J. Cryptology (1993) 6: 169–180

Journal of
CRYPTOLOGY
© 1993 International Association for Cryptologic Research

Modifications to the Number Field Sieve

Don Coppersmith
IBM Research Division, T. J. Watson Research Center,
Yorktown Heights, NY 10598, U.S.A.

Communicated by Andrew M. Odlyzko

Received 19 November 1990 and revised 4 November 1992

Abstract. The Number Field Sieve, due to Lenstra *et al.* [LLMP] and Buhler *et al.* [BLP], is a new routine for factoring integers. We present here a modification of that sieve. We use the fact that certain smoothness computations can be reused, and thereby reduce the asymptotic running time of the Number Field Sieve. We also give a way to precompute tables which will be useful for factoring any integers in a large range.

Key words. Factoring, Sieve methods.

1. Introduction

The Number Field Sieve, due to Lenstra *et al.* [LLMP] and Buhler *et al.* [BLP], with contributions from Adleman [A], is a new routine for factoring integers. Currently its running time is estimated at

$$L[\tfrac{1}{3}, 1.923] = e^{(1.923+o(1))(\log N)^{1/3}(\log\log N)^{2/3}},$$

where N is the number to be factored. We begin in Section 2 with a brief description of the sieve. In Section 3 we describe a modification which reuses the computations of the initial sieve. This modification reduces the exponent in the above expression from 1.923 to 1.902. In Section 4 we use the same ideas to describe a way to precompute tables which will be useful in factoring any integers in a large range; after the precomputation, an individual integer can be factored in time $L[\frac{1}{3}, 1.639]$.

Notation

We follow the traditional notation:

$$L[\alpha, \beta] \stackrel{\text{def}}{=} L_N[\alpha, \beta] \stackrel{\text{def}}{=} e^{(\beta+o(1))(\log N)^{\alpha}(\log\log N)^{1-\alpha}}, \qquad 0 \le \alpha \le 1.$$

The expression $o(1)$ in the exponent denotes a quantity tending to 0 as $N \to \infty$. It should be emphasized that this $o(1)$ hides a lot; this notation is only intended as a first-order approximation to the real computational complexity.

An integer z is said to be *smooth with respect to the bound* y (or *y-smooth*) when all the prime factors of z are bounded by y. The symbol $\Psi(x, y)$ denotes the number of integers z in the range $1 \le z \le x$ which are y-smooth. If $x = L[a, b]$ and $y = L[c, d]$ with $a > c$ and $b, d > 0$, then we have

$$\frac{1}{x}\Psi(x, y) = 1/L[a - c, b(a - c)/d].$$

See, for example, [CEP] and [D]. This can be viewed as the probability that an integer z chosen uniformly at random from the range $1 \le z \le x$ is y-smooth. We generate integers z in the range $1 \le z \le x$ by another process, not uniformly, but we use the same probability estimate, so that all of the running time estimates we obtain are only heuristics.

2. The Number Field Sieve

Here we give a brief description of the Number Field Sieve as developed by Buhler *et al.* [BLP] for general integers; this was predated by a similar sieve by Lenstra *et al.* [LLMP] for integers of a special form. This description is based on notes from a lecture of H. W. Lenstra in October 1990.

Given an integer N which we wish to factor, we select an integer

$$d \approx \delta\left(\frac{\log N}{\log\log N}\right)^{1/3}, \qquad \delta = 3^{1/3}$$

(where "$\approx$" denotes approximate equality, in this case rounding to the nearest integer), and an integer

$$m \approx N^{1/d} = L\left[\tfrac{2}{3}, \frac{1}{\delta}\right].$$

Next we find a monic polynomial $f(\tau)$ of degree d, with coefficients bounded by about m, such that

$$f(m) = 0 \pmod N, \qquad f(m) \neq 0,$$

$$f(\tau) = \sum_{i=0}^{d} c_i \tau^i, \qquad |c_i| \le L\left[\tfrac{2}{3}, \frac{1}{\delta}\right].$$

A convenient choice for f is the base-m representation of N:

$$N = \sum c_i m^i, \qquad 0 \le c_i < m.$$

For most N, substantially smaller values for (m, c_i) than these cannot be found, because of a counting argument.

