ЛЕКЦИЯ 15

ПРОСТЫЕ ЧИСЛА

Натуральное число p, больше единицы называется простым, если оно делится нацело только на 1 и на себя.

Теорема (Эвклид). Множество простых чисел бесконечно.

Обозначим через $\pi(x)$ функцию, которая равна числу простых чисел p в интервале $1 < x \le p$. Российский математик П. Л. Чебышев в 1850г. показал [7], что $0{,}921\, x/\ln x < \pi(x) < 1{,}106\, x/\ln x$.

Простые числа являются важным понятием в криптографии. Многие современные криптографические системы строятся на базе простого числа. Поэтому алгоритмы генерации простых чисел и проверки на простоту сформированного числа являются важными инструментами при создании криптографической системы.

Простые числа встречаются довольно часто. Заметим, что существует около 10^{151} простых чисел длиной от 1 до 512 бит включительно [10], а количество простых чисел меньших 2^{512} приблизительно равно 2^{503} [4]. Для чисел близких n вероятность произвольно выбранному числу оказаться простым числом, равна $1/\ln n$. При случайном выборе двух простых чисел в диапазоне от 1 до 151 бита вероятность совпадения этих чисел ничтожно мала. Простые числа играют важную роль в современной криптографии. Многие современные криптографические системы с открытыми (или не симметричными) ключами формируются с применением простых чисел.

Для простых чисел будем рассматривать три задачи:

- построение простых чисел;
- проверка чисел на простоту;
- факторизация (разложения) чисел на простые множители.

На самом деле все эти три задачи фактически дают

ответ на один вопрос: является ли рассматриваемое число простым. Но для каждой из этих задач применяются свои методы.

Для первой задачи, используя необходимые условия простоты, можно давать ответы типа:

- заданное число n не простое;
- вероятность того, что заданное число n не простое, меньше заданного числа ε.

Для второй задачи можно строить некоторую последовательность чисел специального вида. И для чисел данной последовательности применять некоторые тесты до тех пор, пока не найдем среди них простое число.

Приведем некоторые определения, теоремы, алгоритмы, которые связаны с вопросами решения поставленных задач (см., например [3–5]; [7]; [10–11]).

Определение 1. Числа $F_k = 2^{\alpha} + 1$, $\alpha = 2^k$, $k = 0, 1, 2, \ldots$, называются числами Ферма.

Теорема 1. Число Ферма $n = F_k$ при $k > 0$ является простым тогда и только тогда, когда $3^{(n-1)/2} \equiv -1 \bmod n$.

Определение 2. Пусть p – простое число. Числа вида $M_p = 2^p - 1$ называются числами Мерсенна.

Кстати, все четные совершенные числа имеют вид $2^{p-1}M_p$, где M_p является числом Мерсенна. Напомним, совершенным числом называется число, которое равно сумме всех своих делителей, меньших, чем оно само. Например, $28 = 1 + 2 + 4 + 7 + 14$.

Числа Мерсенна редки. В 2001году было найдено тридцать девятое число Мерсенна для $p = 1\ 3466\ 917$. Для проверки простоты чисел Мерсенна применяется следующая теорема [7].

Теорема 2. Пусть n – простое число, $n > 2$, $M_n = 2^n - 1$. Рассмотрим последовательность $L_0, L_1, \ldots$, которая определяется соотношениями

$$L_0 = 4,\ L^2_{j+1} = L^2_j - 2 \bmod n,\ 0 \le j < n.$$

Число M_n – простое тогда и только тогда, когда $L_{q-2} \equiv 0$

mod n.

ПРОВЕРКА НА ПРОСТОТУ

Простые числа необходимы для большинства криптографических систем с открытыми ключами. Теоретический материал к вопросу построения больших простых чисел можно найти в [7] и [11]. Здесь будут сформулированы только некоторые практические подходы к формированию больших простых чисел. Для генерации больших простых чисел могут быть использованы следующие два подхода:

- формируются случайные числа заданного порядка, и при помощи существующих тестов проверяется, являются ли они простыми.
- по определенному алгоритму генерируются простые числа, и при помощи определенных тестов производится проверка чисел на простоту.

Сначала рассмотрим те тесты, которые используются при реализации первого подхода формирования простого числа.

Пробное деление

Один из самых простых способов проверки числа p на простоту состоит в последовательном делении числа p на все нечетные числа, которые содержатся в интервале [2, $\sqrt{p}$]. Если в процессе деления получим целый результат, то число p – составное. Если же при переборе всех нечетных чисел из интервала [2, $\sqrt{p}$] разделить число p на эти числа нацело нельзя, то число p – простое. Данный метод называется *пробным делением*. Этот метод трудоемок по числу арифметических операций, и он используется в основном для проверки небольших простых чисел.

