The Solovay-Strassen Primality Test

12 October, 1993
Burt Rosenberg
Revised: 6 October, 2000

1 Introduction

We describe the Solovay-Strassen primality test. There is quite a bit of number-theoretic background necessary to the full understanding of the algorithm, however, in practice it is very simple. It is also curious because it works incredibly quickly to give you a probably correct answer, however no one has found a less than exponential-time algorithm to tell you *for certain* whether a number is prime.

The algorithm works by selecting random integers and computing large powers of them in the ring $\mathbf{Z}/n\mathbf{Z}$, where n is the number you want to test. Also, the so called Jacobi symbol is calculated for these integers. If ever these calculations disagree, then n is composite. For if n were prime, the Jacobi symbol would in fact be the Legendre symbol, and for the Legendre symbol equality of the two methods of calculation is a theorem.

It is not necessary for the two calculations to disagree when n is composite, but it is likely. Half of the integers between 1 and $n-1$ which are relatively prime to n will make the calculations disagree. (If we happen to choose an integer which is not relatively prime to n, we are even better off: we not only know n is composite, we have a non-trival factor!) Hence, the probablity that after k choices of a random integer you would wrongly proclaim a composite to be prime is less than $1/2^k$. In practice, the results are even better.

We begin by explaining the Legendre Symbol, then extend the definition to the Jacobi Symbol. From there we apply the Jacobi Symbol to primality testing. Finally, a Pascal program is presented.

2 Quadratic Residues

Suppose p is an odd prime. In $(\mathbf{Z}/p\mathbf{Z})^\times$, the group of invertible elements mod p, that is to say, $\mathbf{Z}/p\mathbf{Z}$ without 0, half of the integers are squares and the rest are not. This is quickly seen by considering the map $x \mapsto x^2$. Each element a in the range of this map receives exactly two elements, namely, if $b^2 = a$ then $(-b)^2 = a$, that is, if we can assume that $b \neq -b$, which is equivalent to assuming $p \neq 2$. The elements which are squares are called quadratic residues, the rest are quadratic non-residues.

Definition 1 (Legendre Symbol) *For p a prime, and b a positive integer, The Legendre Symbol is defined by,*

$$\left[\frac{b}{p}\right] = \begin{cases} 0 & \textit{if } b \textit{ and } p \textit{ are not relatively prime,} \\ 1 & \textit{if } b \textit{ is a quadratic residue mod } p, \\ -1 & \textit{if } b \textit{ is a quadratic non-residue mod } p \end{cases}$$

If p is two, then the value of the Legendre Symbol is one for any odd b and 0 else. For p an odd prime, we can use this theorem:

Theorem 1 *For any odd prime p and any positive integer b,*

$$\left[\frac{b}{p}\right] = b^{(p-1)/2} \pmod{p}.$$

PROOF: If $(b,p) \neq 1$ then $b = 0 \pmod p$ and the equality follows. We henceforth consider integers b relatively prime to p.

It is quickly seen that the set,

$$A = \left\{ a \in (\mathbf{Z}/p\mathbf{Z})^\times \,\middle|\, a^{(p-1)/2} = 1 \pmod p \right\},$$

form a subgroup of $(\mathbf{Z}/p\mathbf{Z})^\times$. Any quadratic residue $b = a^2$ is in A, since,

$$b^{(p-1)/2} = a^{2(p-1)/2} = 1 \pmod p,$$

by Little Fermat. We remarked above that half the elements of $(\mathbf{Z}/p\mathbf{Z})^\times$ are quadratic residues, hence A is of size either $(p-1)$ and $(p-1)/2$ (the order of a subgroup must divide the order of the group). The subgroup A does not include any generator of the group, hence its size is not $(p-1)$. Therefore A contains exactly the quadratic residues.

