УДК 004.056 /002:006.354

АЛГОРИТМ БЫСТРОЙ ГЕНЕРАЦИИ КЛЮЧЕЙ В КРИПТОГРАФИЧЕСКОЙ СИСТЕМЕ RSA.

А.А. Балабанов[1)], А.Ф. Агафонов[1)], В.А. Рыку[2)]

1) Технический университет Молдовы

2) Международный независимый университет

г.Кишинев, Республика Молдова

Аннотация

Создание и анализ эффективных алгоритмов используются в шифровальных системах и, особенно, в алгоритме RSA. Часто, решение таких задач требует много времени, огромных интеллектуальных и финансовых затрат. Вот почему решение проблемы в этой области заслуживает специального внимания. В статье, предлагается один из вариантов модернизации генератора криптоключей на основе вероятностного алгоритма определения простоты числа как модификации классического алгоритма Ферма-Эйлера.

Abstract:

Works under creation and analysis of the effective algorithms used in cryptographic systems and especially for generation key in RSA algorithm. Frequently, the solution of such problems needs huge intellectual efforts, a lot of time and financial expenses. That is why solution or problems setting in this area deserves, in our opinion, special attention. In this article one variant of modernization generation key probability algorithm of a number simplicity definition is examined as a modification of a classical Fermat-Euler's algorithm.

Ключевые слова: криптография, криптосистема RSA, тесты простоты числа, генераторы криптоключей, скорость алгоритма, операционная емкость алгоритма, теория чисел, сравнительный анализ, сопоставительный анализ, формантный анализ, форманта числа

1. Введение. На протяжении нескольких последних десятилетий, человечество сталкивается с проблемой передачи все больших и больших объемов информации. И все актуальнее становится проблема защиты этой информации от несанкционированного доступа или просмотра. Среди наиболее эффективных и часто применяемых для защиты информации криптографических систем являются системы RSA, использующие в качестве криптоключа составное число в виде произведения двух простых чисел.

Математической основой современной криптографии является теория чисел. При всех значительных достижениях в данной области за последние десятилетия (усовершенствования имеющихся алгоритмов, появление новаторских идей, разработка и реализация совершенно новых алгоритмов), все же наиболее актуальным вопросом, присутствующим в каждой практической задаче в области криптографии и криптоанализа является задача проверки произвольного числа на простоту, которая в значительной степени определяет и скорость генерации криптоключей RSA.

2. Классификация алгоритмов тестирования чисел на простоту. На сегодняшний день существует множество способов (алгоритмов) проверки - является число простым или нет. Среди них присутствуют тесты и для чисел определенного вида или структуры, и для произвольных чисел. Последние представляют наибольший интерес, как с теоретической, так с практической точек зрения.

Все существующие алгоритмы тестирования чисел на простоту можно разделить на два класса:

- так называемые *детерминированные* тесты, которые в результате дают гарантированно точный ответ простое ли исследуемое число или нет
- *вероятностные тесты*, результат выполнения которых является достоверным лишь с некоторой достаточно высокой вероятностью.

2.1. Детерминированные тесты на простоту числа. Существует много методов, которые позволяют определить с вероятностью, равной единице, т.е. гарантированно достоверно, является ли заданное число *N* простым. Но, к сожалению, некоторые из них хорошо работают только для малых значений *N*; некоторые требуют полного разложения на множители числа *N-1*; а некоторые позволяют проверять только числа специального вида.

2.2. Вероятностные тесты. Наиболее распространенным и часто применимым на сегодняшний день является второй класс тестов простоты. Это вызвано, в первую очередь, значительно меньшим временем выполнения тестирования числа по сравнению с детерминированными тестами. Немаловажным является и тот факт, что именно такой подход в большинстве случаев используется на практике.

Требования высокой криптостойкости системы вынуждают применять на практике вероятностные тесты с высокой степенью достоверности. Еще совсем недавно это было, однако, весьма затруднительно из-за несовершенства технических характеристик вычислительных систем. Однако рост производительности вычислительной техники, наряду с повышением эффективности разрабатываемых алгоритмов дешифрования делают очерченную задачу реализуемой.

Большинство современных вероятностных методов тестирования чисел на простоту базируется на различных вариациях малой теоремы Ферма.

