Полагая в (5)–(7) $K = R + N$, $M = RN$,

$$a_{\alpha\beta} = \begin{cases} 1, \text{ если } 1 \leqslant \alpha \leqslant R,\ \psi_3(\beta) = \alpha, \\ 1, \text{ если } R+1 \leqslant \alpha \leqslant R+N,\ \psi_1(\beta) = \alpha - R, \\ 0 \text{ в остальных случаях;} \end{cases}$$

$$b_\alpha = \begin{cases} m_i, \text{ если } 1 \leqslant \alpha \leqslant R,\ \alpha = i, \\ p_k, \text{ если } R+1 \leqslant \alpha \leqslant R+N,\ \alpha - R = k, \end{cases}$$

получим производящие функции для числа целых неотрицательных решений системы (2) без ограничений, а также соответственно при ограничениях (3) и (4).

ЛИТЕРАТУРА

1. *Яблонский С. В.* Введение в дискретную математику. М.: Наука, 1986. 384 с.
2. *Гоцуленко В. В.* Формула для числа сочетаний с повторениями при ограничениях и её применение // Прикладная дискретная математика. 2013. № 2(20). С. 71–77.

УДК 519.6

АЛГОРИТМ ГЕНЕРАЦИИ ПАРЫ ПРОСТЫХ ЧИСЕЛ СПЕЦИАЛЬНОГО ВИДА

К. Д. Жуков, А. С. Рыбаков

Рассматривается алгоритм генерации пары простых чисел p и q, таких, что числа $g = \frac{1}{2}(p-1, q-1)$ и $h = \frac{1}{2g}(pq-1)$ также простые. Такие простые числа впервые рассмотрены в 2006 г. М. Дж. Хинеком в связи с предложенной им модификацией криптосистемы RSA, устойчивой к атакам на малые секретные экспоненты. Приводятся экспериментальные данные о времени работы алгоритма.

Ключевые слова: *простые специального вида, Common Prime RSA.*

В 2006 г. М. Дж. Хинек предложил вариант криптосистемы RSA, устойчивой к атакам на малую секретную экспоненту, которая была названа Common Prime RSA. Простые сомножители p и q модуля Common Prime RSA выбираются такими, чтобы числа

$$g = \frac{1}{2}(p-1, q-1), \quad h = \frac{1}{2g}(pq-1) \tag{1}$$

были также простыми, причём число g должно быть достаточно большим.

Система Common Prime RSA не получила распространения. Этот факт связан с малым количеством публикаций по её анализу. Большинство атак на данную разновидность RSA были также предложены М. Дж. Хинеком (см., например, [2]). Не способствует распространению и отсутствие острой необходимости использовать малые секретные экспоненты; другим сдерживающим фактором использования Common Prime RSA является долгая генерация ключей.

Простейшая версия алгоритма генерации простых сомножителей модуля криптосистемы Common Prime RSA, предложенная в [1], описана ниже.

Алгоритм 1. (Хинек, [1])

Вход: натуральные n и m, $m < n$

Выход: пара n-битовых простых чисел p и q, таких, что g и h, определяемые равенствами (1), простые, а g имеет битовый размер m

1: Выбрать случайное простое m-битовое число g.
2: **Повторять**
3: Выбрать случайные положительные целые $(n-m-1)$-битовые числа a и b;
4: $p := 2ga + 1$, $q := 2gb + 1$, $h := 2gab + a + b$
5: **Пока** p, q, h — не простые или $(a, b) \neq 1$;
6: **Вывести** p, q.

В работе [1] отмечается, что алгоритм 1 не оптимизирован. Отметим, что простые числа можно генерировать в двух независимых подциклах. Кроме того, эксперименты показывают, что неудачный выбор простого числа g может привести к тому, что время работы алгоритма будет существенно больше среднего времени работы для заданных параметров n и m. Отсюда целесообразно генерировать простое число g внутри цикла.

Заметим, что самой частой и трудоёмкой операцией алгоритма является тест на простоту. На его первом этапе проверяется, что число не делится на малые простые. Этот этап можно ускорить, учитывая специальный вид простых. Например, вместо проверки условия $r|(2ga+1)$ нужно проверять условие $a \equiv (-2g)^{-1} \pmod{r}$ для малого простого r.

