4. *Bernstein D. J.* Factoring into coprimes in essentially linear time // J. Algorithms. 2005. No. 54(1). P. 1–30.
5. *Bernstein D. J.* Scaled reminder trees // `http://cr.yp.to/papers.html#scaledmod`

УДК 519.688

ОПТИМИЗАЦИЯ $(p-1)$-АЛГОРИТМА ПОЛЛАРДА

А. С. Климина

Приведены критерии выбора параметров $(p-1)$-алгоритма Полларда и рассмотрен метод его оптимизации.

Ключевые слова: $(p-1)$*-алгоритм Полларда, факторизация чисел.*

Рассмотрим $(p-1)$-алгоритм Полларда факторизации числа N [1]. Алгоритм состоит из следующих шагов.

Ш а г 1. Выбираем число k.

Ш а г 2. Выбираем произвольное a, $1 < a < N$.

Ш а г 3. Вычисляем $d = (a, N)$. Если $d > 1$, получили нетривиальный делитель N.

Ш а г 4. Вычисляем $D = (a^k - 1, N)$. Если $1 < D < N$, то D — нетривиальный делитель N. Если $D = 1$, возвращаемся к шагу 1, а при $D = N$ — к шагу 2.

Вопрос оптимального выбора параметра k не исследован до настоящего времени с нужной полнотой. Автором реализованы следующие подходы к выбору числа k: k — произведение нескольких случайных чисел; k — факториал некоторого числа, k — произведение степеней простых чисел; k — наименьшее общее кратное нескольких чисел. После анализа работы программы выбран третий метод, поскольку он работает быстрее остальных и даёт больше удачных разложений [2].

Зафиксируем $a = 2$ и рассмотрим работу $(p-1)$-алгоритма Полларда при следующих допущениях:

1) k — произведение нескольких первых простых чисел в заданной степени;

2) исследуемое число N представляет собой произведение двух простых множителей, p и q.

Пусть $D = (a^k - 1, N)$; $\mathrm{O}_p(a)$, $\mathrm{O}_q(a)$ — показатели числа a по модулям p и q соответственно. Возможны три случая:

1) k кратно ровно одному из чисел $\mathrm{O}_p(a)$, $\mathrm{O}_q(a)$. В этом случае получим $1 < D < N$, и нетривиальный делитель D найден;

2) k не кратно ни одному из $\mathrm{O}_p(a)$, $\mathrm{O}_q(a)$. В этом случае $D = 1$;

3) k кратно и $\mathrm{O}_p(a)$, и $\mathrm{O}_q(a)$. В этом случае $D = N$.

Таким образом, для удачного нахождения делителя N нужно выбрать параметр k не слишком большим и не слишком маленьким. Заметим, что k кратно $\mathrm{O}_p(a)$ для любого a, если k кратно $p - 1$. Это, в свою очередь, гарантированно выполняется, если k является произведением всех простых чисел $p \leqslant \sqrt{N}/2$ в степенях $\log_p(\sqrt{N}/2)$. Такой выбор k хорошо работает для сравнительно небольших чисел N, однако при увеличении числа N становится неприемлемым.

Предлагается следующий алгоритм выбора k в процессе работы программы.

Пусть $k = p_1^{\alpha} p_2^{\alpha} \dots p_l^{\alpha}$, где $p_1, \dots, p_l$ — первые l простых, α — некоторая константа. Будем рассматривать изменение k только за счёт изменения количества простых l. Идея выбора l состоит в том, чтобы выбрать его не слишком большим и не слишком малым, исходя из результатов работы программы (при слишком малом значе-

нии l результатом является 1, а при слишком большом — N). При этом l выбирается методом, похожим на половинное деление. При каждом значении l вычисляется $D(l) = (a^k - 1, N)$, $k = p_1^{\alpha} p_2^{\alpha} \dots p_l^{\alpha}$. Параметр l выбирается следующим образом:
Ш а г 1. $l_{\min} := 1$, $l_{\max} := 1$.
Ш а г 2. $l_{\max} := l_{\max} \cdot 2$; $l = l_{\max}$.
Ш а г 3. Вычислить $D = D(l)$;
— если $D = 1$, то перейти к шагу 2;
— если $D = N$, то перейти к шагу 4;
— если $1 < D < N$ — выход, делитель найден.
Ш а г 4. $l_{\min} := l_{\max}/2$; $l_{\text{midl}} := l_{\min} + (l_{\max} - l_{\min})/2$; $l := l_{\text{midl}}$.
Ш а г 5. Вычислить $D = D(l)$;
— если $D = 1$, то $l_{\min} := l_{\text{midl}}$; $l_{\text{midl}} := l_{\min} + (l_{\max} - l_{\min})/2$; $l := l_{\text{midl}}$; перейти к шагу 5;
— если $D = N$, то $l_{\max} := l_{\text{midl}}$; $l_{\text{midl}} := l_{\min} + (l_{\max} - l_{\min})/2$; $l := l_{\text{midl}}$; перейти к шагу 5;
— если $1 < D < N$ — выход, делитель найден.

Проведены эксперименты на заранее сформированных массивах чисел, представляющих собой произведение двух простых, с разрядностью 20, 40 и 60 десятичных знаков. Из 10000 чисел разрядностью 20 десятичных знаков были разложены 4250, среднее время на обработку одного числа составило 1,42 с. Разрядность максимального числа, которое было разложено на множители, составляет 60 десятичных знаков:

$$105280800840041764587660602798986728548772087134305755177808 3 =$$
$$= 123547896523698521452369860571 \cdot 852145635841296358745632596847 3,$$

время разложения — 29 мин 23 с.

ЛИТЕРАТУРА

1. *Маховенко Е. Б.* Теоретико-числовые алгоритмы в криптографии. М.: Гелиос АРВ, 2006.
2. *Климина А. С.* Оптимизация выбора параметров для алгоритма Полларда // IV ОМНТК «Молодежь. Техника. Космос». СПб.: БГТУ, 2012. С. 285–286.

УДК 519.688

ПАРАЛЛЕЛЬНЫЙ АЛГОРИТМ ВЫЧИСЛЕНИЯ ФУНКЦИЙ РОСТА В КОНЕЧНЫХ ДВУПОРОЖДЁННЫХ ГРУППАХ ПЕРИОДА 5

А. С. Кузнецова, А. А. Кузнецов, К. В. Сафонов

Представлена параллельная версия алгоритма для вычисления функций роста в конечных двупорождённых группах периода 5.

Ключевые слова: *функция роста группы, диаметр Кэли, параллельный алгоритм.*

Пусть p — простое число, G — конечная группа экспоненты p. Это значит, что $g^p = e$ для всех $g \in G$. Так как G нильпотентна, то можно найти цепочку подгрупп $G = G_1 \supset \supset G_2 \supset \dots \supset G_n \supset G_{n+1} = e$, в которой G_i нормальны в G, а факторы G_i/G_{i+1} имеют порядок p и лежат в центре G/G_{i+1}.

Пусть для $1 \leqslant i \leqslant n$ элемент $a_i \in G_i$, но $a_i \notin G_{i+1}$, тогда каждый элемент группы $g \in G$ можно однозначным образом записать в виде

$$g = a_1^{\gamma_1} a_2^{\gamma_2} \dots a_n^{\gamma_n}, \quad 0 \leqslant \gamma_i < p. \tag{1}$$