К. Г. Когос, А. М. Коренева, А. А. Краснопевцев, В. М. Фомичев

МОДЕЛЬ ПРОГРАММНОГО КОМПЛЕКСА ДЛЯ СИСТЕМЫ РАСПРЕДЕЛЕННЫХ ВЫЧИСЛЕНИЙ[1]

Под распределенными вычислениями подразумевается выполнение трудоемких расчетов с помощью разделения на части вычислительного алгоритма и распределения между множеством компьютеров вычислительной нагрузки при реализации алгоритма.

В ходе выполнения предыдущих этапов НИР был проведен анализ различных математических моделей распределенных вычислений, оценена их эффективность для решения ряда задач криптографического анализа, связанных с трудоемкими переборными задачами (с опробованием большого множества ключей). Метод получения оценок заключается в построении алгоритма решения криптографической задачи с учетом ее реализации в системе распределенных вычислений и подсчете количества единиц времени, необходимого для реализации построенного алгоритма. Для некоторых задач проанализированы возможности оптимизации времени реализации алгоритмов в условиях различных моделей распределенных вычислений [1–3].

Данная работа посвящена вопросам моделирования программного комплекса для распределенных вычислений и исследования программного стенда для оптимизации схем разделения алгоритмов опробования криптографических ключей в условиях различных моделей распределенных вычислений.

Описание программного комплекса. Важным элементом программного комплекса для реализации является система моделирования функционирования системы распределенных вычислений. Разработанное программное обеспечение позволяет моделировать процесс опробования криптографических ключей в рамках системы распределенных вычислений.

Разработанное программное обеспечение реализовано в качестве набора модулей, что позволяет добиться определенной «гибкости» реализации, т. е. возможности изменения элементов приложения (целевой криптографической системы, процесса функционирования координатора вычислений). Реализация приложения допускает замену ряда модулей приложения, не требуя при этом перекомпиляции. Общая архитектура системы моделирования изображена на рис. 1.

Важная особенность системы моделирования — наличие плагинов. Подразумевается, что исследуемый алгоритм шифрования является плагином, т. е. отдельным программным модулем. Это дает возможность динамической загрузки алгоритмов шифрования без остановки функционирования системы опробования. Данный программный комплекс реализован в виде клиент-серверного приложения. В качестве серверной части выступает координатор системы распределенных вычислений, в качестве клиента — вычислительный узел сети.

Опишем состав системы более подробно.

Ядро системы. Реализует задачи коммуникации между модулями, представляет собой общую шину, посредством которой осуществляется взаимодействие остальных модулей. При реализации функций координатора распределенных вычислений ядро загружает остальные модули и при необходимости передает им управление. Альтернативное решение состоит в использовании менеджера модулей, в функции которого входит стартовая загрузка модулей с последующей передачей на них управления. Однако использование общей шины является более предпочтительным и с точки зрения простоты реализации, отладки и тестирования всей разработанной системы.

Модуль работы с данными. Выполняет операции с данными, которыми располагает координатор. При решении криптографических задач такими данными могут быть множества

[1] Работа выполнена в рамках мероприятия 1.2.1 Федеральной целевой программы «Научные и научно-педагогические кадры инновационной России» на 2009–2013 г. по направлению «Распределенные вычислительные системы».

опробуемых ключей криптосистемы и шифрматериал (открытый и шифрованный тексты). Результатом вычислений могут быть описание характеристик ключевого подмножества вкупе с характеристиками вычислителей. В качестве системы управления данными может быть использована любая из известных систем (SQL СУБД, NoSQL, XML-файлы и т. д.).

В зависимости от реализации модуля могут применяться различные системы хранения данных. В рамках данной НИР использован файл для хранения необходимой информации. Этот модуль должен быть реализован как хранилище, предоставляющее более широкий набор функций по работе с данными приложения.

