УДК 681.3.06

АНАЛИЗ ВЫЧИСЛИТЕЛЬНОЙ СЛОЖНОСТИ АЛГОРИТМОВ ТЕСТИРОВАНИЯ НА ПРОСТОТУ ЧИСЕЛ МНОГОКРАТНОЙ ТОЧНОСТИ

КАЧКО Е.Г., СВИНАРЕВ А.В., МЕЛЬНИКОВА О.А.

В статье проводится сравнительный анализ вычислительной сложности методов тестирования чисел на простоту, применяемых в криптографии. Указываются размеры таблицы делителей, обеспечивающие минимальную вычислительную сложность комбинированного теста, приводятся теоретические и экспериментальные оценки вычислительной сложности комбинированного теста.

1. Введение

В настоящее время в системах защиты информации широко используются несимметричные криптографические преобразования. Среди них алгоритм цифровой подписи и шифрования RSA, класс подписей Эль-Гамаля (алгоритмы DSS и ГОСТ Р 34.10-94), алгоритм Диффи-Хеллмана и другие. В несимметричной криптографии в качестве ключевых параметров повсеместно применяются простые числа. В алгоритме RSA модуль преобразования N представляет собой произведение двух простых чисел p и q. В набор общесетевых параметров алгоритмов класса Эль-Гамаля входят простые числа p и q, причем q делит p-1. Одним из общесетевых параметров алгоритма Диффи-Хеллмана является простое число p.

Отметим, что для обеспечения требуемого уровня стойкости криптографических систем необходимо использовать простые числа большой разрядности (256, 512, 1024 и более бит) и, следовательно, трудоемкие операции арифметики многократной точности. Кроме того, используемые простые числа во многих криптографических схемах должны удовлетворять еще ряду довольно жестких требований и ограничений для затруднения ведения криптоанализа, поскольку стойкость несимметричных систем основана, как правило, на сложности факторизации многоразрядного числа либо на сложности решения задачи дискретного логарифмирования по модулю большого простого числа. Подобные ограничения создают дополнительные трудности при формировании многоразрядных простых чисел, поскольку ведут к значительным вычислительным затратам на повторную перегенерацию при неудовлетворительном качестве сформированных чисел. В криптографических системах вычислительная сложность алгоритмов является, как правило, критическим параметром, поэтому особую актуальность приобретает проблема быстрого формирования простых чисел с заданными свойствами, т.е. создания быстрого комплексного теста проверки простоты задаваемых чисел, обеспечивающего требуемый уровень достоверности.

2. Обзор известных методов тестирования на простоту чисел

Разработан ряд методов проверки простоты чисел [1–4]. Их можно разделить на три группы. К первой относятся математически строгие методы, которые позволяют однозначно ответить на вопрос о простоте числа, ко второй — вероятностные методы, позволяющие подтвердить предположение о простоте числа только с определенной вероятностью, сколь угодно близкой, но не равной единице. К третьей группе относятся методы, базирующиеся на использовании гипотез, в частности, на предположении истинности очень правдоподобной, но не доказанной гипотезы.

К математически строгим (аналитическим) методам относятся: метод пробных делений, решето Эратосфена, методы прямой факторизации с использованием различных алгоритмов, чисел специального вида (Ферма, Мерсенна и других), а также методы, использующие известные разложения чисел $N \pm 1$.

Основными из вероятностных методов являются: тест, основанный на использовании теоремы Ферма, тесты Лемана, Соловея-Штрассена, Рабина-Миллера.

К основным методам, базирующимся на использовании предположения о справедливости определенной гипотезы (в частности, расширенной гипотезы Римана), относятся следующие тесты: Полларда, Миллера (обычный и полиномиальный), Адлемана-Румели, Ленстры-Коэна и другие.

В данной работе исследуются вероятностные методы, наиболее широко применяемые в криптографии с точки зрения их вычислительной сложности.

3. Анализ вычислительной сложности вероятностных тестов

При работе с криптографическими системами основным критерием предпочтительности выбора того или иного метода проверки простоты чисел является минимизация вычислительной сложности, а следовательно, и времени построения простых чисел. Подчеркнем, что здесь подразумевается построение больших простых чисел (длиной не менее 256 бит). Далее мы проведем сравнительный анализ трех вероятностных методов с точки зрения их вычислительной сложности, при этом не будем накладывать никаких дополнительных требований на простое число, например, на вид разложения чисел $p \pm 1$.

