ISSN 2079-3316 ПРОГРАММНЫЕ СИСТЕМЫ: ТЕОРИЯ И ПРИЛОЖЕНИЯ № 3(17), 2013, с.25–42

УДК 004.056.5

Л. К. Бабенко, Е. А. Ищукова, И. Д. Сидоров

Применение параллельных вычислений при решении задач защиты информации[1]

Аннотация. В настоящей работе рассматриваются вопросы применения распределенных многопроцессорных вычислений для сокращения времени анализа современных криптографических систем защиты информации. Приводятся экспериментальные данные, полученные на основе реализации параллельных алгоритмов анализа симметричных и ассиметричных алгоритмов шифрования.

Ключевые слова и фразы: криптография, криптоанализ, секретный ключ, блочный шифр, стойкость, распределенные многопроцессорные вычисления, методы факторизации, решето числового поля, параллельное просеивание, гауссово исключение.

Введение

Практически все программно-аппаратные методы защиты информации неразрывно связаны с криптографией. Несмотря на то, что криптография существует уже много веков, в том виде, в котором мы ее знаем и используем сейчас, она насчитывает несколько десятилетий. Можно сказать, что основным (хотя и не единственным) направлением развития современной криптографии является создание стойких алгоритмов шифрования. Все существующие шифры по принципу построения и использования секретного ключа разделяют на симметричные и асимметричные. Любой разрабатываемый алгоритм шифрования еще на этапе проектирования подвергается тщательному анализу с целью выявления его слабых мест и возможности взлома. Для того чтобы иметь возможность оценить стойкость используемого шифра необходимо наличие эффективных алгоритмов анализа.

[1] Работа выполнена при поддержке грантов РФФИ № 12-07-00037-а, № 12-07-31120-мол_а, № 12-07-33007_мол_а_вед.

На сегодняшний день существует довольно много различных методов анализа симметричных блочных шифров, основанных на различных подходах. К наиболее известным методам анализа можно отнести такие как, например, метод линейного анализа, метод дифференциального анализа, метод невозможных дифференциалов, бумеранг-атака, алгебраические методы анализа, метод слайдовой атаки.

Для асимметричных криптосистем также существует достаточно большое разнообразие методов. Среди них наиболее известны такие методы как метод Гельфонда, «giant step-baby step», метод встречи на случайном дереве, метод базы разложения, метод решета числового поля, метод Ферма, метод непрерывных дробей, метод квадратичного решета и др. Однако если при анализе симметричных криптосистем различные методы используют различные приемы, такие как линеаризация, рассмотрение пар текстов, составление систем переопределенных уравнений, то при анализе асимметричных криптосистем все методы сводятся к решению двух задач различными способами — задачи дискретного логарифмирования и задачи факторизации больших чисел.

С появлением мощных вычислительных ресурсов задача анализа современных криптосистем превратилась из чисто теоретической в практическую. При этом многие из вышеуказанных методов поддаются распараллеливанию, а значит, могут работать в несколько раз быстрее при использовании соответствующих вычислительных средств. Одним из способов повышения производительности при анализе различных криптосистем является использование распределенных многопроцессорных вычислений (РМВ) для ускорения процесса анализа и скорейшего получения результата. Применение РМВ возможно как при криптоанализе симметричных блочных шифров, так и при использовании методов анализа современных асимметричных криптосистем. Основой использования распределенных вычислений могут служить современные пакеты прикладных программ для кластерных систем. В данной работе рассматриваются реализации на основе использования стандарта «Интерфейс

передачи сообщений» (Message Passing Interface, MPI). MPI выбран за свою многоплатформенность, удобный интерфейс, гибкую конфигурацию и легкую переносимость с одной вычислительной машины на другую. Параллельная программа в модели передачи сообщений представляет собой набор обычных последовательных программ, которые обрабатываются одновременно. Обычно каждая из этих последовательных программ выполняется на своем процессоре и имеет доступ к своей, локальной памяти [[1]]. Явным достоинством при такой организации вычислений является возможность написания и отладки программы на однопроцессорной системе. Также необходимо отметить, что для увеличения быстродействия вычислений необходимо использовать компьютеры, имеющие одинаковую конфигурацию, и соединенные между собой высокоскоростной линией связи.

