УДК 511+519.719.2

Д.В. Кручинин, В.В. Кручинин

Метод построения алгоритмов проверки простоты натуральных чисел для задач защиты информации

Предложен метод построения алгоритмов проверки простоты натуральных чисел с применением аппарата обыкновенных производящих функций и операции суперпозиции функций $\ln(1+F(x))$, где $F(x)$ является обыкновенной производящей функцией с целыми коэффициентами. Показана связь существующих алгоритмов проверки простоты с предложенным методом. Рассмотрены примеры построения новых алгоритмов проверки простоты с использованием предложенного метода.

Ключевые слова: проверка простоты, логарифмическая производящая функция, суперпозиция производящих функций.

Задача построения алгоритмов проверки простоты

Простые числа играют важную роль в современной криптографии и защите информации. 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. Так, в хорошо известной криптографической системе с открытым ключом RSA потребность в выборе простых чисел имеет основополагающую позицию и от выбора простых чисел во многом определяется стойкость шифрования [1].

Все алгоритмы проверки простоты делятся на две большие подгруппы: детерминированные и вероятностные проверки. Алгоритмы первой группы позволяют точно определить, является число простым или составным. Алгоритмы второй группы позволяют это определить, но с некоторой вероятностью ошибки. Многократное их повторение для одного числа, но с разными параметрами, обычно позволяет сделать вероятность ошибки сколь угодно малой величиной. Вероятностные проверки позволяют точно говорить, является ли число составным, если число не удовлетворяет некоторым условиям проверки, поскольку для простых чисел эти условия являются обязательными. Если же число удовлетворяет всем условиям проверки, то это еще не означает, что число является простым, говорят, что «n – простое число с некоторой вероятностью». Зачастую вероятностные проверки являются более быстрыми и обладают меньшей вычислительной сложностью, чем детерминированные.

В настоящее время широко используются вероятностные проверки простоты, так объединенный алгоритм Рабина–Миллера широко используется в криптосистемах с открытым ключом для построения простых ключей длиной 512, 1024 и 2048 бит.

Логарифмические производящие функции

Логарифмической производящей функцией будем называть степенной ряд вида

$$\sum_{n=1}^{\infty} \frac{a(n)}{n} x^n, \tag{1}$$

где $a(n)$ – целочисленная последовательность.

Логарифмическая производящая функция отличается от обычной тем, что в качестве коэффициентов степенного ряда берутся элементы $a(n)$, деленные на порядковый номер, т.е. $\frac{a(n)}{n}$. Еще одно отличие заключается в том, что отсутствует свободный член.

Одно из свойств суперпозиции логарифмических производящих функций гласит, что значение суперпозиции логарифмических производящих функций без n члена для простого n будет числом целым. Обратное утверждение является неверным.

$$z(n) = \sum_{k=1}^{n-1} F^{\Delta}(n,k) r(k) = \sum_{k=1}^{n-1} F^{\Delta}(n,k) \frac{a(k)}{k}, \tag{2}$$

где $F^{\Delta}(n,k)=\sum_{\pi_k\in C_n} f(\lambda_1)f(\lambda_2)...f(\lambda_k)$ – композита производящей функции $F(x)=\sum_{n>0}f(n)x^n$, $R(x)=\sum_{n>0}r(n)x^n$ – логарифмическая производящая функция [2].

Метод построения алгоритмов проверки на простоту

Используя свойства логарифмических производящих функций, становится возможным построение различных алгоритмов вероятностных проверок натурального числа на простоту с применением аппарата обыкновенных производящих функций и операции суперпозиции.

Для построения алгоритмов проверки на простоту будем использовать обыкновенную и логарифмическую производящие функции соответственно $F(x)=\sum_{n>0}f(n)x^n$ и $R(x)=\sum_{n>0}r(n)x^n=\sum_{n>0}\frac{a(n)}{n}x^n$. Далее необходимо получить суперпозицию данных производящих функций:

$$Z(x)=R(F(x)). \tag{3}$$

Для вычисления суперпозиции, согласно [3], первостепенное значение имеет вычисление композиты обыкновенной производящей функции

$$F^{\Delta}(n,k)=\sum_{\pi_k\in C_n} f(\lambda_1)f(\lambda_2)...f(\lambda_k), \tag{4}$$

поскольку справедлива формула

$$z(n)=\sum_{k=1}^{n}F^{\Delta}(n,k)r(k). \tag{5}$$

Зная [2], что значения производной данной суперпозиции для любого *n* будут целыми, произведем дифференцирование, которое позволяет работать с суперпозицией как с целочисленной последовательностью.

Следующим шагом будет упрощение полученного выражения, а именно получение выражения, зависящего только от *n*, без дополнительного суммирования по *k*. Для упрощения можно воспользоваться энциклопедией целочисленных последовательностей [4]. К сожалению, не всегда получается упростить данные выражения, что влияет только на вычислительную сложность. Поэтому такие выражения имеют теоретический интерес, с одной стороны, с другой же стороны, возможно использование приближенных методов оценки выражений, но не всегда целесообразно.

