An implementation of the general number field sieve

J. Buchmann J. Loho J. Zayer

Extended abstract

Fachbereich Informatik
Universität des Saarlandes
66041 Saarbrücken
Germany

Abstract. It was shown in [2] that under reasonable assumptions *the general number field sieve* (GNFS) is the asymptotically fastest known factoring algorithm. It is, however, not known how this algorithm behaves in practice. In this report we describe practical experience with our implementation of the GNFS whose first version was completed in January 1993 at the Department of Computer Science at the Universität des Saarlandes.

1 Introduction

Factoring rational integers into primes is one of the most important and most difficult problems of computational number theory. It was shown in [2] that under reasonable assumptions *the general number field sieve* (GNFS) is the asymptotically fastest known factoring algorithm. It is, however, not known how this algorithm behaves in practice. In this report we describe practical experience with the first version of our implementation of the GNFS. For our implementation we used the methods described in [2], [3], and [7]. In the course of the implementation we have found several improvements which we will describe in the full version of this paper. In this extendend abstract we restrict ourselves to the presentation of a brief sketch of the algorithm and the numerical results.

2 The GNFS

Let $n \in \mathbb{N}$. If one can find two integers x and y with

$$x^2 \equiv y^2 \text{ modulo } n \tag{1}$$

and $x \not\equiv \pm y$ modulo n, then $\gcd(x-y, n)$ is a non trivial divisor of n. Like many other factoring algorithms the GNFS factors n by producing such a pair x, y. This is done in the following way: Let $f(x) = f_0 + f_1 \cdot x + \ldots + f_{d-1} \cdot x^{d-1} + x^d \in \mathbb{Z}[x]$ be an irreducible polynomial for which there exits $m \in \mathbb{Z}$ with $f(m) \equiv 0$ modulo

D.R. Stinson (Ed.): Advances in Cryptology - CRYPTO '93, LNCS 773, pp. 159-165, 1994.

n. Let ρ be a zero of $f(x)$. The algorithm determines a non-empty set S of pairs (a, b) of relatively prime integers with the following properties

$$X = \prod_{(a,b)\in S} (a + bm) = x^2 \text{ with } x \in \mathbb{Z} \tag{2}$$

$$\gamma = \prod_{(a,b)\in S} (a + b\rho) = \delta^2 \text{ with } \delta \in \mathbb{Z}[\rho] \tag{3}$$

The map $\varphi : \mathbb{Z}[\rho] \to \mathbb{Z}/n\mathbb{Z}$, $\rho \mapsto m \bmod n$ is a ring homomorphism. Therefore we have $x^2 \equiv \varphi(\delta^2) \equiv \varphi(\delta)^2 \bmod n$. If we set $y = \varphi(\delta)$ then we have found a congruence of the form (1) which with high probability yields a factorization of n.

The algorithm can thus be divided into three parts: determining the polynomial, finding the squares and extracting the square roots. In the remaining sections we describe our implementation of those parts and we give numerical examples. For background and details we refer to [2], [3] and [7].

3 Determining the polynomial

The first step of GNFS is to find an irreducible polynomial $f(x) \in \mathbb{Z}[x]$ of degree d and a rational integer m, such that $f(m) \equiv 0 \bmod n$. For $n \leq 10^{60}$ we use $d = 3$ and for $10^{60} < n < 10^{180}$ we use $d = 5$. We choose $i \in \mathbb{Z}$ such that for $m = \lfloor n^{\frac{1}{d}} \rfloor + i$ there is an expansion $n = m^d + f_{d-1}m^{d-1} + \ldots + f_1 m + f_0$ with $-m/2 \leq f_j < m/2$. We determine that expansion and we set $f(x) = x^d + f_{d-1}x^{d-1} + \ldots + f_1 x + f_0$. There are various ways of modifying f. We can, for example, replace f by $f + \sum_{j=1}^{d-1} c_j(x^j - mx^{j-1})$. It is still an open question how an optimal polynomial f can be found. We intend to use our implementation of the GNFS to study this question in detail. A few remarkable experimental results can be found in section 6.

4 Finding the squares

To find the set S of coprime pairs $(a, b) \in \mathbb{Z}^2$ satisfying (2) and (3) we use the *standard sieve* which is described in [2] or the lattice sieve which was suggested in [7].