Remark. For earse of presentation we have assumed that f is monic, although the case where f is not monic presents no real difficulties. Similarly, the root $m \in \mathbb{Z}$ of f mod N could just as easily be replaced by $p/q \in \mathbb{Q}$, in the sense that $N | q^d f(p/q) = \sum c_i p^i q^{d-i}$. Both generalizations are treated in [BLP].

In what follows we assume that f is irreducible, as is usually the case; if not, a factorization of f will probably yield a factorization of N.

Consider the number field

$$K = \frac{\mathbb{Q}[\tau]}{(f(\tau))} = \mathbb{Q}(\alpha),$$

and let $\mathbb{Z}[\alpha] = \mathbb{Z}[\tau]/(f(\tau))$ be the subring generated by α, a root of f. The description of the algorithm that follows is only correct if we assume that $\mathbb{Z}[\alpha] = O_K$, the ring of integers in K. For the changes that have to be made if this is not assumed, we refer the reader to [BLP]. These changes are irrelevant if it is only wished to understand the difference between the original number field sieve and the present modification.

Define the homomorphism φ from $\mathbb{Z}[\alpha]$ to the ring $\mathbb{Z}/N\mathbb{Z}$ by $\varphi(\alpha) = m$.

Define two smoothness bounds,

$$B_1 = L[\tfrac{1}{3}, \beta], \qquad \beta = \frac{2}{3^{2/3}} \approx 0.961,$$

$$B_2 = L[\tfrac{1}{3}, \gamma], \qquad \gamma = \frac{2}{3^{2/3}} \approx 0.961,$$

and a coefficient range

$$E = L[\tfrac{1}{3}, \varepsilon], \qquad \varepsilon = \frac{2}{3^{2/3}} \approx 0.961.$$

(It happens here that $B_1 = B_2 = E$ in the original Number Field Sieve, but they are unequal in our modification.) Now sieve through pairs of integers (a, b) in the range $|a| < E$, $|b| < E$, to find pairs satisfying

$$\text{g.c.d.}(a, b) = 1,$$

$$(a - mb) \text{ is } B_1\text{-smooth},$$

$$\text{Norm}(a - \alpha b) \text{ is } B_2\text{-smooth},$$

where

$$\text{Norm}(a - \alpha b) = b^d f\left(\frac{a}{b}\right) = \sum c_i a^i b^{d-i}.$$

Notice that

$$|a - mb| \lesssim mE = L\left[\tfrac{2}{3}, \frac{1}{\delta}\right],$$

the contribution of E being negligible and accounted for by the $o(1)$, and

$$|\text{Norm}(a - \alpha b)| \lesssim dmE^d = L\left[\tfrac{2}{3}, \frac{1}{\delta}\right] L[\tfrac{2}{3}, \varepsilon\delta] = L\left[\tfrac{2}{3}, \varepsilon\delta + \frac{1}{\delta}\right].$$

The number of pairs (a, b), satisfying both smoothness conditions and the g.c.d.

condition, which we expect to find, is about

$$E^2 \frac{1}{mE} \Psi(me, B_1) \frac{1}{dmE^d} \Psi(dmE^d, B_2) = \frac{L[\frac{1}{3}, 2\varepsilon]}{L\left[\frac{1}{3}, \frac{1}{3\delta\beta}\right] L\left[\frac{1}{3}, \frac{\varepsilon\delta + 1/\delta}{3\gamma}\right]}$$
$$= L\left[\frac{1}{3}, 2\varepsilon - \frac{1}{3\delta\beta} - \frac{\varepsilon\delta}{3\gamma} - \frac{1}{3\delta\gamma}\right].$$

