Reprinted from Series A, volume 91, no. 4, December 19, 1988

proceedings

# Fast and rigorous factorization under the generalized Riemann hypothesis

by A.K. Lenstra

*Department of Computer Science, The University of Chicago, 1100 E 58th Street, Chicago, Illinois 60637, U.S.A.*

MATHEMATICS Proceedings A 91 (4), December 19, 1988

# Fast and rigorous factorization under the generalized Riemann hypothesis

by A.K. Lenstra

*Department of Computer Science, The University of Chicago, 1100 E 58th Street, Chicago, Illinois 60637, U.S.A.*

Communicated by Prof. J.H. van Lint at the meeting of November 30, 1987

ABSTRACT

We present an algorithm that finds a non-trivial factor of an odd composite integer $n$ with probability $\geq 1/2 - o(1)$ in expected time bounded by $e^{(1+o(1))\sqrt{\log n \log \log n}}$. This result can be *rigorously* proved under the sole assumption of the generalized Riemann hypothesis. The time bound matches the *heuristic* time bounds for the continued fraction algorithm, the quadratic sieve algorithm, the Schnorr-Lenstra class group algorithm, and the worst case of the elliptic curve method. The algorithm is based on Seysen's factoring algorithm [14], and the elliptic curve smoothness test from [12].

## INTRODUCTION

In this paper we show that combination of results of Seysen [14] and Pomerance [12] leads to a probabilistic algorithm that factors an odd composite integer $n$ in an expected number of bit operations that is bounded by $e^{(1+o(1))\sqrt{\log n \log \log n}}$. The storage required is $e^{(1/2+o(1))\sqrt{\log n \log \log n}}$ bits. The algorithm is not considered to be practical, but has the advantage that the bound on its running time can be rigorously proved, under the assumption of the generalized Riemann hypothesis (GRH). This confirms a prediction of both Seysen [14, Section 6] and Pomerance [12, Section 1], who both mention the likelihood of this result, although a clear statement was lacking.

Deterministic factoring algorithms all have a running time that is exponential in log $n$. The fastest known one is the Pollard-Strassen algorithm that runs in time $O(n^{1/4+\varepsilon})$ for any $\varepsilon > 0$ (cf. [15]). Under the assumption of the GRH one can do slightly better, namely time $O(n^{1/5+\varepsilon})$, for any $\varepsilon > 0$, using Shanks' class group method or Shanks' infrastructure method (cf. [13]).

The fastest fully proved probabilistic integer factoring algorithm is Dixon's random squares algorithm, when combined with the elliptic curve smoothness test. As shown by Pomerance in [12] it runs in expected time $e^{(\sqrt{2}+o(1))\sqrt{\log n \log\log n}}$ and storage $e^{(\sqrt{1/2}+o(1))\sqrt{\log n \log\log n}}$. This result is based on a combination of the elliptic curve method [9], a recent result of Friedlander and Lagarias [5], and Wiedemann's coordinate recurrence method [16] to solve sparse systems of linear equations. The fastest fully proved probabilistic factoring algorithm under the assumption of the GRH was Seysen's class group method [14], which runs in expected time $e^{(\sqrt{5/4}+o(1))\sqrt{\log n \log\log n}}$. The present algorithm is based on Seysen's algorithm, extended with the techniques introduced by Pomerance.

Some definitions and results on *class groups* and *solving sparse systems of linear equations* are reviewed in Section 1. What we need about *smoothness* and *testing for smoothness* is presented in Section 2. The algorithm, an adapted version of the algorithm from [14, Section 8], is described in Section 3. Its expected running time and its probability of success are analyzed in Section 4. The paper heavily draws on [14], where proofs of many of the results we use can be found. It is a detailed version of [8, Section (3.12)], from which most of sections 1 and 2 was taken.

An informal outline of the algorithm is as follows. Given an odd composite integer $n$ to be factored, we consider the class group $C_\Delta$ of discriminant $-n$ or $-3n$. Because ambiguous forms in $C_\Delta$ might lead to a factorization of $n$, we attempt to find an ambiguous form in the following way. Using a small set of 'small prime forms' generating $C_\Delta$, we randomly select elements of $C_\Delta$ with a known factorization in $C_\Delta$. By means of the elliptic curve smoothness test we can easily decide which of these elements can also be factored in another way, using somewhat bigger but still reasonably small prime forms. An element for which this second factorization can be found yields a factorization of the unit element of the class group into reasonably small prime forms. Given sufficiently many factorizations of the unit element, we combine them into a factorization of the unit element in which the exponents of all prime forms are even. Dividing all exponents by two, we now find an ambiguous form, and possibly a non-trivial factorization of $n$. The proper combination of factorizations is found by solving a system of linear equations over $\mathbb{Z}/2\mathbb{Z}$.

The rigorous analysis of the algorithm hinges on several points. In order to be able to prove that the elliptic curve smoothness test works sufficiently often, we have to restrict the class of allowable prime forms. This should be done in such a way that the probability that a random form is built up from these prime forms is high enough. That this is possible follows by combining results from [14] and [12]. Then we have to prove that the forms generated in the algorithm behave approximately as random forms. This is a consequence of the fact that we use a set of generators of $C_\Delta$. Because the latter set has a small cardinality the resulting system is sparse, which makes a fast solution possible. Finally, it has to be shown that the ambiguous form leads to a non-trivial factorization of $n$ with probability $\geq 1/2 - o(1)$.

