REFINED ANALYSIS AND IMPROVEMENTS ON SOME FACTORING ALGORITHMS

C.P. Schnorr
Fachbereich Mathematik
Universität Frankfurt*

Extended Abstract

Abstract. By combining the principles of known factoring algorithms we obtain some improved algorithms which by heuristic arguments all have a time bound $O(\exp \sqrt{c \ln n \ln \ln n})$ for various constants $c \geq 3$. In particular, Miller's method of solving index equations and Shanks's method of computing ambiguous quadratic forms with determinant -n can be modified in this way. We show how to speed up the factorization of n by using preprocessed lists of those numbers in $[-u,u]$ and $[n-u,n+u]$, $0<<u<<n$ which only have small prime factors. These lists can be uniformly used for the factorization of all numbers in $[n-u,n+u]$. Given these lists, factorization takes $O(\exp[2(\ln n)^{1/3}(\ln \ln n)^{2/3}])$ steps. We slightly improve Dixon's rigorous analysis of his Monte Carlo factoring algorithm. We prove that this algorithm with probability 1/2 detects a proper factor of every composite n within $o(\exp \sqrt{6 \ln n \ln \ln n})$ steps.

1. A Refined Analysis of Dixon's Probabilistic Factoring Algorithm.

So far the asymptotically fastest run time of a factoring algorithm has been proved by Dixon (1978). Given a composite number n, this algorithm finds a proper factor of n with probability 1/2 within $O(\exp(4\sqrt{\ln n \ln \ln n}))$ steps. ln denotes the "logarithmus naturalis" with the Eulerian number e as base and exp is the inverse function to ln. Dixon mainly applies the method of "combining congruences" to generate solutions of $x^2 = y^2 \bmod n$. In Sections 2 and 3 we will see that this technique can well be combined with factoring algorithms proposed by J.C.P. Miller (1975) and D. Shanks (1971). We give an outline of Dixon's algorithm with an improved analysis. We decrease the constant 4 in

* This work has been started in Summer 1980 during a stay at the Stanford Computer Science Department. Preparation of this report was supported in part by National Science Foundation grant MCS-77-23738 and by the Bundesminister für Forschung und Technologie.

Dixon's bound to $\sqrt{6}$. The improved theoretical time bound results from a tighter lower bound on the number of quadratic residues mod n which can be completely factored over small primes (Lemma 1) and a specific method for detecting small prime factors. Here we do not focus on designing the most practical algorithm but we like to prove a rigorous asymptotical time bound as small as possible. We do not assume any distribution on the input data but we assume that some intermediate data are chosen at random.

Dixon's algorithm.

begin input n

stage 1 $\nu = \lfloor n^{1/2r} \rfloor$
comment the optimal choice of $r \in \mathbb{N}$ will be made below.
Form the list P of all primes $\leq \nu$: $P = \{p_1, \ldots, p_{\pi(\nu)}\}$.
if $\exists\, p_i \in P : p_i \mid n$ then print p_i stop
$B := \emptyset$

stage 2 Choose $z \in [1, n-1]$ at random and independently from previous choices of z.
if $\gcd(z,n) \neq 1$ then print $\gcd(z,n)$ stop
$w := z^2 \bmod n$ with $0 \leq w < n$

stage 3 Compute $\underline{a} = (a_i \in \mathbb{N} \mid 1 \leq i \leq \pi(\nu))$ and w^* with $w = w^* \prod_{i \leq \pi(\nu)} p_i^{a_i}$ and $\forall p \in P : p$ does not divide w^*.

test 1 if $w^* \neq 1$ then goto stage 2
$B := B \cup \{\underline{a}\}$, $z_{\underline{a}} := z$
Try to find a nontrivial solution of

$$\sum_{\underline{a} \in B} f_{\underline{a}} \underline{a} = \underline{0} \bmod 2; \quad f_{\underline{a}} \in \{0,1\}. \tag{1}$$

test 2 if there is no nontrivial solution then goto stage 2

$$x := \prod_{f_{\underline{a}}=1} z_{\underline{a}}, \qquad y := \prod_{i \leq \pi(\nu)} p_i^{(\sum_{\underline{a} \in B} f_{\underline{a}} a_i)/2}$$

comment The construction implies $x^2 = y^2 \bmod n$; in case $x \neq \pm y \bmod n$, $\gcd(x \pm y, n)$ are proper factors of n.

test 3 if $x \neq \pm y \bmod n$ then print $\gcd(x \pm y, n)$ stop
Choose the first $\underline{a} \in B$ such that $f_{\underline{a}} = 1$.
$B := B - \{\underline{a}\}$, goto stage 2

end

Obviously a proper factor of n has been found as soon as test 3 succeeds. In the following analysis of the algorithm we suppose that n is an odd number with prime factor decomposition:

$$n = \prod_{i=1}^{d} q_i^{l_i} \qquad l_i \geq 1 \text{ and } d \geq 2.$$

Clearly the cases that n is even or a pure prime power can easily be handled in advance. The following facts are due to Dixon.

