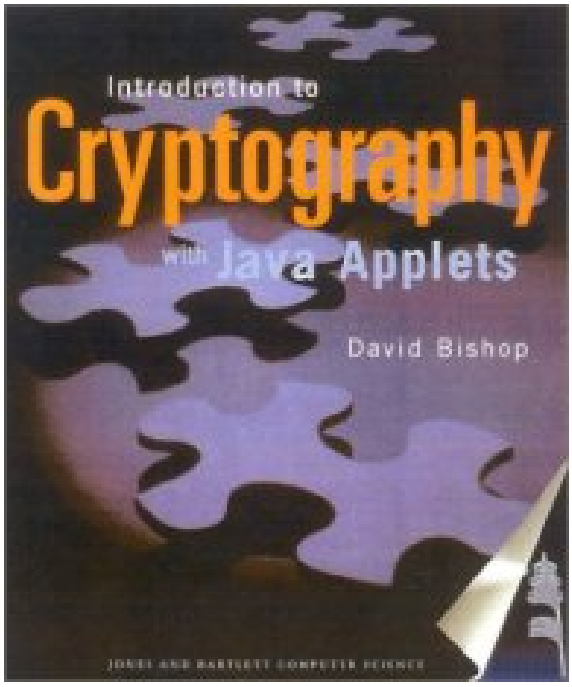


Cryptography With Java Applets
(by David Bishop)

Перевод:
http://northdemon.narod.ru/book.html

ГЛАВА 11

Проверка Чисел На Простоту

Заметим, что теперь мы используем криптосистемы, которые требуют от нас поиска и использования больших простых чисел. Как мы находим большие простые числа? Мы выбираем большое случайное нечетное целое число и пробуем разложить его на множители? Нет. Разложение на множители отбирает слишком много времени, поэтому это не хороший путь определения действительно ли целое число является простым.

По сути, это как раз то, чем занимается шифр Рабина, и другие подобные ему средства защиты. Если разложение на множители открытого (публичного) ключа n не было бы чрезвычайно трудным, любой мог бы его разложить и получить значения закрытого (секретного) ключа p и q (множители n).

Вы, возможно, предположили, что попытаться разложить число на множители - единственный способ определить, действительно ли число является простым. Это не так. Сейчас мы изложим альтернативные методы для проверки чисел на простоту.

Первым делом вспомним Малую Теорему Ферма (МТФ): Если p простое

$$p \nmid a, a^{p-1} \equiv 1 \pmod{p}.$$

(Обратите внимание, что, если

$$a^{p-1} \not\equiv 1 \pmod{p},$$

тогда p должен быть составным.) Как обстоит дело с обратным утверждением из МТФ? То есть, если

$$a^{p-1} \equiv 1 \pmod{p},$$

можем ли мы заключить, что p простое? Удивительно, но это часто достоверно. Мы можем проследить за этим, если возведем 2 в некоторые целые степени, как показано в Таблице 11.1.

Таблица 11.1

n	$2^{n-1} \equiv x \pmod{n}$	Is n prime?
3	$2^2 \equiv 1 \pmod 3$	Yes
4	$2^3 \equiv 0 \pmod 4$	No
5	$2^4 \equiv 1 \pmod 5$	Yes
6	$2^5 \equiv 2 \pmod 6$	No
7	$2^6 \equiv 1 \pmod 7$	Yes
8	$2^7 \equiv 0 \pmod 8$	No
9	$2^8 \equiv 4 \pmod 9$	No
10	$2^9 \equiv 2 \pmod{10}$	No
11	$2^{10} \equiv 1 \pmod{11}$	Yes

Кажется, что, когда n простое, мы всегда получаем значение, сравнимые с 1, когда n составное, мы получаем значение, не сравнимые с 1. Однако такое происходит не всегда. Существуют составные целые числа n, для которых

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

Возьмем составное число 341 = 11·31. Когда мы возводим 2 в степень 340, мы получаем наименьший неотрицательный вычет 1 по модулю 341. (Проверьте). Если бы обратное утверждение из МТФ было верно, мы могли бы заключить, что 341 простое; но это, очевидно не так, поэтому обратное утверждение из МТФ – не верно.