## 1. PRELIMINARIES

(1.1) *Class groups.* For details and proofs of the following results about class groups we refer to [1, 13]; most of these results are due to Gauss. A polynomial $aX^2+bXY+cY^2 \in \mathbb{Z}[X,Y]$ is called a *binary quadratic form*, and $\Delta = b^2-4ac$ is its *discriminant*. We denote a binary quadratic form $aX^2+bXY+cY^2$ by $(a,b,c)$. A form for which $a>0$ and $\Delta<0$ is called *positive*, and a form is *primitive* if $\gcd(a,b,c)=1$. Two forms $(a,b,c)$ and $(a',b',c')$ are *equivalent* if there exist $\alpha,\beta,\gamma,\delta\in\mathbb{Z}$ with $\alpha\delta-\beta\gamma=1$ such that $a'U^2+b'UV+c'V^2=aX^2+bXY+cY^2$, where $U=\alpha X+\gamma Y$, and $V=\beta X+\delta Y$. Two equivalent forms have the same discriminant.

Now fix some negative integer $\Delta$ with $\Delta\equiv 0$ or $1 \bmod 4$. We will often denote a form $(a,b,c)$ of discriminant $\Delta$ by $(a,b)$, since $c$ is determined by $\Delta=b^2-4ac$. The set of equivalence classes of positive, primitive, binary quadratic forms of discriminant $\Delta$ is denoted by $C_\Delta$. The existence of the form $(1,\Delta)$ shows that $C_\Delta$ is non-empty.

(1.2) *Reduction algorithm.* Each equivalence class in $C_\Delta$ contains precisely one *reduced form*, where a form $(a,b,c)$ is reduced if

$$\begin{cases} |b|\leq a\leq c \\ b\geq 0 \text{ if } |b|=a \text{ or if } a=c. \end{cases}$$

These inequalities imply that $a\leq\sqrt{|\Delta|/3}$, so that $C_\Delta$ is finite. For any form $(a,b,c)$ of discriminant $\Delta$ the reduced form equivalent to it can be determined by means of the following reduction algorithm:

(1) Replace $(a,b)$ by $(a,b-2ka)$, where $k\in\mathbb{Z}$ is such that $-a<b-2ka\leq a$.

(2) If $(a,b,c)$ is reduced, then stop; otherwise, replace $(a,b,c)$ by $(c,-b,a)$ and go back to step (1).

It is easily verified that this is a polynomial-time algorithm.

The *class number* $h_\Delta$ of $\Delta$ is defined as the cardinality of $C_\Delta$. It follows from the Brauer-Siegel theorem (cf. [7, Ch. XVI]) that $h_\Delta=|\Delta|^{1/2+o(1)}$ for $\Delta\to-\infty$. Furthermore, $h_\Delta<(\sqrt{|\Delta|}\log|\Delta|)/2$ for $\Delta<-3$.

(1.3) *Composition algorithm.* The set $C_\Delta$, which will be identified with the set of reduced forms of discriminant $\Delta$ because of (1.2), is a finite abelian group, the *class group*. The group law, which we will write multiplicatively, is defined as follows. The inverse of $(a,b)$ follows from an application of the reduction algorithm to $(a,-b)$, and the unit element $1_\Delta$ is $(1,1)$ for $\Delta$ odd, and $(1,0)$ for $\Delta$ even. To compute $(a_1,b_1)\cdot(a_2,b_2)$, we use the Euclidean algorithm to determine $d=\gcd(a_1,a_2,(b_1+b_2)/2)$, and $r,s,t\in\mathbb{Z}$ such that $d=ra_1+sa_2+t(b_1+b_2)/2$. The product then follows from an application of the reduction algorithm to $(a_1a_2/d^2, b_2+2a_2(s(b_1-b_2)/2-tc_2)/d)$, where $c_2=(b_2^2-\Delta)/(4a_2)$. It is again an easy matter to verify that this is a polynomial-time algorithm.

(1.4) *Ambiguous forms.* A reduced form is *ambiguous* if its square equals $1_\Delta$; for an ambiguous form we have $b=0$, or $a=b$, or $a=c$. From now on we assume that $\Delta\equiv 1 \bmod 4$. For these $\Delta$'s there is a bijective correspondence between ambiguous forms and factorizations of $|\Delta|$ into two relatively prime factors. For relatively prime $p$ and $q$ the factorization $|\Delta|=pq$ corresponds to the ambiguous form $(p,p)$ for $3p\le q$, and to $((p+q)/4,(q-p)/2)$ for $p<q\le 3p$. Notice that the ambiguous form $(1,1)$ corresponds to the factorization $|\Delta|=1\cdot|\Delta|$, and that $h_\Delta$ is even if and only if $|\Delta|$ is not a prime power.

(1.5) *Prime forms.* For a prime number $p$ we define the Kronecker symbol $\left(\frac{\Delta}{p}\right)$ by

$$\left(\frac{\Delta}{p}\right)=\begin{cases} 1 \text{ if } \Delta \text{ is a quadratic residue modulo } 4p \text{ and } \gcd(\Delta,p)=1\\ 0 \text{ if } \gcd(\Delta,p)\neq 1\\ -1 \text{ otherwise.}\end{cases}$$

For a prime $p$ for which $\left(\frac{\Delta}{p}\right)=1$, we define the *prime form* $I_p$ as the reduced form equivalent to $(p,b_p)$, where $b_p=\min\{b\in\mathbb{Z}_{>0}: b^2\equiv\Delta \bmod 4p\}$. It follows from the effective Chebotarev density theorem in [6] that, if the generalized Riemann hypothesis holds, then there is an effectively computable constant $c$, such that $C_\Delta$ is generated by the prime forms $I_p$ with $p\le c\cdot(\log|\Delta|)^2$, where we only consider primes $p$ for which $\left(\frac{\Delta}{p}\right)=1$ (cf. [13, Cor. 6.2]); apparently no explicit value for the constant $c$ has been published.

(1.6) *Factorization of forms.* A form $(a,b,c)$ of discriminant $\Delta$, with $\gcd(a,\Delta)=1$, for which the prime factorization of $a$ is known, can be factored into prime forms in the following way. If $a=\prod_{p \text{ prime}} p^{e_p}$ is the prime factorization of $a$, then $(a,b)=\prod_{p\text{ prime}} I_p^{s_p e_p}$, where $s_p\in\{-1,+1\}$ satisfies $b\equiv s_p b_p \bmod 2p$, with $b_p$ as in (1.5). Notice that the prime forms $I_p$ are well-defined because the primes $p$ divide $a$, $\gcd(a,\Delta)=1$, and $b^2\equiv\Delta \bmod 4a$.