Fact 1. prob(x=±y mod n within test 3)=2^{1-d} and the corresponding events for distinct passes of test 3 are mutually independent.

Let T(n) be the total time of the algorithm and let $T_3(n)$ be the time till the first pass of test 3. We count arithmetical steps mod n as single steps. T(n), $T_3(n)$ are random values depending on the random variables z of stage 2. Fact 1 immediately implies:

Fact 2. $E[T(n)] = (1-2^{1-d})^{-1} E[T_3(n)] \leq 2E[T_3(n)]$.
Here E[X] denotes the expectation of the random value X. Let $T_1(n)$ ($T_2(n)$, resp.) be the time spent from any entering of stage 2 till the first pass of test 1 (test 2, resp.) without counting the steps used to solve the various linear systems of equations (1). Since a linear dependence of the $\underline{a}$ with $\underline{a} \in B$ must exist as soon as $\#B \geq \pi(\nu)+1 = O(\nu/\ln\nu)$ it follows that there are at most $\pi(\nu)+1$ passes of test 2 before the first pass of test 3. Hence

Fact 3. $E[T_3(n)] \leq (\pi(\nu)+1)E[T_2(n)]+O(\pi(\nu)^3)$.
Here $O(\pi(\nu)^3)$ bounds the steps to solve all the linear systems (1) occurring in the various passes of stage 3. Indeed this task amounts to solve one system of linear equations with $\pi(\nu)+1$ unknowns. In order to analyze $E[T_2(n)]$ we define

$$Q := \{\text{set of quadratic residues mod } n\} \cap \mathbb{Z}_n^*$$
$$T(n,\nu) := \{r \in [1,n] : \text{all prime factors of } r \text{ are} \leq \nu\}$$
$$M(n,\nu) := \{z \in [1,n] : z^2 \bmod n \in Q \cap T(n,\nu)\}.$$

Let $\varphi(n) = \#\mathbb{Z}_n^*$ be the Eulerian function.

Fact 4. $E[T_2(n)] \leq O(E[T_1(n)]\,\varphi(n)/\#M(n,\nu))$.
Proof. We clearly have prob(w =1)=$\#M(n,\nu)/\varphi(n)$. Hence test 1 will atmost be passed about $\varphi(n)/\#M(n,\nu)$ times between two passes of test 2. □

$T_1(n)$ depends on how the factorization of w over the prime base P is done. An obvious bound is as follows:

Fact 5. $E[T_1(n)] \leq \pi(\nu)+\log n$.

Here log n bounds the number of multiple prime factors of n according to their multiplicity.

So far Facts 1-5 yield under the assumption $\log n \le \pi(\nu)$:

$$E[T(n)] \le O\left(\pi(\nu)^2\left[\frac{n}{\#M(n,\nu)} + \pi(\nu)\right]\right) \qquad (2)$$

and it remains to prove a sharp lower bound on $\#M(n,\nu)$. This will be our main improvement over Dixon's analysis. Let $\kappa: \mathbb{Z}_n^* \to \{\pm 1\}^d \approx \oplus_{i=1}^d \mathbb{Z}_2$ be the quadratic character, defined as follows. For $a \in \mathbb{Z}_n^*$ let $\kappa(a) = (e_1,\ldots,e_d)$ with $e_i = \left(\frac{a}{q_i}\right)$. By definition the Jacobi symbol $\left(\frac{b}{q}\right)$ is 1,(-1, resp.) if b is a quadratic residue (non-residue) mod q. It is well known that $\kappa: \mathbb{Z}_n^* \to \oplus_{i=1}^d \mathbb{Z}_2$ is a group homomorphism and $a \in Q$ iff $\kappa(a)$ is the group unit $(1,1,\ldots,1) \in \{\pm 1\}^d$.

Lemma 1. $\#M(n,\nu) \ge \pi(\nu)^{2r}/(2r)!$ for all natural numbers r with $\nu^{2r} \le n$ provided all prime factors of n are $>\nu$.