1. Тест Соловея-Штрассена. Основан на теореме [3]: для любого нечетного составного числа N количество чисел A, для которых выполняются условия:

$$1)\ 1 < A < N;\ 2)\ \text{НОД}(A, N) = 1;\ 3)\ A^{\frac{N-1}{2}} \equiv \left(\frac{A}{N}\right) \bmod N,$$

где $\left(\frac{A}{N}\right)$ – символ Якоби, не превышает $N/2$.

Из теоремы следует, что если для случайно выбранного A, $1 < A < N$ выполняются условия 2) и 3), то N является простым с вероятностью не менее $1/2$. Если условия выполняются для k различных значений A, число N может быть составным с вероятностью не более 2^{-k}.

Вычислительная сложность теста определяется так:

$$I(k) \approx k(I_{pow}(l) + I_J(l)),$$

где l – количество разрядов в тестируемом числе N; $I_J(l)$ – вычислительная сложность определения символа Якоби для чисел разрядности l; $I_{pow}(l)$ – вычислительная сложность модульного возведения в степень чисел разрядности l.

2. Тест Лемана. Согласно малой теореме Ферма [3]: если N – простое, НОД(N,A) = 1, то

$$A^{N-1} \equiv 1 \pmod N. \qquad (1)$$

Выражение (1) для нечетных N можно представить в виде

$$\left(A^{\frac{N-1}{2}}+1\right)\left(A^{\frac{N-1}{2}}-1\right) \equiv 0 \pmod N. \qquad (2)$$

Тест Лемана проверяет условия (2):

1) выбрать случайное число A < N;

2) вычислить $Z := A^{\frac{N-1}{2}} \bmod N$;

3) если $Z \not\equiv \pm 1 \pmod N$, то N – составное, в противном случае – простое с вероятностью 50 %.

Если проверка выполнена для k различных значений A, то N может быть составным с вероятностью не более 2^{-k}.

Вычислительная сложность теста

$$I(k) \approx kI_{pow}. \qquad (1)$$

3. Тест Рабина-Миллера. Назовем число $n = 2^t n_0+1$ сильным псевдопростым по основанию a, если оно удовлетворяет одному из условий:

1) $a^{n_0} \equiv 1 \pmod n$;

2) существует j, $0<j<t$, такое, что

$$a^{2^j n_0} \equiv -1 \pmod n.$$

Справедлива теорема [3]: если $n > 9$ и n является составным, то количество чисел a, $1<a<n$, взаимно простых с n, таких, что n является сильным псевдопростым по основанию a, не превышает $n/4$.

Приведем алгоритм, основанный на данной теореме [4,5]:

1) выбрать h и M такие, что $N = 2^h M + 1$, где h – наибольшая степень 2, делящая N – 1;

2) выбрать случайное число A < N;

3) установить j := 0; $Z := A^M \bmod N$;

4) если $Z \in \pm 1 \pmod N$, то N может быть простым;

5) если j > 0 и Z = 1, то N – составное;

6) установить j := j + 1;

7) если j = h и $Z \neq N - 1$, то N – составное;

8) если Z = N – 1, то N может быть простым;

9) установить $Z = Z^2 \bmod N$, перейти на шаг 5.

Повторение теста k раз позволяет сделать вывод о том, что число N может быть составным с вероятностью не более 4^{-k}.

Вычислительная сложность оценивается как

$$I(k) \approx k(I_{pow}(l) + hI_{msqr}(l)), \qquad (3)$$

где $I_{msqr}(l)$ – вычислительная сложность модульного возведения в квадрат чисел разрядности l.

Пусть вероятность того, что величина h примет значение $h_0 > 0$, равна 2^{-h_0}. Тогда математическое ожидание h определяется из выражения

$$\bar{h} = \sum_{h_0=1}^{\infty} h_0 2^{-h_0} = 2.$$

Подставляя $\bar{h}$ в (3), получаем окончательную оценку вычислительной сложности:

$$I(k) \approx k(I_{pow}(l) + 2I_{msqr}(l)).$$

Для сравнительной оценки вероятностных методов будем использовать математическое ожидание их вычислительной сложности:

$$\bar{I} = \sum_{i=1}^{\infty} p_i i I_c = I_c \sum_{i=1}^{\infty} i p_i. \qquad (4)$$

Здесь p_i – вероятность отсеивания составного числа на i-м цикле вероятностного теста; I_c – вычислительная сложность одного цикла теста.