(1.7) *Solving sparse systems of linear equations.* Let $A$ be an $m\times(m+1)$-matrix over a finite field, for some positive integer $m$. Suppose we want to find a non-zero vector $x$ over the field such that $Ax=0$, in the situation where $A$ is sparse, i.e., if the number of non-zero entries in $A$ is very small. Straightforward application of Gaussian elimination would need $O(m^3)$ field operations, but then we do not take advantage of the sparseness of $A$.

A faster method is provided by Wiedemann's coordinate recurrence method [16]. Let $B$ be an $m\times m$-matrix, and let $y$ be an $m$-dimensional vector, both over the field in question. Let $w(B)$ be the number of non-zero entries in $B$. By means of the coordinate recurrence method we can compute a vector $x$ such that either $Bx=y$ or $x$ is non-zero and $Bx=0$ [16, Section 3]. This takes an expected number of field operations $O(m^{1+\varepsilon}w(B))$, for $\varepsilon>0$ arbitrary, and storage for $O(m)+w(B)$ field elements. This algorithm can easily be used to solve our problem. Let $B$ be the matrix consisting of the first $m$ columns of $A$, let $y$ be the last column of $A$, and find $x=(x_i)_{i=1}^m$ such that either $Bx=y$ or $x\neq 0$ and $Bx=0$. In the first case put $x_{m+1}=-1$, and in the latter case put $x_{m+1}=0$, then the vector $(x_i)_{i=1}^{m+1}$ is a non-zero solution to the original problem.

For details about this algorithm we refer to [16]. A deterministic version of the coordinate recurrence method has the same (deterministic) running time, but needs storage for $O(m^2)$ field elements. Other algorithms for the solution of sparse systems over finite fields can be found in [3] and [11].

## 2. TESTING FOR SMOOTHNESS

(2.1) *Smoothness.* An integer is *smooth with respect to y*, or *y-smooth*, if all its prime factors are $\le y$. The function $\psi(x,y)$ is defined as the number of positive integers $\le x$ that are smooth with respect to $y$. From [2] and [4] we have that for a fixed arbitrary $\varepsilon>0$, and for $x\ge 10$ and $u\le(\log x)^{1-\varepsilon}$,

$$\psi(x,x^{1/u})=x\cdot u^{-u+f(x,u)},$$

for a function $f$ that satisfies $f(x,u)/u\to 0$ for $u\to\infty$ uniformly in $x$. It follows that for fixed $\alpha,\beta\in\mathbb{R}_{>0}$ and for $n\to\infty$

$$\psi(n^\alpha, n^{\beta\sqrt{(\log\log n)/\log n}})$$
$$=n^\alpha\cdot((\alpha/\beta)\sqrt{\log n/\log\log n})^{-(1+o(1))(\alpha/\beta)\sqrt{\log n/\log\log n}}.$$

With $L(n)=e^{\sqrt{\log n\log\log n}}$, this becomes

$$\psi(n^\alpha, L(n)^\beta)=n^\alpha\cdot L(n)^{-\alpha/(2\beta)+o(1)}. \tag{2.2}$$

We find that a random positive integer $\le n^\alpha$ is smooth with respect to $L(n)^\beta$ with probability $L(n)^{-\alpha/(2\beta)+o(1)}$, for $n\to\infty$.

For $\beta\in\mathbb{R}$ we will often write $L_n[\beta]$ for $L(n)^\beta$, and we will abbreviate $L_n[\beta+o(1)]$ to $L_n[\beta]$, for $n\to\infty$. Notice that in this notation $L_n[\alpha]+L_n[\beta]=L_n[\max(\alpha,\beta)]$, and that $\pi(L_n[\beta])=L_n[\beta]$, where $\pi(y)$ is the number of primes $\le y$.

(2.3) *Testing for smoothness.* Given an integer $x\le n$ and some fixed $\beta\in\mathbb{R}_{>0}$, how many operations does it take to test $x$ for $L_n[\beta]$-smoothness, and to find the complete factorization of $x$ in case of smoothness? Clearly, both tasks can be completed in $L_n[\beta]$ bit operations by trial division up to $L_n[\beta]$. A faster method is provided by the elliptic curve method (cf. [9]). This method finds a factor $p$ of $x$ in expected time $O((\log x)^2 L_p[\sqrt{2}])$. Unfortunately, this running time can only be rigorously proved under the assumption that a random integer in the interval $(p-\sqrt{p}+1, p+\sqrt{p}+1)$ is $L_p[\sqrt{1/2}]$-smooth with probability at least $L_p[-\sqrt{1/2}]$, the probability that would be expected on the basis of (2.2). If this assumption holds for the primes $\leq L_n[\beta]$ dividing $x$, then $x$ can be tested for $L_n[\beta]$-smoothness in expected time $L_n[0]$; this time includes the time needed to factor $x$ in case of smoothness.

In order to get a rigorous smoothness test we have to restrict our attention to primes $p$ for which the interval $(p-\sqrt{p}+1, p+\sqrt{p}+1)$ contains sufficiently many smooth numbers. Pomerance has shown in [12] how this can be achieved. Define for a real number $y$ the set $S(y)$ as the set of primes $p$, $3<p\leq y$, for which the interval $(p-\sqrt{p}, p+\sqrt{p})$ contains at least $\sqrt{p}\cdot e^{-((\log y)^{1/7}\log\log y)/6}$ numbers that are $e^{(\log p)^{6/7}}$-smooth. It then follows from a result of Friedlander and Lagarias (cf. [5]) that

$$(2.4)\qquad \pi(y)-\#S(y)=O(y\cdot e^{-((\log y)^{1/6})/2}),$$

where $\pi$ is the prime counting function (cf. [12, Theorem B′]).