Proof. Let $T_r(m,\nu) := \{w \in [1,m] \mid w = \Pi_{p_i \le \nu} p_i^{a_i} \wedge \Sigma_i a_i = r\}$. Since all prime factors of n are $>\nu$ we have $T_r(\sqrt{n},\nu) \in \mathbb{Z}_n^*$. We partition $T_r(\sqrt{n},\nu)$ into classes $T_i, i=1,\ldots,2^d$ according to the 2^d possible values of κ. Then

$$\bigcup_{i=1}^{d} T_i T_i \subset T_{2r}(n,\nu) \cap Q.$$

Since for each $w \in T_{2r}(n,\nu) \cap Q$, $\#\{z \in \mathbb{Z}_n^* \mid z \bmod n\} = 2^d$ it follows

$$\begin{aligned} \#M(n,\nu) &\ge 2^d \#(T_{2r}(n,\nu) \cap Q) \\ &\ge 2^d \Sigma_{i=1}^{2^d} \#T_i^2 \frac{r!^2}{(2r)!} \end{aligned} \qquad (3)$$

Here $(\#T_i)^2$ counts the number of ordered pairs $(w_1,w_2) \in T_i \times T_i$ and $(2r)!/(r!)^2$ bounds for each $w \in Q$ the number of distinct pairs $(w_1,w_2) \in \cup_i T_i \times T_i$ that yield the product $w_1 w_2 = w$. The Cauchy Schwarz inequality implies

$$\Sigma_{i=1}^{2^d} (\#T_i)^2 \ge 2^{-d} (\Sigma_i \#T_i)^2 = 2^{-d} \#T_r(\sqrt{n},\nu)^2 \qquad (4)$$

(use $\Sigma_i u_i^2 \cdot \Sigma_i v_i^2 \ge (\Sigma_i u_i v_i)^2$ with $u_i = \#T_i, v_i = 1$).

Obviously we have $\#T_r(\sqrt{n},\nu) = \binom{\pi(\nu)+r-1}{r} \ge \pi(\nu)^r/r!$, since $\binom{\pi(\nu)+r-1}{r}$ is

the number of possibilities of choosing with repetitions r elements out of $\pi(\nu)$. Finally we obtain from (3), (4):

$$\#M(n,\nu) \geq \#T_r(\sqrt{n},\nu)\ \frac{r!^2}{(2r)!} \geq \frac{\pi(\nu)^{2r}}{r!^2}\ \frac{r!^2}{(2r)!} = \frac{\pi(\nu)^{2r}}{(2r)!} \quad \blacksquare$$

Putting (2) and Lemma 1 together we obtain

$$E[T(n)] = O\left(\pi(\nu)^2 \left[\frac{n(2r)!}{\pi(\nu)^{2r}} + \pi(\nu)\right]\right)$$

provided $\log n \leq \pi(\nu)$ and $\nu^{2r} \leq n$. Using $\nu = n^{1/2r}$, $\nu \ln\nu \leq \pi(\nu) \leq 2\nu/\ln \nu$ (which follows from the prime number theorem) and $(2r)! = O(\sqrt{2r}(2r)^{2r}e^{-2r})$ (which follows from Stirling's formula) we obtain

$$E[T(n)] = O\left(\frac{(4r)^2 n^{1/r}}{(\ln n)^2}\left[\sqrt{2r}\, e^{-2r}(\ln n)^{2r} + \frac{4rn^{1/2r}}{\ln n}\right]\right) \quad (5)$$

We choose $r \in \mathbb{N}$ as to minimize $n^{1/r}(\ln n)^{2r}$. This implies

$$r = \frac{1}{\sqrt{2}}\sqrt{\frac{\ln n}{\ln \ln n}} + \varepsilon\,, \quad |\varepsilon| \leq 1/2$$

and

$$n^{1/r}(\ln n)^{2r} = O(\ln n \exp\sqrt{8 \ln n \ln \ln n}).$$

This finally yields the

<u>Proposition 1.</u>

$$E[T(n)] = O\left(\frac{\sqrt{2r}\, e^{-2r}}{\ln \ln n} \exp\sqrt{8 \ln n \ln \ln n}\right)$$

$$= o(\exp\sqrt{8 \ln n \ln \ln n}).$$

The asymptotic behaviour of this bound is quite attractive for excessively large n: n can be factored within $n^{\varepsilon(n)}$ steps with $\varepsilon(n) \to 0$ for $n \to \infty$. However, for reasonably sized values the exponent $\varepsilon(n)$ is not much smaller than 0.5 and the algorithm is not practical.

Can the above analysis of Dixon's algorithm still be refined leading to a constant in the exponent which is smaller than $\sqrt{8}$? We discuss two main points, (a) the tightness of our lower bound on $\#M(n,\nu)$ in Lemma 1, (b) the use of more sophisticated factoring algorithms for factoring w over the prime base P in stage 2.

We clearly have $\#M(n,\nu) \leq \psi(n,\nu) := \#\{w \in [1,n]$: all prime factors of w are

$\leq \nu\}$ The asymptotic behavior of $\psi(n,\nu)$ has been analyzed for a long time, see De Bruijn (1966) and Knuth, Trabb Pardo (1976). However, no exact values of $\psi(n,n^{1/2r})$ have been published for large n, say $n=2^{2^\nu}$ $\nu=7,8,9$ and reasonable r, say $4\leq r\leq 10$.