For the particular values given above, this number is

$$L\left[\frac{1}{3}, \frac{2}{3^{2/3}}\right] \approx B_1.$$

Next we build a matrix M over $GF(2)$, whose rows are indexed by these satisfying pairs (a, b) and whose columns fall into three categories:

- $\pi(B_1) + 1$ columns indexed by integer primes $p < B_1$, including, for convenience, $p = -1$ as a special case.
- About $\pi(B_2)$ columns indexed by pairs (p, c), where p is a prime integer less than B_2 and c is a residue mod p such that $f(c) = 0 \bmod p$, so that $(p, c - \alpha)$ is a prime ideal in $\mathbb{Z}[\alpha]$.
- $O(\log N)$ columns (Adleman's "character columns") indexed by selected pairs (p, c) as above, but with $p > B_2$.

In the row corresponding to the pair (a, b) and the column corresponding to p we place the parity of the power of p dividing $a - mb$; in the column corresponding to the pair (p, c) we place the parity of the power of p dividing $\text{Norm}(a - \alpha b)$ if $a - cb = 0 \bmod p$, and 0 otherwise (this is also the power of the ideal $(p, c - \alpha)$ dividing the ideal $(a - \alpha b)$ in $\mathbb{Z}[\alpha]$); and in the character column corresponding to (p, c) we place 1 if $a - cb$ is a quadratic nonresidue mod p and 0 if it is a residue.

The resulting matrix M is roughly square, with side

$$\pi(B_1) + \pi(B_2) + O(\log N) < B_1.$$

In fact, the original choice of parameters (specifically, the $o(1)$ components in the L expressions defining B_1, B_2, E) should be such that M has a few more rows than columns. Also, M is sparse, with $O(\log N)$ nonzero entries per row.

The next step in the solution of a sparse system of linear equations is to find a linear combination of the rows of M which vanishes over $GF(2)$. Since M is sparse, this can be done in time about B_1^2. See, for example, [W] or [Co]; the latter paper speeds up the Wiedemann algorithm by a factor of 32 by partial parallelization.

This linear combination represents a subset S of the pairs (a, b). The product $\prod(a - mb)$ taken over S contains each small prime p to an even power (since the corresponding exponents sum to 0 mod 2), so that $\prod(a - mb)$ is a square of an integer, say

$$\prod_S (a - mb) = y^2.$$

On the other hand, the ideal generated by the product $\prod(a - \alpha b)$ is the square of an ideal, since each prime ideal $(p, c - \alpha)$ corresponding to a small prime $p < B_2$

shows up to an even power. The cancellation of the entries in the character column make us expect that this ideal is principal, generated by some $g(\alpha) \in \mathbb{Z}[\alpha]$, so that we have (as ideals)

$$\left(\prod (a - \alpha b)\right) = (g(\alpha))^2,$$

or, for some unit u, we have an equation on elements:

$$\prod (a - \alpha b) = g(\alpha)^2 u.$$

The cancellation of the entries in the character columns gives us hope that the unit u is in fact a square, so incorporating its square root into $g(\alpha)$ we have

$$\prod (a - \alpha b) = g(\alpha)^2.$$

To compute the element $g(\alpha)$, we can select a prime q which is inert for f (such a q should not be hard to find), calculate an approximation $g_0(\tau)$ satisfying

$$\prod (a - \tau b) = g_0(\tau)^2 \mod(f(\tau), q)$$

with the Berlekamp factoring algorithm, and do Hensel lifting: find successive approximations $g_k(\tau)$ satisfying

$$\prod (a - \tau b) = g_k(\tau)^2 \mod(f(\tau), q^{2^k}),$$

until k is sufficiently large that we have found g satisfying

$$\prod (a - \tau b) = g(\tau)^2 \mod f(\tau).$$

We then take the image $x \equiv \varphi(g) = g(m) \bmod N$, and obtain the sequence of congruences

$$x^2 \equiv g(m)^2 \mod N,$$

$$g(m)^2 \equiv \prod_S (a - mb) \mod f(m),$$

$$\prod_S (a - mb) = y^2.$$

Because $f(m)$ is a multiple of N, this yields

$$x^2 \equiv y^2 \mod N.$$

As in the Fermat method, we then take

$$\text{g.c.d.}(N, y - x)$$

and hope to find a nontrivial factor of N.

