A deterministic version of Pollard’s $p - 1$ algorithm

Bartosz Źrałek *

May 26, 2009

Abstract

We give a deterministic version of Pollard’s $p - 1$ integer factoring algorithm. More precisely, we prove that the integers factorable (partially or completely) in random polynomial time by the $p - 1$ algorithm are in fact factorable in deterministic polynomial time. We also give a simple arithmetic characterization (in terms of multiplicative structure) of a large class of integers $n$ (almost all) that can be factored in deterministic polynomial time if $\phi(n)$ is given and (partially) factored ($\phi$ denotes Euler’s totient function). Finally, we show that $O(\ln n)$ oracle calls for values of $\phi$ are sufficient to completely factor any integer $n$ in less than $\exp[(1 + o(1))(\ln n)^{\frac{2}{3}}(\ln \ln n)^{\frac{1}{3}}]$ deterministic time.

1 Introduction

The design and implementation of probabilistic algorithms confronts us not only with mathematical and philosophical, but also practical questions. Such algorithms can in theory run forever, failing to find a proper result. Moreover we do not even know if truly random events occur in a real life. Finally, generating "long" pseudorandom number sequences is a difficult task on standard hardware. That is why the derandomization of probabilistic algorithms (hopefully without assuming hypotheses such as the Extended Riemann Hypothesis - ERH) so much occupies computational complexity theory and, if we restrict ourselves to number-theoretic algorithms, computational number theory. Admittedly the most spectacular result in this direction is the Agrawal-Kayal-Saxena deterministic polynomial time algorithm for primality proving (AKS).

*Institute of Mathematics, Polish Academy of Sciences, 00-956 Warsaw, Poland. email: b.zralek@impan.gov.pl
In this article we show how to derandomize Pollard’s $p - 1$ integer factoring algorithm [12]. More precisely, we propose a deterministic algorithm, that given an integer $n$ divisible by a prime $p$ such that $p - 1$ is smooth, finds in polynomial time a nontrivial divisor (or a proof of the primality) of $n$. By iterating our algorithm and using the AKS test (to be precise the the Lenstra-Pomerance variant [9]), we get an algorithm that factors completely in deterministic polynomial time integers having at most one prime divisor $p$ such that $p - 1$ is not smooth (corollary 4.4). The proofs of correctness rely on techniques proposed by Pohlig and Hellman [11] (theorem 4.2) and Fellows and Koblitz [5] (theorem 4.3). The latter was developed by Konyagin and Pomerance [7].

Let $\phi$ denote Euler’s totient function. In the article we also point a simple arithmetic characterization (in terms of multiplicative structure) of a large class of integers $n$ (almost all) that can be factored in deterministic polynomial time if $\phi(n)$ is given and (partially) factored (theorems 6.4, 6.6, 6.7). The strong assumption that $\phi(n)$ is (partially) factored is motivated by two reasons. First, although it can be shown that an algorithm due to Miller [10] (see also [11]) factors completely in deterministic polynomial time almost all integers $n$ given solely the value of $\phi(n)$ (it is proved to work for every integer $n$ under the ERH), these integers cannot be simply described (without using Dirichlet characters). Second, one observes easily that the $k$-th iteration $\phi^k(n) = 1$ for some $k \leq 1 + \log_2 n$. Our first idea was therefore to suppose that we are given not only $\phi(n)$, but also the iterations $\phi^2(n), \ldots, \phi^k(n)$. Then we can try to factor $n = \phi^0(n)$ by induction, that is, by factoring $\phi^{k-1}(n)$ given the factorization of $\phi^k(n)$ and so on. We show that there is a deterministic algorithm based on this idea whose running time is bounded by $\exp[(1 + o(1))(\ln n)^{\frac{2}{3}}(\ln \ln n)^{\frac{1}{3}}]$ for every integer $n$ (corollary 7.4). Of course, we are very far from solving the original problem - settling the existence of an unconditional, deterministic polynomial time reduction of integer factoring to computing $\phi$. Still our result seems to be of some interest, as we are unaware of any previously proposed non-exponential deterministic reduction of this type.

2 Notation

$n = p_1^{e_1} \cdots p_k^{e_k}$ is the complete factorization of the odd integer $n$.

$\langle A \rangle_G$ is the subgroup of $G$ generated by $A$. $\text{ord}_G(a)$ is the order of $a$ in the group $G$. If $G = \mathbb{Z}_m^*$ we simplify these notations and write $\langle A \rangle_m$ and $\text{ord}_m(a)$.

2
$C_m$ is the cyclic group with $m$ elements.

$a_i$ denotes the $i$-th coordinate of $a \in \mathbb{Z}_n^* \cong \bigoplus_{j=1}^k \mathbb{Z}_{p_j}^*$.

$\mathbb{P}$ is the set of all prime numbers. $p, q, s$ denote prime numbers.

$p_-(m)$, respectively $p_+(m)$, is the least, respectively the largest, prime dividing $m$.

$(a, b)$, respectively LCM$(a, b)$, denotes the greatest common divisor, respectively the least common multiple, of the integers $a, b$.

$v_s(m)$ is the integer satisfying $s^{v_s(m)} \parallel m$.

$u \geq 1$ is a parameter.

Isprime$(\cdot)$ denotes the Lenstra-Pomerance variant of the AKS test.

$\phi$ denotes Euler’s $\phi$ function.

$\omega(m)$ is the number of distinct prime factors of $m$.

$\Omega(m)$ is the number of prime divisors of $m$ counted with multiplicity.

$\psi(x, z) = \# \{m \leq x : \forall p \in \mathbb{P} \ p \nmid m \Rightarrow p \leq z \}$.

We will make frequent use of the following theorem proved in [7].

**Theorem 2.1 (Konyagin, Pomerance)** If $n \geq 4$ and $2 \leq \ln^c n \leq n$ then

$$\psi(n, \ln^c n) > n^{1-\frac{1}{c}}$$

and always assume that its hypotheses are satisfied when $c$ is fixed (this is natural in the task of factoring $n$). In the last section another estimation of $\psi$ will be applied.

**Theorem 2.2 (Canfield et al.)** $\psi(x, x^{-u}) = xu^{-u+o(u)}$ uniformly for $u = u(x) \to +\infty$ and $u < (1 - \epsilon)\frac{\ln x}{\ln \ln x}$.

### 3 Pollard’s $p - 1$ factoring algorithm

We first sketch the ideas behind the probabilistic version of Pollard’s $p - 1$ factorization method. Let $n$ be an odd integer, not a prime power. Assume that we are given an integer $M$ such that $p-1 \mid M$ for some $p \mid n$ (for the moment we do not consider the issue of finding a suitable $M$). Choose $b \in \mathbb{Z}_n^*$. By Fermat’s little theorem we have $b^M = 1(p)$ and thus $d := (b^M - 1, n) > 1$. If additionally $d < n$ then $d$ is a nontrivial divisor of $n$. But what if $d = n$ i.e. $b^M = 1(n)$? We can pick another element of $\mathbb{Z}_n^*$. We can also hope to find a nontrivial factor of $n$ in the sequence $(b^{M_{l=1}} - 1, n)_{l=1}^{v_2(M)}$, as all square roots of 1 in $\mathbb{Z}_n^*$ are of the form $(\pm 1, \ldots, \pm 1) \in \mathbb{Z}_n^* = \bigoplus_{q \mid n} \mathbb{Z}_{\varphi(q)}^{v_q(n)}$.