Instead of using within stage 2 the straightforward factoring algorithm that leads to Fact 5 we could use one of Pollard's algorithms that finds actors $\leq\nu$ of n in about $O(\sqrt{\nu})$ steps. By computational experience, Pollard's ρ-method (1975) detects factors $\leq\nu$ of n in $O(\sqrt{\nu \ln \nu})$ arithmetical steps mod n, see Guy (1975) and Knuth (1980). This method is highly practical although no rigorous theoretical time bound is known so far. Recently Brent succeeded in factoring $F_8=2^{2^8}+1$ by a variant of this method. Pollard (1974) also proposed a second method with a rigorous time bound. He computes for sufficiently many small $a\in\mathbb{Z}_n^*$, $\gcd(\Pi_{\nu\leq\sqrt{\nu}}(a^{\nu\sqrt{\nu}}-a^{-\mu}),n)$ for $\mu=1,2,\ldots,\nu$. For fixed a, these gcd-values can be computed by the fast Fourier transform within $O(\sqrt{\nu}(\ln\nu)^2\ln\ln\nu)$ steps. In total, Pollard obtains a worst case time bound $O(\nu^{0.5+\varepsilon})$ for arbitrarily small $\varepsilon>0$, but the constant factor, expresses by O, increases in an unknown way as ε decreases. We give a similar but slightly stronger result, see Schnorr (1980) for a detailed proof, also compare Straßen (1976).

Lemma 2. The smallest prime factor $\leq\nu$ of n can be found in $O(\sqrt{\nu}(\ln\nu)^2\ln\ln\nu)$ arithmetical steps mod n, provided $\ln n=O(\ln\nu)^2$.

Using the above procedure in stage 3 of Dixon's algorithm for factoring w over primes $\leq\nu$ clearly improves Fact 5 to

Fact 6. $T_1(n) = O(\nu(\ln\nu)^2\ln\ln\nu)$.

This finally improves the bound of proposition 1 to $E[T(n)] = (\exp\sqrt{\ln n \ln\ln n})$. Thus we obtain the

Theorem 1. For each composite n let $E[T(n)]$ be the expected time that the above algorithm takes to find a proper factor of n. Then for all n

(1) $E[T(n)] = o(\exp\sqrt{6 \ln n \ln\ln n})$.

(2) The event that the algorithm does not find a proper factor of n within $kE[T(n)]$ steps has probability $\leq 2^{-k}$.

Statement (2) is an immediate consequence of the fact that the distinct events of "test 3" (test 1,resp.) failing" are mutually independent. A more practical factoring algorithm is obtained if the quadratic re-

sidues w in stage 2 are produced via the continuous fraction method (see Morrison and Brillhart, 1975) which implies $w=O(\sqrt{n})$ and if Pollard's ρ-method is used for detecting small prime factors of w.

Under the assumption

(AO) the continuous fraction of $\sqrt{n}$ generates quadratic residues mod n which are uniformly distributed in $[1,O(\sqrt{n})]$

the time bound (5) transforms into a time bound

$$E[T(n)] = O\left(n^{3/4r} \ln n\, e^{-r}(\ln n)^r + n^{3/2r}\left(\frac{2r}{\ln n}\right)^3\right) \quad (8)$$

with r even, for the Morrison-Brillhart method. By choosing

$$r = 2\left\lceil \frac{1}{4}\sqrt{\frac{3 \ln n}{\ln \ln n}} \right\rceil$$

we obtain

$$n^{3/4r}(\ln n)^r = O((\ln n)^2 \exp\sqrt{3 \ln n \ln \ln n})$$

$$n^{3/2r} = O(\exp\sqrt{3 \ln n \ln \ln n}).$$

By (8) this implies

Corollary 1. [Assume (AO)]. The Morrison-Brillhart method runs in average time $o(\exp\sqrt{3 \ln n \ln \ln n})$.

This last method is really practical. Wunderlich (1979) obtained average runtimes around $322n^{0.152} \approx n^{0.21}$ for $n \approx 10^{40}$.

2. An Analysis and Revision of J.C.P. Miller's Factoring Method.

J.C.P. Miller (1975) proposed a factoring method based on the computation of indices. We shall develop a slightly improved version of Miller's method which turns out to be quite similar to the previously analyzed Dixon algorithm. Under reasonable heuristic assumptions the runtime of our version of Miller's algorithm will be $O(\exp\sqrt{4.5 \ln n \ln \ln n})$. In particular Miller's method does not yield an independent factoring algorithm but merely a specific modification of the method of "combining congruences mod n". However, as we shall point out, this modification has some decisive advantages in the case that one likes to factor many numbers in the same range. So far all known factoring algorithms collect data which are only useful for factor-

ing one specific number. For instance the congruences collected in Dixon's algorithm cannot be used for different n's. This observation also applies to the factoring algorithms of Morrison-Brillhart (1975), Schroeppel (unpublished, see Monier 1980), Shanks (1971, 1974), and Pollard (1974, 1975). In our version of Miller's method we will collect products of small prime numbers which are near to the number n to be factored. These products of small primes can be uniformly used for factoring all numbers near to n. For the connection to Miller's method, see Schnorr (1980).