При разработке программ для тестирования алгоритмов шифрования с помощью PMB, необходимо учитывать такие особенности, как число раундов, используемых в тестируемом алгоритме и количество данных, требуемое для успешного проведения атаки. Как правило, криптоаналитические атаки проходят в два этапа. На первом этапе выполняется первичная обработка параметров алгоритма и подготовка всех данных для проведения анализа, которые обычно осуществляет один процессор, называемый главным. Второй этап состоит в непосредственном анализе алгоритма, что в большинстве случаев сводится к нахождению секретного ключа, использованного для шифрования данных с помощью исследуемого алгоритма. При этом должна быть организована правильная и грамотная взаимосвязь частей программы, выполняющих вышеуказанные этапы. Кроме того, программа должна позволять использовать в вычислениях любое число процессоров, при этом с помощью разработанного алгоритма данные для анализа должны распределяться равномерно.

В настоящей статье мы постараемся подвести итог многолетней работы по исследованию различных криптографических алгоритмов с использованием PMB.

1. Анализ симметричных алгоритмов шифрования

1.1. Краткие сведения о методе дифференциального анализа

Метод дифференциального криптоанализа впервые был предложен в начале 90-х годов прошлого века Э. Бихамом и А. Шамиром для анализа алгоритма шифрования DES [[2], [3]]. В общем виде дифференциальный анализ блочных алгоритмов шифрования сводится к следующим основным пунктам:

(1) Нахождение для алгоритма шифрования характеристик, обладающих максимальными вероятностями. Поиск характеристик ведется на основе дифференциальных свойств нелинейных криптографических примитивов, входящих в состав алгоритма шифрования;
(2) Поиск правильных пар текстов с использованием найденных характеристик;
(3) Анализ правильных пар текстов и накопление статистики о возможных значениях секретного ключа шифрования.

Первый пункт, заключающийся в поиске лучших характеристик для большинства алгоритмов, выполняется единожды и является теоретической задачей. Значения характеристик полностью зависят от структуры алгоритма шифрования и используемых криптографических примитивов. Иначе дело обстоит лишь с теми алгоритмами, которые обладают нефиксированными элементами. К таким алгоритмам можно, например, отнести алгоритм шифрования ГОСТ 28147-89, у которого S-блоки замены могут выбираться произвольным образом. Для таких алгоритмов поиск характеристик необходимо каждый раз начинать сначала, основываясь на дифференциальных свойствах выбранных S-блоков. Для автоматизации процесса анализа можно разработать алгоритм поиска лучших характеристик, основываясь на алгоритмах поиска по дереву. Для таких алгоритмов можно использовать параллельные модели для ускорения поиска характеристик.

Второй шаг анализа является вычислительно стойкой задачей для любого алгоритма шифрования, при этом не важно, обладает

он фиксированными или нефиксированными элементами. Анализ заключается в опробовании большого числа пар текстов с целью определения являются ли они правильной парой текстов, то есть той парой текстов, которую в дальнейшем можно использовать для анализа с целью поиска секретного ключа шифрования. Данный шаг может и должен быть легко представим в виде параллельных вычислений для сокращения времени анализа.

Последний шаг легко реализуем, требует гораздо меньше вычислений в сравнении со вторым шагом. Он может быть реализован как отдельно в виде последовательного алгоритма, так и быть включенным в состав параллельных алгоритмов по поиску правильных пар текстов. В последнем случае при нахождении правильной пары текстов сразу можно провести ее анализ по накоплению статистики о возможном значении секретного ключа.

Более подробно о дифференциальном криптоанализе различных блочных алгоритмов шифрования, а также о различных производных данного метода можно почитать в работах [[2]–[5]].

1.2. Дифференциальный криптоанализ алгоритма

На основе методов и подходов, предложенных Э. Бихамом и А. Шамиром [[2], [3]] было разработано семейство детальных программно-ориентированных последовательных и параллельных алгоритмов анализа алгоритма шифрования DES с целью поиска секретного ключа шифрования. Более подробно с данными алгоритмами можно ознакомиться в работе [[6]].