It turns out that the expected number of random $b \in \mathbb{Z}_n^*$ needed to split $n$ does not exceed 2.
Theorem 3.1 (Rabin) Let \( n > 2 \) be odd, \( M \) even, \( F(M) = \{ b \in \mathbb{Z}_n^* : b^M \neq 1(n) \} \), \( S(M) = \{ b \in \mathbb{Z}_n^* : \exists 1 \leq l \leq v_2(M) < (b^M - 1, n) < n \} \). Then \( \#(F(M) \cup S(M)) \phi(n) \geq 1 - 2^{1-\omega(n)} \).

Note that we want not only \( M \) to be a multiple of \( p - 1 \) for some (a priori unknown) \( p \mid n \), but also \( \ln M \) to be relatively small (e.g. bounded by a fixed power of \( \ln n \)), so that raising to the power \( M \) (or \( M^2 l - 1 \)) modulo \( n \) does not take too much time. Suppose that \( n \) has a prime divisor \( p \) such that \( p - 1 \) is smooth, say \( p_+(p - 1) \leq \ln^u n \). Set \( M = \prod_{q \leq \ln^u n} q^{\lfloor \ln n / \ln q \rfloor} \). Then \( M \) satisfies the two conditions, since \( \ln M \leq \sum_{q \leq \ln^u n} \ln n / \ln q \ln q = \pi(\ln^u n) \ln n = O(\ln^{u+1} n / \ln \ln \ln n) \) from Chebyshev’s theorem. By contrast, there is no efficient method of finding \( M \) if \( n \) is not divisible by a prime \( p \) as above.

As before suppose that \( n \) is odd, divisible by at least two different primes \( p \) and \( q \). It is well known that if a multiple \( M \) of \( p - 1 \) is given then the previously described search for a nontrivial factor of \( n \) can be derandomized under the ERH. Without loss of generality assume that \( b^M = 1(n) \) for all \( b < 2 \ln^2 n \).

Theorem 3.2 (Bach) Suppose that the ERH is true. Let \( n \geq 2 \), \( \chi \) be a nonprincipal character modulo \( n \). There is an integer \( b < 2 \ln^2 n \) such that \( \chi(b) \neq 1 \).

Using this theorem, we can easily prove the existence of \( b < 2 \ln^2 n \) such that for some \( l \), \( b^{M'} - 1 \) is divisible by \( q \) or \( p \), but not both. We apply it with \( \chi \) induced by \( \left( \frac{\cdot}{p} \right) \), \( \left( \frac{\cdot}{q} \right) \), \( \left( \frac{\cdot}{pq} \right) \) when \( v_2(p - 1) > v_2(q - 1), v_2(p - 1) < v_2(q - 1), v_2(p - 1) = v_2(q - 1) \) respectively.

4 A deterministic variant of Pollard’s \( p - 1 \) factoring algorithm

Our basic framework is as follows. Let \( B = \{2, 3, \ldots, \lfloor \ln^2 n \rfloor\} \). Assume that we are given an integer \( M \) together with its complete factorization such that \( b^M = 1(n) \) for every \( b \in B \). We want to find a simple and not restrictive condition on \( n \) under which \( n \) is factorable in deterministic polynomial time in \( \ln n \) and \( \ln M \). The starting point is a reformulation of the primality criterion from [5]. We restate the argument for the completeness and clarity of exposition.
Theorem 4.1 (Fellows-Koblitz) \( n \) is prime if and only if the following conditions are satisfied.

(i) \( \text{ord}_p(b) = \text{ord}_n(b) \) for every \( b \in \mathcal{B} \) and \( p \mid n \).

(ii) \( \text{LCM}_{b \in \mathcal{B}}(\text{ord}_n(b)) > \sqrt{n} \).

**Proof.** Suppose \( n \) is prime. Condition (i) is then a tautology. We check condition (ii). Let \( H = \langle \mathcal{B} \rangle_n \). \( H \) is cyclic, since \( n \) is prime. Therefore \( \text{LCM}_{b \in \mathcal{B}}(\text{ord}_n(b)) = \#H \geq \psi(n, \ln^2 n) > \sqrt{n} \), where the last inequality follows from theorem [2.1].

Assume now that conditions (i) and (ii) are satisfied. Let \( p = p_-(n) \). We then have \( \text{ord}_p(b) = \text{ord}_n(b) \) for all \( b \in \mathcal{B} \) and thus \( \text{LCM}_{b \in \mathcal{B}}(\text{ord}_p(b)) = \text{LCM}_{b \in \mathcal{B}}(\text{ord}_n(b)) > \sqrt{n} \). However \( \text{LCM}_{b \in \mathcal{B}}(\text{ord}_p(b)) \mid p - 1 \). Consequently \( p > \sqrt{n} \), hence \( n \in \mathbb{P} \). \( \square \)

Let \( b \in \mathbb{Z}_n^*, p \mid n \). Recall that \( \text{ord}_p(b) < \text{ord}_n(b) \) is equivalent to \( p \mid b^{\text{ord}_n(b)} - 1 \) for some \( s \mid \text{ord}_n(b) \). If \( (b^{\text{ord}_n(b)} - 1, n) > 1 \) for some \( s \mid \text{ord}_n(b) \) then we say that \( b \) is a Fermat-Euclid witness for \( n \). Checking conditions (i) and (ii) therefore reduces to factoring the orders of the elements of \( \mathcal{B} \), which can be done efficiently under our assumption on \( M \). Now our task being not primality testing, but factorization, we would like conditions (i) and (ii) to be effective, that is violation of one of them to lead to a nontrivial divisor of \( n \). As noted above condition (i) is effective, unfortunately condition (ii) is not. However it is easy to see that if \( \text{LCM}_{b \in \mathcal{B}}(\text{ord}_n(b)) \leq \sqrt{n} \) then \( \langle \mathcal{B} \rangle_n \) is not cyclic.

Suppose for greater generality that \( \mathcal{B} \) is any "small" subset of \( \mathbb{Z}_n^* \) whose elements have "smooth" orders in \( \mathbb{Z}_n^* \). We will describe below an efficient deterministic algorithm that finds a nontrivial divisor of \( n \) if \( \langle \mathcal{B} \rangle_n \) is not cyclic or a generator of \( \langle \mathcal{B} \rangle_n \) otherwise. By induction, it is sufficient to restrict our attention to the case \#\( \mathcal{B} = 2 \), say \( \mathcal{B} = \{a, b\} \).