Согласно малой теореме Ферма, если число N - простое, то для любого целого a, не делящегося на N, имеет место сравнение $a^{N-1} \equiv 1(\mathrm{mod}\, N)$. Из этой же теоремы следует, что если данное уравнение неразрешимо хотя бы для одного числа a в интервале {1, 2, …, N-1}, то N - составное число. Обратное утверждение не является верным.

Задача повышения достоверности вероятностных тестов стала актуальной для криптографии, после того как были обнаружены «числа Кармайкла» и стало ясно, что тест на основе Малой теоремы Ферма «даёт сбои». Хотя таких чисел и не много (всего – то 16! в ряду первых 100 000 натуральных чисел). Первым алгоритмом, лишённым недостатка теста Ферма, был тест, предложенный Леманном. Этот тест «реагировал» на числа Кармайкла и отсеивал их как составные, чем они на деле и являются.

Этот недостаток, присущий тесту Ферма, устраняется в вероятностных тестах Соловея-Страссена и Миллера-Рабина. Делается это путём смены критерия отсева, который является более сильным, чем Малая Теорема Ферма.

Тест Соловея-Страссена базируется на критерии Эйлера и на понятии квадратичного вычета числа. *Квадратичным вычетом* по модулю p называется целое число a, которое представляет собою остаток от деления квадрата любого числа на модуль p, что можно записать в виде следующего сравнения

$$x^2 \equiv a(\mathrm{mod}\ p).$$

Если это сравнение не разрешимо, то число a называется *квадратичным невычетом* по модулю p.

Критерий Эйлера: Пусть p - простое число и пусть $a \in Z_p^*$. Тогда a является квадратичным вычетом, если и только если разрешимо сравнение:

$$a^{\frac{p-1}{2}} \equiv 1(\mathrm{mod}\ p)$$

Тест Соловея-Страссена обладает решающим преимуществом по сравнению с тестом Ферма: он является менее трудоемким и, после выполнения теста k раз, вероятность не отбраковки составного числа (вероятность определения простоты числа) не превосходит 0,5.

Но на практике степень достоверности теста Соловея-Штрассена не является достаточной.

Более высокую достоверность дает **вероятностный тест Миллера-Рабина.** Он получен путем объединения теста Ферма $a^{N-1} \equiv 1(\mathrm{mod}\ N)$, отсеивающего типичные составные числа, со следующим сравнением $a^{2^s t} \equiv 1(\mathrm{mod}\ p)$, где $2^s t = N - 1$ (N, t - нечетные), с помощью которого исключаются другие составные числа. После повторения данного теста k раз, вероятность не отбраковки составного числа не превосходит $\left(\frac{1}{4}\right)^k$.

Стандартная процедура использования теста Миллера-Рабина для нахождения простых чисел из назначенного ряда чисел или чисел заданного десятичного порядка, согласно [6], производится несколько раз и состоит в следующем.

Для испытуемого числа N произвольно выбирается небольшое число a и, если число N идентифицируется тестом, как составное, после первого же прохода, то оно отбрасывается, и переходят к испытанию другого числа N. Если же после первого прохода число идентифицировано как простое, то выбирается другое небольшое число a и прохождение теста повторяется. Рекомендуется брать не более k (чаще всего пяти) таких чисел a. Если на каком-то проходе, меньшем k, число N идентифицируется как составное, то тест прекращается и испытуемое число отбрасывается. Затем на испытание берётся следующее за N нечётное число того же десятичного порядка.

Заметим, что если следовать указанной процедуре и выбирать числа a из начального ряда простых чисел, то существует большая вероятность идентифицировать ***составное число*** как ***простое,*** поскольку максимальное число a окажется меньше критического [4,5] для данного массива испытуемых чисел.

Алгоритмы определения простоты числа (например, алгоритм Миллера-Рабина) используются в системе RSA, в частности, при генерации ключей.

3. Некоторые заметки о системе RSA. При всех достоинствах системы RSA ей присущи два существенных недостатка. А именно:

1) при одной и той же степени криптостойкости системы защиты скрываемой информации, длина *ключа RSA значительно превышает длину ключа в симметричных криптографических системах;*

2) *для генерации ключей RSA требуется намного больше времени*, чем для генерации ключей в симметричных системах.