На основе разработанных алгоритмов проведения дифференциального криптоанализа DES была реализована программа, позволяющая осуществлять анализ любого числа раундов шифрования с помощью PMB. Данная программа разрабатывалась в среде программирования Microsoft Visual C++ 6.0 в связи с требованиями используемого для многопроцессорных вычислений пакета MPICH 1.2.5. С помощью данной программы был проверен алгоритм анализа 6 раундов шифра DES с использованием наиболее вероятных дифференциалов. Экспериментальные данные показали, что анализ 6-раундового алгоритма DES всегда дает правильный

результат. Время анализа на 2-процессорной системе (с частотой процессоров 2,67 ГГц) в среднем занимает 7,5 минут, на 16-процессорной — 56 секунд. Проверка проводилась для разных значений секретного ключа. Время анализа для разных значений секретного ключа на n-процессорной системе разнилось в среднем на 3–5 секунд, что обусловлено используемой операционной системой и средой передачи данных, и является естественной погрешностью времени вычислений. Второе проводимое испытание — это полный анализ алгоритма DES, состоящего из 8, 10, 12, 14 и 16 раундов на 2-х, 3-х, 4-х и 5-ти процессорной системе с частотой процессоров 2,67 ГГц. Результаты экспериментов приведены в Таблица 1. Результаты анализа алгоритма шифрования DES

Таблица 1. Результаты анализа алгоритма шифрования DES

Число раундов	Число процессоров	Время анализа	Количество найденных пар текстов	Количество правильных пар текстов
8	2	125 часов 17 минут	253	240
	3	103 часа 11 минут		
	4	78 часов 47 минут		
	5	60 часов 31 минута		
10	2	131 час 28 минут	113	97
	3	105 часов 58 минут		
	4	81 час 35 минут		
	5	63 часа 43 минуты		
12	2	137 часов 18 минут	37	34
	3	108 часов 16 минут		
	4	84 часа 53 минуты		
	5	65 часов 48 минут		
14	2	142 часа 37 минут	5	4
	3	111 часов 57 минут		
	4	88 часов 27 минут		
	5	67 часов 13 минут		
16	2	148 часов 23 минуты	1	1
	3	115 часов 23 минуты		

	4	90 часов 15 минут		
	5	71 час 12 минут		

Так как каждый эксперимент (особенно при малом числе используемых процессоров) представляет собой достаточно длительный процесс, то эксперименты проводились для одного значения секретного ключа K_L=2882400171, K_R=3455036365. Из Таблица 1. Результаты анализа алгоритма шифрования DES видно, что при увеличении числа процессоров, принимающих участие в вычислениях, наблюдается практически линейный рост ускорения времени анализа. Это же можно видеть и из графика, приведенного на Рис. 1. Этот график построен по данным Таблица 1. Результаты анализа алгоритма шифрования DES. По оси абсцисс отложено число процессоров, а по оси ординат — время, затрачиваемое на анализ.

Для 16-процессорного кластера с частотой процессоров 1,41 ГГц время работы программы для полного 16-раундового алгоритма DES составило 24 часа 13 минут.

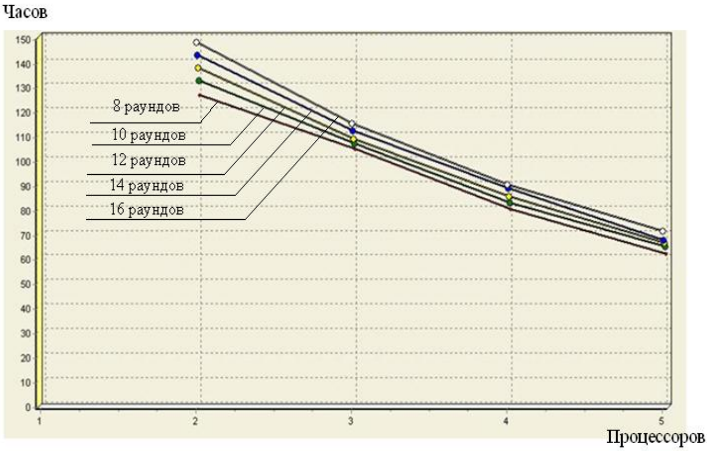


Рис. 1. Время анализа n-раундов алгоритма DES

1.3. Дифференциальный анализ алгоритма ГОСТ 28147-89

Отличительной особенностью алгоритма ГОСТ является наличие в его составе нефиксированных блоков замены, которые каж-

дый раз могут быть разными. В связи с этим для алгоритма ГОСТ первый этап дифференциального криптоанализа по поиску характеристик с максимальными вероятностями необходимо начинать сначала. В результате работы нами был разработан параллельный алгоритм поиска характеристик, обладающих максимальными вероятностями. Подробно с данным алгоритмом можно ознакомиться в работе [[6], [7]].