We assume temporarily that \( \text{ord}_n(a) = s^v \), \( b^{s^v} = 1 \), where \( s \in \mathbb{P} \). There exist an \( i \), \( 1 \leq i \leq k \), such that \( \text{ord}_n(a) = s^v = \text{ord}_{p_i}(a_i) \). Since \( \mathbb{Z}_{p_i}^* \) is cyclic, \( b_i \in \langle a_i \rangle_{p_i} \). Write \( b_i = a_i^{l_0 + l_1 s + \ldots + l_{v-1} s^{v-1}} \) for some (uniquely determined) \( l_m, 0 \leq l_m < s \). Then \( b_i^{s^{v-1}} = a_i^{l_0 s^{v-1}} \), that is \( p_i \mid b_i^{s^{v-1}} - a_i^{l_0 s^{v-1}} \).

We can thus find a \( j \), \( 0 \leq j < s \), such that \( d := (b_i^{s^{v-1}} - a_i^{j s^{v-1}}; n) > 1 \). If \( d < n \) then \( d \) is a nontrivial divisor of \( n \), so suppose that \( d = n \). Therefore \( b_i^{s^{v-1}} = a_i^{j s^{v-1}} \). In particular \( b_i^{s^{v-1}} = a_i^{j s^{v-1}} \) and \( j = l_0 \). Now reason by induction. Let \( m < v - 2 \), \( c = b a^{-(l_0 + \ldots + l_m s^m)} \); assume that \( c^{s^{v-m-1}} = 1 \). Then \( c_i = a_i^{l_m + s^{m+1} + \ldots + l_{v-1} s^{v-1}} \), hence \( p_i \mid c_i^{s^{v-m-2}} - a_i^{j s^{v-1}} \). As before, either we find a \( j \), \( 0 \leq j < s \), such that \( 1 < (c^{s^{v-m-2}} - a_i^{j s^{v-1}}, n) < n \), or
\[ c^{v-m-2} = a^{l_{m+1}} s^{v-1}. \] If this procedure does not lead to a nontrivial factor of \( n \) then \( b = a^{b_0 + \ldots + l_{v-1}} \) and thus \( b \in \langle a \rangle_n \). More formally we use the ensuing algorithm.

\[
\text{PH}(n, a, b, s, v, w) \; \{ a, b \in \mathbb{Z}_n^*, s \in \mathbb{P}, \text{ord}_n(a) = s^v, \text{ord}_n(b) = s^w \}
\]

1. If \( w > v \) then interchange \( a \) and \( b \)
2. \( c = b \)
3. For \( j = 1 \) to \( s - 1 \) compute \( a^{j s^{v-1}} \)
4. For \( m = 0 \) to \( v - 1 \) do
   \[ j = 0 \]
   \[ \text{While } (a^{j s^{v-1}} - c^{e^{v-m-1}}, n) = 1 \text{ do } j = j + 1 \]
   \[ \text{If } (a^{j s^{v-1}} - c^{e^{v-m-1}}, n) \neq n \text{ then return } (a^{j s^{v-1}} - c^{e^{v-m-1}}, n) \]
5. \( 0 \leq s \leq \min\{v, w\} \) and \( s \mid (A, B) \) finds a nontrivial factor of \( n \) then \( \langle a, b \rangle_n \) is cyclic. This algorithm uses \( O((\max(v, w) + s) \max(v, w) \ln^2 n \ln s) \) operations.

(i) Let \( a, b \in \mathbb{Z}_n^*, A = \text{ord}_n(a), B = \text{ord}_n(b) \). If none of the procedures \( \text{PH}(n, a, b, s, v, w) \), \( s \in \mathbb{P} \) and \( s \mid (A, B) \), finds a nontrivial factor of \( n \) then \( \langle a, b \rangle_n \) is cyclic.

**Proof.** (i) The correctness of \( \text{PH}(n, a, b, s, v, w) \) follows from the precedent discussion. Step (3) requires \( \sum s^{-1} j=1 O(\ln^2 n \ln(s^{v-1}))) = O(vs \ln^2 n \ln s) \) operations. The total number of operations used by step (6) in the loop (4) is \( O(v^2 \ln^2 n \ln s) \). Step (8) takes on the whole in the loop (4) \( \sum_{m=0}^{v-1} O(\ln^2 n \ln s^{m+1}) = O(v^2 \ln^2 n \ln s) \) operations. Hence the stated running time.

(ii) Suppose that none of the procedures \( \text{PH}(n, a, b, s, v, w) \), \( s \mid (A, B) \), finds a nontrivial factor of \( n \). We see from (i) that \( v_s(a) = v_s(b) \), else \( c_s = a^{v_s(A)} b^{v_s(B)} \) (\( s \in \mathbb{P}, s \mid AB \)). We claim that \( \langle a, b \rangle_n = \langle c_s : s \mid AB \rangle_n \). The inclusion \( \langle c_s : s \mid AB \rangle_n \subset \langle a, b \rangle_n \) being obvious, we justify the reverse. Let \( s \mid A \). If \( s \nmid B \) then \( v_s(A) \geq 1 > 0 = v_s(B) \) and \( a^{v_s(A)} = c_s \). If \( s \mid B \) then \( \langle c_s = a^{v_s(A)} \rangle \) holds. As the elements \( a^{v_s(A)} (s \mid A) \) have pairwise relatively prime orders in \( \mathbb{Z}_n^* \), we get \( \text{ord}_n(\prod_{s \mid A} a^{v_s(A)}) = \prod_{s \mid A} s^{v_s(A)} = A \). Therefore \( \langle a \rangle_n = \langle c_s : s \mid AB \rangle_n \).
\[ \langle \prod_{s \mid A} a^{\frac{4}{\varphi(s)}} \rangle_n \subset \langle c_s : s \mid AB \rangle_n. \] In a similar fashion \( b \in \langle c_s : s \mid AB \rangle_n \). Hence \( \langle a, b \rangle_n = \langle c_s : s \mid AB \rangle_n \). Since the elements \( c_s \in \mathbb{Z}_n^* \) have pairwise relatively prime orders, it follows that the group \( \langle a, b \rangle_n \) is cyclic, generated by
\[ c := \prod_{s \mid AB} c_s. \square \]

If we compute in step (3) \( a^{js^{v-1}}, a^{js^{v-1}\sqrt{s}} \) for \( j = 1, \ldots, \lfloor \sqrt{s} \rfloor \) and use Shanks’s baby-steps, giant-steps method in (6) to find \( j \) such that \( (a^{js^{v-1}} - c^{s^{v-1}} n) > 1 \) then the running time bound of \( \text{PH}(n, a, b, s, v, w) \) can be improved to \( O((\max(v, w) + \sqrt{s}\ln s)\max(v, w)\ln^2 n\ln s) \).

