

Algorithmic Number Theory 4

Notes by Al Sethuraman

October 30th 2000

In this note, we will look at Pollard's $p-1$ factorization algorithm, and its generalization to Lenstra's Elliptic Curve factorization algorithm.

Pollard's algorithm is extremely good at finding factors of an integer n which have the property that for some prime divisor p of n , $p-1$ has only "small" prime factors. We first make the notion of "small" more precise:

Definition 0.1 *Let B be a finite set of primes. An integer n is said to be B -smooth if every prime factor of n comes from the set B .*

Typically, B is chosen to consist of the first t primes, for some computationally convenient t . (According to the 1997 Handbook of Applied Cryptography, B is typically chosen to contain all primes up to about 10^5 or 10^6 .) Having fixed B , Pollard's algorithm "almost always" finds a factor of n if n is divisible by some prime p such that $p-1$ is B -smooth.

Here is how the algorithm works. Given n , we first fix a B depending on our computational resources. We quickly check if any of the primes in B divide n ; if any of the primes in B already divides n , we have one factor of n already, and our input number has just become smaller. (Each divisibility check is only an $O(r^2)$ process, where $r = \log_2(n)$.) Next, we pick an integer a at random, with $2 \leq a \leq n-1$, and we check if $\gcd(a, n) \neq 1$. Once again, if the \gcd is greater than 1, we have already found a factor.

Next, for each prime q in B , we let α_q denote the greatest integer in $\log n / \log q$. Then, q^{α_q} is the highest power of q that is less than n . Let k denote the product of all the α_q , for the various q in B . We compute $a^k \pmod n$ by the repeated squaring and reduction mod n algorithm we discussed earlier in the third set of notes. Next, we compute $\gcd(a^k \pmod n - 1, n)$. If this \gcd turns out to be 1, our test has failed, and we move on to some other test. However, if there is a prime divisor p of n such that $p-1$ is B -smooth, then "almost always" this \gcd will be greater than 1, and we will find a nontrivial factor of n .

Here is the reason. Since $p-1$ is B -smooth, $p-1$ factors as the product of q^{β_q} for various q 's coming from B . The maximum that β_q can be is only α_q , as $p-1$ is less than n , and q^{α_q+1} exceeds n . It follows that $p-1$ must divide k , as k is defined to be the product of the various q^{α_q} . Hence, in the group $U(\mathbf{Z}/p)$, we must find $a^k = 1$, so $a^k - 1$ is a multiple of p . In particular, $a^k \pmod n - 1$ is also a multiple of p (since we have only subtracted off multiples of n , and thus multiples of p , to go from a^k to $a^k \pmod n$). As long as $a^k \pmod n - 1$ is not zero, i.e., as long as $a^k \pmod n - 1 = bp$ for some nonzero integer b , we'd have found a nontrivial divisor of n . (Note that $a^k \pmod n - 1$ is anyway less than n , so if $a^k \pmod n - 1$ is not zero, then we will indeed find a divisor of n other than n itself.)

So, to understand what "almost always" means, we just have to understand under what situations we will find $a^k \pmod n - 1$ to be zero. Note that this means that in the group

$U(\mathbf{Z}/n)$, the order of a must divide k . But the order of a must also divide $\phi(n)$. If $n = p_1^{l_1} \cdots p_s^{l_s}$, then $\phi(n) = p_1^{l_1-1}(p_1 - 1) \cdots p_s^{l_s-1}(p_s - 1)$. The order of a cannot be divisible by any of the p_i , as none of the p_i equal any of the primes q that divide k —this has already been ruled out at the very beginning. In the abstract decomposition of $U(\mathbf{Z}/n)$ as $C_{p_1^{l_1-1}} \times C_{p_1-1} \cdots \times C_{p_s^{l_s-1}} \times C_{p_s-1}$, if a has a factor along any of the $C_{p_i^{l_i-1}}$, then the order of a will be divisible by p_i —hence, for $a^k = 1 \pmod{n}$ to occur, a must have factors only along the groups of the form C_{p_i-1} . There is no way of knowing ahead if n is square free, but on the off chance that it is, one would simply repeat the Pollard test a few times with different choices of a to see if one gets an a with a factor along one of the C_{p_i-1} , and then, the Pollard algorithm will work. The proportions of a 's for which this will work doesn't seem to me deducible up front, without knowing the factorization of n .

Of course, n could turn out to be square free. In that case, $U(\mathbf{Z}/n)$ is just abstractly the product of the various groups C_{p_i-1} . If at least one of these p_i were not B -smooth, then by choosing different a at random and repeating the Pollard algorithm, one would hit an a whose order does not divide k , and then for that a , the algorithm would work. Once again, the proportions of a 's for which this strategy will work doesn't seem to me deducible up front, without knowing the factorization of n .

