Advanced Algorithms	Lecture 3 : March 14, 1995
Lecturer: Johan Håstad	*Scribe: Johan Håstad*

3.1 More integer factorization

We want to solve the equation $x^2 = y^2$ mod N where $x \neq \pm y$ mod N. The reason is that in such a case

$$0 = x^2 - y^2 = (x+y)(x-y) \quad mod\ N,$$

and since neither factor is 0 mod N, $gcd(x+y, N)$ will be nontrivial.

Already Fermat tried this and his approach was to find q such that q^2 was small mod N. The idea is that it is more likely that a small number is a square. Let us heuristically estimate this probability. Since the distance between s^2 and $(s+1)^2$ is about $2s$ then the probability that a random number of size roughly x is a square is about $\frac{1}{2\sqrt{x}}$.

One approach for getting numbers q with q^2 small mod N is choosing an integer close to $\sqrt{N}$ or any number of the form $\lceil \sqrt{aN} \rceil + b$ where a and b are small. In this case we get that q^2 mod N is on the order of magnitude of $\sqrt{N}$ and hence heuristically the probability that is a square is on the order of $N^{-1/4}$ and thus we get a factoring algorithm that runs in time $O(N^{1/4})$.

It is still unknown how to find a general method to generate $Q > \sqrt{N}$'s such that Q^2 mod N is significantly smaller than $\sqrt{N}$. Before we continue let us describe another method for obtaining number Q such that Q^2 mod N is small.

3.2 Continued fractions

Continued fractions is a method for obtaining good rational approximations to a real number Θ. Suppose $\Theta > 1$ then we write

$$\Theta = a_1 + \frac{1}{\Theta_1}$$

where a_1 is the largest integer smaller than Θ and Θ_1 is whatever makes the equation correct. By construction $\Theta_1 > 1$ and hence we can repeat the process with Θ_1 etc

$$\Theta = a_1 + \frac{1}{a_2 + \frac{1}{\Theta_2}} = a_1 + \frac{1}{a_2 + \frac{1}{a_3+\frac{1}{\Theta_3}}}.$$

Replacing $\frac{1}{\Theta_i}$ by 0 gives a rational number $\frac{p_i}{q_i}$ which turns out to be a good approximation of Θ. For instance the famous approximation $\frac{22}{7}$ for π can be obtained by using

$$\pi = 3 + \frac{1}{7 + \frac{1}{\pi_2}}$$

and then ignoring $\frac{1}{\pi_2}$. In general it is true that

$$|\Theta - \frac{p_i}{q_i}| \leq \frac{1}{q_i^2}.$$

The numbers p_i and q_i can be obtained very efficiently given an approximation of Θ.

Let us see how to use this to obtain numbers Q such that Q^2 is small. We use the continue fraction expansion of $\sqrt{N}$ and let $\frac{p_i}{q_i}$ be one obtained fraction. Since the derivative of x^2 at $\sqrt{N}$ is $2\sqrt{N}$ we have that $\frac{p_i^2}{q_i^2}$ is within $2\sqrt{N}q_i^{-2}$ of N and hence p_i^2 mod N is $O(\sqrt{N})$.

3.3 Combining equations

Consider the following. Let $N = 4633$ be a number to be factored. Then

$$67^2 = -144 \quad mod\ N$$

and

$$68^2 = -9 \quad mod\ N.$$

Neither of these equations can be used directly to obtain a factorization of N. However, They can be combined to yield

$$(67 \cdot 68)^2 = -144 \cdot -9 = (12 \cdot 3)^2 \quad mod\ N.$$

Since $67 \cdot 68 = (-77)$ mod N we obtain the factorization $4633 = 41 * 113$ since $gcd(-77 + 36, 4633) = 41$ and $gcd(-77 - 36, 4633) = 113$.

We want to expand this to a more general approach. Take a number B and consider all primes smaller than B together with the special "prime" -1. These primes will be called our factor base. We generate Q such that Q^2 mod N has small absolute value and hope that it factors in the factor base. In such a case we remember it and pick a new Q and try again. This way we generate equations of the form

$$Q_i^2 = \prod_{p<B} p^{a_{i,p}}$$

for some $i = 1, 2 \ldots s$. We want to combine these by choosing a suitable subset S and multiplying the equations in this subset. This yields an equation of the form

$$\prod_{i \in S} Q_i^2 = \prod_{i \in S, p<B} p^{a_{i,p}} = \prod_{p<B} p^{\sum_{i \in S} a_{i,p}}.$$

The left hand side is always a square while the right hand side is a square if $\sum_{i \in S} a_{i,p}$ is even for any p. Thus all that remains is to find such an S. However this is just linear algebra mod 2. In particular, we start with the vectors

$$\begin{array}{cccc}
(a_{1,-1}, & a_{1,2}, & a_{1,3}, & \ldots a_{1,B}) \\
(a_{2,-1}, & a_{2,2}, & a_{2,3}, & \ldots a_{2,B}) \\
(a_{3,-1}, & a_{3,2}, & a_{3,3}, & \ldots a_{3,B}) \\
\vdots & \vdots & \vdots & \vdots \\
(a_{s,-1}, & a_{2,2}, & a_{s,3}, & \ldots a_{s,B})
\end{array}$$

Where each coefficient just is considered a number mod 2 and we want to find a linear dependence. If there are more vectors than dimensions there is always such a dependence and it can be found efficiently by Gaussian elimination. Combining this set of equations gives a relation $x^2 = y^2$ mod N as described above. There is no guarantee that it will be nontrivial (i.e. $x \neq \pm y$) and there is not even a proof that that it is nontrivial with high probability. However, heuristically it should be nontrivial with probability $1/2$ and this turns out to be true in practice.

