заданий из списка: i-е задание $(i=1,...,n-1)$ отсылается на вычислительный процесс с номером $i+1$. Каждый вычислительный процесс приступает к обработке полученного задания.

Первые два этапа являются подготовительными, выполняются быстро, и время их выполнения не вносит значительного вклада в общее время работы.

Этап 3. После выполнения этапов 1-2 управляющий процесс переходит в состояние ожидания решений заданий с вычислительных процессов. Если на управляющий процесс приходит ответ «UNSAT», то задание, ответ на которое был прислан, становится связанным завершенным. На приславший данный ответ вычислительный процесс отправляется очередное свободное задание из списка. Программа завершает свою работу, если на управляющий процесс приходит ответ «SAT» (в этом случае исходная КНФ выполнима) или если получены ответы «UNSAT» на все задания.

В этапах 1 - 3 используются только блокирующие функции обмена MPI_Send и MPI_Recv. Приведенная схема решения проста и присуща многим задачам, допускающим крупноблочное распараллеливание, но ее описание полезно для понимания работы PD-SAT в режиме прогнозирования.

Реализация режима прогнозирования трудоемкости параллельного решения SAT-задачи

Данный режим основан на процедуре 2, приведенной выше. Как и в режиме решения SAT-задачи, вычисления в режиме прогнозирования разделены на три этапа. Основные отличия режимов - в функционировании на третьем этапе.

В режиме прогнозирования заданием является конкретная SAT- задача (в отличие от режима решения, где заданием является список SAT-задач).

Далее используются обозначения, введенные в работе [11]. Через $d^{\min}$ и $d^{\max}$ обозначаются натуральные числа, определяющие соответственно нижнюю и верхнюю границы интервала, в котором изменяются значения d, где d - размерность декомпозиционного множества. Через q обозначается жестко заданное значение, определяющее число КНФ в произвольной случайной выборке, а через r - число вычислительных единиц РКС, относительно которого строится прогноз. В режиме прогнозирования на вход PD-SAT подаются параметры $d^{\min}$, $d^{\max}$, q, r.

Этап 1. Как и в режиме решения SAT-задачи, PD-SAT использует n процессов. На управляющем процессе формируется список заданий. Для каждого значения $d\in\{d^{min},\ldots,d^{\max}\}$ строится отдельная выборка заданий. В случае $2^d>q$ в выборку заданий включаются SAT-задачи для q случайным образом выбранных КНФ из декомпозиционного семейства. Если $2^d\le q$, то в выборку включаются SAT-задачи для всех КНФ из декомпозиционного семейства. Задания из всех выборок объединяются в единый параллельный список, притом первыми в списке располагаются задания из выборки, полученной при $d=d^{\max}$. Далее задания располагаются по убыванию значения d, последними в списке расположены задания из выборки, полученной при $d=d^{\min}$.

Этап 2. Подобно режиму решения SAT-задачи, на данном этапе с управляющего процесса отсылаются первые $n-1$ свободных заданий из списка: i-е задание $(i=1,...,n-1)$ отсылается на вычислительный процесс с номером $i+1$.

Этап 3. На данном этапе осуществляется параллельная обработка различных выборок заданий с целью построения прогнозов трудоемкости решения исходной SAT-задачи при использовании соответствующей декомпозиции. Основной на данном этапе является процедура прогнозирования GetPredict, работающая на управляющем процессе. Данная процедура, во-первых, определяет параметры лучшего на текущий момент прогноза трудоемкости решения исходной SAT-задачи. Во-вторых, она определяет, обработка каких выборок заданий должна быть прервана ввиду превышения соответствующими процессами текущих ограничений на время работы. Процедура GetPredict запускается через малые временные интервалы. На входе GetPredict получает следующие массивы:

- cnf_real_time_arr - в массиве содержится получаемая от вычислительных процессов информация о времени обработки связанных завершенных заданий;
- cnf_appr_time_arr - массив строится на управляющем процессе и содержит информацию о времени обработки связанных незавершенных заданий;
- cnf_status_arr - массив статусов заданий;
- set_status_arr - массив статусов выборок заданий.

