Factoring Numbers Using Singular Integers

Leonard M. Adleman*

Department of Computer Science
University of Southern California

Abstract

Recently, A.K. Lenstra, H.W. Lenstra, Jr., M.S. Manasse and J.M. Pollard [5,6] have introduced a new algorithm for factoring integers of special form. Based on earlier work of Coopersmith, Odlyzko and Schroeppel [2] and of Pollard [10], the new algorithm, the 'number field sieve', is the fastest known for factoring integers of the form $r^e \pm s$ where e and s are small. In [5], the authors raise the issue of generalizing the number field sieve to produce an efficient algorithm for all numbers. Beginning with the author's suggestions, together with those of Buhler and Pomerance as reported in [5], we produce such a general purpose number field sieve along with an heuristic argument that it factors numbers in random time:

$$e^{(c+o(1))(\log n)^{1/3}(\log\log n)^{2/3}}$$

where $c < 2$.

* Research supported by NSF through grant CCR-8911662.

Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the ACM copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Association for Computing Machinery. To copy otherwise, or to republish, requires a fee and/or specific permission.

© 1991 ACM 089791-397-3/91/0004/0064 $1.50

1 Overview

The description in this note will be informal. It will be assumed that the reader is familiar with [6]. For information on algebraic number theory see for example [7] or [3].

Assume that we wish to factor a rational integer n. As with the number field sieve and many other factoring algorithms, our goal will be to find $x, y \in Z_{>0}$ such that $x^2 \equiv y^2 \bmod n$. This is accomplished in the number field sieve by working in an algebraic extension of the rationals. However, working in such an extension introduces the usual complications: units, non-unique factorization, etc. These complications are sufficiently onerous that generalization of the number field sieve to arbitrary numbers becomes problematic. Our idea is to finesse these complications by the use of singular integers.

For all number fields K and all $\sigma \in O_K$ (the ring of integers of K), σ is a singular integer (more properly a quadratic singular integer with respect to O_K) iff there exists an ideal $I \subseteq O_K$ such that $(\sigma) = I^2$. Note that I may be principal.

The number field sieve proceeds by working

in a field $K = Q(\alpha)$ to find a set S of pairs of rational integers such that:

$$\prod_{<a,b>\in S}(a+\alpha b) = ((\prod_{u\in U} u^{\bar{l}_u})(\prod_{g\in G} g^{\bar{v}_g}))^2 \quad (1)$$

and

$$\prod_{<a,b>\in S}(a+mb) = (\prod_{p\ prime, p\leq B} p^{\bar{w}_p})^2$$

Where B is a positive real number (the 'smoothness' bound). U is a set of generators for the units of O_K, G is a set of generators for the prime ideals of norm less than B and residue class degree one (O_K was assumed to be a PID in [5,6]). f is the minimal polynomial for α and $m \in Z_{>0}$ is such that $f(m) = n$ (or a small multiple of n).

Computing S, U, G and the associated exponents as above is tantamount to factoring n. However, the 'real' goal of (1) is to find a set of pairs S and a $\delta \in O_K$ such that:

$$\prod_{<a,b>\in S}(a+\alpha b) = \delta^2 \quad (2)$$

and

$$\prod_{<a,b>\in S}(a+mb) = (\prod_{p\ prime, p\leq B} p^{\bar{w}_p})^2$$

In the new algorithm, this goal is accomplished in two steps and in a way which avoids the complications referred to above.

First we will be content with constructing a set of integer pairs S such that:

$$\prod_{<a,b>\in S}(a+\alpha b) = (\prod_{\mathcal{P}\in T} \mathcal{P}^{\bar{s}_P})^2 \quad (3)$$

and

$$\prod_{<a,b>\in S}(a+mb) = (\prod_{p\ prime, p\leq B} p^{\bar{w}_p})^2$$

where T is the set of prime ideals of small norm (e.g. norm less than B) and residue class degree one.

Thus (3) is establishing that $\sigma = \prod_{<a,b>\in S}(a+\alpha b)$ is a singular integer as opposed to the square sought in (1). Notice that finding such a σ requires only that we deal with ideals and not the units and generators used in the number field sieve.

Next, we define two singular integers to be equivalent iff their ratio is a square in K. The set of equivalence classes form an Abelian group with the class containing the squares as identity. This group has exponent 2 and hence is a vector space over F_2. Further we expect that the dimension h of this space will be small. Hence by gathering h singular integers we can be sure that the product of some subset of them will be a square. We will need to find that subset.