Нет ничего уникального в выборе числа 2 в качестве основания, поэтому мы могли бы просто взять другое основание. Например, мы применяем “тест Ферма” для 341, используя число 3 в качестве основания. Мы получаем

$$3^{340} \equiv 56 \pmod{341}.$$

Это немедленно устанавливает, что 341 является составным числом. Таким образом, мы

могли бы предположить, что сможем обойти неудачу теста Ферма, пробуя различные основания по модулю *n* до тех пор, пока мы либо

1. Получим наименьший неотрицательный вычет, не равный 1, и сделаем вывод, что *n* составное.

2. Не получим вычет, сравнимый с 1 по модулю *n* после большого количества попыток с различными основаниями, и заключим, что *n,* вероятно, простое.

Действительно, это не плохая идея, если бы не существование чисел Кармайкла; они - очень редкие составные целые числа, которые обманывают тест Ферма по любому основанию *b* взаимно простому с *n*. Целое число 561 = 3·11·17 – число Кармайкла, и мы можем доказать это следующим образом:

Возьмем любое основание *b* взаимно простое с 561;
так НОД (*b*, 3) = НОД (*b*, 11) = НОД (*b*, 17) = 1.
МТФ сообщает нам тогда, что $b^2 \equiv 1 \pmod 3$, $b^{10} \equiv 1 \pmod{11}$, и $b^{16} \equiv 1 \pmod{17}$. Это говорит о том, что

$$b^{560} = (b^2)^{280} \equiv 1 \pmod 3,$$

$$b^{560} = (b^{10})^{56} \equiv 1 \pmod{11},$$

$$b^{560} = (b^{16})^{35} \equiv 1 \pmod{17}.$$

Утверждение 26 теперь подразумевает, что $b^{560} \equiv 1 \pmod{561}$ для любого основания *b* такого, что НОД(*b*, 561) = 1, и таким образом, 561 является числом Кармайкла.

Хотя числа Кармайкла встречаются очень редко (гораздо реже, чем простые числа), их существует бесконечное множество. Мы не будем доказывать этого. Факт того, что они существуют уже достаточен, чтобы избегать использовать тест Ферма для проверки чисел на простоту, особенно, когда мы можем разработать более лучшие тесты, которые не смогут одурачить числа Кармайкла. Пример такого теста – тест Миллера. Он основан на тесте Ферма, но немного обходит его. Чтобы доказать, что тест Миллера работает, нам понадобится следующее Утверждение, которое Вы должны уметь легко доказать.

Утверждение 34

Пусть *p* - простое и $x^2 \equiv 1 \pmod p$. Тогда $x \equiv 1 \pmod p$ или $x \equiv -1 \pmod p$.

Утверждение 34 говорит о том, что квадратные корни из 1 по модулю простого числа равны только 1 и -1. Этот факт будет очень полезен. Теперь мы сможем обсудить тест Миллера, который основан на Утверждении 34.

11.1 ТЕСТ МИЛЛЕРА

Определение (Тест Миллера)

Пусть n положительное целое число и $n - 1 = 2^s t$, где s - неотрицательное целое число, t - нечетное положительное целое число. Мы скажем, что n проходит тест Миллера по основанию b, если или $b^t \equiv 1 \pmod{n}$ или $b^{kt} \equiv -1 \pmod{n}$ для некоторого $k = 2^j$, $0 \leq j \leq s - 1$.

Давайте подробно обсудим, как работает тест Миллера. Предположим, что Вы проверяете целое число n на простоту, и получаете $b^{n-1} \equiv 1 \pmod{n}$. Это не говорит Вам о том, простое n или составное, таким же образом рассмотрите величину $y = (n - 1)/2$, и вычислите $x \equiv b^y \pmod{n}$.

Если n - простое, мы должны получить $x \equiv 1 \pmod{n}$ или $x \equiv -1 \pmod{n}$, это следует из Малой Теоремы Ферма $x^2 = x \cdot x = (b^{(n-1)/2})(b^{(n-1)/2}) = b^{(n-1)} \equiv 1 \pmod{n}$ и из Утверждения 34.

Таким образом, при вычислении x, мы должны предвидеть следующие случаи:

1. x не сравнимо ни с 1, ни с -1 по модулю n. В этом случае, x имеет квадратный корень, который не сравним ни с 1, ни с -1; следовательно, n не может быть простым по Утверждению 34 и не проходит испытание.

2. $x \equiv -1 \pmod{n}$. Этот случай говорит о том, что n может быть простым числом. Мы не можем продолжать испытания, как только мы получаем вычет равный -1, поэтому мы заключаем, что n проходит испытание.