On the other hand, by Little Fermat, the square of $b^{(p-1)/2}$ is 1, hence for those integers relatively prime to p but outside of A, the power must evaluate to the only other root of one, that is, -1. □

There are several rules for computing with Legendre symbols, for instance: that the Legendre symbol depends only on the residue of b mod p, that the Legendre symbol of 1 over p is always 1, and that,

$$\left[\frac{b_1 b_2}{p}\right] = \left[\frac{b_1}{p}\right]\left[\frac{b_2}{p}\right].$$

These are easy to verify. Two rules which are more difficult to show are:

Theorem 2 *For p an odd prime,*

$$\left[\frac{2}{p}\right] = (-1)^{(p^2-1)/8} = \begin{cases} 1 & \textit{if } p = \pm 1 \pmod 8 \\ -1 & \textit{if } p = \pm 3 \pmod 8 \end{cases}$$

Theorem 3 (Law of Quadratic Reciprocity) *For p and q odd primes,*

$$\left[\frac{p}{q}\right]\left[\frac{q}{p}\right] = (-1)^{(p-1)(q-1)/4} = \begin{cases} -1 & \textit{if } p = q = 3 \pmod 4 \\ 1 & \textit{else.} \end{cases}$$

For proofs see Neal Koblitz's *A Course in Number Theory and Cryptography*, or for an enchantingly elementary proof, André Weil's *Number Theory for Beginners.*

3 Jacobi Symbols

The Jacobi Symbol extends the definition of the Legendre Symbol to "denominators" other than primes. In doing so, it loses the number-theoretic interpretation, it no longer indicates which integers are quadratic residues, and it is no longer possible to calculate it by taking the numerator to a certain power mod the denominator.

Definition 2 (Jacobi Symbol) *For any positive inger n, we define the Jacobi symbol according to the prime decomposition of n by,*

$$\textit{if } n = \prod_{i=1}^{r} {p_i}^{\alpha_i}, \ \textit{then} \ \left(\frac{m}{n}\right) = \prod_{i=1}^{r} \left[\frac{m}{p_i}\right]^{\alpha_i}.$$

Note immediately that the Jacobi symbol has values either 1, -1 or 0, and it is zero only if one of its factors is zero, that is, m and n are not relatively prime. Since the Euclidean algorithm efficiently determines if two integers are relatively prime, from the point of view of calculating the Jacobi symbol, attention focuses on the case of m and n relatively prime.

Theorem 4 *For integers n and m, and factorizations of $n = n_1 n_2$ and $m = m_1 m_2$, the Jacobi symbols obeys:*

$$\left(\frac{m_1 m_2}{n}\right) = \left(\frac{m_1}{n}\right)\left(\frac{m_2}{n}\right),$$

and

$$\left(\frac{m}{n_1 n_2}\right) = \left(\frac{m}{n_1}\right)\ \left(\frac{m}{n_2}\right),$$

Proof: For the first equivalence, write out the Jacobi symbol as a product of Legendre symbols, apply the rule,

$$\left[\frac{n_1 n_2}{m}\right] = \left[\frac{n_1}{m}\right]\left[\frac{n_2}{m}\right],$$

for Legendre symbols, then rearrange and collect terms. For the second equivalence, write out the Jacobi symbol according to its definition, then collect terms. □

Theorem 5 *For n and m relatively prime and odd,*

$$\left(\frac{n}{m}\right)\left(\frac{m}{n}\right) = (-1)^{(m-1)(n-1)/4}.$$

Proof: Let i be the number of prime factors in n and j the number of prime factors in m, counting multiplicity. We use induction on i and j. When $i = j = 1$, the basis case, the theorem is exactly the law of quadratic reciprocity. Assume now that the theorem is true for any $i < I$ and $j < J$, with I and J greater than one. We show it is true for any $i < I + 1$ and $j < J + 1$.