To do this we introduce the idea of the 'quadratic signature'. First we will find a short sequence $\mathcal{P}_1, \mathcal{P}_2, ..., \mathcal{P}_z$ of prime ideals which are not in the set T (i.e. not in our 'factor base'). $z = 2h$ is probably sufficient. Let R denote the set of $\gamma \in O_K$ which are relatively prime to $\prod_{i=1}^{z} \mathcal{P}_i$. For all $\gamma \in R$ associate the vector $< r_1, r_2, ..., r_z >$ of zeros and ones such that $r_i = 0$ iff γ is a quadratic residue modulo $\mathcal{P}_i$. This is a homomorphism from R/R^2 onto the z dimensional vector space over F_2. Since we know that some product of the h singular integers is a square, it follows that their quadratic

signatures must be dependent. We will find a dependency and it appears that with high probability this will correspond to a subset of the singular integers whose product is a square. From this we will easily get the desired set of integer pairs S such that:

$$\prod_{<a,b>\in S}(a+\alpha b)=\delta^2 \qquad (4)$$

and

$$\prod_{<a,b>\in S}(a+mb)=(\prod_{p\ prime,p\leq B}p^{\overline{w}_p})^2$$

We will now find δ by taking the square root (in O_K) of $\tau=\prod_{<a,b>\in S}(a+\alpha b)$.

2 Algorithm

To begin the factorization of n, we will select an $m\in Z_{>0}$ such that, in the notation of [5,6], m is an integer near $L_n[2/3,y]$, where $y\in\Re_{>0}$ will be determined later.

Set $d=(1/y)(\log n/\log\log n)^{1/3}$. Then write $n-m^d$ base m as $\sum_{i=0}^{d-1}a_im^i$. Next let $f(x)=x^d+\sum_{i=0}^{d-1}a_ix^i$. Then $f(m)=n$ and $f(1)\leq dm$. This f is as suggested by Buhler and Pomerance in [5]. We can assume f is irreducible (lest we immediately factor n). We will work in the field $K=Q(\alpha)$ where α is a root of f.

Set the smoothness bound $B\in\Re$ to be $L_n[1/3,z]$, where $z\in\Re_{>0}$ will be determined later. It is possible to set two different smoothness bounds, one for the rational primes and one for the prime ideals; however, we will omit this generality here.

PART I

Consider $a,b\in Z_{>0}$ such that both are bounded by $L_n[1/3,(g-1)y^2]$, where $g\in\Re_{>0}$ will be determined later. Proceeding in a manner analogous to that in [6], find sets of integer pairs $S_1,S_2,...,S_h$ (where $h\in Z_{>0}$ will be determined later) such that for $i=1,2,...,h$:

$$\prod_{<a,b>\in S_i}(a+\alpha b)=(\prod_{\mathcal{P}\in T}\mathcal{P}^{\overline{s}_{\mathcal{P},i}})^2 \qquad (5)$$

and

$$\prod_{<a,b>\in S_i}(a+mb)=(\prod_{p\ prime,p\leq B}p^{\overline{w}_{p,i}})^2$$

where T is the set of prime ideals of residue class degree one and norm less than B.

PART II

Find $2h$ prime ideals $\mathcal{P}_i$ of residue class degree one and norm greater than B. Produce the 'quadratic signatures' described above for the singular integers $\sigma_i=\prod_{<a,b>\in S_i}(a+\alpha b)$ and find a dependency. From this, find a set of integer pairs S such that there exists a $\delta\in O_K$:

$$\prod_{<a,b>\in S}(a+\alpha b)=\delta^2 \qquad (6)$$

and

$$\prod_{<a,b>\in S}(a+mb)=(\prod_{p\ prime,p\leq B}p^{\overline{w}_p})^2$$

PART III

Calculate $\tau=\prod_{<a,b>\in S}(a+\alpha b)$ and calculate $\delta=\tau^{1/2}$.

At this point proceed as in [6].

3 Singular Integers

Let σ_1 and σ_2 be singular integers. Define $\sigma_1 \sim \sigma_2$ iff exist γ, $\delta \in O_K$ such that $\gamma^2\sigma_1 = \delta^2\sigma_2$. The set of equivalence classes form an Abelian group with identity the class containing the squares. Further, this group has exponent 2 and hence is a vector space over F_2. Consider the map which takes the class containing σ into the ideal class containing I where $I^2 = (\sigma)$. Then this is an homomorphism onto the 2-part of the ideal class group of O_K with kernel the set of classes which contain a unit. In fact, the kernel is isomorphic to the units modulo their squares. It follows that the dimension of the whole space $h = d_U + d_C$ where d_U is the rank of the unit group of O_K and d_C is the rank of the 2-part of the ideal class group. By Dirichlet's unit theorem, $d_U \leq d$.

