Раздел IV. Методы и средства криптографии и стеганографии

УДК 681.3.06

Л.К. Бабенко, И.Д. Сидоров

ПАРАЛЛЕЛЬНЫЙ АЛГОРИТМ ДИСКРЕТНОГО ЛОГАРИФМИРОВАНИЯ МЕТОДОМ РЕШЕТА ЧИСЛОВОГО ПОЛЯ*

Задача дискретного логарифмирования является одной из фундаментальных в криптоанализе систем с открытым ключом, на сложности её решения основана стойкость криптосистем Диффи-Хеллмана, Эль-Гамаля и многих других. В конечной группе целых чисел Fp эта задача формулируется следующим образом: пусть задано основание a, степень b и модуль p, связанные соотношением $a^x \equiv b(\mathrm{mod}\ p)$. Необходимо найти показатель x (mod r), где r – порядок циклической группы <a>, образованной элементом a.

Известно много алгоритмов решения данной задачи с различными оценками временной сложности, однако на практике интерес представляют субэкспоненциальные алгоритмы, основанные на методах базы разложения или решета числового поля[1]. Метод общего решета числового поля (GNFS) является самым асимптотически быстрым из известных на сегодняшний день. К тому же, по сравнению с методом базы разложения, у него намного меньше емкостная сложность.

Метод общего решета числового поля для вычисления дискретного логарифма был предложен Гордоном [2] и развит Вебером [3]. Метод использует однозначность разложения на простые идеалы в кольцах целых алгебраических чисел. Кратко, поиск дискретного логарифма с помощью GNFS выглядит следующим образом [1]:

1. Подготовительный этап. Выберем базу разложения $D_1=\{p_1,p_2,\ldots,p_n\}$ – простые числа, меньшие некоторой границы, включая число -1. Выберем неприводимый над Q полином $f(x)=X^n+a_{n-1}X^{n-1}+\ldots+a_0$, целое число m, такое, что $f(m) \equiv 0(\mathrm{mod}\ p)$ и комплексный корень полинома α. Фактически, α является целым алгебраическим числом. Сформируем расширенное кольцо $Z[\alpha]$ как простое расширение кольца Z элементом α. Каждый идеал в этом кольце однозначно раскладывается в произведение простых идеалов.

Сформируем базу разложения D2 , состоящую из первых степеней простых идеалов $Z[\alpha]$ с нормой, меньшей некоторой границы. На практике для этого генерируем алгебраические числа $g+ha$, где g, h - взаимно простые, и $N(g+ha)$ меньше заданной границы.

2. Найдём не менее чем $\#D_1+\#D_2+n$ пар (c_i,d_i) таких, что целое число $c_i + d_i m(\mathrm{mod}\ p)$ является D_1-гладким, и число $c_i + d_i\alpha$ является D_2-гладким. Число называют D-гладким, если его можно представить как произведение элементов базиса D. Проверить число на D_1-гладкость можно, найдя НОД числа и

* Работа выполнена при поддержке гранта РФФИ №06-07-89010-а

произведения $\prod_{i=2}^{n} p_i^{\left\lfloor \frac{\log p}{\log p_i} \right\rfloor}$ (первый элемент базиса не участвует, так как он равен -1, и его показатель может принимать только значения 0 и 1, фактически, рассматривается гладкость проверяемого числа и его же, но со знаком минус по модулю, p). Если НОД числа и такого произведения равен самому числу, число является D-гладким, и можно пробным делением найти его разложение по элементам базиса D.

Проверка числа на D_2-гладкость можно осуществить аналогичным образом, с той разницей, что вместо целых чисел проверяется гладкость нормы идеала $N(c_i + d_i\alpha)$ относительно норм простых идеалов, входящих в базис разложения D_2. Однако, на практике можно не генерировать числа c_i,d_i и проверять гладкость $c_i + d_i m (\mathrm{mod}\ p)$ по базису D_1 и $c_i + d_i\alpha$ по базису D_2; можно сгенерировать разложение по базису D_2, найти алгебраическое число $c_i + d_i\alpha$, соответствующее произведению элементов базиса D_2 с выбранными степенями, и проверить число $c_i + d_i m (\mathrm{mod}\ p)$ на гладкость по базису D_1.

