УДК 519.7

ПРИМЕНЕНИЕ ДОБРОВОЛЬНЫХ ВЫЧИСЛЕНИЙ К РЕШЕНИЮ КРИПТОГРАФИЧЕСКИХ ЗАДАЧ[1]

О. С. Заикин, М. А. Посыпкин, А. А. Семенов

Добровольные вычисления — это распределённые вычисления, в которых используются вычислительные ресурсы, предоставляемые т. н. «добровольцами» (волонтёрами). В роли добровольцев обычно выступают частные лица, располагающие собственными персональными компьютерами (ПК), ресурсы которых они предоставляют для решения разнообразных научных задач. Компьютеры добровольцев (вообще говоря, находящиеся в географически удалённых друг от друга точках) объединяются в грид под управлением некоторой среды. Такие грид-системы создаются под конкретные задачи, на решение которых могут уходить месяцы и даже годы, и называются проектами добровольных вычислений.

Сегодня функционируют более 70 проектов добровольных вычислений, в которых решаются задачи поиска новых космических объектов (Einstein@home, Milkyway@home), новых лекарств от пока неизлечимых болезней (World Community Grid, Rosetta@home), а также задачи из многих других областей. Несколько проектов направлены на решение криптографических задач. Исторически первым криптографическим проектом стал distributed.net, запущенный в 1997 г. С тех пор в данном проекте решены несколько задач криптоанализа различных модификаций шифра RC5 с использованием метода «грубой силы». В настоящий момент в distributed.net решается задача криптоанализа шифра RC5 с 72-битным секретным ключом. Проект DistrRTgen, запущенный в 2008 г., направлен на построение rainbow-таблиц, предназначенных для криптоанализа хэш-функций. В DistrRTgen разработана версия расчётного приложения для видеокарт, благодаря чему производительность проекта достигла 1,2 пентафлопс.

В 2007 г. запущен проект Enigma@home, направленный на криптоанализ трёх сообщений, закодированных криптографической машиной Энигма и перехваченных в северной Атлантике в 1942 г. Криптоанализ двух сообщений был успешно осуществлён в первые несколько месяцев работы проекта, и с тех пор решается задача криптоанализа третьего сообщения, которое было принято с помехами. При разработке подавляющего большинства действующих проектов добровольных вычислений использована программная платформа BOINC [1], прототипом которой стали технологии, применённые при создании одного из самых крупных проектов добровольных вычислений SETI@home. Эта платформа стала свободно распространяться в 2004 г. Из всех перечисленных проектов только distributed.net не использует платформу BOINC.

В докладе представлено описание проекта добровольных вычислений SAT@home [2], созданного ИДСТУ СО РАН и ИСА РАН и предназначенного для распределённого решения задачи о булевой выполнимости (SAT). Проект был запущен 29 сентября 2011 г. При его создании была использована платформа BOINC [1] и библиотека DC-API [3]. Распределённое приложение проекта состоит из управляющей и расчётной частей. Управляющая часть отвечает за создание заданий в базе данных проекта, а также за обработку результатов выполнения заданий, присылаемых с ПК участников. Отправкой заданий на ПК участников и получением результатов занимаются стандартные

[1]Работа выполнена при поддержке РФФИ (гранты № 11-07-00377-а и № 10-07-00301-а) и Седьмой Рамочной программы Европейского Союза (FP7/2007-2013), грант № 261561 (DEGISCO).

службы BOINC. Основу расчётной части составляют SAT-решатели minisat-1.14.1 и minisat-2.0, модифицированные с учётом особенностей КНФ, кодирующих задачи обращения дискретных функций [4]. Исполняемые файлы расчётной части взаимодействуют с BOINC-клиентом при помощи функций библиотеки DC-API, что позволяет, в частности, реализовать периодическое сохранение промежуточных данных вычислений в виде контрольных точек. На момент написания данной работы в проекте SAT@home принимает участие более тысячи активных добровольцев и задействовано более двух тысяч активно работающих ПК. Средняя производительность проекта за весь период работы составляет 1,7 терафлопс.

В декабре 2011 г. в SAT@home был запущен эксперимент по решению задачи криптоанализа генератора ключевого потока A5/1. На сегодняшний день наиболее эффективным методом криптонализа данного генератора является «rainbow-метод» [5]. Однако доступные rainbow-таблицы [5] покрывают ключевое пространство примерно на 88 % (при «реалистичных» ограничениях на условия криптоанализа). Нами были сгенерированы 10 тестов, не поддающихся криптоанализу с использованием этих rainbow-таблиц. На момент написания данной работы в проекте SAT@home успешно решены 9 из них (в среднем на решение каждого теста уходило 2 недели работы проекта).

ЛИТЕРАТУРА

1. Платформа BOINC для организации добровольных вычислений. `http://boinc.berkeley.edu/`
2. Проект добровольных вычислений SAT@home. `http://sat.isa.ru/pdsat/`
3. *Balaton Z., Gombas G., Kacsuk P., et al.* Sztaki desktop grid: a modular and scalable way of building large computing grids // 21th Intern. Parallel and Distributed Processing Symposium. Long Beach, California, USA, 2007. P. 1–8.
4. *Semenov A., Zaikin O., Bespalov D., and Posypkin M.* Parallel logical cryptanalysis of the generator A5/1 in BNB-Grid system // LNCS. 2011. V. 6873. P. 473–483.
5. A5/1 Cracking project. `http://reflextor.com/trac/a51/wiki`

УДК 004.431:004.432:004.436

АНАЛИЗ ОБЪЕКТНЫХ ФАЙЛОВ DELPHI С ИСПОЛЬЗОВАНИЕМ СПЕЦИФИКАЦИИ СЕМАНТИКИ МАШИННЫХ КОМАНД

А. А. Михайлов

Процесс декомпиляции является важной, а при решении некоторых задач (таких, как поддержка программного обеспечения без возможности использования исходного кода или восстановление исходного кода) и неотъемлемой частью разработки программного обеспечения. В общем случае (для произвольных исполняемых файлов) эта задача является очень сложной, например требуется разделить память программы на код и данные. В объектных файлах Delphi программа оказывается более структурированной, например выделены блоки памяти, соответствующие коду каждой процедуры; имеется информация о типах данных; может присутствовать отладочная информация. Таким образом, при работе с файлами `dcu` [1] задача декомпиляции становится более достижимой.

В общем виде формат файла `dcu` выглядит следующим образом: сначала идёт заголовок, в котором содержится общая информация о файле, такая, как размер, время