3. $x \equiv 1 \pmod{n}$. Это также говорит, что n может быть простым числом, и, кроме того, мы можем продолжить проверку n на простоту следующим путем:

a. Если $2 \mid y$, разделите y на 2 (снова), и вычислите $x \equiv b^y \pmod{n}$. Затем рассмотрите

полученный результат в соответствии с тремя вышеизложенными случаями.

b. Если y не делится на 2, последнее значение x было сравнимо с 1 по модулю n.
Мы не можем проверять тест далее, и заключаем, что n проходит испытание на простоту.

Заметим, что предыдущая процедура должна в конечном итоге закончиться, так как

- мы должны, в конечном счете, получить вычет, не равный 1, или
- в течение каждого повторения мы делим значение y пополам, и при некотором значении y, это деление должно остановиться.

Это должно разъяснить Вам, что, при запуске теста Миллера для простого числа он сработает.

Утверждение 35

Если n - простое число и b - положительное целое число, такое что

$$n \nmid b,$$

тогда n проходит тест Миллера по основанию b.

Доказательство.

Если в описанном выше алгоритме n - простое число, то Вы должны в конечном итоге

1. Получить значение $x \equiv -1 \pmod{n}$, или
2. Быть не в состоянии продолжить деление y на 2.

Любым из путей простое n проходит тест. Из Утверждения 34 следует, что нет смысла в получении квадратного корня из 1, который не был бы сравним ни с 1, ни с -1 по модулю n.

ПРИМЕР.

1. Возьмем простое число $n = 29$ и основание $b = 5$; пронаблюдаем, что n проходит тест Миллера. Начнем со степени $y = (n - 1)/2 = (29 - 1)/2 = 14$ и вычислим $5^{14} \equiv 1 \pmod{29}$. До тех пор, пока это выполняется; делим y на 2, чтобы получить $y = 7$, и вычислим $5^7 \equiv 28 \equiv -1 \pmod{}$

29). Этот результат также подходит, и мы не можем продолжить дальнейшую проверку, так как получили вычет -1. (Если бы мы получили вычет равный 1, мы также не смогли бы продолжить проверку, так как y не делится дальше на 2; невзирая ни на что, $n = 7$ проходит тест Миллера по основанию 5.)

2. Возьмем простое число $n = 257$ и основание $b = 22$, обратите внимание на последовательность теста Миллера в Таблице 11.2.

ТАБЛИЦА 11.2

Exponent y	$22^y \equiv ?(\text{mod } 257)$	Conclusion
128	1	Pass–continue
64	1	Pass–continue
32	–1	Pass–STOP

3. Мы повторяем вышеупомянутое испытание для $n = 257$, но воспользуемся другим основанием $b = 17$. (См. Таблицу 11.3.)

ТАБЛИЦА 11.3

Exponent y	$17^y \equiv ?(\text{mod } 257)$	Conclusion
128	1	Pass–continue
64	1	Pass–continue
32	1	Pass–continue
16	–1	Pass–STOP

4. Мы ещё раз повторяем испытание для $n = 257$, но используем основание $b = 4$. (См. Таблицу 11.4.)

ТАБЛИЦА 11.4

Exponent y	$4^y \equiv ?(\text{mod } 257)$	Conclusion
128	1	Pass–continue
64	1	Pass–continue
32	1	Pass–continue
16	1	Pass–continue
8	1	Pass–continue
4	–1	Pass–STOP

Призываем Вас проверить значения, полученные здесь. Когда целое число *n* не проходит тест Миллера, *n* определенно составное, но если оно проходит тест, мы все еще не знаем наверняка, действительно ли оно простое. Тем не менее, здесь не существует никаких составных целых чисел, которые могут одурачить тест Миллера по любому основанию *b*; даже числа Кармайкла должны, в конечном счете, не пройти тест Миллера по некоторому основанию *b*. Мы не будем доказывать это, сформулируем лишь утверждение к этому результату.

Утверждение 36

Предположим, что *n* – нечетное, составное положительное целое число. Тогда *n* не проходит тест Миллера, по крайней мере, в 75 процентах испытаний по основанию *b*, где $1 \leq b \leq n - 1$.

Например, возьмем число Кармайкла 561; оно проходит тест Фермата, но не проходит тест Миллера по основанию 2. (См. Таблицу 11.5.)

ТАБЛИЦА 11.5

Test	Status
$2^{560} \equiv 1 \pmod{561}$	Passes Fermat's test
$2^{280} \equiv 1 \pmod{561}$	Pass
$2^{140} \equiv 67 \pmod{561}$	FAIL–STOP

Неудача теста Миллера точно устанавливает, что число 561 является составным.