Running Time

The sieve takes time

$$O(E^2 \log\log B_1) = L[\tfrac{1}{3}, 2\varepsilon] = L\left[\tfrac{1}{3}, \frac{4}{3^{2/3}}\right] \approx L[\tfrac{1}{3}, 1.923].$$

The solution of sparse linear equations takes time

$$B_1^2 = L[\tfrac{1}{3}, 2\beta] \approx L[\tfrac{1}{3}, 1.923].$$

The computation of $g(\alpha)$ and $g(m)$ takes time about $B_1 \log B_1$ with fast multiplication techniques, or B_1^2 with traditional ones. So the sieve and the solution of linear equations are asymptotically the most time-consuming operations.

3. Reusing the Computation

Our first modification is based on the idea that several polynomials f will work with the same pair of integers m and N. We can select any convenient integer m of the right magnitude, say $m = 2^{80}$, and select a range E and a smoothness bound B_1, and then sieve all pairs (a, b) in the range $|a|, |b| < E$ to find pairs satisfying

$$\text{g.c.d.}(a, b) = 1,$$

$$(a - mb) \text{ is } B_1\text{-smooth.}$$

Call the resulting pairs the *Stage-1 pairs*. Now, given N, select several different polynomials f of degree d with coefficients bounded by about $N^{1/d}$ such that $f(m) = 0 \bmod N$. The number of different polynomials we want is

$$\frac{B_1'}{B_2'} = L[\tfrac{1}{3}, \beta - \gamma],$$

where

$$B_1' = \pi(B_1) + 1$$

and

$$B_2' = \pi(B_2) + O(\log N).$$

(The notation is chosen so that B_j and B_j' agree up to the $o(1)$ components of the L expressions defining them.)

We can select one polynomial $f(\tau) \stackrel{\text{def}}{=} f_0(\tau)$ by taking the coefficients from the canonical expression of N in base m:

$$N = \sum c_i m^i, \qquad 0 \le c_i < m,$$

and obtain other polynomials $f_i(\tau)$ as

$$f_i(\tau) = f_0(\tau) + i(\tau - m), \qquad 0 \le i \le \frac{B_1'}{B_2'}.$$

The bound on the coefficients c_i needs to be relaxed somewhat, to mB_1'/B_2', but this will not affect the first-order estimate of the running time. For each polynomial f, we run through all the Stage-1 pairs to find those for which

$$\text{Norm}_f(a - \alpha b) \text{ is } B_2\text{-smooth.}$$

To each surviving pair (a, b) attach the polynomial f explicitly, to give a *Stage-2 triple* (a, b, f).

Caveat. We cannot sieve this time, but must run the smoothness tests individually, since the pairs (a, b) are irregularly spaced. We can use the elliptic curve smoothness test due to Lenstra [L], which again will not increase the first-order estimate of

running time, but in practice this test is more expensive than the sieve, with asymptotic cost $L[\frac{1}{6}, \ldots]$ as opposed to $O(\log \log N)$ per integer in the sieve. See [P] for examples of such use of the elliptic curve method as an auxiliary procedure for integer factorization.