A restricted definition of smoothness now leads to a rigorous smoothness test. We say that an integer is $(x, y)$-smooth if it is $\leq x$ and built up from primes $p$ such that $p\leq e^{64(\log\log x)^6}$ or $p\in S(y)$. Define $\psi_1(x, y)$ as the number of $(x, y)$-smooth integers (cf. [12, Section 3]). From (2.4) it follows that for $\beta\in\mathbb{R}_{>0}$ fixed and for $x\to\infty$

$$(2.5)\qquad \psi_1(x, L_x[\beta])=\psi(x, L_x[\beta])\left(1+O\left(\frac{(\log\log x)^{11/2}}{\sqrt{\log x}}\right)\right)$$

(cf. [12, proof of Lemma 3.1]), so that, with (2.2),

$$(2.6)\qquad \psi_1(x, L_x[\beta])=x\cdot L_x[-1/(2\beta)].$$

From (2.6) it follows that $(n, L_n[\beta])$-smooth numbers occur asymptotically about as frequently as ordinary $L_n[\beta]$-smooth numbers, a result that we will not need.

(2.7) *A rigorous smoothness test.* It has been proved in [12] that any $(n, L_n[\beta])$-smooth number can be recognized with high probability in time $L_n[0]$. This is done as follows. First remove the prime factors $\leq e^{64(\log\log n)^6}$ by trial division. If the resulting quotient $a$ is not equal to 1, apply the elliptic curve method (cf. [9]) to find the factors $\leq L_n[\beta]$ of $a$. If $a$ is $(n, L_n[\beta])$-smooth, then all factors in the second stage are actually in $S(L_n[\beta])$. From [12, Theorem 2.1] it follows that, in case of smoothness, the complete factorization will be found with probability at least $1-(\log a)/a$ and in time $L_n[0]$. This finishes the description of the rigorous smoothness test.

(2.8) *Smoothness in class groups.* Now fix a negative integer $\Delta$ with $\Delta\equiv 1$ mod 4, and consider the elements of the class group $C_\Delta$ as in (1.1). We say that a form $(a, b)\in C_\Delta$ with $\gcd(a, \Delta)=1$ is $y$-smooth if $a$ is $y$-smooth (cf. (2.1)); the factorization of the form $(a, b)$ follows from the factorization of $a$ as in (1.6). Consequently, the only primes that can occur in the factorization of $a$ are primes $p$ for which $\left(\frac{\Delta}{p}\right)=1$. For that reason, we define $\psi_\Delta(x, y)$ as the number of positive integers $\leq x$ that are built up from primes $p\leq y$ for which $\left(\frac{\Delta}{p}\right)=1$. In the notation of (2.1) we have from [14, Theorem 5.2] and (2.1) that for fixed $\beta\in\mathbb{R}_{>0}$ and $\Delta\to-\infty$

$$(2.9)\qquad \psi_\Delta(\sqrt{|\Delta|}/2, L_{|\Delta|}[\beta])=\sqrt{|\Delta|}\cdot L_{|\Delta|}[-1/(4\beta)],$$

under the assumption of the generalized Riemann hypothesis. This means that, under the GRH, the probability that a random integer $\leq\sqrt{|\Delta|}/2$ is built up from primes $p\leq L_{|\Delta|}[\beta]$ for which $\left(\frac{\Delta}{p}\right)=1$ is asymptotically about the same as the probability that such an integer is $L_{|\Delta|}[\beta]$-smooth. The extra condition $\left(\frac{\Delta}{p}\right)=1$ on the primes $p$ therefore makes asymptotically no difference. In [14] it is shown as a consequence of (2.9) that a random reduced form is $L_{|\Delta|}[\beta]$-smooth with probability at least $L_{|\Delta|}[-1/(4\beta)]$. We will have no need for this result; instead we need a slightly stronger version.

Testing a reduced form $(a, b)$ for smoothness can be done by testing $a$ for smoothness. In view of the rigorous smoothness test in (2.7) we must restrict the allowable prime factors of $a$ as we have done in (2.3). We say that an integer is $(x, y, \Delta)$-smooth if it is $(x, y)$-smooth and built up from primes $p$ with $\left(\frac{\Delta}{p}\right)=1$; a form $(a, b)$ is $(x, y, \Delta)$-smooth if $a$ is $(x, y, \Delta)$-smooth. Let $\psi_{\Delta,1}(x, y)$ be the number of $(x, y, \Delta)$-smooth integers. Clearly $\psi_\Delta(x, L_x[\beta])-\psi_{\Delta,1}(x, L_x[\beta])$ is bounded from above by $\psi(x, L_x[\beta])-\psi_1(x, L_x[\beta])$. From (2.5) it follows that

$$\psi_\Delta(x, L_x[\beta])-\psi_{\Delta,1}(x, L_x[\beta])=O\left(\psi(x, L_x[\beta])\cdot\frac{(\log\log x)^{11/2}}{\sqrt{\log x}}\right),$$

so that, with (2.2) and (2.9),

$$(2.10)\qquad \psi_{\Delta,1}(\sqrt{|\Delta|}/2, L_{|\Delta|}[\beta])=\sqrt{|\Delta|}\cdot L_{|\Delta|}[-1/(4\beta)],$$

for fixed $\beta\in\mathbb{R}_{>0}$, $\Delta\to-\infty$, and under the assumption of the generalized Riemann hypothesis.

Because the $(n, L_n[\beta], \Delta)$-smooth numbers form a subset of the $(n, L_n[\beta])$-smooth numbers, they can be recognized with the same high probability and in time $L_n[0]$ by the smoothness test in (2.7).