После выполнения данного шага получаем матрицу показателей, с которыми элементы базисов D_1, D_2 входят в разложение чисел $c_i + d_i m (\mathrm{mod}\ p)$, $c_i + d_i\alpha$.

3. С помощью методов линейной алгебры определим показатели e_i, такие, что $\prod_i (c_i + d_i\alpha)^{e_i}$ делится только на a и b, а целое алгебраическое число $\prod_i (c_i + d_i\alpha)^{e_i}$ является r-той степенью в Z[α]. Фактически, для выполнения этих условий можно с помощью метода Гаусса привести в 0 показатели элементов базиса D_2, при этом $\prod_i (c_i + d_i\alpha)^{e_i}$ = 1=1r – r-тая степень в Z[α]. Затем, также с помощью метода Гаусса получить произведение $\prod_i (c_i + d_i\alpha)^{e_i} (\mathrm{mod}\ p)$ делящееся только на a и b. Например, в простейшем случае, когда a и b являются элементами базиса D_1, все показатели элементов базиса, кроме a и b, приводятся в 0, а показатели a и b – в 1.

4. После окончания третьего этапа получим показательное уравнение вида $a^s b^t \equiv 1 (\mathrm{mod}\ p)$, откуда $s + tx \equiv 0 (\mathrm{mod}\ r)$, $x = -st^{-1} (\mathrm{mod}\ r)$.

Если в уравнении $s + xt \equiv 0 (\mathrm{mod}\ r)$ t и r – взаимно простые числа, то решение вида $x = -st^{-1} (\mathrm{mod}\ r)$ очень легко вычислить, используя расширенный алгоритм Евклида. Пусть НОД(t,r)>1, и невозможно найти обратный элемент по модулю r, тогда найдём r'=r/НОД(t,r) и получим $x = -qt^{-1} (\mathrm{mod}\ r')$. Решением исходного уравнения тогда будет одно из чисел $x = x' + ir'$, $i \in 0 \ldots НОД(t,r)$, которое легко найти подстановкой x_i в уравнение. Так как на практике (например, по рекомендации Шнайера [4]) обычно p=2p'+1, где p' – также простое, r=2p, НОД(t,r) не превосходит 2, и необходимо будет проверить только 2 кандидата.

Этапы 2 и 3 являются вычислительно сложными, поэтому для ускорения их выполнения автором были разработаны параллельный алгоритм нахождения D-гладких чисел (просеивания) и параллельный алгоритм Гаусса.

Нахождение D-гладких чисел достаточно просто реализовать параллельно, так как проверки отдельных чисел можно производить независимо. Каждый процесс генерирует комбинацию показателей элементов базиса D2, вычисляет алгебраическое число $c_i + d_i\alpha$ и проверяет гладкость $c_i + d_i m (\mathrm{mod}\ p)$ по базису D1. Независимые процессы заполняют свои участки матрицы показателей за разное время. Для исключения потерь времени, процессы, завершив просеивание определённого количества чисел, подсчитывают, сколько гладких чисел уже найдено, и останавливают поиск, если размер матрицы показателей достиг желаемого. По окончанию вычислений каждый процесс хранит в своей памяти участок (полосу) матрицы показателей.

Параллельный алгоритм устранения ненулевых показателей на основе метода Гаусса работает следующим образом: для каждого элемента базиса находится опорная строка с ненулевой степенью данного элемента в разложении, затем эта строка помечается как использованная и рассылается всем процессам. Процессы исключают данную переменную из своей части матрицы, выполняя для каждой i-й строки матрицы e следующее преобразование: $e_{ij}^{(l+1)} = s_k e_{ij}^{(l)} - e_{ik} s_j, \quad j \in 1...n$. Все операции с элементами матрицы осуществляются по модулю r.