Модуль поддержки сети. Модуль обеспечивает взаимодействие координатора с участниками вычислений. Реализация взаимодействия с помощью отдельного модуля целесообразна в связи с тем, что сетевое взаимодействие «клиент—сервер» может быть достаточно сложным и осуществляться с использованием разнообразных технологий и протоколов, в том числе таких, которые обеспечивают противодействие компрометации результатов вычислений.

Интерфейс оператора. Разработанная система является автоматизированной (но не автоматической), поэтому данный модуль необходим для обеспечения взаимодействия системы с оператором, осуществляющим контроль за процессом выполнения распределенных вычислений. Для многих задач опробования ключей функции оператора весьма просты: запуск процесса опробования и его прерывание. Данный модуль наглядно отображает информацию о ходе и результатах вычислительного процесса.

Менеджер плагинов. Этот модуль обеспечивает возможность работы программного комплекса с различными криптографическими системами, реализуемыми в виде модуля. С помощью такого подхода можно существенно расширить область использования системы.

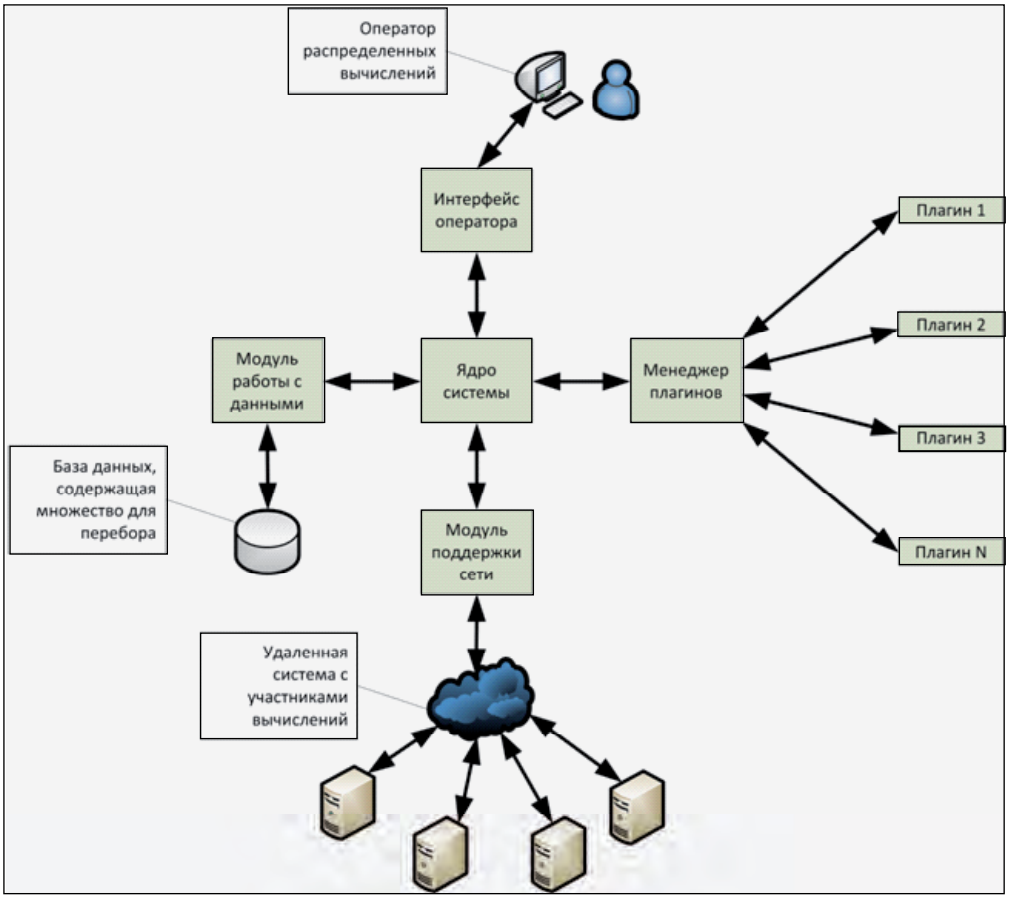

Рис. 1. Архитектура системы моделирования

Рассмотренный принцип построения среды распределенных вычислений позволяет без необходимости доработки всей системы в целом осуществлять ее модификацию. За счет выделенного модуля работы с сетью имеется возможность использовать программный комплекс для реализации различных классов распределенных вычислений. С помощью замены данного модуля система распределенных вычислений может быть реализована и как кластер, и как распределенная Грид-сеть, вычислительные узлы которой географически удалены от координатора вычислений и сетевое взаимодействие осуществляется с использованием технологий и возможностей, предоставляемых сетью Интернет.

Построение модульной архитектуры несет дополнительные возможности в разделении физического расположения модулей, т. е. в расположении отдельных модулей на различных серверах. Тем самым в рамках данной архитектуры достигается возможность горизонтального масштабирования предлагаемой системы.

Экспериментальное исследование ПО. При выполнении распределенных вычислений координатор реализации распределенного алгоритма решает следующие задачи: определяет схему разделения алгоритма на части, определяет порядок обмена данными и результатами вычислений с исполнителями, обеспечивает возможность использования при вычислениях различных вычислительных платформ с целью привлечения наиболее широкого круга участников. Разделение алгоритма на части в общем случае не сводится к распараллеливанию вычислений, допустимо также разделение алгоритма и на последовательные фрагменты в сочетании с принципом конвейеризации вычислений.