When running the algorithm, it is probably best to gather d_U singular integers and proceed to Part II. If no dependency is found then return to Part I and gather another singular integer, etc. Intuition suggests that few returns to Part I would be required. Nonetheless, for the purpose of running time analysis, we now give an upper bound on h.

Using estimates for class numbers [9], and discriminants [8] it is possible to show that for sufficiently large n, $d_C \leq \log_2(H_K) \leq d^6$ (where H_K denotes the class number of the field K). This estimate can no doubt be improved considerably; however, it is sufficient for the purpose of this paper.

Hence we have that for sufficiently large n, $h \leq d^7$.

4 Analysis of Running Time

Our analysis is for sufficiently large n.

For Part I, we must find the sets S_i. Since m is approximately $L_n[2/3, y]$ and $a, b \in Z_{>0}$ are bounded by $L_n[1/3, (g-1)y^2] = L_n[2/3, (g-1)y/d]$, it follows that $N(a+\alpha b) \leq f(1)\max\{a^d, b^d\}$ is bounded by $L_n[2/3, gy]$ and $a + mb$ is bounded by $L_n[2/3, y]$. We will use the assumption in [5,6] that (with regard to being smooth) $am+b$ and $N(a+\alpha b)$ behave as random positive integers below their respective bounds. Since the smoothness bound $B = L_n[1/3, z]$, the probability that $N(a+\alpha b)$ is B-smooth is $L_n[1/3, -(gy/3z)]$ and the probability that $a + mb$ is B-smooth is $L_n[1/3, -(y/3z)]$. Hence the probability that the pair $< a, b >$ will have both $N(a + \alpha b)$ and $a + mb$ B-smooth is $L_n[1/3, -(g+1)y/3z]$. Call such a pair 'good'. There are $L_n[1/3, 2(g-1)y^2]$ pairs. Hence these can be expected to yield $L_n[1/3, 2(g-1)y^2 - ((g+1)y/3z)]$ good pairs. Since $L_n[1/3, z]$ good pairs are needed, we require that $2(g-1)y^2 - ((g+1)y/3z) \geq z$. As in [6] we also have that the time for Part I is $L_n[1/3, 2(g-1)y^2]$.

For Part II, the $2h$ ($\leq 2d^7$ by the previous section) prime ideals $\mathcal{P}_i$ of residue class degree one and norm greater than B can be found when the original ideals in the factor base are found. As in [6] we represent each by a pair of positive rational integers $< p, c_p >$. The p^{th} component of the quadratic signature of $\sigma_i = \prod_{<a,b> \in S_i}(a + \alpha b)$ can be computed by first calculating $\overline{\sigma_i} = \prod_{<a,b> \in S_i}(a + c_p b) \bmod p$, and then calculating the quadratic character of $\overline{\sigma_i}$ in Z/pZ. Since there are less than or equal to

$L_n[1/3, z]$ pairs in S_i, all of the quadratic signatures can be calculated within time $L_n[1/3, z]$. The time needed to find a dependency is negligible. We will assume that with high probability such a dependency will correspond to a square. The time needed to construct the set S is also bounded by $L_n[1/3, z]$. Hence the total time for Part II is at most $L_n[1/3, z]$.

For Part III we first calculate $\tau = \prod_{<a,b>\in S}(a + \alpha b)$. Notice that there are less than or equal to $L_n[1/3, z]$ terms on the right hand side. For all $\zeta \in Z[\alpha]$, $\zeta = \sum_{i=0}^{d-1} a_i \alpha^i$ let ζ' denote $\max\{abs(a_i)\}$. Then a simple analysis shows that τ' has length $L_n[1/3, z]$ and τ can be calculated in time $L_n[1/3, 2z]$ ($L_n[1/3, z]$ if fast integer multiplication [13] is used). To calculate $\delta = \tau^{1/2}$ use, for example, [4] which requires time at most $L_n[1/3, 2z]$ (using fast integer multiplication). Hence the total time required for Part III is at most $L_n[1/3, 2z]$

Thus the total time required is at most the max of $L_n[1/3, 2(g-1)y^2]$ and $L_n[1/3, 2z]$ subject to the constraint that $2(g-1)y^2 - ((g+1)y/3z) \geq z$. Taking $y = (1/3)^{(1/3)} \approx .69$, $g = 3$, $z = 2/3^{2/3} \approx 0.96$ gives a total time of $L_n[1/3, c]$ with $c \approx 1.92 < 2$ as desired.