By careful choice of parameters B_1, B_2, E, d (in particular, choosing the values of the $o(1)$ in the $L[\frac{1}{3}, \ldots]$ in their definitions), we arrange things so that at this point in the computation, for each polynomial f, there are about $2B_2'$ Stage-2 triples (a, b, f). We build a matrix M of size about $2B_1' \times 2B_1'$. Here $B_1' = \pi(B_1) + 1$ of the columns are for integer primes $p < B_1$ as before. For each f there are about $B_2' = \pi(B_2) + O(\log N)$ columns devoted to first-degree primes in $\mathbb{Z}[\alpha]$ and character columns. There will be B_1'/B_2' blocks of such columns, or about B_1' columns in all. Each row corresponds to a triple (a, b, f). Nonzero entries in a given row will be in the block of B_2' columns devoted to its own f, and in the B_1' columns devoted to the integer primes p. Figure 1 illustrates the layout of this matrix.

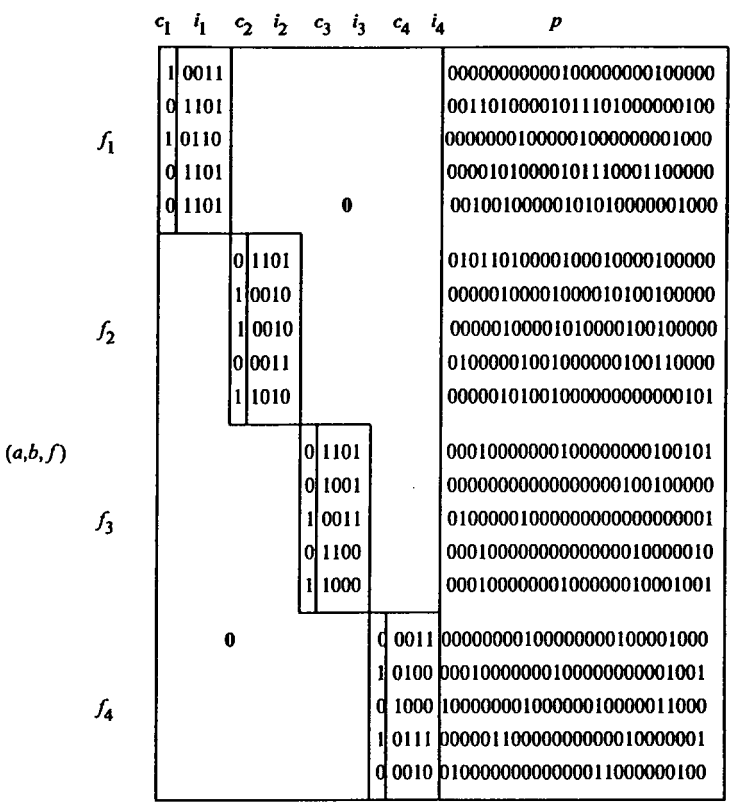

Fig. 1. Layout of modified matrix. f_j: polynomial defining the jth field; (a, b, f_j): row corresponding to Stage-2 triple; c_j: character columns for f_j; i_j: ideals in $\mathbb{Z}(\alpha_j)$; p: primes in $\mathbb{Z}$; **0**: blocks of zeros.

When the matrix is filled, we solve a sparse system of linear equations as before, finding a linear combination of rows that vanishes mod 2. Let S denote the corresponding set of Phase-2 triples (a, b, f), and let S_f be the subset of S belonging to a particular polynomial f.

For each f we find an element $g_f(\alpha)$ so that

$$\prod_{S_f} (a - \alpha b) = g_f(\alpha)^2,$$

that is,

$$\prod_{S_f} (a - \tau b) \equiv g_f(\tau)^2 \mod f(\tau).$$

(We require that, for each of $L[\frac{1}{3}, \ldots]$ values of f, the product

$$\prod_{S_f} (a - \alpha b)$$

has a square root in the corresponding ring. To this end, we increase the number of character columns, so that, for each f, we can bound by $O(1/L[\frac{1}{3}, \ldots])$ the probability that such a square root $g_f(\alpha)$ fails to exist.)