Turning back to our main question, we propose the following deterministic algorithm for splitting \( n \) given an integer \( M \) as in the beginning of this section.

\[
\text{Split}(n, M, s_1, v_1, \ldots, s_t, v_t) \quad \{ M = s_1^{v_1} \cdots s_t^{v_t} \text{ is the complete factorization of } M \}
\]

(1) For \( b = 2 \) to \( \lfloor \ln^2 n \rfloor \) do

(2) If \( b^M \neq 1(n) \) then return (‘failure’) and stop

(3) Compute \( B = \text{ord}_n(b) \)

(4) For every prime \( s \) dividing \( B \) do

(5) If \( (b^\frac{B}{s} - 1, n) > 1 \) then return \( (b^\frac{B}{s} - 1, n) \)

(6) If \( \text{LCM}(\text{ord}_n(2), \ldots, \text{ord}_n(\lfloor \ln^2 n \rfloor)) \approx \sqrt{n} \) then return \( n \text{ is prime} \)

(7) Let \( a = 2, A = \text{ord}_n(2) \)

(8) For \( b = 3 \) to \( \lfloor \ln^2 n \rfloor \) do

(9) For every prime \( s \) dividing \( (A, B) \), where \( B = \text{ord}_n(b) \), do

(10) \( \text{PH}(n, a^{\frac{4}{\varphi(s)}}, b^{\frac{B}{s}}, s, v_s(A), v_s(B)) \)

(11) If a factor \( d \) of \( n \) has been found then return \( d \)

(12) Let \( a = \prod_{\substack{s \mid A \mid B \atop \varphi(s) \neq 0}} c_s \), where

\[ c_s = b^{\frac{B}{s}} \text{ if } v_s(A) \leq v_s(B), \text{ else } c_s = a^{\frac{4}{\varphi(s)}}, \]

let \( A = \text{LCM}(A, B) \)

\textbf{Theorem 4.3} Let \( M = s_1^{v_1} \cdots s_t^{v_t} \) be the complete factorization of the integer \( M \), \( s_0 = \max\{s \mid M : \forall q \mid n \quad s \mid q - 1\} \) if \( \{s \mid M : \forall q \mid n \quad s \mid q - 1\} \neq \emptyset \), else \( s_0 = 0 \). Suppose that \( a^M = 1(n) \) for all \( a \in \{2, 3, \ldots, \lfloor \ln^2 n \rfloor\} \). Then the algorithm \( \text{Split}(n, M, s_1, v_1, \ldots, s_t, v_t) \) finds a nontrivial divisor (or a proof of the primality) of \( n \) in \( O(\ln^4 n(\ln^2 M + \ln^2 n + s_0 \ln n)) \) deterministic time.

\textbf{Proof.} Step (3) requires \( O(\ln^2 n \ln^2 M) \), step (4) \( O(\ln^4 n) \) operations. By
Step (2) requires, as was shown in \[9\], \(O(\ln^6 n \ln^c \ln n)\) operations for some constant \(c\). We have \(\ln M = O(\frac{\ln^{u+1} n}{\ln \ln n})\). From theorem \[4.3\] the running time of \(\text{Split2}(n, M, s_1, v_1, \ldots, s_t, v_t)\) is therefore \(O(\ln^{2u+6} n)\) operations. (ii) We obtain the complete factorization of \(n\) by iterating \(\text{Split2}(n, M, s_1, v_1, \ldots, s_t, v_t)\).
Factor$(n, M, s_1, v_1, \ldots, s_t, v_t)$

1. Let $d = \text{Split2}(n, M, s_1, v_1, \ldots, s_t, v_t)$
2. If $d = (x \text{ 'is prime'})$ then return $x$
3. Factor$(d, M, s_1, v_1, \ldots, s_t, v_t)$
4. Factor$(n/d, M, s_1, v_1, \ldots, s_t, v_t)$

Factor$(n, M, s_1, v_1, \ldots, s_t, v_t)$ calls Split2$(n, M, s_1, v_1, \ldots, s_t, v_t)$ $O(\ln n)$ times. Hence Factor$(n, M, s_1, v_1, \ldots, s_t, v_t)$ runs in $O(\ln^2 n + 7 n)$ time. □

Let us briefly compare the running times of the original Pollard $p - 1$ algorithm with the new version. The original algorithm finds a non-trivial divisor of $n$ in $O(\ln u + 3 n)$ random time under the assumption of corollary 4.4 (i). Our deterministic version also runs in polynomial time, but is considerably slower and is thus rather of theoretical than practical interest.

Of course, the obtained running time bound of Split$(n, M, s_1, v_1, \ldots, s_t, v_t)$ is polynomial in $\ln n$ and $\ln M$ for more integers $n$ than those considered in corollary 4.4. Let $D(n, u) = \max_{q > \ln^u n} \# \{ p \mid n \colon q \mid p - 1 \}$. We should expect that the integers $n$ for which $D(n, u) > 1$ are rare. This is in fact true. We prove slightly more than needed to motivate the ideas of section 6.

**Theorem 4.5** Let $l \in \mathbb{N}$. The number $B(x, u, l)$ of integers $n \leq x$ such that $D(n, u) > l$ is bounded above by $cx^{2u \ln^{l+1} \ln x}/\ln^u x$ where the constant $c$ does not depend upon $u$.

**Proof.** We have:

\[
B(x, u, l) \leq \sqrt{x} + \sum_{\sqrt{x} \leq n \leq x} \sum_{q > \ln^u n} \sum_{p_1 < \ldots < p_{l+1}} 1 \leq \sqrt{x} + \sum_{q > \ln^u x} \sum_{p_1 < \ldots < p_{l+1}} \sum_{n \leq x} 1
\]

\[
\sum_{n \leq x} \sum_{p_1 \ldots p_{l+1}} 1 = \sum_{p_1 \ldots p_{l+1} \leq x} \left[ \frac{x}{p_1 \ldots p_{l+1}} \right] \leq x \sum_{p_{l+1} \leq x} \frac{1}{p_1 \ldots p_{l+1}}
\]

\[
\leq x \left( \sum_{p \leq x} \frac{1}{p} \right)^{l+1} \leq c_1 x \ln^{l+1} \ln x / (q - 1)^{l+1}.
\]
where the last inequality follows from the uniform bound

\[
\sum_{p \leq x, p \equiv 1 (d)} \frac{1}{p} \leq \frac{c_0}{\varphi (d)} \ln \ln x
\]

(use summation by parts and apply the Brun-Titchmarsh inequality). Hence

\[
\sum_{q > 2^{-u}} \ln^u x \sum_{x < n \leq x} \prod_{p \mid n, p_i = 1 (q)} 1 \leq c_1 x \ln^{l+1} \ln x \sum_{q > 2^{-u}} \ln^u x \frac{1}{(q-1)^{l+1}}
\]

\[
\leq c_2 x \frac{2^{lu} \ln^{l+1} \ln x}{\ln^{lu} x}
\]

Thus

\[
B (x, u, l) \leq c_3 x \frac{2^{lu} \ln^{l+1} \ln x}{\ln^{lu} x}
\]

□

5 Some known reductions of factoring to computing \( \phi \)

Taking \( M = \phi (n) \) in theorem 3.1 we get the following classical result.

**Theorem 5.1 (Rabin)** Given \( \phi (n) \) we can completely factor \( n \) in \( O (\ln^4 n) \) expected time.

For reasons already explained at the end of section 3, substituting \( M = \phi (n) \) also gives

**Theorem 5.2 (Miller)** If the ERH holds, then given \( \phi (n) \) we can completely factor \( n \) in \( O (\ln^6 n) \) deterministic time.