В результате работы было реализовано две программы на основе разработанного параллельного алгоритма: одна с использованием статического распределения данных, другая — с использованием динамического. Работа обеих программ была протестирована с помощью 16-процессорного кластера (с частотой процессоров 1,41 ГГц). В ходе тестовых экспериментов проводились замеры скорости вычислений для разных начальных условий: числа процессоров, участвующих в вычислениях, количества раундов шифрования и начального значения пороговой вероятности. В условиях реального времени удалось получить результаты для 8-раундового алгоритма шифрования ГОСТ 28147-89. При этом для каждого n-раундового алгоритма (где $n \geq 8$) проводились замеры времени для двух разных начальных значений пороговой вероятности: равной 0 и отличной от 0.

Результаты экспериментов для программы со статическим распределением данных сведены в графики, представленные на Рис. 2. Процесс №0 отображает общее время работы программы в целом. Если сопоставить рост его времени вычислений с остальными процессами, то становится видно, что до 5 раундов шифрования объем возможных выходных данных увеличивается не так резко, как для всех последующих, что и отражает график. На графиках не представлены точки для 7-ми и 8-ми раундового алгоритма шифрования, так как время их вычислений значительно выше времени для 2-х, 3-х, 4-х, 5-ти и 6-ти раундов. Поэтому, если представить полученные данные все вместе, не будет видно динамики изменения скорости вычисления для алгоритмов шифрования с числом раундов меньше 7. Из Рис. 2 также видно, что до 5-ти ра-

ундов для разных процессов (в частности для 4-го, 6-го и других) время вычислений не всегда увеличивается с ростом числа раундов шифрования, а в некоторых случаях даже существенно уменьшается. Это связано с несколькими факторами, в частности: со сравнительно небольшим приростом вычислений при увеличении числа раундов шифрования, с погрешностью передающей среды, со спецификой межпроцессорного обмена пороговыми вероятностями. Как показали эксперименты, наиболее интенсивный межпроцессорный обмен происходит в первые 15 секунд вычислений (что как раз охватывает время вычислений до 5-раундового алгоритма включительно). В дальнейшем обмен происходит лишь изредка, когда один из процессов находит вероятность, превышающую пороговую. При анализе алгоритма шифрования с числом раундов больше 7 таких скачков не наблюдается.

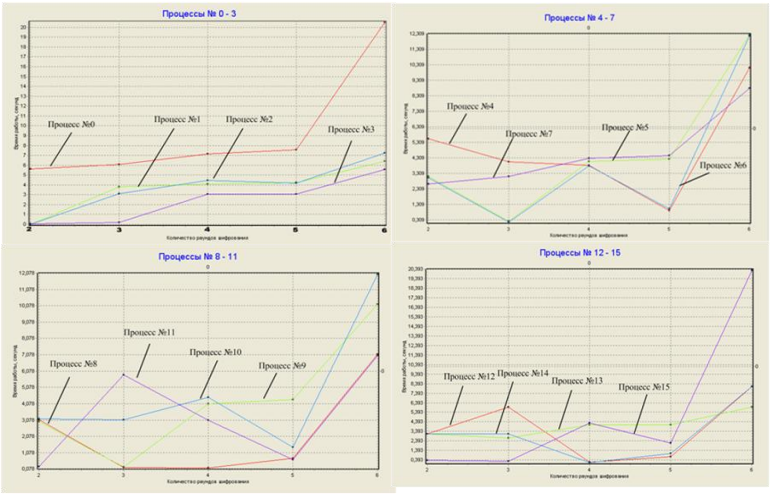


Рис. 2. Результаты экспериментов для программы со статическим распределением данных

Результатом работы программы является одна или несколько пар входная выходная вероятность, имеющая максимальную вероятность. Для 7-раундового алгоритма с фиксированными блоками

замены была найдена всего одна такая пара с вероятностью p = 1.591252e-008.