Решето Эратосфена

Если мы хотим составить таблицу всех простых чисел среди чисел
2, 3,..., N, то надо последовательно вычеркнуть все числа, которые делятся

- на 2, кроме 2;
- на 3, кроме 3;
- на 5 кроме 5;
- на следующее число, которое не вычеркнуто, кроме этого числа;

и т. д.

В итоге среди чисел от 1 до N останутся лишь простые числа. Для реализации метода нужен большой объем памяти ЭВМ, однако для составления таблиц простых чисел он является наилучшим [11]. Более того, разрабатываются специальные процессоры, на которых операции «просеивания» выполняются очень эффективно [7].

Замечание. Пробное деление и решето Эратосфена можно применять при решении задачи разложения целого числа на множители.

Тест на основе малой теоремы Ферма

Малая теорема Ферма [7] утверждает, что если n простое число, то выполняется условие: при всех $a \in \{2, 3, \ldots, n - 1\}$ имеет место сравнение

$$A^{n-1} \equiv 1 \bmod n.$$

На основании этой теоремы можно построить вероятностный алгоритм проверки на простоту числа n. Если для некоторого целого a из интервала
$[2, n]$ соотношение $a^{n-1} \equiv 1 \bmod n$ не выполняется, то число n – составное. Если же теорема выполняется, то вывод, что число n простое, сделать нельзя, так как

теорема дает лишь необходимое условие. Поэтому, если для некоторого a имеет место соотношение $a^{n-1} \equiv 1 \bmod n$, то говорят, что число n является псевдопростым по основанию a [1]. Существует бесконечно много пар чисел a и n, где n – составное и псевдопростое. Вообще для любого $a > 1$ существуют бесконечно много псевдопростых чисел по основанию a [1]. Вообще, справедливы следующие два утверждения.

- Если пара $(2, n)$ удовлетворяют сравнению $a^{n-1} \equiv 1 \bmod n$, то и пара чисел $(2, 2^n - 1)$ также ему удовлетворяют.
- Для любого простого числа n и любого $a > 2$ такого, что $(a^2 - 1, n) = 1$, число $(a^{2n} - 1)/(a^2 - 1)$ является псевдопростым по основанию a.

Определение. Составные числа n, для которых при всех основаниях выполняется сравнение $a^{n-1} \equiv 1 \bmod n$, называются числами Кармайкла.

Схема алгоритма на базе малой теоремы Ферма

Дано число n и параметр $\gamma = 1$ для идентификации результата проверки.

1) делается случайный выбор целого числа a из интервала $[2, n]$;
2) используя алгоритм Эвклида, вычисляется НОД для чисел a и n;
3) если НОД больше 1, то выполняется шаг 7;
4) для числа a проверяется сравнение $a^{n-1} \equiv 1 \bmod n$;
5) если сравнение не выполняется, то определяется параметр $\gamma = 0$ (число составное) переход на шаг 7.
6) если сравнение выполнено, то можно повторить тест;
7) выдать результат проверки ($\gamma = 0$ – число составное).

При завершении работы алгоритма возможны следующие выводы:

- число n – составное ($\gamma = 0$);
- если $\gamma = 1$, то число n является либо простым, либо составным и числом Кармайкла.

Здесь уместно заметить [7], что числа Кармайкла достаточно редки. Так существуют всего 2 163 чисел Кармайкла, которые не превосходят 25 000 000 000, и всего 16 чисел, которые не превосходят числа 100 000.

Тест Соловея – Штрассена

Тест Соловея – Штрассена проверки на простоту числа p базируется на теореме 4.

Теорема 4. Для любого нечетного n следующие условия эквивалентны:

- n – простое;
- для любого $a \in Z_n^*$ выполняется сравнение $a^{(n-1)/2} \bmod n \equiv L(a, n)$, где Z_n^* – мультипликативная группа, элементами которой являются элементы $a \in Z_n$ (Z_n – кольцо вычетов по модулю n).

Для проверки простоты числа p используется алгоритм вычисления символа Якоби.

Схема алгоритма на базе малой теоремы Ферма

Пусть дано нечетное число p. Надо проверить является ли число p простым.

1. Выбирается случайное число a, меньшее p.
2. Если НОД $(a, p) \neq 1$, то p – составное число.
3. Вычисляется сравнение $L(a, p) \equiv a^{(p-1)/2} \bmod p$.
4. Вычисляется символ Якоби $J(a, p)$.
5. Если $L(a, p) \neq J(a, p)$, то число p – составное.
6. Если $L(a, p) = J(a, p)$, то вероятность того, что число p – составное не превышает 50 %.

Если проверка повторяется k раз, то вероятность того, что число p – составное, не превышает $1/2^k$.