Вероятность p_i определяется из выражения

$$p_i = p_c^{i-1}(1-p_c),$$

где p_c – вероятность прохождения составного числа через 1 цикл теста. Тогда уравнение (4) может быть представлено в виде

$$\bar{I} = I_c(1-p_c)\sum_{i=1}^{\infty} i p_c^{i-1}. \qquad (5)$$

Заметим, что для приведенных выше тестов значения вероятности p_c имеют вид 1/m, где m может принимать значение 2 или 4. В этом случае (5) можно упростить:

$$\bar{I} = I_c \frac{m-1}{m} \sum_{i=1}^{\infty} \frac{i}{m^{i-1}} = \frac{I_c m}{m-1}. \qquad (6)$$

По формуле (6) получим значения $\bar{I}$ для рассмотренных вероятностных тестов (табл. 1).

Таблица 1

Сравнение полученных результатов показывает, что $I_3 < I_2 < I_1$, т.е. тест Рабина-Миллера является наиболее эффективным по вычислительной сложности.

4. Анализ вычислительной сложности комбинированного теста

Для дальнейшего снижения вычислительной сложности тестирования в [5] предложен комбинированный метод, включающий в себя детерминированный и вероятностный тест. Сущность его заключается в том, что перед проведением вероятностного теста выполняется деление числа N на малые простые числа и, как показано ниже, на этом этапе отсеивается значительная доля составных чисел. Выигрыш по вычислительной сложности достигается за счет того, что операция деления выполняется значительно быстрее, чем операция возведения в степень [6].

Комбинированный тест содержит пункты:

1) предварительно строится таблица делителей T, в которую заносятся n_p первых простых чисел: 3, 5, 7;

2) для i, принимающего значения от 1 до n_p, выполняется шаг 3;

3) если N делится на T[i], то закончить тест с результатом «N является составным»;

4) число N проверяется по какому-либо вероятностному тесту.

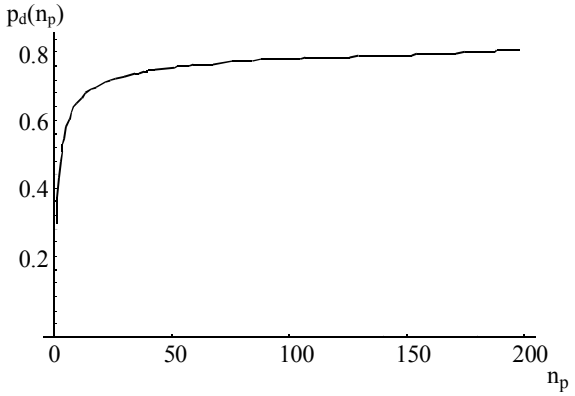
При анализе данного метода возникает задача выбора размера таблицы простых чисел n_p, при котором вычислительная сложность тестирования достигает минимума.

Математическое ожидание вычислительной сложности комбинированного теста можно определить по формуле

$$\bar{I} = I_{d1}(l)\sum_{i=1}^{n_p} i\left(1 - p_d(i)\right) + \left(1 - p_d\left(n_p\right)\right)\bar{I}_{pt}, \qquad (7)$$

где $I_{d1}(l)$ – вычислительная сложность операции деления l-разрядного числа многократной точности на число однократной точности; $p_d(i)$ – вероятность того, что N не делится на первые i простых чисел из таблицы; $\bar{I}_{pt}$ – математическое ожидание вычислительной сложности вероятностного теста, применяемого на шаге 4.

Аналитически определить вероятности $p_d(i)$ сложно, поэтому были проведены их экспериментальные оценки. Результат представлен на рисунке.



Зависимость вероятности p_d от размера таблицы делителей n_p

Если на шаге 4 выполняется тест Рабина–Миллера, то (7) примет вид

$$\bar{I} = I_{d1}(l)\sum_{i=1}^{n_p} i\left(1 - p_d(i)\right) + \frac{4}{3}\left(1 - p_{d\Sigma}\right)\left(I_{pow}(l) + 2I_{msqr}(l)\right). \qquad (8)$$

Для определения значений n_p, обеспечивающих минимум $\bar{I}$, в выражение (8) подставим оценки вычислительной сложности операций арифметики многократной точности для длины машинного слова 32 бита, полученные авторами в [6, 7]:

$I_{d1}(l) \approx 6l$;

$I_{msqr}(l) \approx 14l^2 + 27l$;

$I_{pow}(l) \approx 556l^3 + 1240l^2 + 245l$.

