Testing for Prime Numbers

Introduction

To implement the RSA cryptosystem, we need to produce a pair of large prime numbers. We shall describe one method for doing this called the ***Solovay-Strassen Algorithm***. To be absolutely certain that a given number is prime may take a considerable amount of time due to all the checking that is involved. Our practical need is for a fast method, so in order to gain speed we trade off with certainty. That is, we use a method which quickly determines that a number is prime, *with a high probability*, rather than absolute certainty. In order to describe the method, we need to examine some concepts from Number Theory.

Quadratic Residues mod p

Let p be an odd prime number. An integer x, with $1 \le x \le$ p-1 is defined to be a ***quadratic residue*** modulo p if the congruence $y^2 \equiv x \bmod p$ has a solution. The other non-zero x's are called ***quadratic non-residues***.

For example, with p = 13, the quadratic residues are **1** = $(\pm 1)^2$, **4** = $(\pm 2)^2$, **9** =$(\pm 3)^2$, **3** = $(\pm 4)^2$, **12** = $(\pm 5)^2$ and **10** = $(\pm 6)^2$, while the quadratic non-residues are: 2, 5, 6, 7, 8, and 11.

It is true that for any odd prime p, as in the above example, half of the non-zero elements are quadratic residues and the other half are quadratic non-residues.

Legendre Symbol

For any odd prime p and any integer a ≥ 0, we define the ***Legendre symbol*** as follows:

$$\left(\frac{a}{p}\right) = \begin{cases} +1 & \text{if a mod p is a quadratic residue} \\ 0 & \text{if p divides a} \\ -1 & \text{if a mod p is a quadratic non-residue} \end{cases}$$

It follows from a result of Euler that

$$\left(\frac{a}{p}\right) \equiv a^{\frac{p-1}{2}} \bmod p$$

This means that determining whether or not a number is a quadratic residue can be done computationally. Thus, 2 is a quadratic residue modulo 17, since $2^{(17-1)/2} = 2^8 = 256 \equiv 1 \bmod 17$. And indeed, $2 \equiv 6^2 \bmod 17$. While 5 is a quadratic non-residue mod 7, since $5^{(7-1)/2} = 5^3 = 125 \equiv 6 \equiv -1 \bmod 7$.

Jacobi Symbol

The Legendre symbol can be generalized to the ***Jacobi symbol***. Let n be any positive odd integer and a ≥ 0 any integer.

If the prime decomposition of n is

$$n = p_1^{e_1} p_2^{e_2} p_3^{e_3} \cdots p_k^{e_k},$$

then the Jacobi symbol is defined by:

$$\left(\frac{a}{n}\right) := \prod_{i=1}^{k} \left(\frac{a}{p_i}\right)^{e_i}.$$

Note that the Jacobi symbol, since it is the product of Legendre symbols, can only have the values of 0, +1 or -1.

Properties of the Jacobi symbol

There are several properties of the Jacobi symbol that make its computation fairly easy and, most importantly, do not require that n be factored.

1. If $m_1 \equiv m_2 \bmod n$ then, $\left(\frac{m_1}{n}\right) = \left(\frac{m_2}{n}\right)$.

2. The Jacobi symbol is multiplicative, i.e., $\left(\frac{m_1 m_2}{n}\right) = \left(\frac{m_1}{n}\right)\left(\frac{m_2}{n}\right)$.

In particular, if $m = 2^k t$, where t is odd, then $\left(\frac{m}{n}\right) = \left(\frac{2}{n}\right)^k \left(\frac{t}{n}\right)$.

3. We have, $\left(\frac{2}{n}\right) = \begin{cases} +1 \text{ if } n \equiv \pm 1 \bmod 8 \\ -1 \text{ if } n \equiv \pm 3 \bmod 8 \end{cases}$.

4. Suppose m and n are odd integers. Then $\left(\frac{m}{n}\right) = \left(\frac{n}{m}\right)$,

unless $n \equiv m \equiv 3 \bmod 4$, in which case $\left(\frac{m}{n}\right) = -\left(\frac{n}{m}\right)$.

Example

$$\left(\frac{610}{987}\right)=\left(\frac{2}{987}\right)\left(\frac{305}{987}\right) \text{ by 2}$$

$$=-\left(\frac{305}{987}\right) \text{ by 3}$$

$$=-\left(\frac{987}{305}\right) \text{ by 4}$$

$$=-\left(\frac{72}{305}\right) \text{ by 1}$$

$$=-\left(\frac{2}{305}\right)^3\left(\frac{9}{305}\right) \text{ by 2}$$

$$=-\left(\frac{9}{305}\right) \text{ by 3}$$

$$=-\left(\frac{305}{9}\right) \text{ by 4}$$

$$=-\left(\frac{8}{9}\right) \text{ by 1}$$

$$=-\left(\frac{2}{9}\right)^3 \text{ by 2}$$

$$=-1 \text{ by 3.}$$

Euler Pseudo-primes

Note that if n is a prime number then it follows that for all a

(1) $$\left(\frac{a}{n}\right) = a^{\frac{n-1}{2}} \bmod n .$$

So, if there exists an a for which $$\left(\frac{a}{n}\right) \neq a^{\frac{n-1}{2}} \bmod n$$

then n is definitely not a prime (i.e., n is a ***composite number***). However, there are composite numbers n so that (1) is satisfied for some a. These numbers are called ***Euler pseudo-primes*** with base a. It can be shown however, that for any given composite number n, there are at most n/2 values of a less than n for which n is an Euler pseudo-prime with base a. This is the basis of the Solovay-Strassen algorithm.

The Solovay-Strassen Algorithm

The algorithm is as follows:

1. Let n be the number that is being tested for primality.
2. Randomly choose an integer a, with $1 \leq a \leq n$.
3. If

$$\left(\frac{a}{n}\right) \neq a^{\frac{n-1}{2}} \bmod n$$

then stop and report that n is composite.

4. Otherwise, repeat steps 2 and 3, k times (where k is a preselected integer).

If the algorithm does not report that n is composite, then the probability that n is an Euler pseudo-prime with respect to the k choices of a is $1/2^k$. If k is large enough, this is a very small probability. Put another way, for large enough k, there is a very high probability that n is a prime.

The Solovay-Strassen Algorithm

So, to use this algorithm to find primes, you randomly select odd numbers in the size range you are interested in. Run them through the algorithm with k set at say 100. If the algorithm reports that the number is composite, then choose another random selection, until the algorithm reports that your choice is prime. The prime number theorem from number theory, tells you that the probability of selecting a prime at random is about 2/ln n (selecting only odd numbers), so, for an n in the 512-bit range this is about 2/177. Thus, on average you would expect to select a prime of this size randomly about once every 90 tries.