As an example, let n = 1037

stage 1: Generate many distinct representations of n or multiples of n as a sum or difference of two products of small primes. For instance we have

$$\begin{aligned} 1037 &= 2^8 5 - 3^5 \qquad \text{i.e.} \qquad 2^8 5 = 3^5 \bmod n \\ &= 2\cdot 3\cdot 5^2\cdot 7 - 13 \qquad 2\cdot 3\cdot 5^2\cdot 7 = 13 \bmod n \\ &= 2^2 3^5 + 5\cdot 13 \qquad 2^2 3^5 = -5\cdot 13 \bmod n \\ &= 3\cdot 7^3 + 2^3 \qquad 3\cdot 7^3 = -2^3 \bmod n \end{aligned}$$

We obtain by multiplying the above congruences:

$$2^{11} 3^7 5^3 7^4 = 2^3 3^5 5\cdot 13^2 \bmod n$$

Since no prime of our base divides n, this yields

$$2^8 3^2 5^2 7^4 = 13^2 \bmod n.$$

From $2^4\cdot 3\cdot 5\cdot 7^2 = 353 \bmod n$ we obtain

$$353^2 = 13^2 \bmod n$$

which gives us the proper factors

$$\begin{aligned} &\gcd(353 - 13, n) = 17 \\ &\gcd(353 + 13, n) = 61. \end{aligned}$$

A formal description of our method is as follows.

<u>begin</u> input n
$\nu := n^{1/2r}, u := n^{d/2r}$
<u>comment</u> the optimal choice of r and d will be made below
Form the list $P = \{p_0, p_1, \ldots, p_{\pi(\nu)}\}$ of all primes $\leq \nu$, including $p_0 = -1$

if $\exists p_i \in P, i \geq 1 : p_i \mid n$ then print p_i stop

stage 1 Compute the lists

$$L := \left\{ (w,\underline{a}) \;\middle|\; |w| \leq u,\ w = \Pi_i p_i^{a_i},\ \underline{a} = (a_i \mid 0 \leq i \leq \pi(v)) \right\}$$

$$\tilde{L} := \left\{ (n+w,\underline{b}) \;\middle|\; |w| \leq u,\ w = \Pi_i p_i^{b_i},\ \underline{b} = (b_i \mid 0 \leq i \leq \pi(v)) \right\}$$

$$B := \{ (\underline{a},\underline{b}) \mid \exists\, w : (w,\underline{a}) \in L \wedge (n+w,\underline{b}) \in \tilde{L} \}$$

stage 2 Find a nontrivial solution $(f_{(\underline{a},\underline{b})} \mid (\underline{a},\underline{b}) \in B)$ of

$$\Sigma_{(\underline{a},\underline{b}) \in B} f_{(\underline{a},\underline{b})} (\underline{a},\underline{b}) = \underline{0} \bmod 2,\ f_{(\underline{a},\underline{b})} \in \{0,1\}.$$

test 2 if no solution exists then increase u goto stage 1

$$x := \prod_{i \leq \pi(v)} p_i^{\left(\Sigma_{(\underline{a},\underline{b}) \in B} f_{(\underline{a},\underline{b})} a_i \right)/2}$$

$$y := \prod_{i \leq \pi(v)} p_i^{\left(\Sigma_{(\underline{a},\underline{b}) \in B} f_{(\underline{a},\underline{b})} b_i \right)/2}$$

comment the construction implies $x^2 = y^2 \bmod n$.

test 3 if $x \neq \pm y \bmod n$ then print $\gcd(x \pm y, n)$ stop
Choose the first $(\underline{a},\underline{b}) \in B$ such that $f_{(\underline{a},\underline{b})} = 1$
$B := B - \{(\underline{a},\underline{b})\}$ goto stage 2.

end

This algorithm is virtually very similar to the one of Dixon, and on the other hand it is an improved version of Miller's method.

The time analysis of this algorithm will be based on the following assumptions.

(A1) The ratio of the number of times of "test 3 failing" to "test 3 succeeding" is bounded.

(A2) The numbers which are completely factorizable over P are independently distributed in $[-u,u]$ and $[n-u,n+u]$. These numbers have about the same frequency in $[n-u,n+u]$ and $[0,n]$ for $0 \ll u \ll n$.