Для программы с динамическим распределением данных скорость работы для n-раундового алгоритма шифрования, где n<7, оказалась несколько лучше, чем для статического распределения (в Таблица 2 представлены экспериментальные данные для 6-раундового алгоритма). При этом все процессоры получили одинаковую нагрузку, а соответственно примерно одинаковое время работы, в отличие от алгоритма статического распределения, где некоторые процессы считали быстрее остальных. Однако при семи раундах время работы значительно превышает время работы программы со статическим распределением. И это при том, что все процессоры имеют одинаковую загрузку.

Таблица 2. Экспериментальные данные для алгоритма с динамическим распределением данных

Номер процесса	1	2	3	4
Кол-во проанализированных значений	6	5	7	4
Время работы, секунд	10.63264	10.78271	11.09175	11.96433
Номер процесса	5	6	7	8
Кол-во проанализированных значений	5	13	6	10
Время работы, секунд	10.27041	11.53331	11.40407	10.84244
Номер процесса	9	10	11	12
Кол-во проанализированных значений	7	4	11	6
Время работы, секунд	10.75531	10.13938	10.22375	10.93916
Номер процесса	13	14	15	16
Кол-во проанализированных значений	6	10	8	11
Время работы, секунд	10.65408	14.16819	10.77713	13.83076

На первый взгляд кажущаяся невозможность такого события объясняется просто: при статическом распределении процессы для анализа получают значения из разных диапазонов. И одно из этих значений позволяет практически сразу определить пару входная –

выходная разность, имеющую вероятность, сильно приближенную к максимально возможной. Вследствие межпроцессорного обмена это значение вероятности становится пороговым для всех процессов. Поэтому дальнейший анализ проходит с большим отсечением заведомо плохих пар, то есть пар, имеющих вероятность ниже текущего порогового значения. При динамическом распределении данных такое хорошее значение вероятности сразу не определяется, поэтому процессам приходится осуществлять перебор практически по полному дереву, что существенно тормозит процесс анализа.

Исходя из этого, можно сделать вывод, что эффективность применения разработанных алгоритмов зависит не только от числа используемых процессоров и количества раундов шифрования, но и от способа распределения данных анализа. Чем быстрее будет найдена вероятность, максимально приближенная к наилучшей, тем быстрее будет проведен анализ. При этом также необходимо помнить, что различные блоки замены имеют различные значения при дифференциальном криптоанализе, поэтому для разных наборов блоков замены, разные входные вероятности будут давать наилучшую вероятность при прохождении через раунды шифрования.

В Таблица 3 представлены данные, полученные для 6-раундового алгоритма шифрования с пороговым значением вероятности, равным 0, с использованием различного числа процессоров.

Таблица 3. Временные показатели работы программы на разном числе процессоров

Число процессоров	2	3	4	5	6	7	8	9
Статическое распределение	54.2	57.1	30.8	36.6	24.4	24.6	24.7	20.8
Динамическое распределение	63.6	40.7	27.2	22.9	21.2	20.2	19.9	17.6
Число процессоров	10	11	12	13	14	15	16	
Статическое распределение	22.06	18.5	15.5	18.4	19.8	18.8	18.8	
Динамическое распределение	15.5	16.5	18.1	15.6	17.3	13.4	14.3	

Для наглядности данные этих экспериментов представлены также в виде графика на Рис. 3. Из Рис. 3 видно, что не всегда большее число процессоров приводит к снижению времени расчетов. Однако при использовании 16 процессоров для анализа с помощью статического распределения данных время вычислений сокращается в 2,88 раза, с помощью динамического — в 4,4 раза по сравнению с такими же расчетами на двухпроцессорной системе.