Let n be an integer with $I + 1$ factors. Write it as the product of two integers $n = n_1 n_2$ each having less than I factors, and apply the induction hypothesis to the factored Jacobi symbols:

$$\begin{aligned}
\left(\frac{n_1 n_2}{m}\right)\left(\frac{m}{n_1 n_2}\right) &= \left(\frac{n_1}{m}\right)\left(\frac{n_2}{m}\right)\left(\frac{m}{n_1}\right)\left(\frac{m}{n_2}\right) \\
&= (-1)^{(n_1-1)(m-1)/4}(-1)^{(n_2-1)(m-1)/4} \\
&= (-1)^{(n_1+n_2-2)(m-1)/4}.
\end{aligned}$$

Hence our product is -1 if both m and $n_1 + n - 1$ are 3 mod 4, and 1 else. Note that, since n_1 and n_2 are both odd, $(n_1 - 1)(n_2 - 1)$ is divisible by 4. Multipling this out, we get $n = n_1 + n_2 - 1$ mod 4. So the product is -1 if n and m are 3 mod 4, and 1 else. That is,

$$(-1)^{(n_1+n_2-2)(m-1)/4} = (-1)^{(n-1)(m-1)/4},$$

which proves the theorem for $i = I$ and and any $j < J$.

Swapping the role of n and m gives that the theorem is true for $i = I$ and $j = J$. Continuing by induction, the theorem is true for all i and j. □

In calculating the Jacobi symbol we use the previous theorem and reduction of the numerator modulo the denominator to reduce step by step the calculation, except if the numerator or denominator is even. If both are even, then the result is zero. We can throw out all factors of two of the denominator, since any odd is a quadratic residue mod 2. If the numerator is even, its factors of two are treated using the following theorem.

Theorem 6 *For any odd integer n,*

$$\left(\frac{2}{n}\right) = (-1)^{(n^2-1)/8}.$$

PROOF: Similar to the previous theorem, we use an induction on the number of elements in the prime decomposition of n. The basis case is n is a prime, when the Jacobi symbol equals the Legendre symbol and the theorem is true by definition.

Suppose the theorem is true for all n which are the product of less than I primes. Write n, a product of I primes, as $n = n_1n_2$ and calculate, using the induction hypothesis:

$$\begin{aligned}\left(\frac{2}{n_1n_2}\right) &= \left(\frac{2}{n_1}\right)\left(\frac{2}{n_2}\right) \\ &= (-1)^{({n_1}^2+{n_2}^2-2)/8}\end{aligned}$$

Note that an odd integer is either 1 or 3 mod 4, so any square of an odd integer is 1 mod 4. Therefore $({n_1}^2-1)({n_2}^2-1)$ is zero mod 16. Multipling this out we find,

$$n^2 = {n_1}^2 + {n_2}^2 - 1 \pmod{16}.$$

Recall that $m^2 - 1$ is divisible by 8 for any odd m, so $n^2 - 1$ and ${n_1}^2 + {n_2}^2 - 2$ are equal and both either 0 or 8 mod 16. In any case,

$$(-1)^{({n_1}^2+{n_2}^2-2)/8} = (-1)^{(n^2-1)/8},$$

proving the theorem for any integer a product of I primes.

Proceeding the induction, we have the theorem for all odd n. □

4 Application to Primality Testing

The Jacobi symbol is used to test for primality of a given integer n by testing for agreement between two calculations,

$$\left(\frac{b}{n}\right) = (?)\ b^{(n-1)/2} \pmod{n},$$

which, if n is a prime, is an identity for the Legendre symbol. If n is not a prime, however, either b will not be relatively prime to n, or the two calculations might not agree. How often they do not agree is discussed next.