It follows from (2.10) that a random positive integer $\leq\sqrt{|\Delta|}/2$ is $(\sqrt{|\Delta|}/2, L_{|\Delta|}[\beta], \Delta)$-smooth with probability $L_{|\Delta|}[-1/(4\beta)]$, for $\Delta\to-\infty$ and under the assumption of the GRH. Now consider how likely it is that a random reduced form is $(\sqrt{|\Delta|}/2, L_{|\Delta|}[\beta], \Delta)$-smooth. Let $F(\Delta)$ be the number of $(\sqrt{|\Delta|}/2, L_{|\Delta|}[\beta], \Delta)$-smooth reduced forms. As in [14, Lemma 5.1] one then easily shows with (1.2) and (1.6) that

$$(2.11)\qquad F(\Delta)\geq\psi_{\Delta,1}(\sqrt{|\Delta|}/2, L_{|\Delta|}[\beta]).$$

The probability that a random reduced form is $(\sqrt{|\Delta|}/2, L_{|\Delta|}[\beta], \Delta)$-smooth is $\geq F(\Delta)/h_\Delta$. With (2.11) and the upper bound on $h_\Delta$ from (1.2) we find that this probability is at least

$$\frac{\psi_{\Delta,1}(\sqrt{|\Delta|}/2, L_{|\Delta|}[\beta])}{\sqrt{|\Delta|}\log|\Delta|}$$

(cf. [14, proof of Proposition 4.4]). Application of (2.10) now yields that a random reduced form is $(\sqrt{|\Delta|}/2, L_{|\Delta|}[\beta], \Delta)$-smooth with probability at least $L_{|\Delta|}[-1/(4\beta)]$, for $\Delta \to -\infty$ and under the assumption of the generalized Riemann hypothesis. As was the case for integers, this smoothness can be recognized with high probability in time $L_{|\Delta|}[0]$ by the smoothness test in (2.7).

## 3. THE ALGORITHM

We describe a probabilistic algorithm to factor an integer $n$ that is based on the algorithm from [14, Section 8], and that makes use of the smoothness test from (2.7).

(3.1) *The algorithm*

(1) Let $n$ be an odd composite integer that is not a power of a prime number. Put $\Delta = -n$. If $\Delta \equiv 3 \bmod 4$, then replace $\Delta$ by $3\Delta$.

(2) Define $P_c$ for a positive constant $c$ as the set of primes $p$ with $p \leq c \cdot (\log|\Delta|)^2$ and $\left(\frac{\Delta}{p}\right) = 1$, where $c$ is chosen such that the prime forms $(I_p)_{p\in P_c}$ generate $C_\Delta$ (cf. (1.5)); this is possible under the assumption of the GRH. Define $P_S$ as the set of primes $p$ for which $p \leq e^{64(\log\log(\sqrt{|\Delta|}/2))^6}$ or $p \in S(L_{|\Delta|}[1/2])$ (cf. (2.3)), and for which $\left(\frac{\Delta}{p}\right) = 1$. Let $P = P_c \cup P_S$, and put $i = 0$. (Notice that asymptotically $P_c$ is contained in $P_S$.)

(3) For all $p \in P_c$ randomly and independently draw $e_p \in \{0, 1, \ldots, |\Delta| - 1\}$, and compute the reduced form

$$(a, b) = \prod_{p\in P_c} I_p^{e_p} \in C_\Delta.$$

(4) Use the smoothness test from (2.7) to test whether $a$ is $(\sqrt{|\Delta|}/2, L_{|\Delta|}[1/2], \Delta)$-smooth, i.e., can completely be factored using the primes in $P_S$. If not, go back to step (3). Otherwise, use the factorization of $a$ to determine an integral vector $(t_p)_{p\in P_S}$ such that

$$(a, b) = \prod_{p\in P_S} I_p^{t_p}.$$

Put $r_{p,i} = e_p - t_p$, where $e_p = 0$ for $p \in P \setminus P_c$ and $t_p = 0$ for $p \in P \setminus P_S$, then

$$\prod_{p\in P} I_p^{r_{p,i}} = 1_\Delta.$$

(5) If $i < \#P$, then replace $i$ by $i+1$ and go back to step (3). Otherwise, we have $\#P + 1$ vectors $(r_{p,i})_{p\in P}$. Let $A$ be the $\#P \times (\#P + 1)$-matrix having the $(r_{p,i} \bmod 2)_{p\in P}$, for $i = 0, 1, \ldots, \#P$, as columns. Apply the algorithm described in (1.7) to the matrix $A$ to compute a non-zero solution $x = (x_i)_{i=0}^{\#P}$ to $Ax = 0$ over $\mathbb{Z}/2\mathbb{Z}$, and determine the integral vector $(v_p)_{p\in P}$ defined by $(\sum_{i=0}^{\#P} r_{p,i}x_i)_{p\in P} = (2v_p)_{p\in P}$. Notice that $\prod_{p\in P} I_p^{2v_p} = 1_\Delta$.

(6) Compute the ambiguous form $\prod_{p\in P} I_p^{v_p}$ and attempt to factor $n$ using this ambiguous form (cf. (1.4)).

This finishes the description of the factoring algorithm.

## 4. THE ANALYSIS

In this section we give an analysis of the probability of success and of the expected running time of Algorithm (3.1). We need the following lemma, which is a slightly modified version of [14, Lemma 4.5].