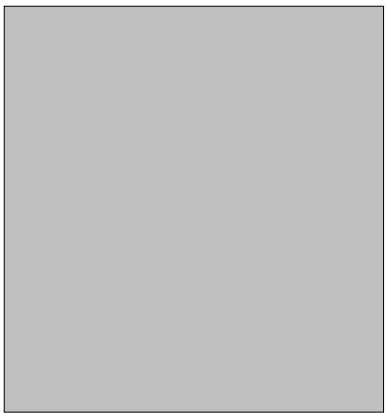
Результаты расчетов для часто используемых длин простых чисел приведены в табл. 2.

5. Особенности тестирования на простоту общесетевых параметров алгоритма ГОСТ Р 34.10-94

Алгоритм цифровой подписи ГОСТ Р 34.10-94 предполагает использование в качестве общесетевых параметров двух простых чисел: простого числа q длиной 256 бит и простого числа p длиной 512 или 1024 бита, причем

$p = qN + 1$, где N – четное число.

Таблица 2



Для построения простого числа в соответствии с ГОСТ Р34.10-94 выбирается минимальное простое число длиной 32 бита и выполняется постепенное удвоение длины числа до тех пор, пока не будут получены числа требуемых длин.

Рассмотрим алгоритм формирования простого числа q_{i+1} длиной 2l по заданному простому числу q_i длиной l.

1. Вычисляется минимальное четное число k, такое что kq_i имеет требуемую длину 2l.

2. Формируется случайная сдвижка g для обеспечения возможности формирования простого во всем диапазоне числа длиной 2l:

3. Определяется случайное число r длиной 2l и число $g = \frac{r}{2q_i}$.

4. Вычисляется значение $q_{i+1} = Nq_i + 1$, где $N = k + g$. Если длина q_{i+1} превосходит требуемую, перейти на шаг 2.

5. Пока q_{i+1} не простое, выполнять шаг 5.

6. Вычислить $q_{i+1} = q_{i+1} + 2q_i$; $N = N + 2$.

Для проверки числа q_{i+1} на простоту используется аналитический метод, основанный на проверке выполнения условий малой теоремы Ферма:

$$2^{q_{i+1}-1} \equiv 1\left(\mathrm{mod}\, q_{i+1}\right) \qquad (9)$$

и следствия из теоремы Люка [3]:

$$2^N \not\equiv 1\left(\mathrm{mod}\, q_{i+1}\right). \qquad (10)$$

Число считается простым, если выполняются оба эти условия.

Вычислительная сложность проверки

$$I_g = \frac{3}{2} I_{pow}(l) \qquad (11)$$

Для получения 256-битного q и 512-битного p требуется выполнить 4 шага увеличения длины ($32 \to 64 \to 128 \to 256 \to 512$), а для 1024-битного p требуется 5 шагов. На каждом шаге выполняется поиск простого числа с использованием одной операции возведения в степень при неблагополучном исходе и двух операций возведения в степень – при благополучном. Количество итераций увеличивается пропорционально длине простого числа и на длине 512 бит оно составляет порядка 100. Для длины 1024 бита эти значения еще больше, поэтому вычислительная сложность генерации простых p и q достаточно велика.

Для уменьшения вычислительной сложности предлагается использовать комбинированный тест совместно с проверками (9) и (10) на каждой итерации, при этом сохраняется справедливость со-

отношения (7), имеющего в данном случае следующий вид:

$$\bar{I}_g = I_{d1}(l)\sum_{i=1}^{n_p} i\left(1 - p_d(i)\right) + \frac{3}{2}\left(1 - p_d\left(n_p\right)\right)I_{pow}(l).$$

Комбинированный тест дает выигрыш по вычислительной сложности по сравнению с (11):

$$\Delta\bar{I}_g = \frac{3}{2}p_{d\Sigma}I_{pow}(l) - I_{d1}(l)\sum_{i=1}^{n_p} i\left(1 - p_d(i)\right).$$

6. Результаты экспериментальных исследований вычислительной сложности

Комбинированный тест на простоту был реализован программно на 32-разрядной ПЭВМ. В ходе экспериментов получены значения времени выполнения теста с вероятностью необнаружения составного числа 2^{-100} для различных вероятностных тестов и комбинированного метода (табл. 3).

Таблица 3

Проведены эксперименты по определению эффективности использования комбинированного теста для обеспечения минимального времени генерации общесетевых параметров алгоритма ГОСТ Р 34.10 для длины 512 и 1024 бита. На длине тестируемого числа 512 бит при использовании таблицы из 100 делителей время генерации уменьшается в 2 раза. При длине тестируемого числа 1024 бита и таблице из 1000 делителей время генерации уменьшается более чем в 5 раз.