Define the homomorphism φ_f from $\mathbb{Z}[\alpha] = \mathbb{Z}[\tau]/(f(\tau))$ to $\mathbb{Z}/N\mathbb{Z}$ by

$$\varphi_f(\alpha) = m,$$

noticing that m is independent of f, and compute the element

$$x_f \equiv \varphi_f g_f(\alpha) \equiv g_f(m) \mod N,$$

so that

$$x_f^2 \equiv \prod_{S_f} (a - mb) \mod N.$$

Now compute

$$x = \prod_f x_f.$$

As before,

$$\prod_S (a - mb)$$

is the square of an integer, say

$$\prod_S (a - mb) = y^2,$$

where this product is over all of S. As before, we obtain

$$x^2 \equiv y^2 \mod N$$

and hope that

$$\text{g.c.d. } (N, y - x)$$

gives a nontrivial factor of N.

Parameters and Running Time

The number of pairs (a, b) surviving the first sieve (Stage-1 pairs) is about

$$E^2 \frac{1}{mE} \Psi(mE, B_1) = \frac{L[\frac{1}{3}, 2\varepsilon]}{L[\frac{1}{3}, 1/3\delta\beta]} = L\left[\tfrac{1}{3}, 2\varepsilon - \frac{1}{3\delta\beta}\right].$$

For each f, the number surviving the second smoothness test (Stage-2 triples) is

$$L\left[\tfrac{1}{3}, 2\varepsilon - \frac{1}{3\delta\beta} - \frac{\varepsilon\delta}{3\gamma} - \frac{1}{3\delta\gamma}\right].$$

We need these to number at least

$$2B_2' = 2L[\tfrac{1}{3}, \gamma] = L[\tfrac{1}{3}, \gamma],$$

where the factor of 2 is absorbed into the $o(1)$ in the definition of L. This gives rise to the condition

$$2\varepsilon - \frac{1}{3\delta\beta} - \frac{\varepsilon\delta}{3\gamma} - \frac{1}{3\delta\gamma} \geq \gamma. \qquad (*)$$

The sieve takes time

$$O(E^2 \log\log(B_1)) = L[\tfrac{1}{3}, 2\varepsilon].$$

The later smoothness tests take time

$$\frac{B_1'}{B_2'} L\left[\tfrac{1}{3}, 2\varepsilon - \frac{1}{3\delta\beta}\right] = L\left[\tfrac{1}{3}, \beta - \gamma + 2\varepsilon - \frac{1}{3\delta\beta}\right].$$

To solve the system of linear equations takes time

$$B_1^2 = L[\tfrac{1}{3}, 2\beta].$$

Finding the square roots $g(\alpha)$ takes time $(B_1/B_2) \times B_2^c$ where $c = 1$ or 2, depending on the algorithm used, so that it should be bounded by

$$B_1 B_2 = L[\tfrac{1}{3}, \beta + \gamma].$$

We find that the choice of

$$\varepsilon = \beta = \left(\frac{46 + 13\sqrt{13}}{108}\right)^{1/3} \approx 0.95094,$$

$$\gamma = \varepsilon \times \left(\frac{\sqrt{13} - 1}{3}\right) \approx 0.82591,$$

$$\delta = \varepsilon \times \left(\frac{4\sqrt{13} - 10}{3}\right) \approx 1.40175$$

minimizes the total asymptotic running time subject to the condition (*), giving a total running time of

$$L[\tfrac{1}{3}, 2\varepsilon] \approx L[\tfrac{1}{3}, 1.902].$$

The major contributions to this running time are the sieve and the solution of the system of linear equations. The later smoothness tests and square-root calculations are asymptotically less time-consuming.

4. The Factorization Factory

Our second modification is based on the idea that m can be chosen independently of N. It entails a massive precomputation, during which all pairs (a, b) of integers bounded by E are searched for smoothness of $a - mb$ with respect to B_1, with the

results placed into a permanent file. Subsequently, to factor an individual N we use this file instead of the first stage of sieving. We can ask for a tradeoff between the running time of the precomputation and the running time for an individual value of N.