Решение задачи по определению схемы разделения алгоритма на части, решаемой координатором, зависит от выбранной математической модели распределенных вычислений и не должно занимать слишком много времени. Следовательно, определение схемы разделения алгоритма на части и оптимизацию этой схемы целесообразно автоматизировать. Важным компонентом программного обеспечения является приложение, осуществляющее построение разбиений ключевого множества с целью разделения задания между участниками. Основная задача таких программ — построить ключевые подмножества для опробования участниками в условиях заданной модели распределенных вычислений. Входными данными для разработанного приложения являются данные о ключевом множестве (число ключей, вычислительная сложность опробования отдельного ключа) и данные о вычислительных ресурсах системы распределенных вычислений, к которым относятся в основном количество участников вычислений и их производительность.

Для определенного класса алгоритмов опробования криптографических ключей был разработан набор утилитарного программного обеспечения. В состав программного стенда входят программы, оптимизирующие схемы разделения алгоритма на части в смысле минимизации времени работы алгоритма в целом. Во многих случаях разделение алгоритма на части представляет собой разбиение опробуемого ключевого множества.

Оптимизация разбиения ключевого множества между участниками состоит в том, чтобы время работы каждого участника было примерно одинаковым. В этом случае каждый вычислитель распределенной сети завершает процесс опробования приблизительно в один и тот же момент времени. Для минимизации времени опробования ключей системой в целом было разработано соответствующее программное обеспечение.

С целью оценки качества разработанного ПО был выполнен ряд вычислительных экспериментов. Экспериментальные исследования направлены на оценку вычислительных возможностей системы в условиях различных моделей распределенных вычислений.

Для описания полученных результатов используем обозначения: N — число участников системы; K — мощность ключевого множества; ϖ_i — производительность процессора i-го вычислителя, $I = 1,\ldots,N$; T — время построения оптимального разбиения.

При различных ограничениях на данные параметры проанализировано время построения разбиения ключевого множества и время опробования ключей.

Результаты (в зависимости от K и количества участников, сгруппированных по производительности) представлены в таблице 1.