Литература: 1. *Кнут Д.* Искусство программирования для ЭВМ/Пер. с англ. М.: Мир, 1977. Т.2. 728 с. 2. *Миллер Г.Л.* Гипотеза Римана и способы проверки простоты чисел // Кибернетический сборник. М.: Мир, 1985. Вып. 23. С. 31-50. 3. *Уильямс Х.* Проверка чисел на простоту с помощью вычислительных машин // Кибернетический сб. М.: Мир. 1985. Вып. 23. С. 51-99. 4. *Василенко О.Н.* Современные способы проверки простоты чисел // Кибернетический сб. М.: Мир, 1988. Вып. 25. С. 162-188. 5. *Schneier B.* Applied Cryptography Second Edition: protocols, algorithms and source code in C. New York: John Wiley & Sons. 1996. 758 p. 6. *Горбенко И.Д., Качко Е.Г., Свинарев А.В., Мельникова О.А.* Программирование операций многократной точности **//** Безопасность информации. 1995. №1. С. 18-21. 7. *Свинарев А.В.* Методы ускорения процедур цифровой подписи класса Эль-Гамаля // Радиотехника. 1997. Вып. 104. С. 173—178.

Поступила в редколлегию 22.03.98

Качко Елена Григорьевна, доцент кафедры ПОЭВМ УРЭ. Научные интересы: криптология, системное ограммное обеспечение. Адрес: 310726, Украина, Харьв, пр. Ленина, 14, тел. 30-24-52, 45-53-45.

Свинарев Андрей Владимирович, аспирант кафедры М ХТУРЭ. Научные интересы: криптографические тоды защиты информации в компьютерных системах. Адрес:310726, Украина, Харьков, пр. Ленина, 14. тел. 30-24-52.

Мельникова Оксана Анатольевна, ассистент кафедры ЭВМ ХТУРЭ. Научные интересы: криптографические методы защиты информации в компьютерных системах. Адрес: 310726, Украина, Харьков, пр. Ленина, 14, тел. 30-24-52,38-46-92.

УДК 681.3.04

ПОСТРОЕНИЕ ШТРИХОВЫХ КОДОВ С ПОВЫШЕННОЙ ПОМЕХОУСТОЙЧИВОСТЬЮ

ДИЧКА И.А., СУЛЕМА Е.С.

В статье рассматривается задача построения шриховых кодов с повышенной помехоустойчивостью. Задача решается с использованием методов теории помехоустойчивого кодирования, а именно V-кодов. Рассмотрены примеры построения штриховых кодов с повышенной помехоустойчивостью, а также алгоритм декодирования таких штриховых кодов.

Использование ШК подразумевает три этапа: 1) изготовление ШК-этикетки; 2) хранение информации в виде ШК; 3) считывание ШК с последующим декодированием исходной информации.

На каждом из этих этапов возможно возникновение ситуаций, приводящих к искажению ШК-символов, а значит, и к ошибкам в исходной информации. К ним можно отнести следующее:

1) сбой устройства печати, а также дефекты печати (неровности, светлые и темные пятна, неконтрастное изображение, сворачивание (расплывание) краски); 2) непродуманное оформление ШК-этикетки (в качестве носителя ШК выбрана глянцевая бумага, материал типа полиэтилена с неустойчивой краской); 3) загрязнение ШК-этикетки в процессе хранения и использования; 4) сбой устройства считывания.

Основная идея построения ШК с обнаруживающей и корректирующей способностью знаков состоит в следующем: перейти от контроля правильности всего ШК-символа (ШК-символ — совокупность данных, представленных в виде ШК, построенная по определенным правилам) в целом, осуществляемого путем добавления к исходному слову контрольного символа, к контролю правильности каждого ШК-знака (под знаком понимают совокупность штрихов и пробелов, соответствующую определенному алфавитно-цифровому символу). Тогда можно построить ШК, который позволяет исправлять, например, все однократные ошибки, возникающие в пределах каждого знака. Значит, при представлении n-символьного исходного слова такой ШК теоретически позволит исправить n-кратную ошибку, возникшую в пределах всего ШК-символа.

Проведенные авторами исследования показали, что наиболее вероятными являются ошибки вставки разрядов в двоичном векторе знака (компоненты двоичного вектора знака — двоичные нули и едини-