If each of these $p_i - 1$ is B smooth, then $\phi(n)$ will divide k , so a^k will always equal 1 mod n . Thus, randomly varying a any number of times will always produce $a^k = 1 \pmod{n}$. But then, in this situation here is another method that will definitely work: since each p_i is B -smooth, each $p_i - 1$ is a product of prime powers of the form q^{γ_q} with the q 's from B , and the $\gamma_q \leq \alpha_q$. We simply form the finitely many products of the various $q^{\gamma_q} \pmod{n}$, add 1, and check the gcd of the result with n . One of these products, with 1 added, will yield a p_i , and then the gcd process will exhibit this p_i .

Note that we go through these gyrations only when we find that $a^k = 1 \pmod{n}$, at which point we know that k and $\phi(n)$ have a nontrivial common factor.

Let us pick the key features of this process. First, one starts by accepting that one has to have some notion of smoothness: after all, if one were to search among a set of primes for divisors of an integer n , this set should be small to be computationally effective. (Else, one might as well do a straightforward trial division by integers less than \sqrt{n} .) Next, one hopes that the order of one of the groups $U(\mathbf{Z}/p)$, for various primes p dividing n , will be B smooth. (This algorithm fails if this were not the case.) Finally, notice that when one of the primes p is such that $p - 1$ is B smooth then this algorithm produces an expression (namely, $a^k - 1 \pmod{n}$) which yields, in “most” cases, a nontrivial gcd with n .

The problem, of course, is that n may fail to satisfy the condition that one of the groups $U(\mathbf{Z}/p)$ is B smooth. Lenstra's algorithm dispenses with the groups $U(\mathbf{Z}/p)$ altogether, and substitutes them with various elliptic curves mod p . One has a huge number of such curves to select from, and the order of at least one of them will be B -smooth, at least, if one searches “long enough.” The moment one has an elliptic curve whose order mod p is B -smooth, Lenstra's algorithm also provides an expression that will yield a nontrivial gcd

with n . (Of course, one doesn't know in advance what this p is modulo which the elliptic curve has B smooth order, but the algorithm finesses this point quite cleverly, by working modulo n instead.)

First, let us quote, for the record, a theorem of Hasse:

Theorem 0.2 *Let k be a finite field with q elements. The number N of k -rational points on an elliptic curve defined over k satisfies the inequality $|N - (q + 1)| \leq 2\sqrt{q}$. (In other words, $(q + 1) - 2\sqrt{q} \leq N \leq (q + 1) + 2\sqrt{q}$.)*

So what is Lenstra's algorithm? It works as follows: Given an integer n to be factored, one first selects at random an elliptic curve $C : y^2 = x^3 + ax + b$ with integer coefficients, and a random point (x, y) on this curve. This would be done, for instance, by selecting a, x , and y to be random integers, and letting $b = y^2 - x^3 - ax$. Next, one checks that the discriminant of this curve, namely $4a^3 + 27b^2$, is not only nonzero, but also, that $\gcd(4a^3 + 27b^2, n) = 1$. What this guarantees is that not only is the curve C nonsingular as a curve over the rationals, but also, that for any prime $p \geq 5$ that divides n , the curve C_p obtained by reducing the coefficients of $y^2 = x^3 + ax + b \pmod{p}$ is nonsingular as a curve over the field \mathbf{Z}/p . (This is an example about how one finesses the fact that one doesn't yet know which primes divide n —once for all, if you make sure that the discriminant is relatively prime to n , then you guarantee that the discriminant stays relatively prime to any prime divisor of n .)

(Of course, if $\gcd(4a^3 + 27b^2, n) > 1$, we would have found a nontrivial divisor of n . Note that $\gcd(4a^3 + 27b^2, n) = n$ is ruled out since we have checked that $4a^3 + 27b^2 \neq 0$, hence, $\gcd(4a^3 + 27b^2, n) < n$.)

As in Pollard's algorithm, we fix our set B of small primes, and we compute the integer k as the product of all the q^{α_q} over all the primes q in B , where α_q is chosen so that $q^{\alpha_q} < n$ but $q^{\alpha_q+1} > n$. (As in that test, one first checks that none of the primes in B already divide n .)

With $P = (x, y)$, the random point on C , we compute kP , that is, the point P added to itself repeatedly $k - 1$ times. The computation of kP proceeds in stages, and at each stage in the computation, we mod the result by n . If this computation proceeds without a hitch, we need to pick another elliptic curve and a point on it, but if it does have a hiccup in it, then we are onto something interesting! All this requires a bit of explanation:

First, just as with exponentiation, where we compute b^r by "repeated squaring," we compute can compute kP for an elliptic curve by "repeated doubling." The process is the same, except that we are thinking of our elliptic curve as an additive group, instead of multiplicative group. Thus, we write $k = a_t 2^t + a_{t-1} 2^{t-1} + \dots + a_0$ with $a_t = 1$ and the other a_i being either 1 or 0, and we repeatedly compute $2P, 4P, \dots, 2^t P$ by doubling the previous computation. In parallel, as we perform these repeated doublings, we simply add together those $2^i P$'s for which $a_i = 1$.