There are $\pi(B) \approx \frac{B}{\ln B}$ primes less than B and thus we need at least that many equations. This leads to the following algorithm which is parameterized by B.

1. Find all primes $\leq B$.
2. Find numbers Q of the form $\lceil \sqrt{aN} \rceil + b$ where a and b are small and see if Q^2 mod N factor in primes $\leq B$. Repeat until we have $\pi(B) + 2$ successes.
3. By Gaussian elimination find a subset S of the equations such that when the corresponding equations are multiplied we get a relation $x^2 = y^2$ mod N. If this

is a trivial then find another number Q such that $Q^2 \bmod N$ factors in primes at most B and repeat until we get a nontrivial relation (and hence a successful factorization).

Let us try to analyze this algorithm. One key to performance is the choice for B. Let $S(A, B)$ be the probability that a random number around A factors in prime factors bounded by B. Then heuristically the second step runs in time about

$$B^2 S(\sqrt{N}, B)^{-1}.$$

This follows since we need to find about B numbers that factor and for each success we expect to examine $S(\sqrt{N}, B)^{-1}$ numbers and to investigate whether one number factors can be done in time about B. The third step runs in time about $O(B^3)$. To balance the above factors B should be chosen around $2^{c\sqrt{\log N \log\log N}}$ which gives a running time of $2^{d\sqrt{\log N \log\log N}}$ for some constants c and d. In practice one has to be very careful when choosing B. In the recent success in factoring a 129 digit number B was chosen to be around 10^7.

3.3.1 Implementation tricks

To make the given algorithm run fast a number of implementation tricks can be used and let us mention one. Suppose we fix a. Whether $(\lceil\sqrt{aN}\rceil + b)^2 \bmod N$ is divisible by a specific prime p depends only on $b \bmod p$. This suggests the following algorithm

1. Store a floating point approximation of $\log Q_b$ where $Q_b = (\lceil\sqrt{aN}\rceil + b)^2 \bmod N$ in cell b of an array.
2. For each p find possible values of $b \bmod p$ such that Q_b is divisible by p. There are either no such b or two possible values which we call b_1 and b_2. In the latter case subtract $\log p$ from cells with addresses $b_1 + ip$ and $b_2 + ip$ for all possible values of i.

Since this implementation is very similar to a sieve, the algorithm is called the quadratic sieve, see [6], [1].

3.4 Other methods

There are a number of other methods for factoring and let us mention their names and a couple of sentences on each method.

1. **Elliptic curves, [2], [4].** Generate a random elliptic curve and hope that the number of points on the curve is a number that has only small prime factors. This algorithm runs in expected time

$$2^{d\sqrt{\log N \log\log N}}$$

   for some constant d. This algorithm has the property that it finds small prime factors faster than large prime factors.

2. **Continued fraction, [5].** Similar method to the one described here. It uses a similar factor base and the only difference is that it generates the numbers $Q^2 \bmod N$ to be factored using continued fractions as described in an earlier section. This algorithm also runs in time

$$2^{d\sqrt{\log N \log\log N}}$$

   for some constant d.

3. **Number field sieve, [3].** This is especially efficient for numbers N of the form $a^b + c$ where a is small. Suppose we want to factor $7^{200} + 11$. Let $\alpha = 7^{40}$. Then $\alpha^5 = -11 \bmod N$ and thus can be considered as an algebraic number. Also this algorithm has a factor base. It contains the small standard primes but also numbers of the form

$$a_0 + a_1\alpha + a_2\alpha^2 + a_3\alpha^3 + a_4\alpha^4$$

   where this is a prime with small norm when considered as an algebraic number. The algorithm then generates numbers of the form $a\alpha + b$ and tries to factor them both as a standard integer and as an algebraic number. As usual it collects successful factorizations and then finds a dependence mod 2. This algorithm has been implemented and found to be the most efficient existing algorithm for the the special numbers mentioned above. There is also hope that the algorithm can be extended to work for general numbers with an expected running time of

$$2^{d(\log N)^{1/3}(\log\log N)^{2/3}}$$

   for some constant d.

References

[1] Gerver: *Factoring Integers with a Quadratic Sieve.* Math. Comp. (41) 1983.

[2] Lenstra: *Factoring Integers with Elliptic Curves.* Annals of Math. (2) 1987.

[3] Lenstra & Lenstra: *The Development of the Number Field Sieve.* Springer Verlag, 1993.

[4] Montgomery: *Speeding the Pollard and Elliptic Curve Methods of Factorization.* Math. Comp. (48) 1987.

[5] Morrison & Brillhart: *A Method of Factoring and the Factorization of* F_7. Math. Comp. (29) 1975.

[6] Pomerance: *Analysis and Comparison of Some Integer Factoring Algorithms.* Comp. Methods in Number Theory, Math. Centrum, Amsterdam, 1982.