Theorem 7 *For any prime p, there is a generator for $(\mathbf{Z}/p^2\mathbf{Z})^\times$.*

PROOF: For any prime p, there is a generator g for $(\mathbf{Z}/p\mathbf{Z})^\times$. Let h equal g or $g(1+p)$, depending on whether or not

$$g^{p-1} = ?\ 1 \pmod{p^2}.$$

If $g^{p-1} = 1 \pmod{p^2}$, then

$$(g(1+p))^{p-1} = 1 + (p-1)p + p^2 w = 1 + p(p-1) \pmod{p^2},$$

where w is some integer. Hence h can be chosen so that its $p-1$ power mod p^2 is not one. Since in either case $h = g \pmod{p}$, h is a generator of $(\mathbf{Z}/p\mathbf{Z})^\times$,

We show that h is a generator of $(\mathbf{Z}/p^2\mathbf{Z})^\times$. Let $h^j = 1 \pmod{p^2}$. This this congruence remains true modulo p, therefore $(p-1)\,|\,j$. This being so, we can write j as $j = (p-1)j'$. But j must also divide the order of the group, $j\,|\,p(p-1)$, hence $j'\,|\,p$. Since h^{p-1} is not one, j' cannot be one, so it must be p. Therefore the order of h is the size of the group. □

Theorem 8 *For any odd composite n, there is a b relatively prime to n such that,*

$$\left(\frac{b}{n}\right) \neq b^{(n-1)/2} \pmod{n}.$$

PROOF: If $p^2\,|\,n$, for a prime p, let g generate $(\mathbf{Z}/p^2\mathbf{Z})^\times$. Select a b such that $b = g \pmod{p^2}$ and $b = 1 \pmod{q}$ for any other distinct prime q dividing n. The existence of b is assured by the Chinese Remainder Theorem. If the equation were true, then

$$b^{n-1} = 1 \pmod{n},$$

which being a congruence remaining true in $\mathbf{Z}/p^2\mathbf{Z}$, would imply that $p(p-1)\,|\,(n-1)$. However, then p would divide both n and $n-1$.

So we can suppose n is square free. Let g be a quadratic non-residue in some $\mathbf{Z}/p\mathbf{Z}$ where $p\,|\,n$. Select a b, again by the Chinese Remainder theorem, such that $b = g \pmod{p}$ and $b = 1 \pmod{q}$ for any other prime q dividing n. Then it is impossible for $b^{(n-1)/2} = -1 \pmod{n}$, else this would be true $\mathbf{Z}/q\mathbf{Z}$ for those primes where b is one. However, the rules of calculation for the Jacobi symbol give:

$$\left(\frac{b}{n}\right) = \left(\frac{b}{p}\right)\left(\prod_{q\,|\,n}\left(\frac{b}{q}\right)\right) = -1.$$

We used that n is odd in two places, that there are quadratic non-residues and that $-1 \neq 1 \pmod{q}$. □

Theorem 9 *If n is an odd composite, then for at least half of the integers b relatively prime to n in the interval $[1, n-1]$,*

$$\left(\frac{b}{n}\right) \neq b^{(n-1)/2} \pmod{n}.$$

Proof: Let A be the set of a for which $(a, n) = 1$ and the equality holds. Since there is a b relatively prime to n for which the equality does not hold, take any $a \in A$ and consider ab. It is again relatively prime to n and,

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

because we are inside the group $(\mathbf{Z}/n\mathbf{Z})^{\times}$. Hence we have that all of bA does not satisfy the equality, and hence A cannot account for more than half the elements in $[1, n-1]$. □

