7 Number Theory 2

7.1 Prime Numbers

Prime Numbers
The generation of prime numbers is needed for many public key algorithms:

- RSA: Need to find p and q to compute $N = pq$
- ElGamal: Need to find prime modulus p
- Rabin: Need to find p and q to compute $N = pq$

We shall see that testing a number for primality can be done very fast

- Using an algorithm which has a probability of error
- Repeating the algorithm lowers the error probability to any value we require.

Prime Numbers
Before discussing the algorithms we need to look at some basic heuristics concerning prime numbers.
A famous result in mathematics, conjectured by Gauss after extensive calculation in the early 1800's, is:

Prime Number Theorem The number of primes less than X is approximately $\frac{X}{\log X}$

This means primes are quite common.
The number of primes less than 2^{512} is about 2^{503}

Prime Numbers
By the Prime Number Theorem if p is a number chosen at random then the probability it is prime is about:

$$\frac{1}{\log p}$$

So a random number p of 512 bits in length will be a prime with probability:

$$\approx \frac{1}{\log p} \approx \frac{1}{355}$$

So on average we need to select 177 odd numbers of size 2^{512} before we find one which is prime.
Hence, it is practical to generate large primes, as long as we can test primality efficiently

7.2 Primality Testing

Primality Tests
For many cryptographic schemes, we need to generate large primes. This is usually done as follows:

- Select a random large number
- Test whether or not the number is a prime.

Naive approach to primality testing on n:

- Check if any integer from 2 to n-1 (or better: $\sqrt{n}$) divides n.

An improvement:

- Check whether n is divisible by any of the prime numbers $\leq \sqrt{n}$
- Can skip all numbers divisible by each prime number (Sieve of Eratosthenes)

These methods are too slow.

Sieve of Eratosthenes
To find prime numbers less than M:

- List all numbers $2, 3, 4, \ldots, M-1$
- Cross out all numbers with factor of 2, other than 2
- Cross out all numbers with factor of 3, other than 3, and so on
- Numbers that "fall through" sieve are prime

	2	3	~~4~~	5	~~6~~	7	~~8~~	~~9~~	~~10~~
11	~~12~~	13	~~14~~	~~15~~	~~16~~	17	~~18~~	19	~~20~~
~~21~~	~~22~~	23	~~24~~	~~25~~	~~26~~	~~27~~	~~28~~	29	~~30~~

Primality Tests
Two varieties of primality test:

- Probabilistic
  - Identify probable primes with very low probability of being composite (in which case they are called pseudoprimes).
  - Much faster to compute than deterministic tests.
  - Examples:
    - Fermat
    - Solovay-Strassen
    - Miller-Rabin

- Deterministic
  - Identifies definite prime numbers.
  - Examples:
    - Lucas-Lehmer
    - AKS

7.3 Fermat Primality Test

Fermat Primality Test
Fermat's Little Theorem: if n is prime and $1 \leq a < n$, then:

$$a^{n-1} \equiv 1 \pmod{n}$$

To test if n is prime, a number of random of values for a are chosen in the interval $1 < a < n-1$, and checked to see if the following equality holds for each value of a:

$$a^{n-1} \equiv 1 \pmod{n}$$

If n is composite then for a random $a \in \mathbb{Z}_n^*$:

$$\Pr[a^{n-1} \equiv 1 \pmod{n}] \leq 1/2$$

A composite number n is called a pseudoprime to base a if $a^{n-1} \equiv 1 \pmod{n}$.

Fermat Primality Test

```
Pick random a,  1 < a < n-1
if a^(n-1) (mod n)=1 then
    return PRIME
else
    return COMPOSITE
end
```

This test can be repeated t times to reduce the probability of classifying composites as primes.
If the algorithm outputs COMPOSITE at least once: output COMPOSITE; this will always be correct (a is called a witness).
If the algorithm outputs PRIME in all t trials: output PRIME; this will be an error with probability $(1/2)^t$.
Some composites always pass Fermat's test, and so are falsely identified as prime: the Carmichael Numbers.

Fermat Primality Test
Carmichael numbers are composite numbers n which fail Fermat's Test for every a not dividing n.

- Hence probable primes which are not primes at all.

There are infinitely many Carmichael Numbers

- The first three are 561, 1105, 1729

Carmichael Numbers n have the following properties:

- Always odd
- Have at least three prime factors
- Are square free
- If p divides n then $p-1$ divides $n-1$.

Fermat Primality Test
Example: consider n = 1234567890.

- n is a composite (clearly) with one witness given by $a = 2$.
- $a^{n-1} \pmod{n} = 612861332$

Example: consider $n = 2^{192} - 2^{64} - 1$.

- n is probably prime since we cannot find a witness for compositeness.
- Actually n is a prime, so it is not surprising we did not find a witness.

7.4 Solovay-Strassen Primality Test

Solovay-Strassen Primality Test
Euler's Criterion: if n is an odd prime and $a \in \mathbb{Z}_n^*$ then:

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

- $\left(\frac{a}{n}\right)$ is the Jacobi symbol.
- If n is composite then for a random $a \in \mathbb{Z}_n^*$:

$$Pr[\left(\frac{a}{n}\right) = a^{(n-1)/2}] \leq 1/2$$

Algorithm proposed by Solovay and Strassen (1973):

- A randomized algorithm.
- Never incorrectly classifies primes and correctly classifies composites with probability at least 1/2.

Solovay-Strassen Primality Test

```
Pick random a, 1 < a < n-1
if gcd(a,n)>1 then
   return COMPOSITE
end
if (a/n) = a^((n-1)/2) then
   return PRIME
else
   return COMPOSITE
end
```

This test can be repeated t times to reduce the probability of classifying composites as primes.