Обратите внимание, что Утверждение 36 говорит о том, что для теста Миллера не может существовать ничего похожего на числа Кармайкла. Фактически, Утверждение 36 позволяет нам устанавливать "вероятность" того, что целое число является простым. Предположим, к примеру, что мы берем очень большое целое число *n*, и оно проходит тест Миллера, по некоторому основанию *b*, лежащему между 1 и *n* - 1. Так как *n* может пойти тест Миллера в 75 процентах от всех оснований, то есть, вероятность того, что *n* является составным составляет 25 процентов, это эквивалентно тому, что вероятность получить простое *n* не меньше, чем 75 процентов.

11.2 ТЕСТ РАБИНА - МИЛЛЕРА

Если далее мы будем повторять тест Миллера для числа *n* с различными основаниями, мы сможем или обнаруживать, что *n* составное, или приблизить к 1 вероятность того, что это *n* является простым. Это фактически то, что современные компьютеры делают при поиске больших простых чисел, этот специфический метод обнаружения "вероятно" простых чисел, называется *тестом Рабина - Миллера*.

Кратко: если целое число *n* проходит тест Рабина – Миллера для *q* различных оснований, то вероятность того, что *n* является простым не меньше, чем 1 - ( )^q. Важно обратить внимание, что основания, используемые для теста Рабина – Миллера должны быть выбраны настолько случайно, насколько это возможно.

Java Алгоритм

Java класс BigInteger предусматривает конструкторы для генерации, вероятно, простых чисел. Используемый в них тест на простоту отличен от теста Рабина-Миллера; их вариант после *q* проходов теста устанавливает, что *n* простое с вероятностью 1 - ( )^q. (Наиболее вероятно, что они пользуются тестом на простоту Соловея-Штрассена, в котором используется так называемый символ Якоби; у нас нет никаких оснований для его изучения, так как тест

Рабина-Миллера в отличие от теста Соловея-Штрассена обеспечивает более высокую вероятность за меньшее количество испытаний.)

Мы напишем метод под названием *primeProbability()*, который будет производить тест Миллера-Рабина для целого числа *n*. Этот метод будет возвращать вероятность того, что *n* является простым, для точно установленного числа проходов. Если *n* не пройдет тест при каком-нибудь прогоне, тогда метод *primeProbability()* вернет *0*.