5 Remarks

Other General Purpose Algorithms:

In [5] an outline is given for modifying the number field sieve to handle general integers. In [6], the authors remark that an heuristic analysis suggests that this version would run in random time

$$e^{(c+o(1))(\log n)^{1/3}(\log\log n)^{2/3}})$$

where $c \approx 2.08$.

More recently D. Coopersmith [1] has added further ideas to those found in [5,6] and here to produce a general purpose factoring algorithm that appears to run in random time

$$e^{(c+o(1))(\log n)^{1/3}(\log\log n)^{2/3}})$$

where $c \approx 1.90$.

Factoring Numbers of Special Form In Practice:

The number field sieve was originally defined for numbers of the form $r^e \pm s$ where e and s are small. In particular, for numbers of this form the original number field sieve appears to run in time

$$e^{(c+o(1))(\log n)^{1/3}(\log\log n)^{2/3}})$$

where $c \approx 1.53$ [6].

The ideas in this paper will not improve the asymptotic running time for factoring such numbers. However, these ideas should make it possible to factor all numbers of the form $f(r)$ where $f \in Z[x]$ has small coefficients and small degree in the time given above. Further, it should be possible to carry out the factorizations in this time even in cases where non-unique factorization domains are encountered. Finally, because it is no longer necessary to find and manipulate units and generators for prime ideals, implementation may be easier.

More details concerning implementation can be found in [5,6] and [1].

Factoring General Numbers In Practice/ Implications For Cryptography:

The fastest previously known general purpose factoring algorithms (see [11]) have an heuristic random running time of

$$e^{(1+o(1))(\log n)^{1/2}(\log\log n)^{1/2}})$$

Hence the new algorithms will be more efficient than the old algorithms once numbers of sufficient size are encountered. Where is the crossover? A naive approach to this question is to ignore the $o(1)$'s in the exponents and calculate the least n for which:

$$e^{(\log n)^{1/2}(\log\log n)^{1/2}}$$

$$\geq$$

$$e^{(1.92)(\log n)^{1/3}(\log\log n)^{2/3}}$$

The answer is $n \approx 10^{123}$. Hence one might expect the new algorithms to overtake the old on numbers of approximately 123 decimal digits. However, this may be far too optimistic because it appears that the $o(1)$'s actually favor the old algorithms. Of course where the cross over actually occurs depends on which algorithms are being used and how they are implemented; nonetheless, it seems possible that the new algorithms may begin to overtake the old ones only on numbers of about 330 decimal digits or so. Our reason for believing this is based on the following (very informal) reasoning:

Roughly speaking, both the old and the new algorithms work in the following way. On input n, 'trial elements' (either numbers or pairs of numbers) are generated; only those trial elements which pass a smoothness test are kept; when sufficiently many trial elements have passed, they are put together to form the desired congruence between squares. On a given input n the algorithm which factors n the fastest will most likely be the one which generates trial elements with the fewest digits.

Now consider the problem of factoring a 330 decimal digit number n using one of the old algorithms. These algorithms generate trial elements which are numbers of value slightly larger than $n^{1/2}$. In our case lets say they are between 165 and 200 decimal digits. Now consider the general purpose number field sieve on the same input. For numbers of 330 digits we work in a field of degree 7 over the rationals. The trial elements which are generated have the form $< a+mb, N(a+\alpha b) >$. The analysis of section 4, shows that $a+mb$ will be approximately 47 decimal digits and $N(a+\alpha b)$ will be approximately 141 decimal digits. Hence the trial elements have lengths (equal to the sum of the lengths of $a+mb$ and $N(a+\alpha b)$) of about 188 decimal digits. Hence in this case, whether the old algorithms or the new algorithm would win the factorization race is too close to call.

The most practical of the old algorithms are currently capable of factoring numbers of up to about 110 decimal digits by using several hundred processors running for several weeks. If the above reasoning is essentially correct, then 110 digits is far below the point where the new algorithms begin to have an advantage. Hence as the new algorithms currently exist, they appear to be of no practical value when the number to be factored is not of special form.

