Lecure 4

Notes by Maja Gidlund

October 11, 2014

1 Integer factorization and primality

Given integer N is it prime? If not, what are it factors? Why do we care about this problem? There are at least two reasons as follows.

1. Basic mathematical problem.
2. Our ability to find primes quickly and our inability to factor very large numbers are the key reasons that the RSA encryption system works.

Let us start with a simple question and look at factoring 32-bit integers. The first algorithm that comes to mind is trial division: Try to divide by $2, 3, 4, ...\sqrt{N}$ and see if any division gives an integer answer.

The number of operations required this is about $\sqrt{N}$ which in this case is about $2^{16} \approx 65000$ and as a computer does about 2^{30} operations per second this takes less than a millisecond. If we, on the other hand, want to factor a 100-bit number this requires 2^{50} operations which would need 2^{20} (about two weeks). Factoring a 200-bit number by this method not feasible.

We can consider the following improvement: Only divide by primes $\leq \sqrt{N}$. To evaluate this algorithm we need to know how many primes there are $\leq M$. The prime number theorem tells us that this is roughly $\frac{M}{\ln M}$ and thus the improvement is only by a factor $\ln N$ and is minor and we need to turn to other algorithms. We note for the future that random number with t bits is prime with probability roughly $\frac{1}{0.7t}$.

It turns out that it is much easier to check whether a number is prime than to factor a number that we know is composite so let us turn to this problem first. We have the following theorem.

Theorem 1 *If N is prime then for $1 \leq a \leq N-1$ we have $a^{N-1} \equiv 1$ mod N.*

As example note the following examples.

- N=7, $2^6 = 64 \equiv 1 \bmod 7$ *7 looks prime.*
- N=15, $2^{14} = 16384 \equiv 4 \bmod 15$ *15 is not a prime.*

Note that in the second case we can conclude that 15 is not prime but do not get any indication how to find its factors. To be able to use the Fermat theorem we need two additional properties

1. Need to check whether a^{N-1} mod N equals one quickly.
2. Need to converse, i.e. something similar to that if N is not prime then $a^{N-1} \equiv 1$ mod N only for few or no a.

Answers:

1. Is true.
2. Almost true, there are a few bad N's, but for random N it works fine.

Let us turn to computing a^{N-1} mod N , $N \geq 2^{100}$. Note that a^{N-1} has at least N bits and thus we cannot write down this integer. Crucial observation: If doing some multiplications and we want the answer mod N then we need only know partial results mod N and thus the size of numbers do not have to become large. However, computing a^{N-1} without thinking requires $N-2$ multiplications and we have to be more clever. Let us show how to this by an example and compute 2^{46} mod 47. First note that 46 = 101110 in binary and consider the following numbers in binary

1. 1=1
2. 2=10
3. 5=101
4. 11=1011
5. 23=10111
6. 46=101110

We compute 2 to these powers mod 47.

1. $2^1 = 2$
2. $2^2 = 4$
3. $2^5 = 32$
4. $2^{11} = (2^5)^2 \cdot 2 = 1024 \cdot 2 = 372 = 27$ mod 46
5. $2^{23} = (2^{11})^2 \cdot 2 = 27^2 \cdot 2 = 729 \cdot 2 = 24 \cdot 2 = 1$
6. $2^{24} = (2^{23})^2 = 1^2 = 1$

As each exponent is either twice the previous exponent or that number plus one we can get always get from one to the next by squaring and, if needed, one multiplication. We summarize this fact as follows.

Theorem 2 *a^b mod N for n-digit numbers a, b, N can be computed by n squarings, and $\leq n$ multiplications mod N.*

In practice for 1000-digit numbers this can be done in less than second.

Let us recall a useful fact. For a prime p a number g is a generator if the numbers g, g^2, g^3, $\ldots g^{p-1}$ are all numbers $\neq 0$ mod p. Two examples

- $p = 5$, $g = 2$, 2,4,3,1
- $p = 7$, $g = 3$, 3,2,6,4,5,1

We have $p - 1$ numbers. Looking at g^i is taking every i'th element (and wrapping around).

What is $g^{\frac{p-1}{2}}$ for a generator g? Looks to be $p - 1$ (or equivalently -1), but is this always the case? We have $(g^{\frac{p-1}{2}})^2 = g^{p-1} \equiv 1$ mod p by Fermat. and thus this number is a solution to $x^2 = 1$. For a prime, a degree d equation has at most d solutions and thus for N prime, $x^2 = 1$ mod N has only the obvious solutions 1 and -1. In our case as $g^{\frac{p-1}{2}} \neq 1$ we must have $g^{\frac{p-1}{2}} = -1$.

If N is not prime other things can happen and in particular $x^2 = 1$ mod 8 has the solutions $x = 1, 3, 5, 7$.

As stated above, for most N it is the case that for most a, $1 \leq a \leq N - 1$, $a^{N-1} \neq 1$ mod N. Exceptions are known as "Carmichael numbers" and Gary Miller and Michael Rabin (MR) showed how to take care of all numbers.

Miller-Rabin

1. If N is even output "not prime" unless $N = 2$.
2. Otherwise $N - 1 = 2^\ell \cdot s$, s odd.
3. Pick random a, $1 \leq a \leq N - 1$ and compute $x_0 \equiv a^s$ mod N,
4. $x_{i+1} = x_i^2$ mod N, $i = 0, 1, ..., \ell - 1$.

Note that $x_\ell = a^{2^\ell \cdot s} = a^{N-1}$ mod N. Guess prime iff: $x_\ell \equiv 1$ and $x_0 \equiv 1$ or last $x_i \neq 1$ equals -1. Otherwise we know that N is composite.

We have the following two theorems.

Theorem 3 *If N is prime the MR always says "probably prime".*

Theorem 4 *If N is composite the probability that MR says "probably prime" is $\leq \frac{1}{4}$.*

In other words, $\frac{3}{4}$ of all a witness that N is composite. If we try 50 random a's then the probability of an incorrect answer is at most $2^{-100} = \left(\frac{1}{4}\right)^{50}$. Finally we give the following algorithm for picking your own 1000-bit prime:

1. Pick random own 1000-bit number N.

2. Run MR on N. If composite, try again.

Note that we can me algorithms more efficient by first doing trial division with small factors.