Статус задания и статус выборки заданий - это динамически изменяющиеся параметры, которые в различные моменты вызова GetPredict могут принимать различные значения.

В текущий момент статус задания может принимать следующие значения:

- WAIT, если задание свободное или связанное незавершенное;
- STOP, если задание находится в выборке, обработка которой прерывается. Задания, получившие статус STOP, в дальнейшем не обрабатываются;
- UNSAT, если задание связанное завершенное и соответствующая ему КНФ оказалась невыполнимой;
- SAT, если задание связанное завершенное и соответствующая ему КНФ оказалась выполнимой.

В текущий момент статус выборки заданий может принимать значения:

- WAIT, если в выборке имеются задания со статусом WAIT, но нет ни одного задания со статусом SAT;
- SAT, если хотя бы одно задание из выборки получило статус SAT;
- STOP, если счет для выборки прерван;
- UNSAT, если все задания из выборки имеют статус UNSAT.

На выходе GetPredict выдает измененные массивы cnf_appr_time_arr, cnf_status_arr, set_status_arr, а также массив cnf_to_stop_arr, содержащий номера вычислительных процессов, на которых должна быть прервана обработка текущих заданий.

Прерывание обработки заданий достигается за счет отправки с управляющего процесса неблокирующих сообщений о прерывании на вычислительные процессы с номерами из массива cnf_to_stop_arr.

От вычислительных процессов требуется не только получать и решать задания, но и периодически проверять наличие сообщений о прерываниях. В используемые SAT-решатели были внесены изменения, позволяющие осуществлять такую проверку за счет применения неблокирующих функций MPI_Iprobe. Если сообщение о прерывании есть, то работа SAT-решателя досрочно завершается, выдается ответ «UNSAT». Даже если КНФ была на самом деле выполнимой, для прогнозирования это не важно. Сообщение с ответом «UNSAT» отправляется на управляющий процесс, после чего принимается следующее задание.

Между периодическими запусками процедуры прогнозирования управляющий процесс переходит в состояние ожидания ответов от вычислительных. Если от вычислительного процесса присылается ответ «UNSAT», на приславший этот ответ вычислительный процесс отправляется очередное свободное задание из списка, время решения SAT-задачи заносится в массив cnf_real_time_arr. Процедура прогнозирования завершает свою работу, если управляющий процесс получил ответ «SAT» или если для всех заданий получены ответы «UN- SAT».

Применение неблокирующих обменов позволяет каждому процессу эффективно использовать свое рабочее время: управляющий процесс занимается отправкой заданий и обработкой решений, вычислительные процессы решают SAT-задачи.

Дополнительно отметим, что в PD-SAT предусмотрена процедура отслеживания «опоздавших» сообщений о прерывании: такие сообщения могут возникать вследствие того, что за время обработки данных управляющим процессом на некотором вычислительном процессе было решено задание из выборки, обработку которой необходимо было прервать. В этом случае сообщение о прерывании от управляющего процесса может быть некорректно интерпретировано. Такого рода ситуации исключаются за счет дополнительной проверки статусов сообщений, поступающих на вычислительные процессы от управляющего.

В PD-SAT используются следующие SAT-решатели, основой которых является известный решатель Minisat:

- dminisat, основанный на MiniSat-C, оптимизирован для решения SAT-задач, кодирующих задачи обращения дискретных функций;
- minisat2, без существенных изменений;
- minisat2_mod, основан на minisat2, внесены изменения в ключевые параметры- константы, добавлено увеличение активности ядровых переменных.

Изначально SAT-решатели семейства Minisat предназначены только для работы под Unix-подобными ОС. В исходный код всех используемых в PD-SAT SAT-решателей были внесены изменения, обеспечивающие платформонезависимость (в смысле переносимости на уровне исходного кода). Тем самым, PD-SAT может функционировать как под управлением Unix-подобных ОС, так и под управлением ОС семейства Windows.

