Таблица 2

Один пример при разных λ, $s = 52$ и 9, для которого $EH_0 = 24662{,}7$

№	λ	s	Время решения уравнений (в с)	P_s	HP_s	$H+1$	H_0
1	0	9	14,4	0,0146625	13901	948060	13916
2	1	9	11,5	0,0146625	13901	948060	13916
3	2	9	10,4	0,0146625	13901	948060	13916
4	3	9	10,3	0,0146625	13901	948060	13916
5	4	9	9,5	0,0146625	13901	948060	13916
6	5	9	9,7	0,0146625	13901	948060	13916
7	6	9	20,03	0,0146625	13901	948060	13916
8	4	52	2,47	$5{,}4 \cdot 10^{-11}$	$5{,}2 \cdot 10^{-5}$	948060	1
9	5	52	2,61	$5{,}4 \cdot 10^{-11}$	$5{,}2 \cdot 10^{-5}$	948060	1

Из приведённых таблиц видно, что реальные H_0 и HP_s близки. Отсюда можно сделать вывод, что предположения, при которых выведены формулы (7), верны. Для экспериментальной оценки необходимого размера списка простых была написана программа в пакете символьных вычислений mathematica 6, который предоставляет возможность, благодаря использованию функций трехмерной графики, за одно выполнение программы в приемлемое время получить картину поведения сложных математических функций.

ЛИТЕРАТУРА

1. *Зачёсов Ю. Л., Салихов Н. П.* О методе решения полиномиального сравнения $p(x) = 0 \bmod N$ // Обозрение прикладной и промышленной математики. 2008. Т. 15. Вып. 5. С. 769–784.
2. *Колчин В. Ф., Севастьянов Б. А., Чистяков В. П.* Случайные размещения. М.: Наука, 1976.
3. *Кнут Д.* Искусство программирования для ЭВМ. Т. 3. Сортировка и поиск. М.: Мир, 1978. 844 с.

УДК 519.7

ДИСКРЕТНОЕ ЛОГАРИФМИРОВАНИЕ В ПОДГРУППАХ ПРОСТОГО ПОРЯДКА

И. А. Панкратова

Во многих криптосистемах, например в протоколах идентификации Окамото, Шнорра, в протоколе цифровой подписи Шаума — ван-Антверпена, в хэш-функции CvHP, в американском стандарте цифровой подписи DSS и других, вычисления выполняются в некоторой подгруппе (как правило, простого порядка) мультипликативной группы $\mathbb{Z}_p^*$. Соответственно стойкость таких криптосистем полагается на сложность задачи дискретного логарифмирования в подгруппах.

Пусть p — простое число, q — простой делитель $p-1$, G — циклическая подгруппа порядка q группы $\mathbb{Z}_p^*$, a — порождающий элемент G. Тогда задача дискретного логарифмирования в G заключается в следующем: для данных p, q, a и элемента $b \in G$ требуется найти единственное целое x, $0 \leqslant x \leqslant q-1$, такое, что $a^x = b (\mathrm{mod}\, p)$.

Обсудим возможности применения известных алгоритмов логарифмирования [1] для решения этой задачи.

Алгоритм Гельфонда (*baby-step giant-step algorithm*) и ρ-метод Полларда особенности группы никак не учитывают; их работоспособность зависит только от того, насколько велик её порядок, и время работы составляет $O(\sqrt{q})$. Таким образом, эти алгоритмы могут применяться для логарифмирования в группе G без каких-либо изменений и с тем же успехом, что и в других группах порядков, приблизительно равных q. Алгоритм Полига — Хеллмана совершенно неприменим для логарифмирования в группе простого порядка, так как в этом случае его работа сводится к полному перебору всех возможных значений логарифма. Алгоритм Адлемана (*index calculus algorithm*) имеет сложность $O(c^{(\ln p)^{1/2}})$; одним из его шагов является решение системы сравнений по модулю порядка группы. Поэтому алгоритм Адлемана кажется наиболее пригодным для логарифмирования в группе простого порядка, так как в этом случае приходится решать систему уравнений над полем, а не над кольцом, что много проще. Основная идея алгоритма состоит в том, что в группе G выбирается некоторое подмножество элементов S, называемое факторной базой, и сначала определяются логарифмы элементов этого подмножества. Затем с их помощью вычисляется искомый логарифм.

Алгоритм 1 (Адлемана)
Вход: $G =< a >, |G| = n,\ b \in G$.
Выход: $x = \log_a b$.