Тест Рабина – Миллера

Обоснование алгоритма Рабина – Миллера можно найти в [3]. Здесь дадим только самые общие соображения. Известно, что если число p – простое, то уравнение $x^2 \equiv 1 \bmod p$ имеет лишь два решения: $x \equiv \pm 1 \bmod p$. Итак, пусть p – нечетное целое число, которое надо проверить на простоту. Если p – простое, то по теореме Ферма для любого целого a взаимно простого с p выполняется сравнение $a^{m-1} \equiv 1 \bmod p$. Так как $p - 1$ – четно, то получаем $a^{(m-1)/2} \equiv \pm 1 \bmod p$. Если оказывается, что $(m - 1)/2$ – четно, то можно повторить рассуждение, при котором получим, что $a^{(m-1)/4} \equiv \pm 1 \bmod p$, и т. д.

Поэтому, чтобы проверить на простоту нечетное число p, выбираем случайным образом число a из интервала $[1, p - 1]$ и вычисляем

$$a^t \bmod p,\ a^{2t} \bmod p,\ \ldots,\ a^{\beta t} \bmod p,$$

где t – нечетное и число $\beta = 2^s$. Если одно из этих чисел не совпадает с +1 или –1, то можно сделать вывод, что число p является числом составным. Если значения чисел совпадают с +1 или –1, то повторяем этот тест k раз. После повторения этого теста k раз вероятность того, что составное число p не будет выявлено, не превосходит $1/4^k$. После высказанных соображений перейдем к формулировке алгоритма Рабина – Миллера. В некоторой литературе данный алгоритм называют тестом Миллера – Рабина [4].

Схема алгоритма Рабина – Миллера

Пусть дано нечетное число p. Надо проверить

является ли число p
простым. Далее предположим, что $p - 1 = 2^s t$.

1. Выбираем случайное число a, меньшее p и определяем $k = 0$.
2. Вычисляем с помощью алгоритма Эвклида НОД двух чисел a и p. Если НОД $(a, p) \neq 1$, то p – составное число.
3. Вычисляем $b \equiv a^t \bmod p$. Если $b = 1$ или $b = p - 1$, то число p вероятно простое.
4. Если $b \neq 1$ и $b \neq p - 1$, то вычисляем $b \equiv b^2 \bmod p$ и $k = k + 1$.
5. Если число $b = p - 1$, то число p вероятно простое. Перейти на шаг 7.
6. Пока $k < s$ выполнять пункт 4.
7. Завершить работу алгоритма.

Рассмотрим примеры.

Пример. Пусть $p = 181$. Имеем $p - 1 = 45 \times 2^2$. По представленному разложению определяем значение параметра $t = 45$.

1. Выбираем случайное число $a = 52 < p$, и определяем $k = 0$.
2. Используем алгоритм Эвклида для вычисления НОД двух чисел 52 и 181:
   делим число 181 на число 52, получаем $181 = 52 \cdot 3 + 25$;
   делим число 52 на число 25, получаем $52 = 25 \cdot 2 + 2$;
   делим число 25 на число 2, получаем $25 = 12 \cdot 2 + 1$;
   делим число 2 на число 1, получаем $2 = 1 \cdot 2 + 0$;
   получаем, что НОД двух чисел 181 и 52 равен 1.

Так как НОД не позволяет установить является ли число 181 составным, то продолжаем выполнять алгоритм Рабина – Миллера.

3. Последовательно вычисляем

$$b \equiv 52^t \bmod 181 \equiv 52^{45} \bmod 181:$$
$$52^2 \bmod 181 \equiv 2704 \bmod 181 \equiv 170 \bmod 181,$$
$$52^4 \bmod 181 \equiv 170^2 \bmod 181 \equiv 28900 \bmod 181 \equiv 121 \bmod 181,$$
$$52^8 \bmod 181 \equiv 121^2 \bmod 181 \equiv 14641 \bmod 181 \equiv 161 \bmod 181,$$
$$52^{16} \bmod 181 \equiv 161^2 \bmod 181 \equiv 25921 \bmod 181 \equiv 38 \bmod 181,$$
$$52^{32} \bmod 181 \equiv 38^{2`} \bmod 181 \equiv 1444 \bmod 181 \equiv 177 \bmod 181,$$
$$52^{40} \bmod 181 \equiv (52^{32} \bmod 181)(52^8 \bmod 181) \equiv$$
$$\equiv (177 \times 161) \bmod 181 \equiv 28497 \bmod 181 \equiv 80 \bmod 181,$$
$$52^{41} \bmod 181 \equiv (52^{40} \bmod 181)(52 \bmod 181) \equiv (80 \times 52) \bmod 181 \equiv$$
$$\equiv 4160 \bmod 181 \equiv 178 \bmod 181,$$
$$52^{45} \bmod 181 \equiv (52^{41} \bmod 181)(52^4 \bmod 181) \equiv (178 \times 121) \bmod 181 \equiv$$
$$\equiv 21538 \bmod 181 = 180 \bmod 181.$$

Итак, получили $b = 180 = p - 1$. Откуда следует, что число $p = 181$ вероятно простое.