Our selection of values, minimizing the per-number time at the expense of the precomputation, would be

$$\beta = \gamma = \left(\frac{5 + 2\sqrt{6}}{18}\right)^{1/3} \approx 0.8193,$$

$$\delta = \beta \times (3\sqrt{6} - 6) \approx 1.1048,$$

$$\varepsilon = \beta \times \left(\frac{\sqrt{6}}{2}\right) \approx 1.0034,$$

giving an initial computation cost of

$$L[\tfrac{1}{3}, 2\varepsilon] \approx L[\tfrac{1}{3}, 2.0068],$$

a storage cost of

$$L[\tfrac{1}{3}, 2\beta] \approx L[\tfrac{1}{3}, 1.6386],$$

and a cost per factored integer N of

$$L[\tfrac{1}{3}, 2\beta] \approx L[\tfrac{1}{3}, 1.6386].$$

Since $\beta = \gamma$ we are only using one polynomial f for a given N here. The savings is coming entirely from the precomputed table.

We can select $d = \lceil \log_m N \rceil$, and determine the coefficients of f from the base-m expansion of the integer $\lceil m^d/N \rceil \times N$, so that $f(m) \equiv 0 \bmod N$, as required. This will also give a monic polynomial f; if we do not insist on f being monic, $d = \lfloor \log_m N \rfloor$ will suffice.

This precomputation will suffice for a very large range of values of N, say

$$N_0 < N < N_0 \times N_0^{1/d} = N_0 \times L\left[\tfrac{2}{3}, \frac{1}{\delta}\right].$$

This is because the choice of m and E depends only on the rough magnitude of N.

Schnorr's Scheme

Several years ago Schnorr [S] proposed a factorization scheme with a similar purpose: a large precomputation, after which any number N in a narrow range $|N - N_0| < L[\frac{2}{3}, \varepsilon]$ could be factored easily. His scheme can be viewed as an extreme special case of the present algorhtm. Its cost per individual factorization N is similar to ours, though the other performance characteristics are quite different.

For Schnorr's scheme, let

$$m = N_0,$$

$$d = 1,$$

$$f(\alpha) = \alpha + (N - N_0),$$

$$|a| < E = L[\tfrac{2}{3}, \varepsilon],$$

$$b = -1,$$

$$B_1 = B_2 = L[\tfrac{1}{3}, \beta].$$

The precomputation involves searching for B_1-smooth integers of the form

$$a - mb = a + N_0,$$

where

$$|a| < L[\tfrac{2}{3}, \varepsilon].$$

The smoothness probability in that range is

$$1/L\left[\tfrac{2}{3}, \frac{2}{3\beta}\right],$$

so that as long as

$$\varepsilon > \frac{2}{3\beta}$$

we will find enough smooth numbers (Stage-1 pairs), say

$$L\left[\tfrac{1}{3}, \beta + \frac{\varepsilon}{3\beta}\right]$$

of them.

After this precomputation, we are asked to factor a particular integer N, which is now restricted to the narrow range

$$|N - N_0| < L[\tfrac{2}{3}, \varepsilon].$$

From

$$f(\alpha) = \alpha + (N - N_0)$$

we obtain

$$\mathrm{Norm}_f(a - \alpha b) = c_1 a + c_0 b = (a + N_0) - N.$$

These numbers are bounded by

$$L[\tfrac{2}{3}, \varepsilon],$$

so they will be B_1-smooth with probability

$$1/L\left[\tfrac{1}{3}, \frac{\varepsilon}{3\beta}\right].$$

Arrange parameters so that at least

$$B_1 = L[\tfrac{1}{3}, \beta]$$

of the values $(a + N_0 - N)$ are smooth; these become our Stage-2 triples $(a, -1, f)$. The rest of the scheme is similar to the present one.