Далее идет приведение данного выражения к тождественному выражению (2), а именно деление на *n* и вычитание *n*-го члена суммы из выражения суперпозиции.

В итоге после всех преобразований получается выражение, в идеале зависящее только от *n*, которое будет целым для простых *n*, т.е. алгоритм вероятностной проверки на простоту.

В зависимости от параметров суперпозиции, а именно от самой логарифмической производящей функции, от композиты подставляемой производящей функции, выражение (2) имеет различные числовые и вероятностные характеристики и вычислительные трудности. Вероятность в данных проверках появляется за счет нахождения суммы элементов композиты, т.е. зависит от коэффициентов производящей функции *F(x)*.

Примеры построения алгоритмов проверки простоты

1. Проверка на простоту на основе суперпозиции $\ln\left(\frac{1}{1-F(x)}\right)$, где $F(x)=\frac{1}{1-x}=1+x+x^2+...+x^n...$

Для нахождения суперпозиции необходимо получить выражение для композиты данной производящей функции. Известно [3], что композита производящей функции $F(x)$ имеет следующий вид:

$$F^{\Delta}(n,k)=\binom{n-1}{k-1}, \tag{6}$$

тогда коэффициенты суперпозиции будут равны

$$z(n)=\sum_{k=1}^{n}F^{\Delta}(n,k)r(k)=\sum_{k=1}^{n}\frac{1}{k}\binom{n-1}{k-1}=\frac{2^n-1}{n}. \tag{7}$$

Выражение

$$\sum_{k=1}^{n-1}\frac{1}{k}\binom{n-1}{k-1}=\frac{2^n-2}{n} \tag{8}$$

является целым числом для простых n.

После преобразования получим

$$2^{n-1}\equiv 1 \bmod n, \tag{9}$$

что является тестом на простоту на основе малой теоремы Ферма по основанию 2.

Преимуществами данного теста являются его быстрота и маленькая трудоемкость, поскольку ЭВМ основывается на двоичной системе счисления и 2^{n-1} является последовательностью единиц до разряда n. Поэтому основная трудоемкость заключается в делении на n.

Этот тест эффективно использовать для обнаружения составных чисел, а также на начальной стадии проверки простоты для больших чисел. В зависимости от требуемой погрешности проверки чисел на простоту целесообразно использовать как в криптосистемах, так и для решения прикладных задач.

2. Проверка на простоту на основе суперпозиции $\ln\left(\frac{1}{1-F(x)}\right)$, где $F(x)=\alpha x+\beta x^2$.

В [3] показано, что композита производящей функции $F(x)=\alpha x+\beta x^2$ имеет следующий вид:

$$F^{\Delta}(n,k,\alpha,\beta)=\binom{k}{n-k}\alpha^{2k-n}\beta^{n-k}. \tag{10}$$

Таким образом, для нахождения суперпозиции воспользуемся выражением

$$z(n)=\sum_{k=1}^{n}\binom{k}{n-k}\alpha^{2k-n}\beta^{n-k}\frac{1}{k}. \tag{11}$$

При $\alpha=1,\beta=1$ получим проверку на простоту, основанную на числах Люка [2, 4]: выражение

$$\frac{L_n-1}{n} \tag{12}$$

является целым для простых чисел, или $L_n\equiv 1 \bmod n$, где L_n – числа Люка.

Рассмотрим еще один частный вариант этой последовательности, где один из параметров больше единицы, например $\alpha=2,\beta=1$. Таким образом, получаем

$$z(n)=\sum_{k=1}^{n}\binom{k}{n-k}2^{2k-n}\frac{1}{k}, \tag{13}$$

$$\frac{n}{2}z(n)=[1,3,7,17,41,99,239,577,1393,3363,...]. \tag{14}$$

Данная последовательность является последовательностью целых чисел A001333 [4], откуда формула данной последовательности имеет вид

$$\frac{\left(1-\sqrt{2}\right)^n+\left(1+\sqrt{2}\right)^n}{2}. \tag{15}$$

Преобразуя это выражение, получаем проверку простоты натурального числа: если n простое натуральное число, то выражение

$$\frac{\left(1-\sqrt{2}\right)^n+\left(1+\sqrt{2}\right)^n-2^n}{n} \tag{16}$$

является целым.

3. Проверка на простоту на основе суперпозиции $\ln\left(\frac{1}{1-F(x)}\right)$, где $F(x)=\frac{1-\sqrt{1-4x}}{2x}$.

В [2] рассмотрены данные условия и получена следующая проверка на простоту: если n простое натуральное число, то значение выражения

$$\frac{1}{n}\left(\binom{2n-1}{n-1}-1\right) \tag{17}$$

является целым.

При использовании данной проверки на простоту при небольших значениях n была замечена следующая закономерность: ошибочно определялись простыми только квадраты и кубы самих, причем всех, простых чисел.

4. Проверка на простоту на основе чисел Каталана и $F(x)=\alpha x+\beta x^2$.