Алгоритм 2

Вход: натуральные n и m, $m < n$; натуральный параметр метода k

Выход: пара n-битовых простых чисел p и q, таких, что g и h, определяемые равенствами (1), простые, а g имеет битовый размер m

1: Построить k первых простых чисел $p_1, \ldots, p_k$.
2: **Повторять**
3: Используя технику просеивания, выбрать случайное простое m-битовое число g.
4: Вычислить $g_i := (-2g)^{-1} \mod p_i$ для всех $i = 1, \ldots, k$.
5: **Повторять**
6: Используя технику просеивания, выбрать случайное $(n-m-1)$-битовое положительное целое a, такое, что $g_i \neq a \mod p_i$, $i = 1, \ldots, k$.
7: Вычислить $p := 2ga + 1$
8: **Пока** p не простое
9: Вычислить $h_i := a(-2ga - 1)^{-1} \mod p_i$ для всех $i = 1, \ldots, k$.
10: **Повторять**
11: Используя технику просеивания, выбрать случайное $(n-m-1)$-битовое положительное целое b, такое, что $g_i \neq b \mod p_i$, $h_i \neq b \mod p_i$, $i = 1, \ldots, k$.
12: Вычислить $q := 2gb + 1$
13: **Пока** q не простое и $(a, b) \neq 1$
14: Вычислить $h := 2gab + a + b$
15: **Пока** h не простое
16: **Вывести** p, q

Алгоритмы реализованы на языке программирования C++ с использованием библиотеки NTL [3]. В таблице проводятся результаты экспериментов на компьютере с процессором Intel core i7 с тактовой частой 3,33 ГГц при значении параметра $k = 100$.

Время работы алгоритмов варьируется в пределах, отличающихся на порядок. В связи с этим в таблице указано худшее время в трёх случайных экспериментах.

Результаты экспериментов с 1024-битовым модулем

p и q, биты	g, биты	Время алг. 1, с	Время алг. 2, с
512	256	46	24
	320	51	31
	384	58	19
1024	512	1082	213
	640	908	660
	768	794	98

Из таблицы видно, что, несмотря на предложенное ускорение метода построения специальных простых, выработка модуля криптосистемы Common Prime RSA занимает неприемлемо большое время. Отметим, что выработка пары случайных простых чисел без дополнительных свойств занимает десятые доли секунды.

ЛИТЕРАТУРА

1. *Hinek M. J.* Another look at small RSA exponents // LNCS. 2006. V. 3860. P. 82–98.
2. *Hinek M. J.* Cryptanalysis of RSA and Its Variants. CRC Press, 2009.
3. *Shoup V.* NTL — a library for doing number theory // http://www.shoup.net

УДК 519.688

ПОЛИНОМЫ ХОЛЛА ДЛЯ КОНЕЧНЫХ ДВУПОРОЖДЁННЫХ ГРУПП ПЕРИОДА СЕМЬ[1]

А. А. Кузнецов, К. В. Сафонов

Пусть $B_k = B_0(2, 7, k)$ — максимальная конечная двупорождённая группа периода 7 ступени нильпотентности k. В работе вычислены полиномы Холла для B_k при $k \leqslant 4$.

Ключевые слова: *периодическая группа, собирательный процесс, полиномы Холла.*

Пусть $B_k = B_0(2, 7, k)$ — максимальная конечная двупорождённая группа периода 7 ступени нильпотентности k. В данном классе групп наибольшей является группа B_{28}, порядок которой равен 7^{20416} [1]. Для каждой из B_k получены pc-представления (power commutator presentation) [1].

Пусть $a_1^{x_1} \dots a_n^{x_n}$ и $a_1^{y_1} \dots a_n^{y_n}$ — два произвольных элемента в группе B_k, записанные в коммутаторном виде. Тогда их произведение равно

$$a_1^{x_1} \dots a_n^{x_n} \cdot a_1^{y_1} \dots a_n^{y_n} = a_1^{z_1} \dots a_n^{z_n}.$$

Основой для нахождения степеней z_i является собирательный процесс [2, 3], который реализован в системах компьютерной алгебры GAP и MAGMA. Кроме того, существует альтернативный способ для вычисления произведений элементов группы, предложенный Ф. Холлом [4]. Холл показал, что z_i представляют собой полиномиальные

[1]Работа выполнена при поддержке Министерства образования и науки Российской Федерации, проект Б 112/14.