Schnorr's scheme is simpler than the present one; because f is linear, the relevant number field is just $\mathbb{Q}$. His precomputation is $L[\frac{2}{3}, \varepsilon]$, compared with our $L[\frac{1}{3}, *]$. After the precomputation, his scheme works over a narrow range of N:

$$|N - N_0| < L[\tfrac{2}{3}, \varepsilon],$$

while ours works for a much larger range, say

$$N_0 < N < N_0 \times L\left[\tfrac{2}{3}, \frac{1}{\delta}\right].$$

This is because his polynomials f have only one free coefficient $c_0 = N - N_0$, while ours have d free coefficients.

However, it is remarkable that the time to factor an individual integer N is $L[\frac{1}{3}, *]$ in both his scheme and the present one, because the size of $\text{Norm}_f(a - \alpha b)$ is $L[\frac{2}{3}, *]$ in both cases.

5. Conclusions

Starting from the Number Field Sieve, we have made two observations: that m can be chosen independently of N, and that many different polynomials f can be chosen for the same values of N and m. This allows us to precompute a list of smooth integers of the form $(a - mb)$. This list is then used in two different ways.

First, if we count the precomputation as part of the factorization cost, we can used several different polynomials f, thereby decreasing the size E of the integers a, b, and improving the smoothness probability. The net result is a slight decrease in the running time, for $L[\frac{1}{3}, 1.923]$ to $L[\frac{1}{3}, 1.902]$.

Second, if we amortize the cost of the precomputation over many factorizations N, we can select different parameters (in particular, only one polynomial f for each N to factored), and achieve a substantial decrease in the time to factor an individual integer, from $L[\frac{1}{3}, 1.923]$ to $L[\frac{1}{3}, 1.639]$. This can have implications for cryptanalysis.

Acknowledgment

Two anonymous referees have been of considerable help, particularly in eliminating sloppiness from the original description of the algorithm.

References

[A] L. M. Adleman, Factoring numbers using singular integers, *Proc. 23rd Annual ACM Symposium on the Theory of Computing*, 1991, pp. 64–71.

[BLP] J. P. Buhler, H. W. Lenstra, Jr., and C. Pomerance, *Factoring Integers with the Number Field Sieve*, Springer-Verlag, Berlin, Lecture Notes in Mathematics, to appear.

[CEP] E. R. Canfield, P. Erdös, and C. Pomerance, On a problem of Oppenheim concerning "Factorisatio Numerorum", *J. Number Theory* **17** (1983), 1–28.

[Co] D. Coppersmith, Solving Linear Equations over GF(2) II: Block Wiedemann Algorithm, Research Report RC 17293, IBM T. J. Watson Research Center, Yorktown Heights, NY, 17 October 1991. To appear in *Math. Comput.*

[D] N. G. de Bruijn, On the number of positive integers $\leq x$ and free of prime factors $> y$, II, *Nederl. Akad. Wetersch. Indag. Math.* **38** (1966), 239–247.

[L] H. W. Lenstra, Jr., Factoring integers with elliptic curves, *Ann. of Math.* **126** (1987), 649–673.

[LLMP] A. K. Lenstra, H. W. Lenstra, Jr., M. S. Manasse, and J. M. Pollard, The number field sieve, *Proc 22nd Annual ACM Symposium on the Theory of Computing*, 1990, pp. 564–572.

[P] C. Pomerance, Fast, rigorous factorization and discrete logarithm algorithms, in: D. S. Johnson, T. Nishizeki, A Nozaki, and H. S. Wilf (eds), *Discrete Algorithms and Complexity*, Academic Press, Orlando, FL, 1987, pp. 119–143.

[S] C. P. Schnorr, Refined analysis and improvements on some factoring algorithms, *J. Algorithms* **3** (1982), 101–127.

[W] D. H. Wiedemann, Solving sparse linear equations over finite fields, *IEEE Trans. Inform. Theory* **32** (1986), 54–62.