## Primality Testing

A great deal of modern cryptography is based on the fact that factoring is apparently hard. At least nobody has published a fast way to factor yet. Of course, certain numbers are easy to factor – numbers with small prime factors, for example. So often, for cryptographic purposes, we may want to generate very large prime numbers and multiply them together. How can we find large prime numbers?

We are fortunate to find that prime numbers are pretty dense. (That is, there are an awful lot of them.) Let  $\pi(x)$  be the number of primes less than or equal to x. Then

$$\pi(x)\approx\frac{x}{\ln x},$$

or more exactly

$$\lim_{x\to\infty}\frac{\pi(x)}{\frac{x}{\ln x}}=1.$$

This means that on average about one out of every  $\ln x$  numbers is prime, if we are looking for primes about the size of x. So if we want to find prime numbers of say 250 digits, we would have to check about  $\ln 10^{250} \approx 576$  numbers on average before finding a prime. (We can search smarter, too, throwing out multiples of 2, 3, 5, etc. in order to check fewer numbers.) Hence, all we need is a good method for testing if a number is prime. With such a test, we can generate large primes easily – just keep generating random large numbers, and test them for primality until we find a suitable prime number.

How can we test if a number *n* is prime? The pedantic way is to try dividing *n* by all smaller numbers. Alternatively, we can try to divide *n* by all primes up to  $\sqrt{n}$ . Of course, both of these approaches are quite slow; when *n* is about  $10^{250}$ , the value of  $\sqrt{n}$  is still huge! The point is that  $10^{250}$  has only 250 (or more generally  $O(\log n)$ ) digits, so we would like the running time of the algorithm to be based on the size 250, not  $10^{250}$ !

How can we quickly test if a number is prime? Let us start by looking at some ways that work pretty well, but have a few problems. We will use the following result from number theory:

**Theorem 1** If *p* is a prime and  $1 \le a < p$  then

$$a^{p-1} = 1 \mod p$$
.

*Proof.* There are two nice proofs for this fact. One uses a simple induction to prove the equivalent statement that  $a^p = a \mod p$ . This is clearly true when a = 1. Now

$$(a+1)^p = \sum_{i=0}^p \binom{p}{i} a^{p-1}.$$

The coefficient  $\binom{p}{i}$  is divisible by p unless i = 0 or i = p. Hence,

$$(a+1)^p = a^p + 1 \mod p = a+1 \mod p$$
,

where the last step follows from the induction hypothesis.

An alternative proof uses the following idea. Consider the numbers 1, 2, ..., p-1. Multiply them all by a, so now we have a, 2a, ..., (p-1)a. Each of these number is distinct mod p, and there are p-1 such numbers, so in fact the sequence a, 2a, ..., (p-1)a is the same as the sequence 1, 2, ..., p-1 when considered modulo p, except for the order. Hence

$$1 \times 2 \times \dots \times (p-1) = a \times 2a \times \dots \times (p-1)a \mod p$$
  
=  $a^{p-1} \times 1 \times 2 \times \dots \times (p-1) \mod p$ .

Thus we have  $a^{p-1} = 1 \mod p$ .

This immediately suggests one way to check if a number n is prime. Compute  $2^{n-1} \mod n$ . If it is not 1, then n is certainly not prime! Note that we can compute  $2^{n-1} \mod n$  quite efficiently, using fast exponentiation, which requires only  $O(\log n)$  multiplications! Thus this test is efficient.

But so far this test is just one-way; if *n* is composite, we may have that  $2^{n-1} = 1 \mod n$ , so we cannot assume that *n* is prime just because it passes the test. For example,  $2^{340} = 1 \mod 341$ , and 341 is not prime. Such a number is called a 2-pseudoprime, and unfortunately there are infinitely many of them. (Of course, even though there are infinitely many 2-pseudoprimes, they are not as dense as the primes – that is, there are relatively very few of them. So if we generate a large number *n* randomly, and see if  $2^{n-1} = 1 \mod n$ , we will most likely be right if we then say *n* is prime if it passes this test. In practice, this might be good enough! This is not a good primality test, in general.)

You might think to try a different base, other than 2. For example, you might choose 3, or a random value of *a*. Unfortunately, there are infinitely

many 3-pseudoprimes. In fact, there are infinitely many composite numbers n such that  $a^{n-1} = 1 \mod n$  for all a that do not share a factor with n. (That is, for all a such that the greatest common divisor of a and n is 1.) Such n are called Carmichael numbers – the smallest such number is 561. So a test based on this approach is destined to fail for some numbers.