(4.1) LEMMA. *Let $m$ be a positive integer, and let $\Lambda$ be a lattice in $\mathbb{Z}^m$ of determinant $h$ such that the exponent of $\mathbb{Z}^m/\Lambda$ is at most $d$. Then for any $w \in \mathbb{Z}^m$ and $B \in \mathbb{Z}_{\geq d}$ we have*

$$(4.2)\qquad \frac{1}{h}\cdot\left(1 - \frac{h-1}{B}\right) \leq \frac{\#\{z : z \in \mathbb{Z}_B^m, z \equiv w \bmod \Lambda\}}{B^m} \leq \frac{1}{h}\cdot\left(1 + \frac{h-1}{B}\right),$$

*and*

$$(4.3)\qquad \frac{1}{h}\cdot\left(1 - \frac{d-1}{B}\right)^m \leq \frac{\#\{z : z \in \mathbb{Z}_B^m, z \equiv w \bmod \Lambda\}}{B^m} \leq \frac{1}{h}\cdot\left(1 + \frac{d-1}{B}\right)^m,$$

*where $\mathbb{Z}_B = \{0, 1, \ldots, B-1\}$.*

PROOF. Let $e_i$ be the $i$th basis vector in the standard basis for $\mathbb{Z}^m$, and $\bar{e}_i$ its image in $\mathbb{Z}^m/\Lambda$. The order $h_i$ of $\bar{e}_i$ modulo the subgroup of $\mathbb{Z}^m/\Lambda$ generated by $\bar{e}_1, \bar{e}_2, \ldots, \bar{e}_{i-1}$ satisfies $h_i \leq d$ and $\prod_{i=1}^m h_i = h$.

There is a bijection between any $h_1 \times h_2 \times \cdots \times h_m$ box in $\mathbb{Z}^m$ and $\mathbb{Z}^m/\Lambda$. This implies that, for integers $k_i$, $1 \leq i \leq m$, any $k_1h_1 \times k_2h_2 \times \cdots \times k_mh_m$ box in $\mathbb{Z}^m$ intersects every coset modulo $\Lambda$ precisely $\prod_{i=1}^m k_i$ times. With $k_i = \lfloor B/h_i \rfloor \geq (B - h_i + 1)/h_i$ it follows that the number of times that $\mathbb{Z}_B^m$ intersects a particular residue class is at least

$$\prod_{i=1}^m \lfloor B/h_i \rfloor \geq \frac{B^m}{h}\cdot\prod_{i=1}^m\left(1 - \frac{h_i - 1}{B}\right).$$

The lower bound in (4.3) now follows from $h_i \leq d$, and the lower bound in (4.2) follows by repeated application of

$$\frac{a-1}{B} + \frac{b-1}{B} - \frac{(a-1)(b-1)}{B^2} \leq \frac{ab-1}{B},$$

for $a, b \geq 1$. The upper bounds in (4.2) and (4.3) can similarly be derived by taking $k_i = \lceil B/h_i \rceil \leq (B + h_i - 1)/h_i$. This proves Lemma (4.1).

(4.4) We now prove that the forms generated in step (3) of Algorithm (3.1) behave approximately as random reduced forms in $C_\Delta$. The proof follows the lines of the proof of Proposition 4.3 in [14].

Let $P_c$ be as in step (2) of (3.1), and let $M=\mathbb{Z}^{P_c}$. Define a mapping $\phi$ from $M$ onto $C_\Delta$ that maps $(e_p)_{p\in P_c}\in M$ to $\prod_{p\in P_c} I_p^{e_p}\in C_\Delta$; that $\phi$ is a surjective homomorphism follows from the choice of $c$, and such a $c$ can be chosen under the assumption of the GRH. The kernel $N$ of $\phi$ is a sublattice of the lattice $M$, and $M/N\cong C_\Delta$. The determinant of $N$ equals $h_\Delta$.

Now let $e=(e_p)_{p\in P_c}$ where the $e_p$ are randomly and independently selected from $\{0,1,\ldots,|\Delta|-1\}$, as in step (3) of (3.1). For an arbitrary reduced form $f\in C_\Delta$ we have that $f=\phi(e)$ with probability

$$\frac{\#\{e: e\in\mathbb{Z}_{|\Delta|}^{P_c}, \phi(e)=f\}}{|\Delta|^{\#P_c}},$$

with the notation as in Lemma (4.1). Applying Lemma (4.1) with $m=\#P_c$, $\Lambda=N$, $h=h_\Delta$, $d\le h$, $w=f$, and $B=|\Delta|$, we find from (4.2) that this probability is $(1+o(1))/h_\Delta$ for $n\to\infty$ (cf. (1.2)), so that the forms generated in step (3) of Algorithm (3.1) indeed behave as random reduced forms in $C_\Delta$.

(4.5) *Running time analysis.* From (4.4) and the last paragraph of (2.8) (with $\beta=1/2$), it follows that $a$ in step (4) of Algorithm (3.1) is $(\sqrt{|\Delta|}/2, L_{|\Delta|}[1/2], \Delta)$-smooth with probability at least $L_{|\Delta|}[-1/2]$, under the assumption of the GRH; the time per smoothness test is $L_{|\Delta|}[0]$. Because $\#P\le\pi(L_{|\Delta|}[1/2])=L_{|\Delta|}[1/2]$, we find that it takes expected time $L_{|\Delta|}[1/2]\cdot L_{|\Delta|}[1/2]\cdot L_{|\Delta|}[0]=L_{|\Delta|}[1]$ to generate the matrix $A$ in step (5), under the assumption of the GRH.

As noted in (1.7) the expected time to solve the system $Ax=0$ over $\mathbb{Z}/2\mathbb{Z}$ in step (5) is $L_{|\Delta|}[1/2]\cdot w(A)$, where $w(A)$ is the number of non-zero entries in $A$. Let us analyse how many non-zero entries there can be in any column $(r_{p,i} \bmod 2)_{p\in P}$ of $A$, where $r_{p,i}=e_p-t_p$. The number of non-zero $e_p$'s is bounded by $\#P_c$ (cf. step (3)), and therefore by $c\cdot(\log|\Delta|)^2$. The number of non-zero $t_p$'s is bounded by the number of distinct prime divisors of $a$ in step (4). Because $a\le\sqrt{|\Delta|}/2$, we find that there are at most $\log_2|\Delta|$ non-zero $t_p$'s. It follows that there are $O((\log|\Delta|)^2)$ non-zero entries per column of $A$, so that $w(A)=L_{|\Delta|}[1/2]$. We find that the system can be solved in time $L_{|\Delta|}[1]$. Notice that the GRH plays an important role in this argument, namely to bound the number of generators of $C_\Delta$ in such a way that $A$ becomes sparse.