In both algorithms we must choose two factor bases. The *rational factor base* F_R is the set of all rational primes below some bound $s_R \in \mathbb{R}_{>0}$. The *algebraic factor base* is the set F_A of all degree one prime ideals of $\mathbb{Z}[\rho]$ of norm below $s_A \in \mathbb{R}_{>0}$. The values for s_R and s_A are chosen according to experimental experience. Each prime in F_A is represented by a pair (p, c_p) where c_p is a zero of f modulo p. We also need large prime bounds L_R and L_A which are roughly $100 \cdot s_R$ or $100 \cdot s_A$, respectively.

To apply the standard sieve, we fix bounds $A, B \in \mathbb{Z}_{>0}$ on a and b, respectively. Again those values are chosen according to experimental experience. For each $b \in \{1, 2, \ldots, B\}$ we determine all a with $-A < a < A$ such that $\gcd(a, b) = 1$, all of the prime factors of $a + bm$ except for at most one factor $l_R(a, b)$ belong to F_R and all of the prime ideal factors of $(a + b\rho)\mathbb{Z}[\rho]$ except for at most two factors $l_{A,1}(a, b)$ and $l_{A,2}(a, b)$ belong to F_A. Also, the extra rational prime factors are called *large rational primes* and they must be below L_R. Analogously, the extra algebraic prime factors are called *large algebraic primes* and their norms must be below L_A. Any such pair (a, b) is called a *good pair*. We say that a good pair without large primes is of *type fff*, if there is a large rational prime it is of *type pff*. The definition of the *types fpf, fpp, ppf* and *ppp* is analogous. For a more detailed description of the sieve algorithm see [2].

To use the lattice sieve we divide the factor bases into two parts. The set $F_{r,s}$ of *small rational primes* contains all elements of F_R no larger than s_R/t where t may be chosen between 2 and 10. The set $F_{r,m}$ of *medium primes* is the complement of $F_{r,s}$ in F_R. For $q \in F_{r,m}$ the set $LR_q = \{(a, b) : q | a + bm\}$ is a two dimensional lattice in $\mathbb{Z}^2$. If $(\underline{u}, \underline{v})$ is a basis of LR_q then one can find good pairs (a, b) whose small primes are bounded by q by inspecting the vectors $c\underline{u} + d\underline{v}$ for $c, d \in \mathbb{Z}$, $-C < c < C$, $0 < d < D$ where $C \in \mathbb{R}_{>0}$ and $D \in \mathbb{R}_{>0}$ are chosen according to experimental experience. For any fixed d this can be done by a sieving procedure which is described in [7]. In this procedure we take advantage of the following fact: For $p \geq 2C$ and $d \in \{1, \ldots, D\}$ there is exactly one c_d such that p is a divisor of $a + bm$ for $(a, b) = c_d\underline{u} + d\underline{v}$ and $-p/2 \leq c_d < p/2$. Since $c_d = c_{d-1} + c_1 \bmod p$ those numbers can be very easily computed. It is even possible to determine the interesting values of c_d for which $-C < c_d < C$ immediately. This leads to a significant speed up of the lattice sieve. A similiar trick can be applied to find $a + b\rho$ which factors up to large primes over F_A.

Once sufficiently many good pairs are found, we determine for each good pair (a, b) the decompositions $a + bm = l_R(a, b) \cdot \prod_{p \in F_R} p^{e_p(a,b)}$ and $(a + b\rho)\mathbb{Z}[\rho] = l_{A,1}(a, b) \cdot l_{A,2}(a, b) \cdot \prod_{P \in F_A} P^{e_P(a,b)}$, where $l_R(a, b), l_{A,1}(a, b)$ and $l_{A,2}(a, b)$ also may be 1. We also determine a small set F_Q of degree one prime ideals of $\mathbb{Z}[\rho]$ of norms bigger than L_A and for each $Q \in F_Q$ we set $e_Q(a, b) = 0$ if $a + b\rho$ is a square in $\mathbb{Z}[\rho]/Q$ and $e_Q(a, b) = 1$ otherwise. The large primes are handled by constructing cycles as discribed in [1] and [6]. By calculating a non trivial linear dependency among the vectors $((e_p(a, b))_{p \in F_R}(e_P(a, b))_{P \in F_A}(e_Q(a, b))_{Q \in F_Q})$ over $\mathbb{F}_2$ we determine the subset S of the set of all pairs (a, b) that we are looking for. As noted in [2] it may be necessary to replace γ in (3) by $(f'(\rho))^2\gamma$ to guarantee that the square belongs to $\mathbb{Z}[\rho]$ rather than to the maximal order of the field $\mathbb{Q}[\rho]$.