Авторами была разработана программная реализация данных алгоритмов на языке C++ с использованием свободных библиотек OpenMPI (для обеспечения межпроцессного взаимодействия), NTL и GMP (для работы с целыми числами произвольной длины). При реализации использовался полином вида $x^2+\alpha^2=0$, при этом число α программой подбиралось так, чтобы существовало m, такое, что $f(m) \equiv 0 (\mathrm{mod}\ p)$ для заданного p. Для того, чтобы исключить проверку пар-кандидатов на гладкость по обоим базисам, число $c_i + d_i\alpha$ генерировалось как произведение элементов базиса D_2 с некоторыми случайными индексами, и, следовательно, было D_2-гладким по построению. Оставалось только проверить число $c_i + d_i m (\mathrm{mod}\ p)$ на гладкость по базису D_1.

В табл. 1 приводится зависимость времени просеивания и Гауссова исключения от размера базиса D_2. Размерность задачи (длина модуля p) составляет 42 бита, для вычислений использовалось одно ядро процессора AMD Athlon(tm) 64 X2 Dual Core Processor 3800+.

Таблица 1

Зависимость времени вычислений от размера базиса D2

Размер базиса D2,чисел	Время просеивания, с	Время исключения, с
600	351	211
300	107	58
150	38	22
100	24	15
50	11	8
20	4,6	5,3
10	2,9	3,9
5	1,3	0,3

Из табл. 1 видно, что чем меньше размер базиса, тем меньше время вычислений. Это можно объяснить теоретически, так как чем меньше размер базиса, тем меньше времени тратится на генерацию гладкого алгебраического числа. Время исключения также уменьшается, так как оно зависит от размерности матрицы показателей, напрямую связанной с размерностью базиса.

На следующих графиках приводится зависимость времени вычислений от числа процессоров в многопроцессорной вычислительной системе. Эксперимент проводился на кластере кафедры БИТ ТТИ ЮФУ (20 процессорных ядер Intel Xeon 2,6GHz, 10 Гб ОЗУ, передающая среда Gigabit Ethernet).