Замечание. Для генерации даже небольших простых чисел вычисления довольно громоздки. Поэтому для реальных чисел, несомненно, подобные проверки чисел на простоту надо делать при помощи программ на компьютере.

В [10] дается некоторое руководство при генерации простых чисел для практических приложений. Это руководство сводится к реализации нескольких этапов (шагов).

1. Сгенерируйте случайное n-битовое число p.
2. Установите старший и младший биты равными 1. Старший бит в этом случае гарантирует требуемую длину простого числа, а младший – его нечетность.

3. Убедитесь, что число p не делится на малые простые числа 3, 5, 7, 11 и т. д. Наиболее надежна проверка делимости на все простые числа, меньше 2000.
4. Выполните тест Рабина – Миллера для некоторого случайного числа a. Если p проходит тест, то сгенерируйте другое случайное число a и повторите тест. Для практических приложений достаточно повторить тест Рабина – Миллера пять раз.
5. Если p не проходит один из тестов, надо сгенерировать другое число p и повторить данное руководство снова.

Другое число p, если оно оказалось не простым, можно получить не генерируя новое, а последовательно перебирая все целые, начиная от $p + 1$, $p + 2$, и т. д., пока не найдется простое число.

Перейдем теперь к вопросу о генерировании больших простых чисел.

Построение больших простых чисел и детерминированные алгоритмы проверки чисел на простоту

Рассмотрим еще один способ формирования простых чисел. Этот способ базируется на определенной процедуре генерирования простых чисел, проверка которых осуществляется с помощью тестов на простоту. Такой подход применяется, например, в стандарте электронной цифровой подписи (ЭЦП) ГОСТ Р 34.10–94 и основывается на следующей теореме [12].

Теорема 5. Пусть $p = qN + 1$, где q – нечетное простое число, N – четное число и $p < (2q + 1)^2$. Число p является простым, если выполняются два условия:

$$1)\ 2^{qN} = 1 \bmod p;$$

$$2)\ 2^{N} \neq 1 \bmod p.$$

Генерация простого числа с использованием теоремы 5 осуществляется по несколько упрощенной в принятом стандарте схеме [12]. Пусть требуется сформировать простое число p длины $t \geq 17$ бит. С этой целью строится убывающая последовательность чисел $\{t_i\}$, где $i = 0, 1, \ldots, s$, для которых $t_0 = t$, $t_i = [t_{i-1}/2]$. Далее последовательно вырабатывается последовательность простых чисел $p_s, p_{s-1}, \ldots p_0$, для всех $i = 1, \ldots, s$. Генерация простого числа p_{i-1} осуществляется с использованием следующее формулы $p_{i-1} = p_i N + 1$, где число N удовлетворяет следующим условиям [12]:

- N – четное;
- N – такое, что длина числа $p_{i-1} = p_i N + 1$ в точности должна быть равна t_{i-1}.

В стандарте ГОСТ дается некоторый алгоритм вычисления числа N. Для учебного варианта N – случайное четное число, которое получают с помощью датчика случайных чисел (если N – нечетно, то $N = N + 1$).

Число p_{i-1} считается полученным, если одновременно выполнены следующие два условия:

1) $2^{\beta} = 1 \bmod p_i$, $\beta = p_{i-1} N$;

2) $2^N \neq 1 \bmod p_i$.

Если хотя бы одно из условий не выполняется, то значение числа N увеличивается на 2, и вычисляется новое значение p_{i-1}, которое снова проверяется на простоту по двум условиям. Данный процесс продолжается до тех пор, пока не будет получено простое число p_{i-1} (т. е. условия 1 и 2 алгоритма генерации простого числа будут выполнены).

Заметим, что с 2002 года вышеупомянутый отечественный стандарт ЭЦП заменен на новый ГОСТ Р 34.10–2001 [7].

Проверка чисел Мерсенна на простоту

Напомним, что число $M_s = 2^s - 1$ ($s \geq 2$ – простое число) называется числом Мерсенна. Критерием простоты чисел Мерсенна служит следующее утверждение [3].

Теорема. Число Мерсенна $M_s = 2^s - 1$, где $s \geq 3$ – нечетное число, является простым тогда и только тогда, когда

– число s – простое;
– выполняется сравнение $L_{n-2} \equiv 0 \bmod M_s$, где последовательность $\{L_k\}$ формируется по такому правилу:

$$L_0 = 4,\ L_{k+1} = (L^2_k - 2) \bmod M_s \text{ при } k \geq 0.$$