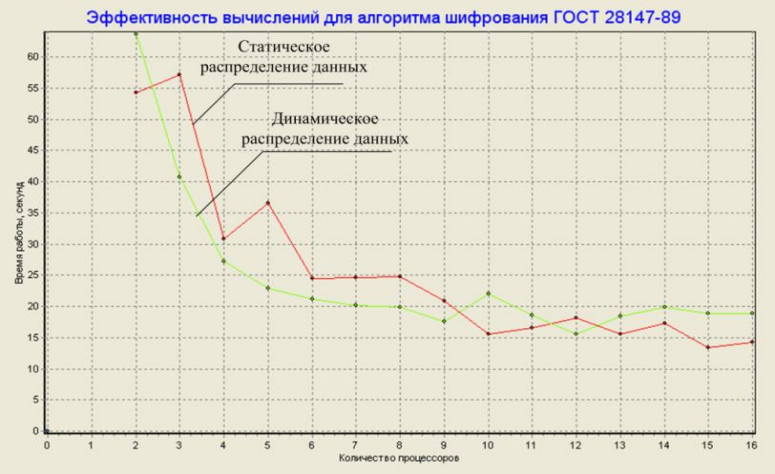


Рис. 3. Зависимость времени вычислений от числа используемых процессоров

Для вычислений с динамическим распределением данных из графика на Рис. 3 видно, что наблюдается практически линейный рост ускорения. Для вычислений с помощью 8 процессоров и меньше значение эффективности находится в пределах от 1,2 до 0,8.

2. Анализ асимметричных криптосистем

2.1. Решение задачи дискретного логарифмирования методом базы разложения

Метод базы разложения для вычисления дискретного логарифма в группе $\mathbf{F}_p^*$ основан на вложении этой группы в полугруппу $\mathbf{Z}^*$ кольца целых чисел. По определению гомоморфизма указанных полугрупп из равенства $AB = C$ в кольце $\mathbf{Z}$ вытекает $AB \equiv C \pmod p$, из равенства $Ax = B$ и сравнения $Ay \equiv B \pmod p$ следует $x \equiv y \pmod r$, где r — простой порядок группы, образованной элементом A [[8], [9]].

В методе базы разложения решения задачи дискретного логарифмирования можно выделить три основных этапа: построение базиса разложения, устранение ненулевых показателей на основе метода Гаусса, решение полученного сравнения. Этапы 1 и 2 являются вычислительно сложными, поэтому для ускорения их выполнения были разработаны параллельный алгоритм нахождения D-гладких степеней (просеивания) и параллельный алгоритм Гаусса для устранения ненулевых показателей при элементах базиса разложения. Подробнее с данными алгоритмами можно ознакомиться в работах [[10]–[11]]. На Рис. 4 представлены результаты экспериментов, показывающие эффективность распараллеливания первого и второго этапов алгоритма.

Эксперименты с помощью программной модели, реализованной на основе разработанных алгоритмов, выполнялись на кластере кафедры безопасности информационных технологий (БИТ) ЮФУ (20 вычислительных ядер Intel Xeon 2.33 GHz, 10 Гб ОЗУ, передающая среда Gigabit Ethernet). Для построения данных графиков находился логарифм в группе, образованной простым числом, содержащим 70 двоичных разрядов. Размер базиса был равен 800.

Из графиков видно, что первый этап вычислений очень хорошо поддаётся распараллеливанию, на втором этапе прирост производительности не столь высок. Фактически, при увеличении числа процессов выше 10 наблюдается снижение производительности. Это вызвано тем, что накладные расходы на передачу опорной строки

процессам превалируют над выигрышем в производительности за счёт параллельной обработки участков матрицы. Уменьшая размер базиса, можно уменьшить размерность матрицы и время выполнения второго этапа, за счёт увеличения сложности просеивания. Также хороший результат должно дать использование более производительной среды передачи данных, например, InfiniBand.