1. Выбираем факторную базу $S = \{p_1, p_2, \ldots, p_t\}$.
2. Ищем значения a^k, разложимые в базе S; для этого:
   2.1. Выбираем случайное $k,\ 0 < k < n$, и вычисляем $y = a^k$.
   2.2. Ищем разложение $y = \prod\limits_{i=1}^{t} p_i^{e_i}$; в случае успеха составляем линейное сравнение $\sum\limits_{i=1}^{t} e_i \cdot x_i = k (\mathrm{mod}\, n)$.
   2.3. Шаги 2.1 и 2.2 повторяем до тех пор, пока не получим систему сравнений, имеющую единственное решение.
3. Решая полученную на шаге 2 систему сравнений, находим значения x_i для всех $i = 1, \ldots, t$.
4. Ищем $x = \log_a b$; для этого:
   4.1. Выбираем случайное $k,\ 0 \leqslant k < n$, и вычисляем $y = b \cdot a^k$.
   4.2. Ищем разложение $y = \prod\limits_{i=1}^{t} p_i^{e_i}$; в случае успеха находим
   $x = \sum\limits_{i=1}^{t} e_i \cdot x_i - k (\mathrm{mod}\, n)$, иначе возвращаемся к п. 4.1.
5. Ответ: x.

Нетрудно видеть, что если $S \subseteq G$, то получаемые в шаге 3 значения x_i — это и есть логарифмы элементов факторной базы S: $x_i = \log_a p_i$; корректность вычисления $x = \log_a b$ в этом случае очевидна. Если $G = \mathbb{Z}_p^*$, то в качестве S рекомендуется брать множество из t первых простых чисел. Если же $G \subset \mathbb{Z}_p^*$, то выбор факторной базы с условием $S \subseteq G$ представляет собой определённую проблему, ведь не обязательно все первые простые числа принадлежат G. В работе [2] предлагается выбирать факторную базу $S \subseteq \mathbb{Z}_p^*$, не обязательно $S \subseteq G$. При условии $p = q \cdot s + 1$, где $(q, s) = 1$, алгоритм 1 остаётся работоспособным, даже если $S \not\subseteq G$.

Утверждение 1. Пусть $p = q \cdot s + 1$, где p простое и $(q, s) = 1$, G — подгруппа $\mathbb{Z}_p^*$ порядка q. Алгоритм 1 корректно решает задачу дискретного логарифмирования в группе G при выборе любого непустого подмножества $\mathbb{Z}_p^* \backslash \{1\}$ в качестве факторной базы.

Доказательство. Заметим, что группа G есть множество всех вычетов степени s по модулю p, т. е. множество тех и только тех элементов группы $\mathbb{Z}_p^*$, логарифмы которых по основанию g кратны s.

Рассмотрим разложение $a^k \bmod p = \prod_{i=1}^{t} p_i^{e_i}$. Пусть $s' = s^{-1} \bmod q$. Тогда $a^{s \cdot s'} = a (\bmod p)$ и $a^k = a^{s \cdot s' \cdot k} = \left(\prod_{i=1}^{t} p_i^{e_i} \right)^{s \cdot s'} = \prod_{i=1}^{t} \left(p_i^{s \cdot s'} \right)^{e_i} (\bmod p)$. Поскольку $p_i^s \in G$, то и $p_i^{s \cdot s'} \in G$. Следовательно, существует $x_i = \log_a p_i^{s \cdot s'}$; именно это значение и будет найдено в шаге 3 алгоритма Адлемана при решении системы сравнений. Для шага 4 проведём те же рассуждения: $y \in G$, откуда $y^{s \cdot s'} = y (\bmod p)$, $b \cdot a^k = \left(\prod_{i=1}^{t} p_i^{e_i} \right)^{s \cdot s'} = \prod_{i=1}^{t} \left(p_i^{s \cdot s'} \right)^{e_i} (\bmod p)$. Тогда $\log_a b = \sum_{i=1}^{t} e_i \cdot \log_a p_i^{s \cdot s'} - k (\bmod q) = \sum_{i=1}^{t} e_i \cdot x_i - k (\bmod q)$. ■

Присмотримся к элементу $p_i^{s \cdot s'}$, логарифм которого вычисляется в шаге 3 алгоритма. Составим и решим сравнение

$$y^s = p_i^s (\bmod p).$$

Получаем

$$s \cdot \log y = s \cdot \log p_i (\bmod p - 1),$$

$$\log y = \log p_i (\bmod q),$$

$$\log y \in \{\log p_i, \log p_i + q, \ldots, \log p_i + (s-1) \cdot q\},$$

$$y \in Y = \{p_i, p_i \cdot g^q, \ldots, p_i \cdot g^{(s-1) \cdot q}\} = \{g^{\log p_i + q \cdot t} : t = 0, \ldots, s-1\}.$$

Множество Y — это все корни s-й степени из p_i^s. Здесь t пробегает полную систему вычетов по модулю s и $(q, s) = 1$, значит, и $\log p_i + q \cdot t$ также пробегает полную систему вычетов по модулю s. Поскольку единственный из показателей $\log p_i + q \cdot t$ кратен s, то один и только один элемент из Y, а именно $p_i^{s \cdot s'}$, принадлежит G. Другими словами, при решении системы сравнений в шаге 3 алгоритма 1 в качестве значения переменной x_i будет вычислен логарифм того корня s-й степени из p_i^s, который принадлежит G. В случае, если $s = 2^k$ (в частности, если p — надёжное простое число; тогда $s = 2$), можно расширить факторную базу S без увеличения количества переменных. А именно: для каждого $p_i \in S$ вычисляем все корни s-й степени из p_i^s по модулю p (это можно сделать, так как вычисление квадратных корней по простому модулю — лёгкая задача) и включаем их в факторную базу, сопоставляя всем им одну и ту же переменную.