While it is an open problem whether factoring unconditionally reduces in deterministic polynomial time to computing Euler’s \( \phi \) function, for some integers such a reduction is particularly easy. The simplest nontrivial examples are integers \( n \) with exactly two prime factors. Suppose first that \( n = pq \). Then \( p+q = n - \phi (n) + 1 \). Given \( \phi (n) \) we compute the right-hand side of this equality and find \( p \) and \( q \) by solving a quadratic equation. Now turn to the general case \( n = p^aq^b \), say \( p < q \). If \( p \nmid q-1 \) then \( \frac{n}{(n, \phi (n))} = pq \)
and \( \frac{\phi(n)}{(n, \phi(n))} = (p - 1)(q - 1) = \phi(pq) \), thus the previous method applies. If \( p \mid q - 1 \) then \( \frac{n}{(n, \phi(n))} = q \) and therefore \( q, \alpha, p \) will be obtained one after the other. It is worth making here a general observation. The value \( \phi(n) \) can be used to check whether a nontrivial factorization of \( n \) is a factorization into primes.

Lemma 5.3 (i) For every \( n \geq 2 \) we have \( \phi(n^\alpha) \leq n^{\alpha-1}(n - 1) \) with equality if and only if \( n \in \mathbb{P} \).

(ii) Suppose that \( n = \prod_{i=1}^{k} n_i^{\alpha_i} \), where the \( n_i \) are pairwise coprime integers greater than \( 1 \). Then \( \phi(n) = \prod_{i=1}^{k} n_i^{\alpha_i-1}(n_i - 1) \) if and only if each \( n_i \) is prime.

Proof. Showing (i) is straightforward. (ii) is a consequence of (i). □

S. Landau [8] has shown that computing the equal order factorization of any integer \( n \), that is the sequence \( n_i := \prod_{p: v_p(n_i) = i} p \), can be done in deterministic polynomial time given a '\( \phi \)-oracle' (this oracle finds instantly the values of Euler’s \( \phi \) function for \( O(\ln n) \)-bit inputs). In fact, if \( \omega(n) \geq 3 \) then \( O(\Omega(n) \ln^2 n) \) bit operations and at most \( \omega(n) - 2 \) oracle calls (including \( \phi(n) \)) are needed. Notice that if \( \omega(n_i) \leq 2 \) for all \( i \) then the additional calls \( \phi(n_i) \) will lead to the complete factorization of \( n \). For instance every integer \( n = p^\alpha q^\beta s^\gamma \), where \( p, q, s \) are distinct primes and \( \alpha, \beta, \gamma \) integers not all equal, can be, given \( \phi(n) \), completely factored in \( O(\ln^3 n) \) deterministic time.

6 Factoring almost all \( n \) in deterministic polynomial time when \( \phi(n) \) is given and (partially) factored

Recall the definition of \( D(n, u) \) from section 4. Fix the parameter \( u \). We have seen that \( D(n, u) \leq 1 \) for almost all integers \( n \). Moreover, \( n \) is factorable in deterministic polynomial time if \( D(n, u) \leq 1 \) and the integer \( \phi(n) \) as well as its complete factorization are known. It is natural to ask whether these undeniably strong assumptions can be weakened. To answer this question we first introduce some supplementary notations.

Let \( M \) be a divisor of \( \phi(n) \) and \( 0 < y \leq 1 \) a parameter. Set \( F(n, M, y) = \#\{p \mid n: (M, p - 1) \geq (p - 1)^y \} \), \( D(n, M, u) = \max_{q \mid \ln^2 n} \#\{p \mid n: q \mid p \} \).
(M, p − 1). From now on suppose that the integer M is given in a completely factored form. Then F(n, M, y) measures the part of the factorization of φ(n) that we know. If d is factor of n then D(n, M, u) measures the efficiency of the algorithm PH(d, a, b, s, v, w): it runs in polynomial time in ln n whenever ω(d) > D(n, M, u), s | M and s | p − 1 for every p | d.

In the three subsections below we will consider the problem of factoring n in deterministic polynomial time for different values of y, F(n, M, y) and D(n, M, u).

6.1 F(n, M, 1) ≥ ω(n) − 2, D(n, M, u) ≤ l (l ∈ N)

Let η > 0 be arbitrary. Assume that n has no prime divisor below ln l + 1 + η. For simplicity, we first suppose that F(n, M, 1) = ω(n). Assume that we have found a divisor d > 1 of n. We now describe a procedure that leads to a nontrivial divisor or a proof of the primality of d in deterministic polynomial time (in ln n). The complete factorization of n can be thus obtained by induction.

Let B_1 = {2, 3, ..., ⌊ln l + 1 + η⌋}, B_2 = {2, 3, ..., ⌊ln l + 1 + η⌋}. By our assumptions on n and M we have (b^M − 1, d) > 1 for every b ∈ B_2. We can suppose that b^M = 1(d) for all b ∈ B_2, because in the contrary case a nontrivial divisor of d is found.

We first check whether the set B_1 contains a Fermat-Euclid witness for d. If this is the case then we get a nontrivial divisor of d, so assume the contrary.

Let A_1 = LCM_{b ∈ B_1}(ord_d(b)). Then A_1 | p − 1 for every p | d. Suppose that p^+(A_1) ≤ ln^u n; note that this inequality is satisfied if ω(d) > l, since D(n, M, u) ≤ l. We follow the procedure from section 4 to test if ⟨B_1⟩_d is cyclic. We can suppose that this is the case, for otherwise we find a nontrivial divisor of d.

Lemma 6.1 Let B_1 = {2, 3, ..., ⌊ln l + 1 + η⌋}. Assume that ⟨B_1⟩_d is cyclic and none of the elements of B_1 is a Fermat-Euclid witness for d. Then ω(d) ≤ l.

Proof. A similar argument to that in the proof of theorem 4.1 shows that p^−(d) > d^{1/ln^u}. □

By the above lemma there is no loss of generality in assuming that ω(d) ≤ l. We test whether there is a Fermat-Euclid witness for d among the elements of B_2 \ B_1. For the same reasons as above, suppose that there is no such witness. We compute A_2 = LCM_{b ∈ B_2}(ord_d(b)). Therefore A_2 | p − 1 for every p | d. In particular, (A_2, d) = 1.
Lemma 6.2 Let $\mathcal{B}_2 = \{2, 3, \ldots, [ln^{l+1+\eta}d]\}$. $A_2 = LCM_{b \in \mathcal{B}_2}(\text{ord}_d(b))$. Assume that $\omega(d) \le l$ and $(A_2, d) = 1$. Then $A_2^{\omega(d)+1} > d^{1+\frac{1}{l(l+1+\eta)}}$.