Under these assumptions we obtain

Theorem 2. [Assume (A1), (A2)] The above algorithm has time bound

$$O(\exp \sqrt{4.5 \ln n \ln \ln n}).$$

One interesting feature of the above algorithm is that the main work in stage 1, namely the construction of the lists $L, \tilde{L}$ is almost independent from n. These lists can be used uniformly for the factorization of all numbers in $[n-u,n+u]$, $u=n^{d/2r}$. In particular, if someone has factored n he already has collected the data to easily factor each number near to n. Considering the problem of factorizing many numbers in $[n-u,n+u]$ we will assume that the lists $L, \tilde{L}$ are built up once for ever and that they are sorted with respect to the first component of the elements $(w,\underline{a})$ and $(n+w,\underline{b})$.

Theorem 3. [Assume (A1), (A2)] Given $L,\tilde{L}$, the time bound of the algorithm is

$$T(n) = O(\exp(2(\ln n)^{1/3}(\ln \ln n)^{2/3})).$$

This theorem can be interpreted as follows. Suppose we like to factor all numbers in $[n-u,n+u]$, $u=n^{d/2r}$ and let the cost to preprocess the lists $L,\tilde{L}$ be uniformly distributed to the numbers in $[n-u,n+u]$. Then the factorization of every specific number in $[n-u,n+u]$ accounts for $O(\exp 2(\ln n)^{1/3}(\ln \ln n)^{2/3})$ steps.

3. Improvements on a Method of Shanks.

Shanks (1971) proposed a factoring method which starts by computing the group of equivalence classes of primitive quadratic forms with discriminant -n and in particular he computes the order h(-n) of this group. Then he factors n by constructing a non-trivial ambiguous class. Under the implicit assumption that the entire group of classes is generated by small "prime" forms, and by neglecting log n factors, Shanks proves a time bound of about $O(n^{1/4})$.

We propose a way to construct ambiguous classes without evaluating h(-n) at all. We exploit the fact that ambiguous forms can be constructed mainly in the same way as we generate solutions of $x^2=y^2 \bmod n$, by the method of combining congruences. Under reasonable assumptions this yields an asymptotical time bound $O(\exp \sqrt{3 \ln n \ln \ln n})$.

We summarize some basic facts on binary quadratic forms. The form

$ax^2+2bxy+cy^2$ with $a,b,c \in \mathbb{Z}$ will be described by the triple (a,b,c). Two forms (a,b,c) and $(\bar{a},\bar{b},\bar{c})$ are equivalent if there exist linear transformations with integer coefficients and determinant 1 transforming the one form into the other. Equivalent forms have the same determinant $D := b^2-ac$. A form (a,b,c) is (properly) primitive if $\gcd(a,2b,c)=1$.

Henceforth we will restrict all considerations to forms with negative determinants $D=b^2-ac<0$. In this case the equivalence classes can be characterized by reduced forms. A form (a,b,c) is reduced if $2|b| \le |a| \le |c|$. There is a gcd-like algorithm which, given (a,b,c) computes an equivalent reduced form within $O(\ln|abc|)$ arithmetical steps.

Theorem 4. [Gauss, Artikel 172.] In every equivalence class H with $D<0$ there is either exactly one reduced form (a,b,c) or exactly two reduced forms $(a,\pm b,c)$. In the latter case, H is called ambiguous.

A form with $D<0$ either satisfies $a,c>0$ or $a,c<0$. It is called positive in the first and negative in the second case. The number of equivalence classes with determinant D is finite since a reduced, positive form (a,b,c) always satisfies $2|b| \le a \le \sqrt{4|D|/3}$.

Gauss (1801) introduced the composition of (binary) quadratic forms and proved that the equivalence classes with fixed determinant D form an abelian group, say $QF(D)$, under composition. Given two classes H_1, H_2 represented by their reduced forms, the reduced form of $H_1 \cdot H_2$ can be computed within $O(\ln|D|)$ arithmetical steps over numbers $\le |D|$. The forms which are primitive and positive generate a subgroup of $QF(D)$ which we call $QFP(D)$. The unit element I of the group is represented by $(1,0,-D)$.

The following assertions are equivalent: (1) H is ambiguous, (2) $H \cdot H = I$, (3) every form (a,b,c) in H is equivalent to $(a,-b,c)$, (4) there is a form (a,b,c) in H such that $a|2b$.

The reduced form of an ambiguous class is of either of the following three types:

$$b = 0 \quad \text{or} \quad a = 2b \quad \text{or} \quad a = c.$$

We call these forms ambiguous, they always represent ambiguous classes. These three types of ambiguous forms yield the following factorizations of the determinant:

$$-D = ac\,, \quad -D = b(2c-b)\,, \quad -D = (a-b)(a+b).$$

In this way the problem of factoring n reduces to the construction of ambiguous forms with determinant -n. It is important that Gauss has established a strong correspondence between the factorizations of n and the ambiguous classes in QFP(-n).