Оценка F **операционной сложности** для системы RSA. Для шифрации информации в системе RSA ее операционная сложность оценивается величиной $F \approx \log^2 N$; при дешифрации же имеем $F \approx \log^3 N$, где N – длина ключа. Наиболее затратной является процедура генерации ключей $F \approx \log^4 N$ [3]. Последнее обстоятельство является камнем преткновения для использования RSA в мультимедийных средствах обработки и передачи информации, где необходима поточная шифрация в реальном времени (телевидение, телефония) [1,2], что исключает ее практическое использование в таких информационных системах. Поэтому актуальной проблемой в теории и практике применения RSA на сегодняшний день считается *поиск путей снижения операционной сложности генерации ключей RSA*. Если такое будет достигнуто, то можно будет строить закрытые линии передачи информации на базе RSA с быстрой сменой коротких ключей [1,2].

На практике **процесс генерации ключей в RSA** начинается с выбора двух простых чисел (p, q). Для этого с помощью генератора случайных чисел находится произвольное нечётное число требуемой разрядности. Далее это число проверяется с помощью того или иного теста на простоту. Если тест на простоту отрицателен, то выбирается следующее случайное число, выдаваемое генератором случайных чисел.

После того, как найдены два простых числа, их проверяют на разложимость, т.е. числа p±1 и q±1 как составные должны содержать один большой множитель. Если найденные числа не удовлетворяют данному условию, то их бракуют. И процедура поиска ключей возобновляется.

Здесь можно поступать и так. Просто, к предыдущему числу добавляется 2 и всё повторяется сначала. Когда, в конце концов, такие числа (p, q) будут найдены, приступают к вычислению закрытого параметра (ключа d).

Процедура отыскания закрытого параметра d основана на решении уравнения «замка системы», а именно,

$$ed = k\varphi(p) \times \varphi(q) + 1 = \varphi(N) + 1, \qquad (1)$$

где $\varphi(p)$ и $\varphi(q)$ - есть целочисленная функция Эйлера.

В случае когда p, q - простые числа $\varphi(p) = p - 1, \varphi(q) = q - 1$. Другими словами, необходимо найти решения диофантового уравнения первой степени:

$$ax = by + 1$$

Далее выбирается достаточно короткий открытый ключ e, но при этом следует иметь в виду, что он должен быть взаимно простым с $\varphi(N) = \varphi(p) \times \varphi(q)$, иначе диофантово уравнение будет неразрешимым. С этой целью используется стандартный алгоритм определения наибольшего общего делителя двух чисел.

Окончательное решение задачи генерации ключей сводится к нахождению неизвестных из указанного выше уравнения (1). Эта вычислительная процедура хорошо известна как алгоритм (метод) Эйлера.

4. Модификация алгоритма Эйлера. Каково может быть нововведение для сокращения времени поиска простых чисел? В чём суть вопроса?

Обычно процедура поиска простого числа использует в качестве отправной базы натуральный ряд чисел, который перенасыщен составными числами. Если каким-либо образом удастся отсеять «лишние» составные числа, то можно будет выбирать простые числа из другого ряда, где ***плотность простых чисел больше, чем в обычном натуральном ряде чисел.***

Здесь мы предлагаем использовать инструмент ***форманты чисел*** [7]. Так, например, числовой ряд, в котором отсутствуют числа, делящиеся на 2 и 3, описывается следующей формантой:

$$N = 6p + (1\ ,\ 5)$$

Если необходимо исключить и числа, делящиеся на 5, то форманта такого числового ряда примет вид:

$$N = 30p + (1\ ,\ 7\quad 11\ \ 13\ \ 17\ \ 19\ \ 23\ \ 29)$$

Пример. Допустим, что следует найти простое число, заключённое в интервале от 180 до 230. Такими числами в соответствии с указанной формантой будут:

181 187 191 193 197 199 203 209 211 217 221 223 227 229. (2)

Как видно, число простых чисел (они подчёркнуты) равно 8 из 14, т.е. с вероятностью больше чем 0.5 мы находим простое число, с первого же «захода».