Next, in our algorithm, since in the end, we only want the coordinates of kP modulo n , we actually compute each of these $2^i P$ modulo n . What would this mean? Let us look at the formulae for addition of two points (x_1, y_1) and (x_2, y_2) first. We have:

$$x_3 = \begin{cases} \left(\frac{y_2 - y_1}{x_2 - x_1}\right)^2 - x_1 - x_2 & \text{if } (x_1, y_1) \neq (x_2, y_2) \\ \left(\frac{3x_1^2 + a}{2y_1}\right)^2 - 2x_1 & \text{if } (x_1, y_1) = (x_2, y_2) \end{cases}$$

and

$$y_3 = \begin{cases} \left(\frac{y_2 - y_1}{x_2 - x_1}\right)(x_1 - x_3) - y_1 & \text{if } (x_1, y_1) \neq (x_2, y_2) \\ \left(\frac{3x_1^2 + a}{2y_1}\right)(x_1 - x_3) - y_1 & \text{if } (x_1, y_1) = (x_2, y_2) \end{cases}$$

We start with our point P which has integer coordinates. It is clear from the above that when we double P , we get coordinates that are rational numbers. Suppose we are given a rational number x/y . We can consider the rational number s/t modulo n only when t is relatively prime to n , and in that case, by $s/t \bmod n$, we simply mean the product $t^{-1}s$ in the group $U(\mathbf{Z}/n)$. If t is not relatively prime to n , $s/t \bmod n$ has no meaning.

So we compute along: starting with our point P , then $2P$, $4P$, etc, and at each stage, we perform our additions ($P + 2P$, or $2P + 4P$, etc.) as necessary (see the repeated doubling algorithm above). At each stage, we take the result of our computation, and we examine the denominators $x_2 - x_1$, or y_1 (depending on whether we just added two unequal points or two equal points). If this appropriate denominator is relatively prime to n , then we can mod both the x and y coordinates by n —thus, we would have computed some $k_1P \bmod n$. We proceed on. But suppose we find that the appropriate denominator is not relatively prime to n , then as long as the gcd is less than n , we are done, we have found a factor of n !

What does it mean to get a denominator that is not relatively prime to n , and how do we know for sure that we'll find ourselves in such a situation? Let us look at the first question. Suppose for now that the gcd of this denominator and n is something less than n (and greater than 1). If p is some prime dividing this gcd , what this means that on the curve C_p obtained by reducing the equation of $C \bmod p$, the point k_1P is the infinite point, i.e, in the group C_p , $k_1P = 0$.

How do we know that we will find ourselves in such a situation, i.e., a situation where the appropriate denominator is not relatively prime to n ? Suppose we have an elliptic curve C whose order mod p , for some prime p dividing n , is B -smooth. Then, somewhere along the computation of kP , we would find $k_1P = 0 \bmod p$, as k is a multiple of the order of B . Why? Note that by Hasse's theorem, the order of C_p is at most $p + 1 + 2\sqrt{p}$, i.e., at most $2p$. Unless n were itself prime (something that has presumably been ruled out ahead of time to high enough probability), $n > 2p > |C_p|$. Thus, for each q in B , the integer α_q definitely has the property that either $q^{\alpha_q} > |C_p|$, or else, $q^{\alpha_q} \leq |C_p|$ and $q^{\alpha_q+1} > |C_p|$. Since k is a product of the various q^{α_q} as q ranges through B , if $|C_p|$ is B -smooth, then $|C_p|$ will definitely divide k . Thus, kP will definitely be zero (if an earlier k_1P is not already zero) in C_p . At this point, the relevant denominator will not be relatively prime to n .

But of course, given an elliptic curve C , we have no guarantee that for any prime p that divides n , the curve C_p will have order that is B -smooth. If it happens that $|C_p|$ is not B -smooth for any prime p , then our product kP (or any of the intermediate products k_1P) is not likely to come out to be zero, so at each stage, the relevant denominators will be

relatively prime to n . When this happens, we generate a new curve and a new point on it, and we repeat the computations on this curve.

The question then arises: how far do you have to go before you find an elliptic curve C for which $|C_p|$ is B smooth for some prime p that divides n ? There is an analysis of this: it is a very detailed analysis, and depends on the distribution of the orders of various elliptic curves mod p in the range $p+1-2\sqrt{p}$ to $p+1+2\sqrt{p}$. It also depends on a conjecture about the number of integers around p that are B smooth, a conjecture that I've seen described as "reasonable." When the dust settles, it turns out that for any positive real number g , this algorithm will find a factor of n with probability at least $1 - e^{-g}$ in time that looks like $O(gK(p)r^2)$, where $r = \log_2(n)$, p is the smallest prime factor of n , and $K(x)$ is a function that looks like $e^{\sqrt{(2+o(1)) \log x \log \log x}}$.

There remains the question of the likelihood that the gcd of the relevant denominator and n turns out to be n itself. This would mean that for the particular k_1 for which the denominator in k_1P has this property, the point k_1P equals zero in the curves C_p for *all* primes p that divide n . According to Koblitz, this is "highly unlikely if n has two or more large prime divisors." I don't know how to analyze this probability.