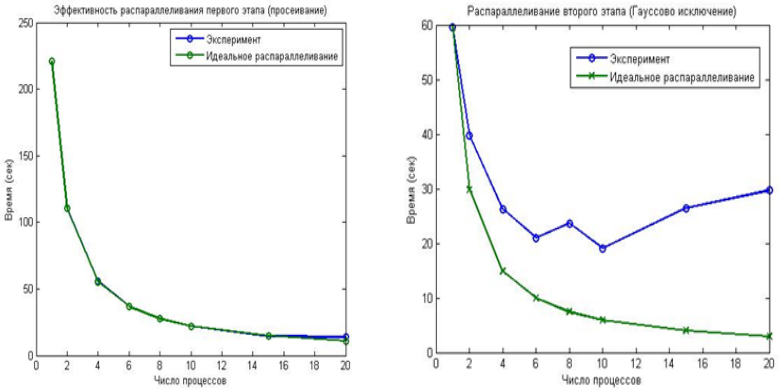


Рис. 4. Распараллеливание первого и второго этапов

2.2. Решение задачи дискретного логарифмирования с помощью метода решета числового поля

Метод общего решета числового поля для вычисления дискретного логарифма был предложен Гордоном и развит Вебером [[8], [9]]. Метод использует однозначность разложения на простые идеалы в кольцах целых алгебраических чисел. Кратко, поиск дискретного логарифма с помощью метода решета числового поля можно свести к следующим этапам: построение базы разложения, просеивание базы разложения, преобразование матрицы показателей на основе метода Гаусса, решить полученное сравнение. Этапы 2 и 3 являются вычислительно сложными, поэтому для ускорения их выполнения были разработаны параллельный алгоритм нахождения D-гладких чисел (просеивания) и параллельный алгоритм

Гаусса. Более подробно с данными алгоритмами можно ознакомиться в работах [[10]–[11]].

В ходе исследования были реализованы разработанные алгоритмы на языке C++ с использованием свободных библиотек OpenMPI (для обеспечения межпроцессного взаимодействия), NTL и GMP (для работы с целыми числами произвольной длины). На графиках, приведенных на Рис. 5, приводится зависимость времени вычислений от числа процессоров в многопроцессорной вычислительной системе. Эксперимент проводился на кластере, состоящем из 20 процессорных ядер Intel Xeon 2,6GHz, 10 Гб ОЗУ, передающая среда Gigabit Ethernet.

Сравнительное тестирование с реализацией метода базы разложения показало, что при относительно небольших размерах модуля p (70–75 бит) метод базы разложения выполняет просеивание заметно быстрее. Время работы реализации решета числового поля сильно зависит от размера базиса, но даже при минимальных его размерах превосходит время работы реализации метода базы разложения. Время, затраченное на преобразования матрицы, у обеих реализаций примерно одинаково. Это согласуется с теоретическими результатами, согласно которым метод решета числового поля начинает работать быстрее метода базы разложения только при достаточно больших размерах модуля.

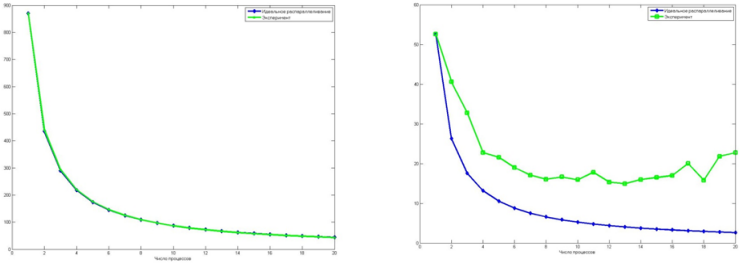


Рис. 5. Прирост производительности при распараллеливании первого (слева) и второго (справа) этапов просеивания и преобразования матрицы

Заключение

В работе рассмотрены вопросы применения распределенных многопроцессорных вычислений для сокращения времени анализа современных криптографических систем защиты информации. Приведены экспериментальные данные, полученные на основе реализации параллельных алгоритмов анализа симметричных и ассиметричных алгоритмов шифрования.

Список литературы

[1] Немнюгин С., Стесик О. Параллельное программирование для многопроцессорных вычислительных систем. — СПб: БХВ-Петербург, 2002.

[2] Biham E., Shamir A. *Differential Cryptanalysis of the Full 16-round DES* // Crypto'92/ — Springer-Velgar, 1998, — P. 487.

[3] Biham E., Shamir A. Differential Cryptanalysis of DES-like Cryptosystems. Extended Abstract // Crypto'90. — Springer-Velgar, 1998. — P. 2.