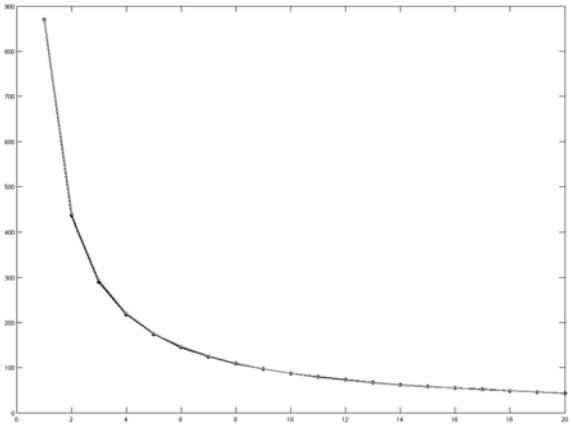

Рис. 1. График зависимости времени просеивания от количества процессоров

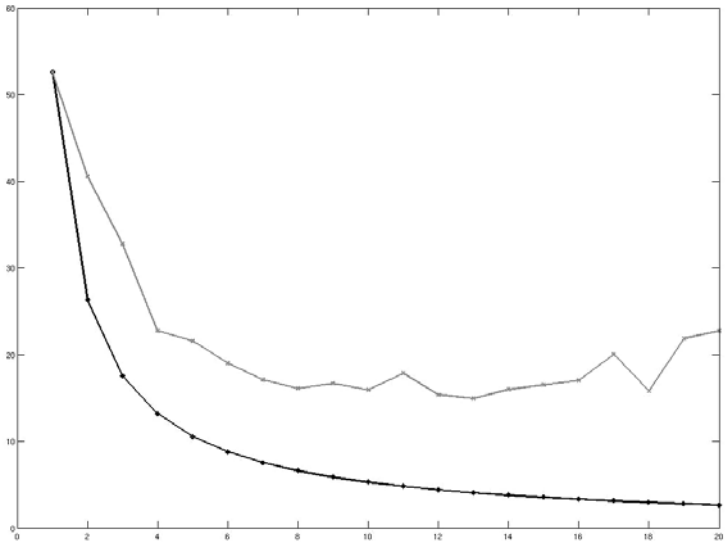

Рис. 2. График зависимости времени исключения от количества процессоров

На графиках нижняя линия (график обратной пропрорциональности) показывает максимально возможное ускорение от распараллеливания. Верхняя линия – это реальные данные, полученные в результате эксперимента.

Из графиков видно, что процесс просеивания распараллеливается практически идеально, процесс Гауссова исключения распараллеливается заметно хуже. Это объясняется тем, что при просеивании межпроцессное взаимодействие практически отсутствует (только лишь учёт количества найденных векторов показателей), а при исключении приходится пересылать достаточно много информации (опорная строка).

При нахождении дискретного логарифма важную роль в оптимизации времени вычислений играет такой параметр, как размер базиса. Если размер базиса будет слишком велик, то можно будет легко найти гладкие числа, но достаточно тяжело будет осуществлять проверки на гладкость. Также сильно возрастёт размер матрицы и соответственно время выполнения исключения. Если размер базиса будет слишком мал, то проверка на гладкость и Гауссово исключение будут легко осуществляться, но сложно будет найти достаточное количество гладких чисел.

Также на скорость выполнения вычислений влияют параметры вычислительной системы – количество и производительность процессоров, а главное – передающая среда.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Ростовцев А.Г., Маховенко Е.Б.* Теоретическая криптография. – СПб.: АНО НПО «Профессионал», 2005. – 480 с.

2. *Gordon D.* Discrete Logarithms in GF(p) using the Number Field Sieve //SIAM Journal on Discrete Mathematics. 1993, Vol. 6 p. 124-138.

3. *Weber D.* Computing discrete logarithms with the number field sieve. //Algorithmic Number Theory: Second international Symposium, ANTS-II. Talence, France, May 1996. Processing, Lecture notes in Computer Science. Springer-Verlag, 1996. Vol 1122, p. 391-403.

4. *Шнайер Б.* Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке Си. – М.: Изд-во ТРИУМФ, 2003. – 816 с.

УДК 004.056

Мкртичян В.В.

ОБ ЭКСПЕРИМЕНТАЛЬНОМ ИССЛЕДОВАНИИ НАДЕЖНОСТИ И ПРИМЕНЕНИИ СХЕМЫ СПЕЦИАЛЬНОГО ШИРОКОВЕЩАТЕЛЬНОГО ШИФРОВАНИЯ

1. Введение и постановка задачи

На практике хорошо известны и широко применяются различные схемы защиты распространяемых данных, использующие криптографическое сокрытие информации и секретное распределение уникальных ключей пользователей [1]. К таким схемам относятся как системы, имеющие аппаратную основу, так и системы, реализованные программно. Отметим, что разработкой этих систем занимаются группы компаний, включающие такие известные фирмы, как IBM, Hitachi, Intel, Matsushita, Sony и Toshiba [2]. К системам первого типа относятся: система защиты телевидения HDTV высокой четкости HDCP, системы защиты CD, DVD-дисков и сменных носителей CPPM и CPRM, система AACS защиты лазерных дисков нового поколения Blu-ray (например, [2]). Примерами систем второго типа являются: система Verimatrix VCAS защиты цифрового интерактивного телевидения IPTV, система DTCP защиты мультимедиа файлов, распространяемых в цифровых сетях, а также программные реализации некоторых из указанных выше аппаратных систем. Отметим, что особенностью систем Verimatrix VCAS и DTCP является тот