Proof. Set $H_2 = \langle \mathcal{B}_2 \rangle_d$. Since $(A_2, d) = 1$, we have also $(\#H_2, d) = 1$. Hence $H_2 \le \bigoplus_{p|d} C_{p-1} \le \mathbb{Z}_d^*$. Therefore $H_2$ contains at most $\omega(d)$ linearly independent elements of order dividing $q^{\nu_q(A_2)}$ for each $q | A_2$. It follows that $A_2^{\omega(d)+1} \ge \#H_2 \ge \psi(d, ln^{l+1+\eta}d) > d^{1-\frac{1}{l(l+1+\eta)}}$. Thus $A_2^{\omega(d)+1} > d^{1+\frac{1}{l(l+1+\eta)}} = d^{1+\frac{1}{l(l+1+\eta)}} \ge d^{1+\frac{1}{l(l+1+\eta)}}$. \(\square\)

If $d < \left(\frac{l}{[l/2]}\right)\frac{ln^{l+1+\eta}}{\eta}$ we factor $d$ by trial division. Suppose that the opposite inequality holds. By lemma 6.2 we have $A_2^{\omega(d)+1} > d^{1+\frac{\eta}{l(l+1+\eta)}} \ge d\left(\frac{l}{[l/2]}\right)$. In the ensuing lemma (whose case $k = 3$ was considered in [7]) we show that the factorization of $d$ can be found by factoring a suitable polynomial in $\mathbb{Z}[x]$. We use the Hensel-Berlekamp algorithm [8, 3] to this end.

Lemma 6.3 Let $d = p_1^{e_1} \cdot \ldots \cdot p_k^{e_k}$. Assume $A \in \mathbb{N}$ divides $p_i - 1$ for $i = 1, \ldots, k$; $p_i = b_i A + 1$. Suppose in addition that $A^{k+1} > \left(\frac{k}{[k/2]}\right)^l d$. Write $d$ in base $A$: $d = 1 + a_1 A + \ldots + a_k A^k$. Then $1 + a_1 x + \ldots + a_k x^k = (b_1 x + 1) \cdot \ldots \cdot (b_k x + 1)$ in $\mathbb{Z}[x]$.

Proof. We have $d = p_1^{e_1} \cdot \ldots \cdot p_k^{e_k} = (b_1 A + 1)^{e_1} \cdot \ldots \cdot (b_k A + 1)^{e_k}$. Since $A^{k+1} > d$, it follows that $e_1 = \ldots = e_k = 1$. Hence $1 + a_1 A + \ldots + a_k A^k = (b_1 A + 1) \cdot \ldots \cdot (b_k A + 1) = 1 + \sum_{j=1}^{k} \omega_{k,j}(b_1, \ldots, b_k) A^j$, where

$$\omega_{k,j}(b_1, \ldots, b_k) = \sum_{1 \le i_1 < \ldots < i_j \le k} b_{i_1} \cdot \ldots \cdot b_{i_j}.$$ 

It is therefore sufficient to show that $0 \le \omega_{k,j}(b_1, \ldots, b_k) < A$ for every $j$, $1 \le j \le k$. By assumption, $A^{k+1} > \left(\frac{k}{[k/2]}\right)^l d$ and thus $b_1 \cdot \ldots \cdot b_k \left(\frac{k}{[k/2]}\right)^l d < b_1 \cdot \ldots \cdot b_k A^{k+1} < d A$. Hence $b_1 \cdot \ldots \cdot b_k \left(\frac{k}{[k/2]}\right)^l < A$ and it follows that $0 \le \omega_{k,j}(b_1, \ldots, b_k) \le \left(\frac{k}{j}\right)^l b_1 \cdot \ldots \cdot b_k \le \left(\frac{k}{[k/2]}\right)^l b_1 \cdot \ldots \cdot b_k < A$. \(\square\)

It remains to treat the slightly more general case $F(n, M, 1) \ge \omega(n) - 2$. We then have $\#\{p \mid n : p - 1 \nmid M\} \le 2$. It is therefore easy to see that iterated use of the preceding algorithm will lead either to the complete factorization of $n$, or a factorization $n = mp^e q^f$, where $(m, pq) = 1$, the complete factorization of $m$ is known, but $p, q, e, f$ are not. In the latter case we can easily compute $\phi(m)$ and $\phi(p^e q^f) = \phi(m) \cdot \frac{\phi(n)}{\phi(m)}$. Finally we get $p, q, e, f$
with the method described in section 5.2. Recall that, by lemma 5.3 (ii), we never had to call Isprime().

**Theorem 6.4** Let \( l \in \mathbb{N} , \eta > 0 \). Suppose that \( \phi(n) \) is given together with a divisor \( M \) of \( \phi(n) \) in a completely factored form, such that \( F(n, M, 1) \geq \omega(n) - 2 \) and \( D(n, M, u) \leq l \). Then the complete factorization of \( n \) can be found in \( O(n^{\log \omega(n) + 1/\eta}) \) deterministic time.

**Proof.** Such an algorithm has just been discussed. Its running time analysis may be easily verified. \( \square \)

### 6.2 Formula for \( F(n, M, \frac{1}{3} + \epsilon) = \omega(n) \), \( D(n, M, u) \leq 1 \)

Choose \( \eta > 0 \) arbitrary. Assume that \( n \) is squarefree and has no prime divisor below \( \ln \frac{1}{3} + \eta \). Let \( d > 1 \) be a divisor of \( n \). We show how to find a nontrivial factor or a proof of the primality of \( d \) in deterministic polynomial time (in \( \ln n \)). Set \( M_p = (M, p - 1), p - 1 = M_p N_p \) for all \( p \mid d \). There is no loss of generality in assuming that \( (M, N_p) = 1 \) and consequently \( (M_p, N_p) = 1 \) for each \( p \mid d \). Hence \( \mathbb{Z}_d^* \simeq G_1 \oplus G_2 \), where \( G_1 = \bigoplus_{p \mid d} C_{M_p} \) and \( G_2 = \bigoplus_{p \mid d} C_{N_p} \). Let \( \mathcal{B} = \{2, 3, \ldots, \lfloor \ln \frac{1}{3} + \eta \rfloor d\} \). As we know \( \phi(n) \) and every prime factor of \( M \) we can compute \( B(b) := \prod_{s \mid M} s^{v_s(\phi(n))} \) for all \( b \in \mathcal{B} \). It is easy to see that \( B(b) = \phi_{G_1}(b) \). We check whether there are \( b \in \mathcal{B} \) and \( s \mid B(b) \) such that \( (\frac{b}{s^{v_s(\phi(n))}}, 1, d) > 1 \). We can suppose that there are no such, since otherwise we get a nontrivial divisor of \( d \). Let \( A = \text{LCM}_{b \in \mathcal{B}}(B(b)) \). Then \( A \mid p - 1 \) for all \( p \mid d \). The following analogue of theorem 4.2 (ii) for the group \( G_1 \) will be of use.

**Theorem 6.5** Let \( \mathbb{Z}_d^* = G_1 \oplus G_2 \), \( (\#G_1, \#G_2) = 1 \), \( a, b \in \mathbb{Z}_d^* \), \( \alpha \) be a multiple of \( \text{ord}_d(a) \) and \( \text{ord}_d(b) \). If none of the procedures

\[
PH(d, a^{\alpha s^{v_s(\alpha)}}, b^{\alpha s^{v_s(\alpha)}}, s, v_s(\text{ord}_d(a)), v_s(\text{ord}_d(b))) \quad \text{where} \ s \in \mathbb{P} \quad \text{and} \ s \mid (\text{ord}_d(a), \text{ord}_d(b), \#G_1),
\]

finds a nontrivial divisor of \( d \) then \( (a, b)_{G_1} \) is cyclic.