[4] Панасенко С. Алгоритмы шифрования. Специальный справочник. — СПб.:БХВ-Петербург, 2009. — 576 с.

[5] Бабенко Л. К. Ищукова Е. А. Современные алгоритмы блочного шифрования и методы их анализа. — М.: Гелиос АРВ, 2006.

[6] Бабенко Л. К., Ищукова Е. А. Анализ современных криптографических систем с помощью метода дифференциального криптоанализа. Актуальные аспекты защиты информации в Южном федеральном университете. Монография. — Таганрог: Изд-во ТТИ ЮФУ, 2011. — С. 102–181.

[7] Бабенко Л. К. Ищукова Е. А. *Применение рекурсивного алгоритма поиска в Б-деревьях для дифференциального криптоанализа алгоритма шифрования ГОСТ 28147-89* // Материалы IX Международной научно-практической конференции «ИБ». Ч. 2. — Таганрог, Изд-во: ТТИ ЮФУ, 2007. — С. 92–97.

[8] Маховенко Е. Б. Теоретико-числовые методы в криптографии: Учебное пособие. — М.: Гелиос АРВ, 2006. — 320 с.

[9] Ростовцев А. Г., Маховенко Е. Б. Теоретическая криптография. — СПб.: АНО НПО «Профессионал», 2005. — 480 с.

[10] Бабенко Л. К., Сидоров И. Д. Параллельные алгоритмы криптоанализа асимметричных систем. Актуальные аспекты защиты информации в Южном федеральном университете. Монография. — Таганрог: Изд-во ТТИ ЮФУ, 2011. — С. 207–252.

[11] Бабенко Л. К., Ищукова Е. А., Сидоров И. Д. *Параллельные алгоритмы факторизации для анализа асимметричных криптосистем* // Материалы международной конференции «Моделирование устойчивого регионального развития». — Нальчик: Изд-во КБНЦ РАН, 2011. — С. 78–84.

Рекомендовал к публикации *Программный комитет* ***Национального суперкомпьютерного форума*** НСКФ-2012

Об авторах:

Бабенко Людмила Климентьевна

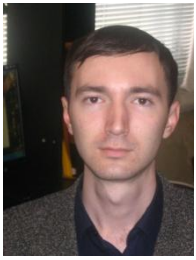
Доктор технических наук, профессор, профессор кафедры Безопасности информационных технологий Южного федерального университета. Автор 294 научных работ. Область научных интересов: организация параллельных вычислений, системы аутентификации, управление доступом, криптографические средства, оценка стойкости защиты информации.

e-mail: blk@tsure.ru

Ищукова Евгения Александровна

Кандидат технических наук, доцент кафедры Безопасности информационных технологий Южного федерального университета. Автор 55 научных работ. Область научных интересов: криптография, криптоанализ, симметричные алгоритмы шифрования, организация параллельных вычислений, оценка стойкости защиты информации.

e-mail: jekky82@mail.ru



Игорь Дмитриевич Сидоров

Кандидат технических наук, доцент кафедры Безопасности информационных технологий Южного федерального университета. Автор 31 научной работы. Область научных интересов: криптография, параллельные вычисления, оценка стойкости защиты информации, искусственные нейронные сети.

e-mail: idsidorov@gmail.com

Образец ссылки на публикацию:

Л. К. Бабенко, Е. А. Ищукова, И. Д. Сидоров. Применение параллельных вычислений при решении задач защиты информации // Программные системы: теория и приложения: электрон. научн. журн. 2013. Т. 4, № 3(17), с. 25–42.

URL: http://psta.psiras.ru/read/psta2013_3_25-42

L. K. Babenko, E. A. Ishchukova, I. D. Sidorov. *Application of parallel calculations at the solution of information protection problems.*

Abstract. Given work considers questions of application of the distributed multiprocessing calculations for reduction of time of the analysis of modern cryptographic systems of protection of information. The experimental data received on the basis of realization of parallel analysis algorithms of symmetric and asymmetric algorithms of enciphering are given.

Key Words and Phrases: cryptography, cryptanalysis, methods of factorization, number field sieve, parallel sieving, Gaussian elimination, secret key, block cipher, strength, multiprocessing calculations.