Таблица 1. Время T для участников, сгруппированных по производительности

Количество участников в группе	**K = 2^{56}**	**K = 2^{128}**	**K = 2^{192}**	**K = 2^{256}**
100	21 мс	38 мс	50 мс	66 мс
200	28 мс	36 мс	41 мс	58 мс
300	26 мс	24 мс	35 мс	49 мс
400	26 мс	38 мс	49 мс	49 мс
500	25 мс	38 мс	49 мс	61 мс
600	26 мс	38 мс	37 мс	67 мс
700	24 мс	41 мс	51 мс	65 мс
800	26 мс	35 мс	42 мс	71 мс
900	26 мс	50 мс	48 мс	53 мс
1000	26 мс	32 мс	57 мс	53 мс

В таблице 2 представлены результаты в случае, когда участники распределенных вычислений имеют одинаковую производительность.

Таблица 2. Время T для участников с одинаковой производительностью

Количество участников в группе	**K = 2^{56}**	**K = 2^{128}**	**K = 2^{192}**	**K = 2^{256}**
10^4	53 мс	57 мс	79 мс	102 мс
10^5	146 мс	185 мс	358 мс	457 мс
10^6	726 мс	1385 мс	1887 мс	2484 мс

В ходе экспериментального исследования ПО построены графики зависимости времени построения оптимального разбиения от числа участников при фиксированной величине K (рис. 2) и от величины K при фиксированном числе участников (рис. 3).