There is a way around this problem, due to Rabin (also called the Miller-Rabin primality test, but the randomized version that we see here is an extension by Rabin to Miller's deterministic test). Let  $n - 1 = 2^t u$ . Suppose we choose a random base a and compute  $a^{n-1}$  by first computing  $a^u$  and then repeatedly squaring. Along the way, we will check to see for the values  $a^u, a^{2u}, \ldots$  whether they have the following property:

$$a^{2^{i-1}u} \neq \pm 1 \mod n, a^{2^{i}u} = 1 \mod n.$$

That is, suppose we find a non-trivial square root of 1 modulo n. It turns out that only composite numbers have non-trivial square roots – prime numbers do not. In fact, if we choose a randomly, and n is composite, for at least 3/4 of the values of a, one of two things will happen: we will either find a non-trivial square root of 1 using this process, or we will find that  $a^{n-1} \neq 1 \mod n$ . In either case, we know that n is composite!

(Intuition behind the Rabin-Miller test: Suppose that  $a^{p-1} = 1 \mod p$ and suppose that  $p-1 = 2^t u$ . Then,  $a^{(p-1)/2} = a^{2^{t-1}u}$  can be one of 1, -1 or some other value modulo p. 1 and -1 are considered trivial square roots of 1. Essentially, if we find an i such that  $a^{2^{i-1}u} \neq \pm 1 \mod n$  and  $a^{2^iu} = 1 \mod n$ then  $a^{2^{i-1}u}$  is a non-trivial square root of 1 and only composite numbers have such non-trivial square roots of 1. This is the reason the Rabin-Miller test uses repeated squaring: because we are interested in determining if there is a non-trivial square root of 1 in that sequence. As an example, suppose that we are interested in checking if 15 is a prime number or not.  $15-1=2^1 \times 7$  giving us t = 1 and u = 7. Consider a = 4.  $a^u = 4^7 = 4 \mod 15$  and  $a^{2u} = 4^{14} = 1$ mod 15. In this case  $4^7$  is a non-trivial square root of 1 modulo 15 and therefore 15 is a composite number. If we only looked at  $4^{14} = 1 \mod 15$  then we would conclude that 15 is a 4-pseudoprime.)

A value of a for which either  $a^{n-1} \neq 1 \mod n$  or the computation of  $a^{n-1}$  yields a non-trivial square root is called a witness to the compositeness of n. We have said that 3/4 of the possible values of a are witnesses (we will not prove this here!). So if we pick a single value of a randomly, and n is composite, we will determine that n is composite with probability at least 3/4. How can we improve the probability of catching when n is composite?

The simplest way is just to repeat the test several times, each time choosing a value of *a* randomly. (Note that we do not even have to go to the trouble of making sure we try different values of a each time; we can choose values with replacement!) Each time we try this we have a probability of at least 3/4 of catching that *n* is composite, so if we try the test *k* times, we will return the wrong answer in the case where *n* is composite with probability  $(1/4)^k$ . For k = 25, the probability of the algorithm itself making an error is thus  $(1/2)^{50}$ ; the probability that a random cosmic ray affected your arithmetic unit is probably higher!

This trick comes up again and again with randomized algorithms. If the probability of catching an error on a single trial is p, the probability of failing to catch an error after t trials is  $(1-p)^t$ , assuming each trial is independent. By making t sufficiently large, the probability of error can be reduced. Since the probability shrinks exponentially in t, few trials can produce a great deal of security in the answer.

## **Examples**

- *n* = 252601.
  - $n 1 = 2^3 \times 31575$ . Suppose we select a = 85132.

$$a^{31575} = 191102 \mod n$$
  
 $a^{2 \times 31575} = 184829 \mod n$   
 $a^{2^2 \times 31575} = 1 \mod n$ 

252601 is a composite number because 184829 is a non-trivial square root of 1. a = 85132 is a witness to this fact.

 $n - 1 = 2^2 \times 26179$ . Suppose we select a = 96152.

$$a^{26179} = 1 \mod n$$

104717 is *probably prime* on the basis of this test alone.

00470

• *n* = 3057601.

 $n - 1 = 2^6 \times 47775$ . Suppose we select a = 99908.

$$a^{47775} = 1193206 \mod n$$
  
 $a^{2 \times 47775} = 2286397 \mod n$   
 $a^{2^2 \times 47775} = 235899 \mod n$   
 $a^{2^3 \times 47775} = 1 \mod n$ 

3057601 is a composite number because 235899 is a non-trivial square root of 1. a = 99908 is a witness to this fact.

• *n* = 577757.

 $n-1 = 2^2 \times 144439$ . Suppose we select a = 314997.

$$a^{144439} = 373220 \mod n$$
  
 $a^{2 \times 144439} = 577756 = -1 \mod n$   
 $a^{2^2 \times 144439} = 1 \mod n$ 

577757 is probably prime on the basis of this test alone because 577756 mod  $n = n - 1 \mod n = -1 \mod n$  is a trivial square root of 1.