**Proof.** Similar to that of theorem 4.2 (ii). In particular, if the procedures \( PH() \) above fail to find a nontrivial divisor of \( d \) then \( (a, b)_{G_1} \) is generated by \( c_s, \) where \( c_s = b^{s^{v_s(\alpha)}} \) if \( v_s(\text{ord}_d(a)) \leq v_s(\text{ord}_d(b)) \), else \( c_s = a^{s^{v_s(\alpha)}} \). \( \square \)
Assume that $p_+(A) \leq \ln^u n$. This inequality holds when $d$ is not a prime power, since $D(n, M, u) \leq 1$. Then we can take $\alpha = \phi(n)$ in the above theorem and test if $\langle B \rangle_{G_1}$ is cyclic by using an inductive procedure as in section 4. We suppose that $\langle B \rangle_{G_1}$ is cyclic, for in the contrary case we get a nontrivial divisor of $d$. If $d$ is a prime power then $G_1$ is a fortiori cyclic.

Therefore $A = \#(\langle B \rangle_{G_1} = \#(\langle B \rangle_{d^nG_2}) \geq \frac{\psi(d, \ln^{\frac{1}{\eta}} \cdot \frac{\epsilon}{\eta} + \frac{1}{2})}{\prod_{p \mid d} N_p} > \frac{d^{1-(\frac{1}{\eta} + \frac{1}{2}) - 1}}{\prod_{p \mid d} p^{1-(\frac{1}{\eta} + \frac{1}{2})}} = d^{\frac{1}{\eta} + \frac{1}{2} + \frac{2}{\eta} + \frac{1}{2}}$. However $A \mid p - 1$ for all $p \mid d$, thus $A < p_-(d)$. Hence $\omega(d) \leq 2$. If $d < 2^{\frac{1}{\eta} + \frac{2}{\eta} + \frac{1}{2}}$ then we factor $d$ by trial division, so assume that the reverse inequality holds. Therefore $A^3 > d^{1 + \frac{3}{2} \cdot \frac{2}{\eta} + \frac{1}{2} + \frac{1}{2}} \geq 2d$. If $\omega(d) = 2$ then the factorization of $d$ is obtained by factoring the polynomial from lemma 6.3. Otherwise, $d$ is a prime power.

**Theorem 6.6** Let $0 < \epsilon \leq \frac{2}{3}, \eta > 0$. Assume that $n$ is squarefree. Suppose furthermore that $\phi(n)$ is given together with a divisor $M$ of $\phi(n)$ in a completely factored form, such that $F(n, M, \frac{1}{3} + \epsilon) = \omega(n)$ and $D(n, M, u) \leq 1$. Then the complete factorization of $n$ can be found in $O_{\eta, \epsilon}(\ln^{\frac{5}{2} + u + \frac{1}{2} + \eta} n)$ deterministic time.

**Proof.** A suitable algorithm has been described above. Its running time analysis may be easily verified. □

**6.3** $F(n, M, \frac{1}{4} + \epsilon) = \omega(n), D(n, M, u) \leq 2$

**Theorem 6.7** Let $0 < \epsilon \leq \frac{3}{4}, \eta > 0$. Suppose that $\phi(n)$ is given together with a divisor $M$ of $\phi(n)$ in a completely factored form, such that $F(n, M, \frac{1}{4} + \epsilon) = \omega(n)$ and $D(n, M, u) \leq 2$. Then there is an algorithm that calls a '$\phi$-oracle' $O(\ln n)$ times to output the complete factorization of $n$ in $O_{\eta, \epsilon}(\ln^{\frac{5}{2} + u + \frac{1}{2} + \eta} n)$ deterministic time.

**Proof.** We follow a similar inductive procedure to the one presented in subsection 6.2. But this time, if $d > 1$ is a divisor of $n$ we first compute $\phi(d)$ using the given '$\phi$-oracle'. Therefore we can assume that $d$ is squarefree (since otherwise $(d, \phi(d))$ is a nontrivial factor of $d$) and $\omega(d) \geq 3$ (for in the contrary case the factorization of $d$ can be found with the elementary method from section 5). The running time analysis may be easily verified. □
7 Factoring all \( n \) in deterministic subexponential time when a 'φ-oracle' is given

We will show that \( n \) can be completely factored in deterministic subexponential time provided that \( \phi(n) \) and its complete factorization are known. Assume that \( n \) has no prime factor below \( n^{\frac{1}{3\varphi(n)}} \), where \( u(x) = \left( \frac{\ln x}{\ln \ln x} \right)^{\frac{1}{3}} \). Let \( d > 1 \) be a divisor of \( n \). Our goal is to find a nontrivial divisor or a proof of the primality of \( d \). Let \( B = \{2, 3, \ldots, [d^{\frac{1}{\varphi(d)}}]\} \). We first check whether there is a Fermat-Euclid witness for \( d \) among the elements of \( B \). Suppose that this is not the case, for otherwise we are done. We compute \( A = \text{LCM}_{b \in B}(\text{ord}_d(b)) \). We further test if \( d \) has a nontrivial factor of the form \( mA + 1 \), where \( m < d^{\frac{1}{\varphi(d)}} \).

**Lemma 7.1** Let \( u(x) = \left( \frac{\ln x}{\ln \ln x} \right)^{\frac{1}{3}} \), \( B = \{2, 3, \ldots, [d^{\frac{1}{\varphi(d)}}]\} \), \( A = \text{LCM}_{b \in B}(\text{ord}_d(b)) \). Suppose that \( \omega(d) > \left( \frac{\ln d}{\ln \ln n} \right)^{\frac{1}{3}} \) and none of the elements of \( B \) is a Fermat-Euclid witness for \( d \). Then \( p_-(d) = mA + 1 \) for some integer \( m < d^{\frac{1}{\varphi(d)}} \) if \( p_-(d) \) is sufficiently large.

**Proof.** Let \( p = p_-(d) \), \( v(x) = \left( \frac{\ln x}{\ln \ln x} \right)^{\frac{1}{3}} \). We have \( p^{\frac{1}{\psi(p)}} = \exp[(\ln p)^{\frac{1}{2}}(\ln \ln p)^{\frac{1}{2}}] \leq \exp[(\ln d)^{\frac{1}{2}}(\ln \ln d)^{\frac{1}{2}}] \leq \exp[(\ln d)^{\frac{1}{2}}(\ln \ln d)^{\frac{1}{2}}] \). Since \( B \) contains no Fermat-Euclid witness for \( d \), it follows that \( A = \text{LCM}_{b \in B}(\text{ord}_d(b)) = \text{LCM}_{b \in B}(\text{ord}_p(b)) = \#(B)_p \geq \psi(p, \text{ord}_p(p)). \) By theorem 2.2, we obtain \( A \geq p \exp[(-\frac{1}{2} + o(1))(\ln p)^{\frac{1}{2}}(\ln \ln p)^{\frac{1}{2}}]. \) Now \( p = mA + 1 \) for some \( m \in \mathbb{N} \). Therefore \( mA < p \leq A \exp[(\frac{1}{2} + o(1))(\ln p)^{\frac{1}{2}}(\ln \ln p)^{\frac{1}{2}}]. \) Hence \( m < \exp[(\frac{1}{2} + o(1))(\ln p)^{\frac{1}{2}}(\ln \ln p)^{\frac{1}{2}}]. \) There is a constant \( n_0 \) such that the term \( o(1) \) in the last expression is less than \( \frac{1}{2} \) when \( p \geq n_0 \). If \( p \geq n_0 \) then we get \( m < \exp[(\ln p)^{\frac{1}{2}}(\ln \ln p)^{\frac{1}{2}}] < d^{\frac{1}{\varphi(d)}}. \)