Выше был рассмотрен пример, где производящая функция чисел Каталана была подставляемой функцией. Теперь же рассмотрим вариант, когда логарифмическая производящая функция будет основываться на числах Каталана. Для этого проинтегрируем производящую функцию чисел Каталана и получим логарифмическую производящую функцию

$$A(x)=\frac{1-\sqrt{1-4x}}{2x}, \tag{18}$$

$$a(n)=\frac{1}{n}\binom{2n-2}{n-1}, \tag{19}$$

$$R(x)=\int A(x)dx=\int\frac{1-\sqrt{1-4x}}{2x}dx=\ln\left(\sqrt{1-4x}+1\right)-\sqrt{1-4x}, \tag{20}$$

$$r(n)=\frac{1}{n^2}\binom{2n-2}{n-1}. \tag{21}$$

Для нахождения коэффициентов суперпозиции

$$Z(x)=\ln\left(\sqrt{1-4F(x)}+1\right)-\sqrt{1-4F(x)} \tag{22}$$

воспользуемся выражением

$$z(n)=\sum_{k=1}^{n}\binom{k}{n-k}\alpha^{2k-n}\beta^{n-k}\frac{1}{k^2}\binom{2k-2}{k-1}. \tag{23}$$

Рассмотрим частный вариант этой последовательности, где $\alpha=1,\beta=1$. Таким образом, получаем

$$z(n)=\sum_{k=1}^{n}\binom{k}{n-k}\frac{1}{k^2}\binom{2k-2}{k-1}. \tag{24}$$

Исходя из полученного выражения, получим проверку на простоту: если n простое натуральное число, то значение выражения

$$z(n)=\sum_{k=1}^{n-1}\binom{k}{n-k}\frac{1}{k^2}\binom{2k-2}{k-1} \tag{25}$$

является целым.

При использовании данной проверки на простоту при небольших значениях n была замечена следующая закономерность: ошибочно определялись простыми только квадраты простых чисел. Поведение теста для больших чисел не рассматривалось ввиду большой трудоемкости алгоритма. Для дальнейшего использования теста необходимо его упрощение, а именно: приведение к выражению, зависящему только от n, и избавление от суммирования.

Заключение

Получен единый аналитический метод генерации алгоритмов проверки на простоту натурального числа, основанный на комбинаторных решениях и производящих функциях.

На основе полученного метода разработано несколько вероятностных проверок на простоту, основанных на числах Люка, Каталана и др. Проанализированы их показатели, такие как количество ошибок и вычислительная сложность.

Показана взаимосвязь полученного метода с существующими тестами на простоту. Так, при помощи полученного метода получен тест на основе малой теоремы Ферма по основанию 2.

Благодаря полученным результатам определены единый подход и математическое обоснование проверок на простоту натурального числа. Появляется возможность генерации более быстрых, качественных и менее трудоемких алгоритмов проверок простоты за счет подбора параметров, используемых в методе генерации. Следовательно, предполагаются улучшения существующих крип-

тографических систем, основанных на генерации и проверке простых чисел, а именно увеличении быстроты работы и уменьшении вычислительной трудности задействованных алгоритмов.

Литература

1. Ященко В.В. Введение в криптографию: учеб. пособие / В.В. Ященко, Н.П. Варновский, Ю.В. Нестеренко. – М.: МЦНМО, 2000. – 272 с.

2. Kruchinin D.V. Superposition's Properties of Logarithmic Generating Functions. – eprint arXiv: 1109.1683, 2011 [Электронный ресурс]. – Режим доступа: http://arxiv.org/PS_cache/arxiv/pdf/1109/1109.1683v1.pdf, свободный (дата обращения: 28.10.11).

3. Кручинин В.В. Комбинаторика композиций и ее приложения. – Томск: В-Спектр, 2010. – 156 с.

4. Sloane J.A. The On-Line Encyclopedia of Integer Sequences [Электронный ресурс]. – Режим доступа: http://www.research.att.com/njas/sequences, свободный (дата обращения: 28.10.11).

Кручинин Дмитрий Владимирович
Студент 5-го курса ТУСУРа
Тел.: +7-913-845-99-04
Эл. почта: KruchininDm@gmail.com

Кручинин Владимир Викторович
Д-р техн. наук, доцент каф. ПрЭ ТУСУРа, зав. лабораторией инструментальных систем моделирования и обучения научного управления Института инноватики ТУСУРа
Тел.: 8 (382-2) 42-30-67
Эл. почта: kru@2i.tusur.ru

Kruchinin D.V., Kruchinin V.V.

The method of constructing algorithms for primality testing natural numbers for the objectives of information security

We offer a method of algorithms development for testing the primality of natural numbers by using the apparatus of ordinary generating functions and operations of superposition of functions ln (1 + F (x)), where F (x) is an ordinary generating function with integer coefficients. The relation is shown between the existing algorithms for testing primality to the proposed method. There are given the examples of the development of new algorithms for testing primality by the proposed method.

Keywords: primality testing, logarithmic generating function, superposition of generating functions.