5 Program

```
program SolovayStrassen (input,output) ;

function gcd( a, b : integer ) : integer ;
var t : integer ;
begin
  if a < 0 then a := - a ;
  if b < 0 then b := - b ;
  if a < b then begin
    t := a ;
    a := b ;
    b := t ;
  end ;
  while b<>0 do begin
    t := a mod b ;
    a := b ;
    b := t ;
  end ;
  gcd := a
end ;

function twoFactor( var a : integer ) : integer ;
var i : integer ;
begin
  i := 0 ;
  while ((a mod 2) = 0 ) do begin
    i := i + 1 ;
    a := a div 2 ;
  end ;
  twoFactor := i ;
end ;
```

```
function jacobi( m, n : integer ) : integer ;
{ assume (m,n)=1, n is odd, 0 < m < n . }
var i, j, d : integer ;
begin

  i := 1 ;
  while (m>1) do begin
  { it could be zero or 1 to exit}

    j := twoFactor( m ) ;
    if ( j mod 2 ) = 1 then begin
      d := n mod 8 ;
      if ( d = 3 ) or ( d = 5 ) then
        i := - i ;
    end ;

    if ( (m mod 4) = 3 ) AND ( (n mod 4 ) = 3 ) then
      i := - i ;

    d := n mod m ;
    n := m ;
    m := d ;

  end ;
  jacobi := i ;

end ;

function multiply( a, b, c : integer ) : integer ;
{ return a * b mod c, without overflow by repeated
  doubling }
{ assume 0 <= a, b < c }

var i : integer ;
begin
  if (a=0) then i := 0
  else begin
    if (a mod 2) = 1 then begin
      i := multiply( (a-1) div 2, b, c ) ;
      if ( (c - i ) > i ) then i := i + i
      else i := ( i - c ) + i ;
      if ( (c - i ) > b ) then i := i + b
      else i := ( i - c ) + b ;
    end else begin
```

```
      i :=  multiply( a div 2, b , c ) ;
      if ( (c - i ) > i ) then i := i + i
      else i := ( i - c ) + i ;
    end
  end ;
  multiply := i ;
end ;

function fastExp( b, j, n : integer ) : integer ;
{ take b to the j mod n }
var i : integer32 ;
begin
  if (j=0) then i := 1
  else if ( j mod 2 ) = 1 then begin
    i := fastExp( b, (j-1) div 2, n ) ;
    i := multiply( multiply( b, i, n ), i, n ) ;
  end else begin
    i := fastExp( b, j div 2 , n ) ;
    i := multiply( i, i, n ) ;
  end ;
  fastExp := i
end ;

function primality( p, i : integer ) : integer ;
{ given an integer p, test for primality,
  using i iterations of some numbers, here they
  are the number 2, 3, ..., i+1.

  Returns 0 if no contradiction between the jacobi
  and legendre symbols was found.
  Else returns the evidence that p is composite,
  either a factor or
  an integer such that the sumbols differ. }

var j : integer ;
    dl, dj : integer ;
    b : boolean ;

begin
  if (p<2) then j := 1 { not prime }
  else if (p=2) then j := 0 { prime }
  else if (p mod 2)=0 then j := 2 { not prime }
  else begin
    {Precondition: p is odd, 3 or larger}
    {Check if i is unnecessarily large, and correct}
    if (i+1)>=p then i := p-2 ;
    j := 1 ;
    b := false ;
    repeat
      j := j + 1 ;
      if (j > (i+1) ) then b := true
      else begin
        { test for non-trivial gcd }
        if gcd(j,p)>1 then b := true
        else { apply tests } begin
          dl := fastExp( j, (p-1) div 2, p ) ;
          if (dl<>1) then dl := dl - p ;
          dj := jacobi( j, p ) ;
          if ( dj<>dl) then b := true ;
        end ;
      end ;
    until b ;
    if (j > (i+1)) then {test ran to completion} j := 0 ;
  end ;
  primality := j ;
end ;

var
 i,j,k : integer ;
 c : char ;
begin
  write(’Quit [y/n]? ’) ; readln(c) ;
  while (c<>’y’) do begin

    writeln(’Primality test of p, i iterations,’) ;
    write(’p? ’) ; readln(j) ;
    write(’i? ’) ; readln(i) ;
    k := primality( j, i ) ;
    if ( k = 0 ) then writeln(j:0,’ might be a prime.’)
    else writeln(j:0,’ is not a prime, fails test using ’,k:0) ;

    write(’Quit [y/n]? ’) ; readln(c) ;
  end ;
end.
```