Typically, the security of public-key cryptosystems rests on the difficulty of factoring numbers without the special form that the new algorithms can take advantage of.

Hence, at this point, the new algorithms appear to provide a major step forward in understanding the computational complexity of factoring while providing no practical advantage for those who would wish to attack existing public-key cryptosystems. There are those who would view this as a most agreeable outcome.

Alternatives:

In Part II the prime ideals $\mathcal{P}_i$ need not be taken with residue class degree one and norm greater than B. Selecting them with residue class degree greater than one will insure that the singular integers constructed in Part I will be relatively prime to $\prod_{i=1}^{z} \mathcal{P}_i$

In Part III the square root of σ can probably be obtained more efficiently by using a special purpose algorithm for square roots rather than the general purpose polynomial factoring algorithm of [4]. An improvement of this kind may eliminate the need for fast integer multiplication.

Proving The Running Time:

Below are three assumptions used in arguing that the general purpose number field sieve factors in the time indicated.

1. The assumption that with regard to being smooth $am+b$ and $N(a+\alpha b)$ behave as random positive integers below their respective bounds.

2. The assumption that when the algorithm is run the resulting congruence of squares $x^2 \equiv y^2 \bmod n$ will be non-trivial (i.e. $gcd(x-y,n) \neq 1$ or $gcd(x+y,n) \neq 1$) with high probability.

3. The assumption that with high probability, the dependency among the quadratic signatures found in *PART II* of the algorithm actually corresponds to a subset of the singular integers whose product is a square.

Assumption (i) seems very difficult to justify rigorously. The norms are clearly not random numbers (e.g. a prime which divides such a norm must have a residue class degree one prime above it in O_K). Assumption (ii) also seems quité difficult to prove simply because the complexity of the construction yields no simple picture of the relationship of x to y. Unfortunately, for both of these assumptions, it appears that the best that can be said at this point is that there are no obvious reasons why they should be wrong and in practice (i.e. actual factorizations [6]) they appear to be correct. Assumption (iii) may be provable. The intuitive idea is that a non-square element $\alpha \in O_K$ should be a quadratic non-residue for about 'half' of the prime ideals of O_K. Hence, each subset of the singular integers whose product is not a square would have about 1 chance in 2 of being 'eliminated' by each of the $\mathcal{P}_j$ (randomly chosen from an appropriate set) used for the calculation of quadratic signatures.

References

[1] Coppersmith D., Modifications to the number field sieve, research report, IBM TJ Watson Research Center, Yorktown Heights.

[2] Coppersmith D., Odlyzko A.M. and Schroeppel R., Discrete logarithms in *GF(p)*, Algorithmica, v. 1, 1986, pp 1-15.

[3] Lang S., *Algebraic Number Theory*, Addison-Wesley, Reading, 1970.

[4] Lenstra A.K. Factoring polynomials over algebraic number fields, rep. IW 213/82, Mathematisch Centrum, Amsterdam; Extended abstract is Proceedings Eurocal '83, European computer algebra conference, LNCS 162, 245-254.

[5] Lenstra A.K., Lenstra H.W., Jr., Manasse M.S. and Pollard J.M. The number field sieve. Manuscript.

[6] Lenstra A.K., Lenstra H.W., Jr., Manasse M.S. and Pollard J.M. The number field sieve. *Proc. 22nd STOC*, 1990, pp. 564-572.

[7] Marcos D.A. , *Number Fields*, Springer-Verlag, New York, 1977.

[8] Mahler K. An inequality of the discriminant of a polynomial. *Michigan Math. J.* 11 (1964), 257-262.

[9] Newman M. Bounds for class numbers. *Proc. Sympos. Pure Math., Vol. VIII, pp. 70-77. Amer. Math. Soc., Providence R.I.*, 1965.

[10] Pollard J.M., Factoring with cubic integers, Draft, August 1988.

[11] Pomerance C., Fast rigorous factorization and discrete logarithm algorithms. *Proc of the Japan-US Joint Seminar June 4-6, 1986* In *Discrete Algorithms and Complexity* Ed. Johnson D.S., Nishizeki T., A. Nozaki and H.S. Wilf, Academic Press, Boston, 1987.

[12] Pomerance C. and Wagstaff S.S., Jr. Implementation of the continued fraction integer factoring algorithm. Congress. Numer., v. 37, 1983, pp 99-118.

[13] Schonhage A. and Strassen V. Schnelle multiplikation grosser zahlen. *Computing*, 7, 1971, 281-292.