We only report the case n odd, since we like to factor only odd numbers.

A pair $(n_1,n_2) \in \mathbb{N}^2$ is an admissible factor pair for n if $n=n_1 \cdot n_2, n_1 < n_2$ and $\gcd(n_1,n_2)=1$. Suppose n has (exactly) l distinct prime factors, then there are (exactly) 2^{l-1} admissible factor pairs for n.

Theorem 5. [Gauss, Artikel 257,258.] Suppose $n \in \mathbb{N}$ is odd and has $l \geq 1$ distinct prime factors. Then there are 2^{l-1} or 2^l ambiguous classes in QFP(-n) according to whether n=3mod4 or n=1mod4. Each of the 2^{l-1} admissible factor pairs of n is obtained by the reduced form of exactly one in case n=3mod4 (two in case n=1mod4) of these ambiguous classes.

The remaining point to be discussed for the factorization of n is how to generate ambiguous classes in QFP(-n). This will be done by exploiting the group structure of QFP(-n). It can easily be seen that

$$[(a,b,c)][(a,-b,c)] = I.$$

Fact 8. Let $[(a,b,c)] \in QFP(-n)$ and let $a=\Pi_i p_i^{a_i}$ be the prime factorization of a, then $[(a,b,c)]=\Pi_i[(p_i^{a_i},b_i,c_i)]$ with $b_i := b \bmod p_i^{a_i}$ and $c_i := (b_i^2+n)/p_i^{a_i}$.

The possibly occurring factors $[(p_i^{a_i},b_i,c_i)]$ in Fact 8 can be characterized as follows.

Lemma 3. Let p be prime, $p \neq 2$, $\gcd(p,n)=1$ and $a \geq 1$. There exists $[(p^a,b,c)] \in QFP(-n)$ with integers b,c iff $(\frac{-n}{p})=1$. If $(\frac{-n}{p})=1$ there are exactly two of these classes, namely $[(p^a,\pm b,(n+b^2)/p^a)]$ for b with $b^2 \equiv -n \bmod p^a$.

We denote one of the classes $[(p^a,\pm b,(n+b^2)/p^a)]$ occurring in Lemma 3 as $I_{p^a,n}$. Then the other class must be $(I_{p^a,n})^{-1}$. It is clear from the multiplication scheme that

$$\{(I_{p,n})^a,(I_{p,n})^{-a}\} = \{I_{p^a,n},(I_{p^a,n})^{-1}\}.$$

This implies that Fact 8 can be rewritten as follows.

Lemma 4. Let $[(a,b,c)] \in QFP(-n)$, a odd and let $a=\Pi p_i^{a_i}$ be the prime factorization of a. Then

$$[(a,b,c)] = \Pi_i (I_{p_i,n})^{a_i \cdot e_i} \text{ with } e_i = \pm 1.$$

In particular, factoring $[(a,b,c)] \in QFP(-n)$ as in Lemma 4 can be done roughly in the time which is necessary to factor a.

By means of Lemma 4 we can generate ambiguous forms with determinant -n mainly in the same way as congruences $x^2=y^2 \bmod n$ are produced by Dixon's factoring algorithm, see Schnorr (1980) for a detailed algorithm.

Example. n = 1037

We choose the factor base P= {3,13}, we have $(-n/5)=(-n/7)=(-n/11)= -1$. The corresponding classes are

$$I_3 = [(3,1,346)] \ , \ I_{13} = [(13,4,81)].$$

One obtains

$$I_3^4 = I_{13}^{-1}$$

$$I_{13}^2 = I_3^2.$$

Hence $I_{13} \cdot I_3^{-1}$ is ambiguous. The reduced form in this class is (34,17,39) which yields the factorization

$$1037 = 17(78 - 17) = 17 \cdot 61.$$

Observe that the factor base in this example is smaller than in the application of Miller's method in Section 2. Dixon's algorithm would require a larger factor base too. Indeed the factor base is so small since the primes p=5,7,11 are excluded because $(\frac{-n}{p}) = -1$.

In our analysis of the algorithm we will use the following heuristic assumptions.

(A3) $\#\{p \leq \nu : p \text{ prime}, (\frac{-n}{p}) = 1\} \geq \frac{\nu}{c \ln \nu}$ with c>0 fixed

(A4) every admissible factor pair of n corresponds to some ambiguous class which is generated by the $I_p, p \leq \nu$.

We choose

$$\nu = n^{1/2r} \ , \ r = 2 \left\lceil \frac{1}{4} \sqrt{\frac{3 \ln n}{\ln \ln n}} \right\rceil$$

and obtain n as a final result.