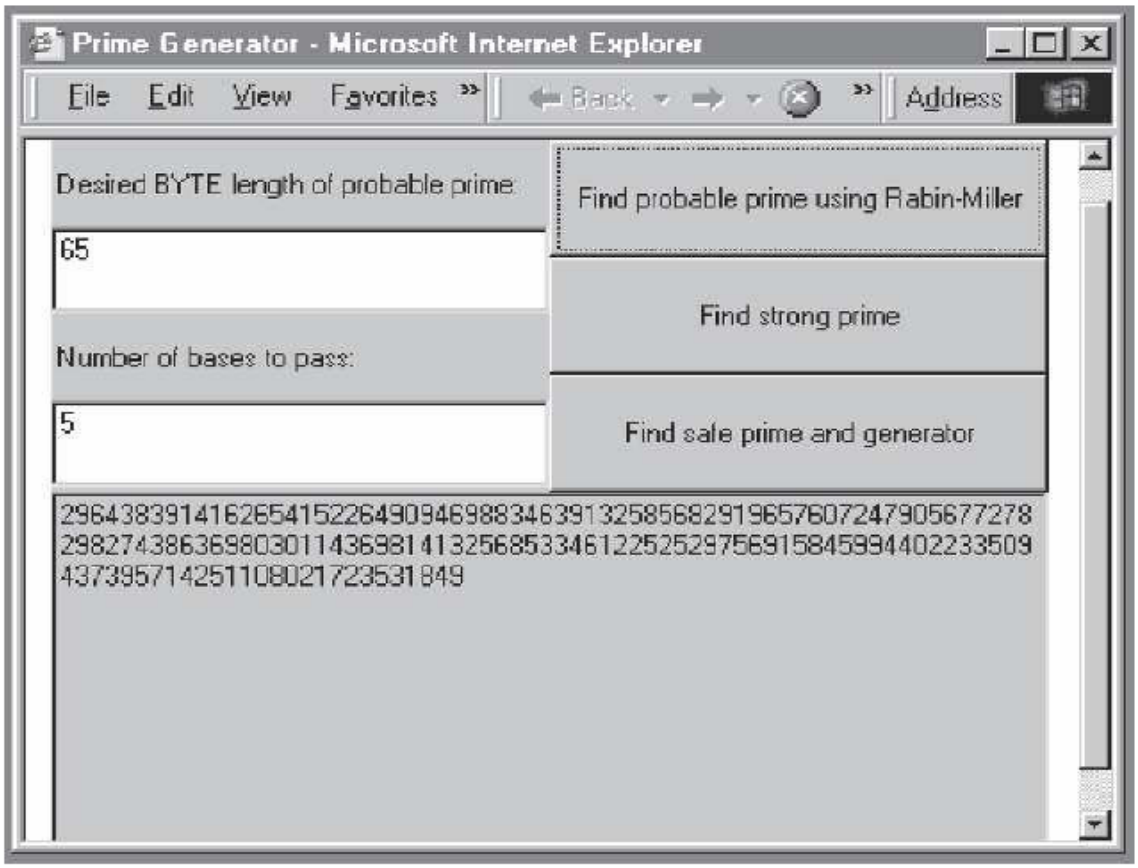
```
import java.math.BigInteger;
import java.security.SecureRandom;
public class BigIntegerMath {
//. . .
static final BigInteger TWO = new BigInteger(929);
//. . .
// Выполнение теста Рабина-Миллера.
// В качестве входного параметра - количество различных оснований,
// по которым проверяем тест.
// Если BigInteger проходит все испытания, возвращаем вероятность,
// того, что число является простым.
// Возвращаем нуль, если BigInteger составное.

public static double primeProbability (BigInteger n, int numPasses) {
      if (numPasses <= 0)
      throw new IllegalArgumentException("Number of bases must be positive!");
      BigInteger b, x;
      SecureRandom sr = new SecureRandom();
      BigInteger nMinusOne = n.subtract(ONE);
      for (int i=0; i < numPasses; i++) {

            //Выбираем случайное основание, меньшее, чем n
            b = new BigInteger(n.bitLength()-1, sr);

            //Первым делом проверяем условие Ферма
            x = b.modPow(nMinusOne,n);
            if (!x.equals(ONE)) return 0.0;//не простое

            //Делим n-1 на 2
            BigInteger[] dr = nMinusOne.divideAndRemainder(TWO);

            //Выполняем проверку корня
            while (dr[1].equals(ZERO)) {
                  x = b.modPow(dr[0], n);

                  //если получили -1, проходит; выходим
                  if (x.equals(nMinusOne)) break; //проходит

                  //Теперь, если не равно -1 или 1, не прошел тест, возвращаем 0
                  if (!x.equals(ONE)) return 0.0; //не простое

                  // Если 1, мы можем продолжить испытания, разделив на 2
                  dr = dr[0].divideAndRemainder(TWO);
            }
      }

      //Единственный способ добраться досюда - пройти все тесты
      return 1.0-Math.pow(0.25,numPasses);
      }
}
```

Апплет под названием *TestPrimeGeneratorApplet* генерирует, вероятно, простые числа, указанной длины в байтах.

Он генерирует случайные числа требуемой длины, и проверяет каждое из них на простоту до первого пропуска, пользуясь числом указанных испытаний.

Снимок экрана, показанный на Рисунке 11.1 содержит апплет, генерирующий, вероятно, простые числа с использованием теста Рабина - Миллера.

Рисунок 11.1



УПРАЖНЕНИЯ

1. Используя тест проверки на простоту Миллера-Рабина, определите вероятность того, что целое число является простым, для каждого из следующих чисел. Выберите ваши собственные основания (так случайно, как Вы сможете). Вы должны уметь делать вычисления без помощи компьютера.

Integer to test	Number of random bases to use
19	3
101	5
103	4
97	3

2. Повторить предыдущее упражнение, используя эти большие числа. Можно воспользоваться программой.

Integer to test	Number of random bases to use
1186913492875024326501274951491498649	10
523987657257476543761243338647825275 1	15
884388835127254389485860540719510049	20
930940866280690607285036868096531589	30
899476440042092355083033089040210287	33

3.Проверьте следующее число на простоту; используйте 5 различных случайных оснований:

1358298529049385849277351428359266778603493846931744549748519669727813092754241848720539208320756059229857826295384738347503872554323492997115554834280062872188576349940639033178286414416468073076683716052622317651279843577212995655335528603220308038077575973232019898509488400406911612308414787543718365846746514894879055274416756 7.