An ambiguous form now follows in $L_{|\Delta|}[1/2]$ applications of the composition algorithm (1.3), so that we conclude that the algorithm runs in an expected number of bit operations that is bounded by $L_n[1]$. The storage required for Algorithm (3.1) is $L_n[1/2]$ as follows from (1.7).

(4.6) *Probability of success.* Let $G$ be the group of ambiguous forms in $C_\Delta$, and let $H$ be the subgroup of $G$ containing the ambiguous forms that lead to a trivial factorization of $n$, i.e., those forms that lead to the factorization $\Delta=1\cdot\Delta$ or, if $-n\equiv3 \bmod 4$, to $\Delta=-3\cdot n$. It is easily seen that $H$ is a subgroup of $G$ (cf. (1.4)), and because $n$ is composite $H$ is a proper subgroup of $G$. We will show that, over all possible runs of Algorithm (3.1), the ambiguous form computed in step (6) of the algorithm equals any given element of $G$ with probability $(1+o(1))/\#G$. It follows that this ambiguous form is contained in $H$ with probability at most $\#H(1+o(1))/\#G$, so that the probability of success of the algorithm is at least $1/2-o(1)$.

To explain how this is proved we use the notation $M$, $\phi$, and $N$ from (4.4), and we introduce the following new notation. Let $U\in M^{\#P+1}$ denote the matrix whose columns are the $\#P+1$ vectors $(e_p)_{p\in P_c}$ that are found in steps (3) and (4) of Algorithm (3.1); so the entries of $U$ are in $\mathbb{Z}_{|\Delta|}=\{0,1,\ldots,|\Delta|-1\}$, and the columns of $U$ have a $(\sqrt{|\Delta|}/2, L_{|\Delta|}[1/2], \Delta)$-smooth image under $\phi$. Notice that the matrix consisting of the $(t_p)_{p\in P_S}$ as in step (4) of (3.1) depends only on the coset of $U$ modulo the sublattice $N^{\#P+1}$ of $M^{\#P+1}$, and that the matrix consisting of the $(e_p \bmod 2)_{p\in P_c}$ depends only on the coset of $U$ modulo $(2M)^{\#P+1}$. This suggests to consider $U$ modulo $K$, where $K=(N\cap 2M)^{\#P+1}$.

Fix a coset $\mathscr{C}$ of $K$ in $M^{\#P+1}$, and assume that some (and hence any) matrix in $\mathscr{C}$ has the property that its columns have a $(\sqrt{|\Delta|}/2, L_{|\Delta|}[1/2], \Delta)$-smooth image under $\phi$. To prove our claim we may restrict our attention, first, to those runs of the algorithm for which $U\in\mathscr{C}$.

The probability that $U$ is equal to a particular matrix in $\mathscr{C}$ with entries from $\mathbb{Z}_{|\Delta|}$ depends only on the images of the columns of that matrix under $\phi$, and therefore only on $\mathscr{C}$. Writing $m=\#P_c(\#P+1)$ and identifying $M^{\#P+1}$ with $\mathbb{Z}^m$, we thus find that $U$ is uniformly distributed over $\mathbb{Z}_{|\Delta|}^m\cap\mathscr{C}$.

The matrix $A$ appearing in (3.1) depends only on $\mathscr{C}$, so it is now fixed. The probability that the non-zero solution $x$ to $Ax=0$ that is found in step (5) of (3.1) is equal to a given solution depends only on $A$, hence on $\mathscr{C}$. Therefore we may, again, restrict attention to those runs of the algorithm for which this solution is equal to a given non-zero solution. We call this solution $x=(x_i)_{i=0}^{\#P}$, with $x_i\in\mathbb{Z}/2\mathbb{Z}$.

For $U\in\mathscr{C}$, we write $\upsilon(U)$ for the ambiguous form that is found in step (6) of (3.1). To describe how $\upsilon(U)$ varies with $U$, we define a mapping $\psi$ from $N\cap 2M$ to the group $G$ by $\psi(y)=\phi(y/2)$, for $y\in N\cap 2M$. This is a well-defined group homomorphism with kernel $2N$; it is surjective because $\phi$ is surjective. Let now $U$, $U'\in\mathscr{C}$, and let the columns of the matrix $U-U'\in K$ be denoted by $k_i$, for $0\le i\le\#P$. Then we have that

$$\upsilon(U)/\upsilon(U')=\prod_{i=0}^{\#P}\psi(k_i)^{x_i}\in G.$$

Denote the mapping from $K$ to $G$ given by the right hand side of this expression by $\psi_x$, and write $J=\ker(\psi_x)$. Because $\psi$ is surjective and $x\ne0$, we have that $K/J\cong G$, so that $\det(J)/\det(K)=\#G$.

The above formula implies that the set of $U\in\mathscr{C}$ for which $\upsilon(U)$ is equal to a given element $\upsilon\in G$ is a coset $\mathscr{D}$ of $J$ in $M^{\#P+1}$ with $\mathscr{D}\subset\mathscr{C}$. We find that the probability that $\upsilon(U)$ equals $\upsilon$ is equal to $\#(\mathbb{Z}^m_{|\Delta|}\cap\mathscr{D})/\#(\mathbb{Z}^m_{|\Delta|}\cap\mathscr{C})$. It remains to be shown that this number is $(1+o(1))/\#G$.