Theorem 4. [Assume (A3), (A4).] Suppose we factor a composite n via the construction of ambiguous forms with determinant -n as above, then for each n a proper factor of n will be found with probability 1/2 within $o(\exp\sqrt{3 \ln n \ln \ln n})$ steps.

The above factoring method can be interpreted as the continuous fraction method in case of negative determinants. Conversely, in case of positive determinants $D=b^2-ac>0$, there is a different concept of reduced forms and there are many equivalent reduced forms. According to Gauss, Artikel 183-187, the equivalent reduced forms can be developed into an even and symmetric period. The recursion for developing this period is the same as that for evaluating the period of the continuous fraction of $\sqrt{D}$. Shanks exploited this coincidence and proposed an algorithm to factor n by constructing an ambiguous form with positive determinant n. Shanks has a way to make giant steps within the period of equivalent reduced forms. This second algorithm of Shanks runs in about $O(n^{1/4})$ steps, see Lenstra (1980) for a more detailed exposition of this method.

Acknowledgement. I am greatly indebted to the Stanford Computer Science Department whose support enabled this work. In particular I thank D. Knuth for many hints and his efficient cooperation. J. Vuillemin communicated to me the thesis of L. Monier. I thank A. Schönhage for his many useful comments.

References

[1] Borodin, A. and Munro, I., The Computational Complexity of Algebraic and Numeric Problems, American Elsevier (1975)

[2] Brent, R.P., "Analysis of some new cycle finding and factorization algorithms." Department of Computer Science, Australian National University, (1979)

[3] de Bruijn, N.G., "On the number of positive integers $\leq x$ and free of prime factors $>y$, II", Nederl. Akad. Wetensch. Proc. Ser. A 69 (1966), 239-247

[4] Diffie, W. and Hellman, M., "New directions in cryptography", IEEE Trans. Information Theory IT-22 (1976), 644-654

[5] Dixon, J-D., "Asymptotical fast factorization of integers". Report Department of Mathematics, Carleton University, Ottawa (1978)

[6] Gauss, C.F., Disquisitiones Arithmeticae, Leipzig (1801). German translation: Untersuchungen über höhere Arithmetik, Springer, Berlin (1889)

|7| Guy, R.K., "How to factor a number", Proc. Fifth Manitoba Conference on Numberical Math. (1975), 49-90

|8| Knuth, D.E., The Art of Computer Programming, Volume 2, Seminumerical Algorithms, Addison-Wesley (1969), second edition (1981)

|9| Knuth, D.E. and Trabb Pardo, L, "Analysis of a simple factorization algorithm", Theoretical Computer Science 3 (1976), 321-348

|10| Legendre, A.M., Theorie des Nombres Tome I, Paris (1978) reprint Blanchard, Paris (1955)

|11| Lenstra, H.W., "On the calculation of regulators and class numbers of quadratic fields". Preprint University of Amsterdam (1980)

|12| Miller, J.C.P., "On factorization, with a suggested new approach", Math. Computation 29 (1975), 155-172

|13| Monier, L., "Algorithmes de factorisation d'entiers", Thèse d'informatique, Université Paris Sud (1980)

|14| Morrison, M.A. and Brillhart, J., "A method of factorization and the factorization of F_7" Math. Computation 29 (1975), 183-205

|15| Pollard, J.M., "Theorems on factorization and primality testing", Proc. Cambridge Phil. Soc. 76 (1974) 521-528

|16| Pollard, J.M., "A Monte Carlo method for factorization", BIT 15 (1975), 331-334

|17| Rabin, M.O., "Probabilistic algorithms in finite fields", SIAM J. Comp. 9, 2 (1980), 273-280

|18| Rivest, R.L., Shamir, A., and Adleman, L., "A method for obtaining digital signatures and public key cryptosystems", Comm. ACM 21,2 (1978), 120-126

|19| Rivest, R.L. and Pinter, R.Y., "Using hyperbolic tangents in integer factoring", MIT report (1979)

|20| Schnorr, C.P., Refined Analysis and Improvements on some factoring algorithms. Preprint. University Stanford, December 1980

|21| Schönhage, A. and Straßen, V., "Schnelle Multiplikation großer Zahlen", Computing 7 (1971), 281-292

|22| Shanks, D., "Class number, a theory of factorization and genera", Proc. Symp. Pure Math., American Math. Soc. 20, (1971) 415-440

|23| Sieveking, M., "An algorithm for division of power series", Computing 10 (1972), 153-156

|24| Straßen, V., "Einige Resultate über Berechnungskomplexität", Jber. Deutsch. Math.-Verein 78, H.1 (1976), 1-8

|25| Wunderlich, M.C., "A running time analysis of Brillhart's continuous fraction method", in Springer, Lecture Notes in Math.751 (1979), 328-342