- If the algorithm outputs COMPOSITE at least once: output COMPOSITE; this will always be correct (a is called a witness).
- If the algorithm outputs PRIME in all the t trials: output PRIME; this will be an error with probability $(1/2)^t$.

Solovay-Strassen Primality Test
Example: Consider $n = 15$.
For $a = 3, 5, 6, 9, 10, 12$ the algorithm will output COMPOSITE
For the other values of a which are relatively prime to n:

a	$(\frac{a}{15})$	$a^7 \pmod{15}$
1	1	1
2	1	8
4	1	4
7	-1	13
8	1	2
11	-1	11
13	-1	7
14	-1	14

The algorithm will output PRIME only for $a = 1$ and $a = 14$.

7.5 Miller-Rabin Primality Test

Miller-Rabin Primality Test
Let 2^k be the largest power of 2 dividing $n-1$.
Thus we have $n-1 = 2^k m$ for some odd number m.
Consider the sequence: $a^{n-1} = a^{2^k m}, a^{2^{k-1} m}, \ldots, a^m$.
We have set this sequence up so that each member of the sequence is a square root of the preceding member.
If n is prime, then by Fermat's Little Theorem, the first member of this sequence $a^{n-1} \equiv 1 \pmod{n}$.
When n is prime, the only square roots of 1 (mod n) are ± 1.

Hence either every element of the sequence is 1, or the first member of the sequence not equal to 1 must be -1 ($\equiv n-1 \pmod{n}$).
The Miller-Rabin test works by picking a random $a \in \mathbb{Z}_n$, then checking that the above sequence has the correct form.

Miller-Rabin Primality Test

```
Pick random a, 1 < a < n-1
b = a^m (mod n)
if b ≠ 1 and b ≠ n-1 then
    i=1
    while i < k and b ≠ n-1
        b = b^2 (mod n)
        if b = 1 then
            return COMPOSITE
        end
        i = i+1
    end
    if b ≠ n-1 then
        return COMPOSITE
    end
end
return PRIME
```

Miller-Rabin Primality Test
For any composite n the probability n passes the Miller-Rabin test is at most 1/4. On average it is significantly less.
The test can be repeated t times to reduce the probability of classifying composites as primes.

- If the algorithm outputs COMPOSITE at least once: output COMPOSITE; this will always be correct (a is called a witness).
- If the algorithm outputs PRIME in all the t trials: output PRIME; this will be an error with probability $(1/4)^t$.

Unlike the Fermat test, there are no composites for which no witness exists.

Miller-Rabin Primality Test
Example: Consider $n = 91$.
$n-1 = 90 = 2 \times 45$, so $k = 1$, $m = 45$.
For a = 1, 9, 10, 12, 16, 17, 22, 29, 38, 53, 62, 69, 74, 75, 79, 81, 82, 90 the algorithm will output PRIME.
These values are called strong liars.
91 is a strong pseudoprime to each of these bases.
For other values of a the algorithm will output COMPOSITE.
These values are called strong witnesses

7.6 Lucas-Lehmer Primality Test

Lucas-Lehmer Primality Test

A Mersenne number is an integer of the form $2^k - 1$, where $k \geq 2$.
If a Mersenne number is a prime, then it is called a Mersenne prime.
Subject of the Great Internet Mersenne Prime Search (GIMPS).
The Mersenne number $n = 2^k - 1$ ($k \geq 3$) is prime if and only if the following two conditions are satisfied:

1. k is prime
2. the sequence of integers defined by $b_0 = 4$, $b_{i+1} = (b_i^2 - 2) \pmod{n}$ $(i \geq 0)$ satisfies $b_{k-2} = 0$.

This is the basis of the Lucas-Lehmer Primality Test.

Lucas-Lehmer Primality Test

```
if k has any factors between 2 and √k
    return COMPOSITE
end
b = 4
for i=1 to k−2 do
    b = (b^2 − 2) mod n
end
if b = 0 then
    return PRIME
else
    return COMPOSITE
```

7.7 AKS Primality Test

AKS Primality Test

AKS algorithm discovered by Agrawal, Kayal and Saxena in 2002.
Result of many research efforts to find a deterministic polynomial-time algorithm for testing primality.
Based on the following property: if a and n are relatively prime integers with $n > 1$, n is prime iff:

$$(x-a)^n \equiv x^n - a \pmod{n}$$

where x is a variable.
Always returns correct answer.
Polynomial time algorithm, but still too inefficient to be used in practice.

AKS Primality Test

```
if n has the form a^b (b > 1) then
    return COMPOSITE
end
r = 2
```

```
while r < n
    if gcd(n,r) ≠ 1 then return COMPOSITE
    if r is a prime > 2 then
        q=largest factor of r−1
        if q > 4∗√r∗log n and n^((r−1)/q) ≠ 1 (mod r) then
            break
        end
        r = r+1
    end
end
for a=1 to 2∗√r∗log n do
    if (x−a)^n ≠ x^n − a (mod gcd(x^r − 1, n)) then return COMPOSITE
end
return PRIME
```

7.8 Primality Testing in Practice

Primality Testing in Practice

The Miller-Rabin test is preferable to the Solovay-Strassen test for the following reasons:

- The Solovay-Strassen test is computationally more expensive.
- The Solovay-Strassen test is harder to implement since it also involves Jacobi symbol computations.
- The error probability for Solovay-Strassen is bounded above by $(1/2)^t$, while the error probability for Miller-Rabin is bounded above by $(1/4)^t$.
- From a correctness point of view, the Miller-Rabin test is never worse than the Solovay-Strassen test.

AKS is a breakthrough result: proves that PRIMES $\in$ P.

- Always gives correct results.
- No practical relevance: prohibitively slow run-times.