Другой способ отсеивания составных чисел основан на использовании *свойства взаимно простых чисел*. Понятно, что *сумма двух взаимно простых чисел не содержит делителей этих чисел*. Возьмём, для цели демонстрации метода, например, такую форманту (формулу ряда)

$$1001 + 30\boldsymbol{к},\ \text{где}\ \boldsymbol{к} = 1\ 2\ 3\ 4\ 5\ 6\ 8.$$

В этом случае генерируется ряд чисел, в которых плотность простых чисел ещё больше, чем в (2). Действительно, полученный ряд для $\boldsymbol{к} = 1\ 2\ 3\ 4\ 5\ 6\ 8$ будет представлен следующими числами

1031 1061 1091 1121 1151 1181 1241 (3)

и содержит в себе 5 простых чисел из 7 (простые числа подчёркнуты).

5. Улучшение алгоритмов простоты. Существенным моментом в деле снижения оперативной ёмкости вычислительных процедур при генерации ключей может стать и улучшение алгоритмов определения простоты числа.

Уже рассмотренный ранее тест простоты Соловея-Штрассена основан на стандартной процедуре вычисления символа Якоби. Эта вычислительная процедура занимает время, эквивалентное величине $\log^2 N$. Нами предложен другой подход [4,5], позволяющий отказаться от этой непростой операции. Суть в том, что в соответствии с теоремой

квадратичной взаимности Гаусса, можно перейти от вычисления величины (a/N)[1] к обратной ей величине, а именно: к вычислению (N/a).

Напомним, что если $(a/N) = 1$, то a - квадратичный вычет N, а в случае, когда $(a/N) = -1$, a есть квадратичный невычет N. Понятно, что при известном (и достаточно малом) a найти вычет/невычет числа N намного проще. Достаточно произвести деление N на a и сверить результат с таблицей квадратичных вычетов a.

Если принять во внимание тот факт, что наиболее употребляемый для проверки на простоту числа стандартный вероятностный алгоритм Миллера-Рабина использует несколько индикативных чисел a, а также, что их достаточно много, если проверять числа N порядка 10^{50} и более, то можно значительно уменьшить (в несколько раз) количество индикативных чисел a, используя найденную нами зависимость количества чисел a от длины сильно простых чисел, которые в свою очередь связанны с величиной проверяемого на простоту числа N [5].

Все эти соображения легли в основу разработанного нами алгоритма генерации ключей, который была программно реализован. В приводимой ниже таблице дается сравнительная оценка быстродействия нашего алгоритма генерации ключей и аналогичного рабочего алгоритма фирмы «ДЕКАРТ», которая на профессиональном уровне занимается подобными работами.

Таблица 1. Время генерации ключей

	N	10^{10}	10^{20}	10^{30}	10^{40}	10^{60}	10^{80}	10^{100}
t, с наш алгоритм		0,000135	0,001044	0,001124	0,001130	0,01003	0,022588	0,025728
t, с алгоритм ДЕКАРТ		0,109	0,125	0.135	0,140	0,172	0,203	0,218

Как видно из таблицы, наш алгоритм намного быстрее генерирует ключи. Но особенно быстро - для весьма коротких ключей. Так, для ключей порядка 30 бит выигрыш составляет 1000 раз и более.

Такой выигрыш на малых длинах ключа особенно значим, поскольку это существенно как раз для создания криптографических систем с быстрой сменой ключей [1].

Литература

[1]. Балабанов А., Агафонов А. Перспективы, проблемы защиты информации в коммерческих системах и повышение надежности охранно-сигнальных систем. Доклад на подсекции Информатики, направление: «Проблемы информационной безопасности» Москва, 23 января 2008 г.

[1] Напомним, что выражение a/N есть символ Якоби, N-проверяемой на простоту число, a – т.н. индикативное число

[2]. Balabanov A., Agafonov A., Svirepov E. Системы мобильной связи с использованием методов криптозащиты на базе криптосистемы RSA. Simpozion stiintific international 2007, ULIM, Chisinau.
[3].Повышение достоверности определения составного числа вероятност-ными методами. Межд.конф. Bit+ “Informational Technologies -2005”.
[4]. Модернизированный тест Соловея-Штрассена.Межд.конф.Bit+ “Informational Technologies -2005”.
[5]. Эффективный алгоритм определения простоты числа для систем информационной безопасности. Межд.конф. Bit+ “Informational Technologies -2006”.
[6]. Шнайер Б. Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке Си. Москва: Триумф, 2002.
[7]. Агафонов А. Лекции по сопоставительному анализу. Кишинёв: Самиздат, 2005

Поступила: 02.07.09.