Both the lattices $K$ and $J$ contain $(2N)^{\#P+1}$, so that the exponents of $\mathbb{Z}^m/K$ and $\mathbb{Z}^m/J$ are bounded by $2h$. Application of (4.3), Lemma (4.1), then yields that

$$\#(\mathbb{Z}^m_{|\Delta|}\cap\mathscr{D})=\frac{|\Delta|^m(1+o(1))}{\det(J)},\quad \#(\mathbb{Z}^m_{|\Delta|}\cap\mathscr{C})=\frac{|\Delta|^m(1+o(1))}{\det(K)},$$

because of the respective sizes of $m$, $h$ and $|\Delta|$. The required result now follows upon division, since $\det(J)/\det(K)=\#G$.

Acknowledgments are due to H.W. Lenstra, Jr. for his valuable assistance while writing this paper, and especially for his formulation of the proof in (4.6). I would also like to thank the Centrum voor Wiskunde en Informatica in Amsterdam, where most of the work on this paper was done, for its hospitality and support.

REFERENCES

1. Borevič, Z.I. and I.R. Šafarevič – Teorija čisel, Moscow 1964. Translated into German, English and French.
2. Canfield, E.R., P. Erdös and C. Pomerance – On a problem of Oppenheim concerning "Factorisatio Numerorum", J. Number Theory **17**, 1-28 (1983).
3. Coppersmith, D., A.M. Odlyzko and R. Schroeppel – Discrete logarithms in *GF(p)*, Algorithmica **1**, 1-15 (1986).
4. de Bruijn, N.G. – On the number of positive integers $\leq x$ and free of prime factors $>y$, II, Indag. Math. **38**, 239-247 (1966).
5. Friedlander, J.B. and J.C. Lagarias – On the distribution in short intervals of integers having no large prime factor, J. Number Theory **25**, 249-273 (1987).
6. Lagarias, J.C., H.L. Montgomery and A.M. Odlyzko – A bound for the least prime ideal in the Chebotarev density theorem, Inventiones Math. **54**, 137-144 (1975).
7. Lang, S. – Algebraic number theory, Addison-Wesley, Reading (1970).
8. Lenstra, A.K. and H.W. Lenstra, Jr. – Algorithms in number theory, in: van Leeuwen, J., A. Meyer, M. Nivat, M. Paterson and D. Perrin (eds) – Handbook of theoretical computer science, to appear; technical report 87-008, Department of Computer Science, The University of Chicago (1987).
9. Lenstra, Jr., H.W. – Factoring integers with elliptic curves, Ann. of Math., **126**, 649-673 (1987).
10. Lenstra, Jr., H.W. and R. Tijdeman (eds) – Computational methods in number theory, Math. Centre Tracts **154/155**, Mathematisch Centrum, Amsterdam (1982).
11. Odlyzko, A.M. – Discrete logarithms and their cryptographic significance, pp. 224-314 in: Beth, T., N. Cot and I. Ingemarsson (eds) – Advances in cryptology, Springer Lecture Notes in Computer Science, Vol. **209** (1985).
12. Pomerance, C. – Fast, rigorous factorization and discrete logarithm algorithms, pp. 119-143 in: Johnson, D.S., T. Nishizeki, A. Nozaki and H.S. Wilf (eds) – Discrete algorithms and complexity, Academic Press, Orlando, Florida (1987).
13. Schoof, R.J. – Quadratic fields and factorization, pp. 235-286 in: [10].
14. Seysen, M. – A probabilistic factorization algorithm with quadratic forms of negative discriminant, Math. Comp. **48**, 757-780 (1987).
15. Voorhoeve, M. – Factorization algorithms of exponential order, pp. 79-87 in: [10].
16. Wiedemann, D. – Solving sparse linear equations over finite fields, IEEE Trans. Inform. Theory, IT-32, 54-62 (1986).

The proof given in [FRF] can be repaired by incorporating a slightly changed version of [POM, Thm B'] in the proof of [SEY, Thm 5.2]. Let $S'(y)$ denote the set of primes $p$ for which $3 < p \leq y$ and $\frac{1}{\sqrt{p}}\psi_0(p, \exp((\log p)^{6/7})) > \exp(-(\log 4p)^{1/7}(\log\log 4p)/6)$, where $\psi_0(v, w)$ is the number of $w$-smooth integers in $(v - \sqrt{v}, v + \sqrt{v})$. So, $S'(y)$ consists of primes $p \leq y$ that have a high probability to be found by the elliptic curve method, in time depending in the usual way on $p$. As in the proof of [POM, Thm B'] one proves that $\pi(y) - S'(y) = O(y \cdot exp(-(\log y)^{1/6}/2))$. It follows that $\pi_\Delta(y) - S'_\Delta(y) = O(y \cdot exp(-(\log y)^{1/6}/2))$, where $\pi_\Delta(y)$ and $S'_\Delta(y)$ are the subsets of $\pi(y)$ and $S'(y)$, respectively, consisting of the primes $p$ for which $(\frac{\Delta}{p}) = 1$.

Let $\psi'_\Delta(x, y)$ be the number of integers $\leq x$ that are entirely built up from primes in $S'_\Delta(y)$. Our goal is to prove [SEY, Thm 5.2] with $\psi_\Delta$ replaced by $\psi'_\Delta$. With [SEY, Thm 5.3] (where one should read 'Li$(x)/2$' instead of 'Li$(x)$'), one gets $|S'_\Delta(x) - \mathrm{Li}(x)/2| = O(x \cdot exp(-(\log x)^{1/6}/2))$. This is a slightly weaker version of [SEY, Thm 5.3], but still sufficiently strong to let both applications of [SEY, Thm 5.3] in Seysen's proof of his [SEY, Thm 5.2] go through in our case. The first application does not require any changes. The second application, in the proof of 'our version' of [SEY, Cor 5.16], causes a slightly different error-term, and as a consequence somewhat different constants in the application of [SEY, Cor 5.16] in the proof of [SEY, (5.17)]. The resulting final inequality [SEY, (5.18)], however, remains unchanged. The proof of the version of [SEY, Thm 5.2] as needed in [FRF] follows.

CORRECTION