Выводы

В работе представлена реализация крупноблочной параллельной технологии решения SAT- задач в виде MPI-программы PD-SAT. Данная программа позволяет осуществлять прогнозирование трудоемкости решения SAT-задач и их непосредственное решение в рамках любой распределенной вычислительной среды с установленной коммуникационной MPI-средой.

На серии численных экспериментов продемонстрировано успешное использование PD-SAT в решении задач логического криптоанализа ряда поточных систем шифрования, последовательный логический криптоанализ в отношении которых не дал приемлемых результатов.

Предполагается дальнейшее развитие представленной в работе технологии и ее применение в параллельном логическом криптоанализе других систем шифрования.

Литература

1. Мирошник, М.А. Разработка средств защиты информации в распределенных компьютерных системах и сетях [Текст] / М.А. Мирошник // Інформаційно-керуючі системи на залізничному транспорті: наук.-техн. журнал. – Харків: УкрДУЗТ, 2015. – №1. – С. 18-25.,
2. Miroshnik, M. Implementation of cryptographic algorithms on FPGA-based digital distributed systems [Text] / M. Miroshnik // Інформаційно-керуючі системи на залізничному транспорті: наук.-техн. журнал. – Харків: УкрДУЗТ, 2015. – № 2 (111). – С. 25-30.
3. Мирошник, М.А. Разработка интеллектуальной диагностической инфраструктуры в распределенных компьютерных системах [Текст] / М.А. Мирошник // Інформаційно-керуючі системи на залізничному транспорті: наук.-техн. журнал. – Харків: УкрДУЗТ, 2015. – №3. – С. 3-9.
4. Мирошник, М.А. Проектирование компьютерных систем с интеллектуальной диагностической инфраструктурой [Текст] / М.А. Мирошник // Радіотехніка: Всеукр. міжвід. наук.-техн. зб. – 2015. – Вип. 180. – С. 64–67.
5. Miroshnik, M.A. Application of software complex for query processing in the database management system with a view of dispatching problem solving in Grid systems [Text] / M.A. Miroshnik, V.G. Kotukh, S.N. Selevko // Telecommunications and radio engineering. – 2013. – Vol.27, № 10. – P. 875-891.
6. Мирошник, М.А. Синтез распределенных компьютерных сред на базе компьютерных сетей [Текст] / М.А. Мирошник // Системи обробки інформації. – 2013. – №7 (114). – С. 86-89.
7. Miroshnik, M.A. Uses of programmable logic integrated circuits for implementations of data encryption standard and its experimental linear cryptanalysis [Text] / M.A. Miroshnik, M.A. Kovalenko // Інформаційно-керуючі системи на залізничному транспорті: наук.-техн. журнал. – Харків: УкрДАЗТ, 2013. – №6. – С. 36-45.,
8. Мирошник, М.А. Методы защиты цифровой информации в распределенных компьютерных сетях [Текст] / М.А. Мирошник // Інформаційно-керуючі системи на залізничному транспорті: наук.-техн. журнал. – Харків: УкрДАЗТ, 2014. – №5. – С. 66-70.
9. Крылова, В.А. Разработка методов оценки эффективности систем защиты информации в распределенных компьютерных системах [Текст] / В.А. Крылова, А.Н. Мирошник // Інформаційно-керуючі системи на залізничному транспорті: наук.-техн. журнал. – Харків: УкрДУЗТ, 2015. – № 2 (111). – С. 43-51.
10. Мірошник, М.А. Розробка методів оцінки ефективності захисту інформації в розподілених комп'ютерних системах [Текст] / М.А. Мірошник // Інформаційно-керуючі системи на залізничному транспорті: наук.-техн. журнал. – Харків: УкрДУЗТ, 2015. – № 4 (113). – С. 39-43.
11. Мирошник, М.А. Проектирование систем искусственного интеллекта с использованием нечеткой логики [Текст] / М.А. Мирошник, В.Г. Котух, Э.Е. Герман // Радиотехника: Всеукр. межвед. науч.-техн. сборник. – Харьков: ХНУРЭ, 2015. – Вып. 182. – С. 42–50.
12. Мирошник, М.А. Применение интеллектуальной диагностической инфраструктуры для управления кибербезопасностью. Часть 1 Интеллектуализация механизмов защиты [Текст] / М.А. Мирошник, В.А. Крылова, А.И. Демичев // Інформаційно-керуючі системи на залізничному транспорті: наук.-техн. журнал. – Харків: УкрДУЗТ, 2015. – № 6 (115). – С. 25-32.
13. Мирошник, М.А. Применение интеллектуальной диагностической инфраструктуры для управления кибербезопасностью. Часть 2 Поддержка жизненного цикла системы киберзащиты [Текст] / М.А. Мирошник, В.А. Крылова, А.И. Демичев // Інформаційно-керуючі системи на залізничному транспорті: наук.-техн. журнал. – Харків: УкрДУЗТ, 2016. – № 1 (116). – С. 16-25.