5 Finding the square roots

Suppose we have found the set S of coprime pairs $(a, b) \in \mathbb{Z}^2$ satisfying (2) and (3). Let $X = \prod_{(a,b) \in S} a + bm$ and let $\gamma = (f'(\rho))^2 \prod_{(a,b) \in S} a + b\rho$. Extracting the

square root x of X is very simple since we know the prime factorization of X. Computing the square root δ of γ is, however, quite difficult since the coefficients in the representation $\delta = \delta_0 + \delta_1 \cdot \rho + \ldots + \delta_{d-1} \cdot \rho^{d-1}$ may be very large. In our implementation we use the method of Couveignes [3]. He suggests to determine a set I of prime numbers which are inert in $\mathbb{Z}[\rho]$ and for each $p \in I$ to compute δ_p such that $\delta_p^2 \equiv \gamma \bmod p$. This can easily be effected by applying a variant of Shanks' RESSOL algorithm [8]. Since we want to apply Chinese remaindering we must determine the image of the same square root for every $p \in I$. Using Newton iteration one can lift any δ_p to a number $\delta_{p^{2^k}}$ such that $\gamma \equiv \delta^2_{p^{2^k}} \bmod p^{2^k}$ where the exponent k is chosen according to experimental experience. Chinese remaindering yields $y = \varphi(\delta)$.
The square γ can be reduced in size by dividing it by some $(a+b\rho)^2$, where (a, b) is a good pair without large primes on the algebraic side. Whether γ is divisible by such a square can be easily checked by inspecting the vectors $((e_P(a,b))_{P\in F_A})$ and $((e_P(\gamma))_{P\in F_a})$. The following table shows the effect of this reduction when used in the factorization of the third number number in section 7.

$\lvert S\rvert$	$\#((a+b\rho)^2)$ reduced	$\lvert I\rvert = \#$ of inert primes *	max. exp. 2^k	maximal # of digits of δ_j	running time in mips h
25022	0	115	256	133777	62.82
25022	7398	60	256	69120	41.01

6 Quality of the polynomials

The least well understood part in the GNFS is how to find the best polynomial f. In this section we illustrate that the algorithm behaves quite differently for different choices of polynomials. Let $n = 6809\,47738\,35969\,19453\,31142\,12277$. Except for m all the parameters were chosen identically as described in the next section. The next two tables show how different polynomials yield a different number of good pairs. For the first table we used the m-adic expansion as described in section 3 to find the polynomial, where $m = \lfloor n^{1/3} \rfloor + i$. From a bigger experiment we present the most interesting results.

* all inert primes about $3 \cdot 10^4$

i	F_A = size of the algebraic factor base	# good pairs of type fff	# good pairs of types $pff \ldots ppp$	# cycles among large primes
−27137	2537	6049	32154	18573
−27139	2532	5019	26906	13801
+23	2524	4811	27812	14665
+13	2492	4365	24790	12139
0	2493	4390	24533	11931
−50467	2498	3552	21016	8985
+27140	2484	3354	19689	7843
−43467	2514	3240	18966	7499
−27142	2454	3181	18552	6998
+27138	2533	2797	16307	5407

For the second table we modified the polynomial $f(x)$ obtained with $m = \lfloor n^{1/3} \rfloor$ by adding g(x).

$g(x)$	$\|F_A\|$	# good pairs of type fff	# good pairs of types $pff \ldots ppp$	# cycles among large primes
$-x^2 + mx$	2535	6014	33224	20213
0	2493	4390	24533	11931
$-x^2 + (m+1)x - m$	2522	3245	18657	7204
$x - m$	2533	3080	16856	5620
$-2(x^2 - (m-1)x - m)$	2348	1780	11312	2339

7 Some full factorizations

The first numbers we factored with GNFS were

1. $n = 6809\,47738\,35969\,19453\,31142\,12277$
   using $f(x) = x^3 + x^2 - 5524\,50799x + 2195\,69758$, $m = 40835\,50467$
2. $n = 82935\,75851\,23433\,22909\,99689\,74960\,03250\,42327$
   using $f(x) = x^3 + 301\,13501\,57913x + 594\,61180\,91613$, $m = 2024\,17135\,03301$
3. $n = 3488\,17079\,74401\,66635\,06963\,23211\,22160\,51028\,26088\,93989$
   using $f(x) = x^3 + 2x^2 + 5\,13769\,39621\,45733x + 2\,78963\,78107\,83197$,
   $m = 15\,16582\,05880\,38497$
4. $n = 9 \cdot 436\,22325\,30202\,01660\,81169\,50834\,54211\,20979\,47919\,09269\,39307$
   $24927\,93753\,70109\,41445\,21495\,39140\,12056\,52499\,95711\,63723\,68586$
   $19995\,36219\,76543\,09529\,71290$
   using $f(x) = x^5 + 9$, $m = 3^{56}$

All relations were found by Pollard's lattice sieve algorithm [7]. The most important datas of these factorizations are summarized in the following table.