Для сравнения заметим, что, например, «Handbook of Applied Cryptography» [1] предлагает решать задачу дискретного логарифмирования в подгруппе простого порядка следующим образом.

Пусть $\mathbb{Z}_p^* =< g >$, $p = q \cdot s + 1$, q простое, $G =< a >$ — подгруппа $\mathbb{Z}_p^*$ порядка q и $b \in G$. С помощью алгоритма Адлемана найдём $y = \log_g a$ и $z = \log_g b$ в $\mathbb{Z}_p^*$. В силу равенств $a^q = g^{y \cdot q} = 1 (\bmod p)$ и $\mathrm{O}_p(g) = p - 1 = q \cdot s$ выполняется условие $(q \cdot s) | (y \cdot q)$, т. е. $s | y$. Аналогично, ввиду $b \in G$, верно, что $s | z$. Пусть $y = s \cdot y_1$, $z = s \cdot z_1$. Запишем:

$a^{z_1} = g^{y \cdot z_1} = g^{y_1 \cdot s \cdot z_1} = g^{y_1 \cdot z} = b^{y_1} (\mathrm{mod}\, p)$, откуда получаем $x = \log_a b = z_1 \cdot y_1^{-1} \mathrm{mod}\, q$. Здесь $y_1^{-1} \mathrm{mod}\, q$ всегда существует в силу условия $0 < y_1 < q$.

При таком подходе нужно дважды решать задачу логарифмирования в $\mathbb{Z}_p^*$, что предполагает решение двух систем линейных уравнений над кольцом, в то время как в методе [2] решается одна система линейных уравнений над полем. Что касается ограничения $(q, s) = 1$, то это условие автоматически выполнено, если простое p построено методом Маурера (так как в этом случае $s < q$), и выполнено с вероятностью $(q-1)/q$, если p построено по алгоритму ГОСТ (поскольку s — случайное число в некотором диапазоне, в котором каждое q-е число кратно q). Таким образом, в большинстве случаев задачу логарифмирования в числовой группе G простого порядка можно решать методом Адлемана, не требуя выбора факторной базы как подмножества G.

ЛИТЕРАТУРА

1. *Menezes A. J., Van Oorshot P. C., Vanstone S. A.* Handbook of Applied Cryptography. N. Y.: CRC Press Series on Discrete Mathematics and Its Applications, 1997.
2. *Белов А. Г.* Исследование алгоритма дискретного логарифмирования Адлемана // Вестник Томского госуниверситета. Приложение. 2005. № 14. С. 45–49.

УДК 519.7

ЗАДАЧИ ЛИНЕЙНОЙ АЛГЕБРЫ, СООТНЕСЕННЫЕ С ЗАДАЧЕЙ ВЫПОЛНИМОСТЬ

Р. Т. Файзуллин

Рассмотрим переход от задачи 3-ВЫПОЛНИМОСТЬ к задаче решения систем линейных алгебраических уравнений. Пусть дана КНФ:

$$L(x) = \prod_{i=1}^{M} C_i, \tag{1}$$

где C_i — дизъюнкты вида $\vee q_{ij}$. Здесь $q_{ij} = x_j$ или $q_{ij} = \bar{x}_j$.

Заметим, что каждому дизъюнкту можно поставить в соответствие уравнение, связывающее уже вещественные переменные. В правой части стоят единицы, а переход от булевых переменных к вещественным осуществляется согласно формулам: $x_j \to y_j$, $\bar{x}_j \to 1 - y_j$. В этом случае мы получаем систему линейных алгебраических уравнений с прямоугольной матрицей:

$$Ay = f. \tag{2}$$

Обратим внимание на то очевидное обстоятельство, что правая часть f_j равна количеству q_{ij}, принимающих значение ИСТИНА в исходной КНФ. Систему можно преобразовать согласно формуле

$$RAy = Bf = Rf = g. \tag{3}$$

Попытаемся построить матрицу B таким образом, чтобы получить в итоге симметричную матрицу. Рассмотрим переменную с индексом j и соответствующие ей уравнения в (3). Будем складывать эти уравнения, аккумулируя неизвестные в j-й строке B, умножая их на -1, если переменная входит в уравнение со знаком минус. Тогда верна следующая лемма.

Лемма. Итоговая матрица B симметрична.