Мірошник М.А., Клименко Л.А., Герман Е.Є. Метод паралельного рішення SAT-задач для реалізації процедур прогнозування трудомісткості. У статті розроблена і реалізована великоблочна паралельна технологія рішення SAT-задач у вигляді MPI-програми в розподілених комп'ютерних системах. У даній технології використовується декомпозиція вихідної SAT- задачі на безліч підзадач. В роботі використовується процедура статистичного прогнозування трудомісткості паралельного розв'язання SAT-задач, яка дозволяє визначити оптимальні прогнозовані параметри декомпозиції. Показано, що використання параметрів декомпозиції, знайдених за допомогою процедур прогнозування, дозволяє успішно вирішувати SAT-задачі, які кодують задачі звернення ряду криптографічних дискретних функцій.

Ключові слова: розподілені комп'ютерні системи, SAT-завдання, SAT-вирішувач, паралельна технологія, MPI-програма, прогнозування, що кодують задачі, криптографічна дискретна функція, КНФ, криптоаналіз.

Miroshnyk Maryna A., Klymenko Lubov A., German Eduard E. SAT-tasks parallel solutions method for realization of procedures complexity predicting. The large-block parallel technology of SAT-tasks solutions as MPI-programs in distributed computing systems was designed and implemented in the paper. The decomposition of the original SAT-tasks into multiple subtasks was used in this technology. The statistical procedure for the parallel SAT-solving tasks complexity predicting, which allows to determine the optimal prediction parameters of decomposition was used in the work. It was shown that the use of decomposition parameters that was found using the prediction procedures allows solving successfully the SAT-encoding task handling cryptographic number of discrete functions.

The realization of large-block parallel technology of SAT-tasks solution as PD-SAT MPI-program was presented in the paper. This program allows predicting the complexity of solving SAT-problems and their immediate solution within any distributed computing environment with an installed MPI-communication environment.

The successful use of PD-SAT in solving the problems of logical cryptanalysis of stream encrypting systems, in respect of which sequential logic cryptanalysis did not give acceptable results was demonstrated in the series of numerical experiments.

Further development of presented technology and its application in parallel logical cryptanalysis in other encryption systems was assumed.

Key words: distributed computing system, SAT-problem, the SAT-solver, parallel technology, MPI-program, prediction encoding tasks cryptographic discrete function, CNF, cryptanalysis.

Рецензент Листровой С.В., д.т.н., профессор, профессор кафедры СКС (УкрГУЖТ)

Поступила 02.06.2016 г.

Мірошник М.А., *д.т.н., професор кафедри СКС, Український державний університет залізничного транспорту, Харків, Україна.*
Клименко Л.А., *к.т.н., доцент кафедри СКС, Український державний університет залізничного транспорту, Харків, Україна.*
Герман Е.Є., *к.т.н., доцент кафедри АХТС та ЕКМ, Національний технічний університет «Харківський політехнічний інститут», Харків, Україна.*

Miroschnyk Maryna*, Dr. of tech. science, Ukrainian State University of Railway Transport, Kharkiv, Ukraine.*
Klymenko Lubov A., *PhD, Ukrainian State University of Railway Transport, Kharkiv, Ukraine.*
German Eduard E., *PhD, National Technical University "Kharkiv Polytechnic Institute", Kharkiv, Ukraine.*