# digits of n	29	40	49	134
factor bases				
biggest prime of the rational factor base	5279	22307	30559	951161
size of the rational factor base	700	2500	3300	75000
bound for the large primes on the rational side	10^5	$6 \cdot 10^5$	10^6	10^8
biggest prime p of the pairs (p,cp) of the algebraic factor base	22291	104729	224737	951109
size of the algebraic factor base	2493	9794	19944	74952
bound for the large prime on the algebraic side	10^5	$1.5 \cdot 10^6$	10^7	10^8
# additional pairs (p,cp) with p bigger than large prime bound	10	10	20	25
finding the squares with the lattice sieve				
sieving bound C	500	500	5000	10000
sieving bound D	50	200	1000	5250
# good pairs of type fff	4390	9133	8010	73798
# good pairs of type pff	4733	13020	16906	184864
# good pairs of type fpf	6515	30937	46531	344560
# good pairs of type ppf	8214	42681	109389	1031253
# good pairs of type fpp	2272	17849	69304	$0^\star$
# good pairs of type ppp	2799	22862	153719	$0^\star$
cycles among large primes	11931	23386	19371	69103
sieving time in mips days	1.75	118	717	41010
extracting the square root				
# inert primes	150	175	240	105
size of inert primes about	$3 \cdot 10^4$	$3 \cdot 10^4$	$3 \cdot 10^4$	$1.1 \cdot 10^5$
max. exponent for lifting	16	64	64	256
# digits of coefficients of the root	~ 8000	~ 51000	~ 69120 **	~ 135500 **
running time in mips hours	7.5	36	41	484.5

---

* only one large prime on each side

** with square reduction

The factorizations are

1. 6809 47738 35969 19453 31142 12277
   $= 1785\,89908\,07069 \cdot 3\,81291\,27547\,91033$
2. 82935 75851 23433 22909 99689 74960 03250 42327
   $= 1301\,67526\,01273\,98757 \cdot 63\,71463\,07169\,01048\,08011$
3. 3488 17079 74401 66635 06963 23211 22160 51028 26088 93989
   $= 22036\,72182\,80384\,74120\,85111 \cdot 15828\,90061\,71597\,82957\,88099$
4. 436 22325 30202 01660 81169 50834 54211 20979 47919 09269 39307 24927 93753 70109 41445 21495 39140 12056 52499 95711 63723 68586 19995 36219 76543 09529 71290
   $= 2 \cdot 5 \cdot 557 \cdot 11\,07553 \cdot 8\,20739\,81221\,45081 \cdot$
   $1\,38579\,05391\,45329\,24856\,06236\,63377\,62045\,74597 \cdot$
   $62\,17073\,56762\,16461\,88942\,98788\,28272\,87720\,85730\,54231\,32773\,87634\,13782\,17457$

References

1. J. Buchmann, J. Loho, J. Zayer, *An implementation of the general number field sieve, full version*, to appear 1993
2. J. P. Buhler, H. W. Lenstra, C. Pomerance, *Factoring integers with the number field sieve*, Lecture Notes in Mathematics 1554, pp. 50 - 94, Springer Verlag, 1993
3. J. - M. Couveignes, *Computing a square root for the number field sieve*, Lecture Notes in Mathematics 1554, pp. 95 - 102, Springer Verlag, 1993
4. D. E. Knuth, *The Art of Computer Programming, vol. 2*, Second Edition, Addison Wesley, 1981
5. A. K. Lenstra, H. W. Lenstra, M. S. Manasse, J. M. Pollard, *The number field sieve*, Abstract: Proc. 22nd Ann. ACM Symp. on Theory of Computing (STOC)(1990),564-572
6. A. K. Lenstra, M. Manasse, *Factoring with two large primes*, preprint 1992
7. J. M. Pollard, *The Lattice Sieve*, Lecture Notes in Mathematics 1554, pp. 43 - 49, Springer Verlag, 1993
8. D. Shanks *Five Number-Theoretic Algorithms* , Proc. Second Manitoba Conference On Numerical Math., 1972, pp. 51-70