If \( d \) has no such divisor then, from the above lemma, either \( d \) is divisible by a 'small' prime (less than some constant), or \( \omega(d) \leq \left( \frac{\ln d}{\ln \ln d} \right)^{\frac{1}{3}} \). Assume the latter. We can also suppose that \( \omega(d) \geq 3 \), since in the contrary case \( d \) is factorable by the elementary method from section 5.

**Lemma 7.2** Let \( u(x) = \left( \frac{\ln x}{\ln \ln x} \right)^{\frac{1}{3}} \), \( B = \{2, 3, \ldots, [d^{\frac{1}{\varphi(d)}}]\} \), \( A = \text{LCM}_{b \in B}(\text{ord}_d(b)) \). Assume that \( 3 \leq \omega(d) \leq \left( \frac{\ln d}{\ln \ln d} \right)^{\frac{1}{3}} \) and \( (A, d) = 1 \). Then \( A^{\omega(d) + 1} > d \left( \frac{\omega(d)}{\lfloor \omega(d)/2 \rfloor} \right) \) if \( d \) is sufficiently large.
Proof. Just as in the proof of lemma 6.2, we have $A^\omega(d) \geq \#\langle B \rangle_d \geq \psi(d, d^{\frac{1}{\omega(d)}})$. Hence $A^\omega(d)^{+1} \geq \psi(d, d^{\frac{1}{\omega(d)}}) - \frac{2}{3} + o(1))(\ln d)^{\frac{1}{d}}(\ln \ln d)^{\frac{1}{d}}$. It follows from theorem 2.2 that $A^\omega(d)^{+1} \geq \psi(d, d^{\frac{1}{\omega(d)}})^{\omega(d)^{+1}}(\ln d)^{\frac{1}{d}}(\ln \ln d)^{\frac{1}{d}}$. Moreover, $\omega(d) \geq 3$, thus $\omega(d)^{+1} \omega(d)^{+1} \leq 4$. Let $0 < \epsilon < \frac{1}{9}$. There exists a constant $n_1$ such that $d \geq n_1$ implies $\exp([\ln d^{\frac{1}{\omega(d)}}] + \omega(d)^{+1}(\ln d)^{\frac{1}{d}}(\ln \ln d)^{\frac{1}{d}}) \geq \exp((\frac{1}{9} - \epsilon)(\ln d)^{\frac{1}{d}}(\ln \ln d)^{\frac{1}{d}})$. The last expression is of course greater than $\exp((\ln d^{\frac{1}{\omega(d)}})^{\frac{1}{d}}(\ln \ln d)^{\frac{1}{d}})$ if $d$ is large enough. This finishes the proof, since $\exp((\ln d^{\frac{1}{\omega(d)}})^{\frac{1}{d}}(\ln \ln d)^{\frac{1}{d}}) = 2^{\omega(d)^{+1}} > \omega(d)^{\omega(d)^{+1}}/2$. □

The complete factorization of $d$ is then obtained by factoring the polynomial from lemma 6.3. We have thus proved

Theorem 7.3 Suppose that $\phi(n)$ is given in a completely factored form. Then the complete factorization of $n$ can be found in less than $\exp((1 + o(1))(\ln n)^{\frac{1}{d}}(\ln \ln n)^{\frac{1}{d}})$ deterministic time.

Corollary 7.4 There is a deterministic algorithm that calls a 'phi-oracle' $O(\ln n)$ times to output the complete factorization of $n$ in less than $\exp((1 + o(1))(\ln n)^{\frac{1}{d}}(\ln \ln n)^{\frac{1}{d}})$ time.

Proof. We use the given 'phi-oracle' to compute the sequence $\phi(n), \phi^2(n), \ldots, \phi^k(n)$, where $k \leq 1 + \log_2 n$ is the least integer such that $\phi^k(n) = 1$. Let $1 \leq m \leq k$. If we have found the complete factorization of $\phi^m(n)$ then, by theorem 7.3, we can obtain the complete factorization of $\phi^{m-1}(n)$ in less than $\exp((1 + o(1))(\ln n)^{\frac{1}{d}}(\ln \ln n)^{\frac{1}{d}})$ deterministic time. The corollary follows by induction. □

References

[1] L. M. Adleman, K. S. McCurley, Open problems in number theoretic complexity II, Algorithmic Number Theory Symposium I (1994), 291-322.

[2] E. Bach, Explicit bounds for primality testing and related problems, Mathematics of Computation, 55 (1990), 355-380.

[3] E. R. Berlekamp, Factoring Polynomials Over Finite Fields, Bell Systems Technical Journal, 46 (1967), 1853-1859.
[4] E. R. Canfield, P. Erdős, C. Pomerance, *On a problem of Oppenheim concerning ‘Factorisatio Numerorum’*, Journal of Number Theory, 17 (1983), 1-28.

[5] M. R. Fellows, N. Koblitz, *Self-Witnessing Polynomial-Time Complexity and Prime Factorization*, Designs, Codes and Cryptography, 2 (1992), 231-235.

[6] K. Hensel, *Neue Grundlagen der Arithmetik*, Journal für die Reine und Angewandte Mathematik, 127 (1904), 51-84.

[7] S. Konyagin, C. Pomerance, *On Primes Recognizable in Deterministic Polynomial Time*, The Mathematics of Paul Erdos, R. L. Graham, J. Nesetril, eds., Springer-Verlag, Berlin, 1997, 176-198.

[8] S. Landau, *Some Remarks on computing the Square Parts of Integers*, Information and Computation, 78, No. 3 (1988), 246-253.

[9] H. W. Lenstra, Jr., C. Pomerance, *Primality Testing with Gaussian Periods*, preliminary version July 20, 2005.

[10] G. L. Miller, *Riemann’s Hypothesis and Tests for Primality*, Journal of Computer and System Sciences, 13 (1976), 300-317.

[11] S. C. Pohlig, M. E. Hellman, *An Improved Algorithm for Computing Logarithms Over GF(p) and its Cryptographic Significance*, IEEE Transactions on Information Theory, 24 (1978), 106-110.

[12] J. M. Pollard, *Theorems on factorization and primality testing*, Proceedings of the Cambridge Philosophical Society, 76 (1974), 521-528.

[13] M. O. Rabin, *Probabilistic algorithm for testing primality*, Journal of Number Theory, 12 (1980), 128-138.