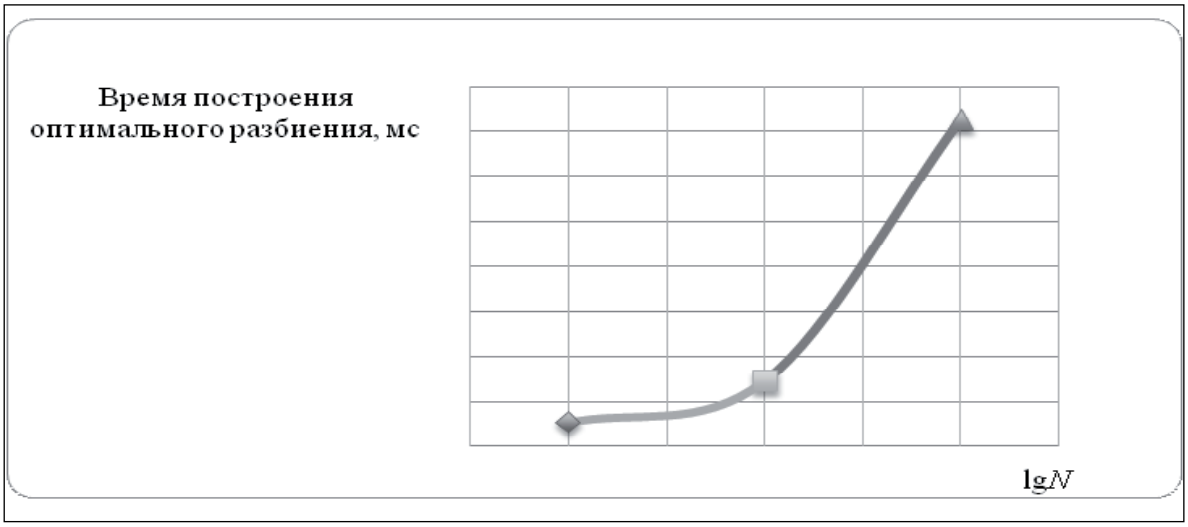

Рис. 2. Зависимость времени T от числа участников при $K = 2^{56}$

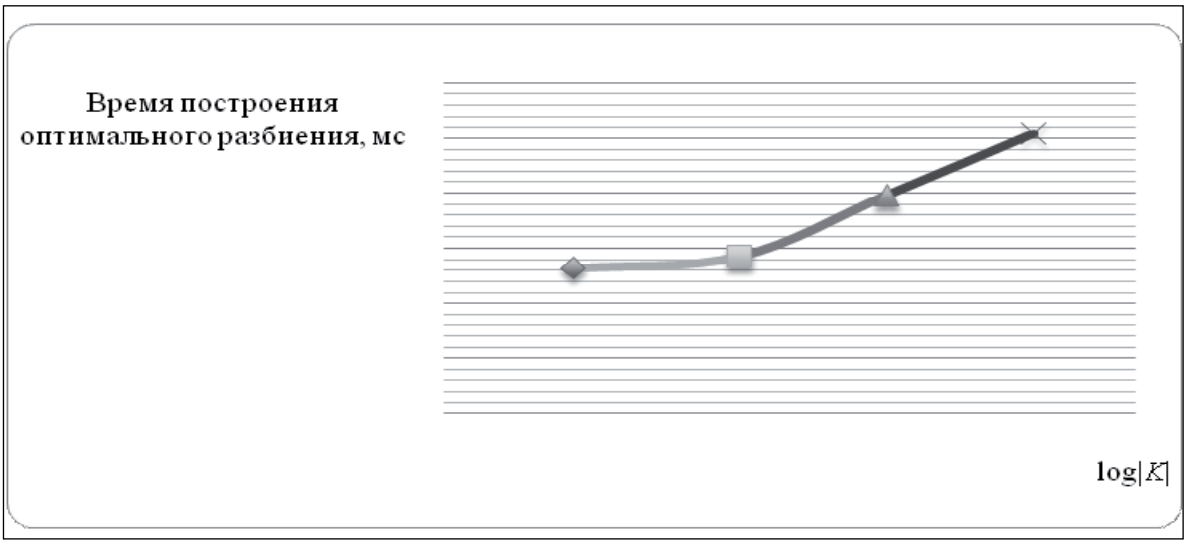

Рис. 3. Зависимость времени T от величины K при $N = 10^4$

Выполнены исследования для усложненной модели распределенных вычислений, в которой при одинаковом количестве $N = 10^6$ участников возможны различные варианты разбиения множества участников по производительностям.

Зависимость времени T от различных вариантов разбиений множества участников на классы по производительности отражена в таблице 3.

Таблица 3. Время T в зависимости от K и от способов разбиения множества участников на группы по производительности

Число классов по производительности	**Число участников в классе**	**$K = 2^{56}$**	**$K = 2^{128}$**	**$K = 2^{192}$**	**$K = 2^{256}$**
2	$5 \cdot 10^5$	714 мс	1314 мс	1871 мс	2466 мс
5	$2 \cdot 10^5$	691 мс	1188 мс	1773 мс	2449 мс
10	10^5	663 мс	1191 мс	1906 мс	2438 мс
10^5	10	805 мс	1300 мс	1971 мс	2531 мс

Выводы

1. Вычислительная сложность построения оптимальных разбиений ключевого множества в рассмотренных задачах линейно зависит от мощности ключевого множества. Аналогичный рост сложности наблюдается и при общем увеличении числа участников вычислений. Вместе с тем сложность построения разбиений не зависит от количества классов, на которые разбиваются участники по производительности.

2. В задачах опробования время построения оптимального разбиения ключевого множества невелико. Это показывает, что в общем объеме распределенных вычислений время, затрачиваемое координатором на составление оптимальных заданий, является несущественным.

СПИСОК ЛИТЕРАТУРЫ:

1. *Фомичев В. М.* О методе согласования для анализа блочных шифров с помощью распределенных вычислений // Безопасность информационных технологий. 2011. № 1. С. 16–20.
2. *Варфоломеев А. А., Коренева А. М., Краснопевцев А. А., Туманов Ю. М., Фомичев В. М.* О реализации метода полного опробования ключей криптосистем в условиях различных математических моделей. Материалы XVIII Всероссийской научно-практической конференции // Безопасность информационных технологий. 2011. № 1. С. 82–84.
3. *Варфоломеев А. А., Когос К. Г., Коренева А. М., Фомичев В. М.* О сложности реализации некоторых алгоритмических методов криптоанализа с помощью распределенных вычислений // Безопасность информационных технологий. 2011. № 4. С. 28–31.