MATHEMATICS OF COMPUTATION
Volume 69, Number 231, Pages 1267–1283
S 0025-5718(99)01137-0
Article electronically published on May 24, 1999

USING NUMBER FIELDS TO COMPUTE LOGARITHMS IN FINITE FIELDS

OLIVER SCHIROKAUER

ABSTRACT. We describe an adaptation of the number field sieve to the problem of computing logarithms in a finite field. We conjecture that the running time of the algorithm, when restricted to finite fields of an arbitrary but fixed degree, is $L_q[1/3; (64/9)^{1/3} + o(1)]$, where q is the cardinality of the field, $L_q[s; c] = \exp(c(\log q)^s(\log\log q)^{1-s})$, and the $o(1)$ is for $q \to \infty$. The number field sieve factoring algorithm is conjectured to factor a number the size of q in the same amount of time.

1. Introduction

Let $q = p^n$, where p is a prime number and n a positive integer, and let $\mathbb{F}_q$ be the field of q elements. Let t and v be nonzero elements in $\mathbb{F}_q$ such that v is in the multiplicative subgroup generated by t, and let x be the smallest nonnegative integer such that $t^x = v$. We call the exponent x the discrete logarithm of v with respect to the base t and write $x = \log_t v$. In this paper we present an algorithm to compute $\log_t v$.

The most succesful methods to date to compute discrete logarithms in a finite field are all descendants of the index calculus method described by Kraitchik in the 1920's (see [14] and [15]) and rediscovered and modified by numerous mathematicians since then (see [25] and [31]). The idea of this approach is to find many multiplicative relations among members of a small subset of $\mathbb{F}_q$ containing t and v. Each such relation gives a linear relation among the logarithms of the elements in the subset. Once enough relations are collected, the values of the logarithms can be obtained using linear algebra.

The two newest members of the index calculus family are the number field sieve and the function field sieve. The first, due to Gordon [12] and modified in [29], is an adaptation of the number field sieve factoring algorithm to the problem of computing logarithms in a prime field. The computations in this method, as with its factoring counterpart, take place in a finite extension of $\mathbb{Q}$. The second, due to Adleman [1] and modified in [3], is most suitable for computing logarithms in fields of small characteristic. In this case, the structure of the number field sieve is transported to a finite extension of $\mathbb{F}_p(X)$.

The algorithm we present in this paper is an extension of the number field sieve algorithm in [29] to the case when $n \geq 1$. Though very similar to its predecessor for prime fields, it differs in one substantial way. The field extension at the heart

Received by the editor July 24, 1997 and, in revised form, July 14, 1998.
1991 *Mathematics Subject Classification.* Primary 11Y40, 11Y16; Secondary 11T71.
Key words and phrases. Finite field, discrete logarithm, number field sieve.

of the algorithm is not built over $\mathbb{Q}$ but instead on top of an extension of $\mathbb{Q}$. As a consequence, many of the routine computations and reductions that are employed in the earlier algorithm are more difficult now.

We begin in §2 with some background information from algebraic number theory. In §3 we describe how to use the ideas of the number field sieve to compute the residue of $\log_t v$ modulo a prime power dividing $q-1$. In conjunction with the Chinese remainder theorem, this method can be used to find $\log_t v$. The analysis given in §4 leads us to conjecture that when t and v are not too large, in a sense to be made precise later, the algorithm of §3 runs in time

$$L_q[1/3; (64/9)^{1/3} + o(1)] \tag{1.1}$$

where

$$L_q[s;c] = \exp\bigl(c(\log q)^s(\log\log q)^{1-s}\bigr),$$

and the $o(1)$ is for $q \to \infty$ with n constant. In §5 we show how to reduce the general discrete logarithm problem to the case that t and v are optimal for the method of §3 and establish (1.1) as the conjectural running time for the general algorithm.

In the case that $n = 1$, our results are not new and can be found in [29]. For $n > 1$, the appearance of the parameter value $1/3$ in (1.1) is an improvement over the constant $1/2$ achieved by earlier algorithms handling fields of arbitrary characteristic and fixed degree ([2], [11], [24]). We note that the quantity (1.1) is conjectured in [6] to be the time needed by the number field sieve to factor an integer the size of q. In [9], Coppersmith proposes a modification of the number field sieve for factoring that uses many number fields simultaneously. As a result, he obtains a conjectural running time of

$$L_q\left[1/3; \frac{(92+26\sqrt{13})^{1/3}}{3} + o(1)\right]$$

to factor a number the size of q. We do not discuss his ideas in the present work and leave it to the reader to show that, when applied to our algorithm, they yield the same improvement.

As indicated, we content ourselves in this paper with asymptotic times for n fixed. For a discussion of recent work on the problem of finding a discrete logarithm algorithm with a running time of $L_q[1/3; c + o(1)]$ for $q \to \infty$ with both p and n varying, we refer the reader to [30], where Adleman's function field sieve is conjectured to have a running time of (1.1) under the constraint that $\log p < n^{1/2}$ and where a modification of our present algorithm is given which is conjectured to run in time (1.1) so long as $\log p > n^{2+\epsilon}$ for some $\epsilon > 0$.

We do not address in this paper questions of practicality. Both the function field sieve in the special form of Coppersmith's algorithm for characteristic 2 and the number field sieve for prime fields have been implemented (see [13] and [33]–[35]). Indeed, Weber, in [34], is able to compute logarithms in a prime field of 85 digits using the number field sieve. It is our belief that for very small n, the algorithm given in this paper is practical and should be able to handle fields near the size of those discussed in [34]. Experiments testing this conviction might be of particular interest to cryptographers working with elliptic curves, as it has been shown that the logarithm problem on a supersingular elliptic curve over $\mathbb{F}_q$ can be reduced in subexponential time to the discrete logarithm problem in a field of small degree over $\mathbb{F}_q$ (see [26]).

2. Preliminaries

Let K be a number field of degree n over $\mathbb{Q}$ and let $\mathcal{O}_K$ be its ring of integers. Let $\sigma_1, \dots, \sigma_n$ denote the embeddings of K into $\mathbb{C}$ and let $||$ be the standard complex absolute value. For $\gamma \in \mathcal{O}_K$, we define the height of γ by $h(\gamma) = \max\{|\sigma_i(\gamma)|\}$.

2.1 Lattices. A lattice in $\mathbb{R}^n$ is the $\mathbb{Z}$-span of a set of $\mathbb{R}$-linearly independent vectors. Let $\{b_1, \dots, b_n\}$ be such a set of vectors and let $L = \sum_{i=1}^n \mathbb{Z}b_i$. We say that $\{b_i\}$ is a basis of L, and define the determinant $d(L)$ of L to be the absolute value of the determinant of the matrix whose columns are the vectors $b_1, \dots, b_n$. The determinant of L is independent of the basis used to calculate it.

Let $\| \, \|$ denote the Euclidean norm on $\mathbb{R}^n$. We call a basis $\{b_i\}_{1 \le i \le n}$ *reduced* if

(i) $\|b_i\| \le 2^{(n-1)/4} \cdot d(L)^{1/n}$ for some i, and
(ii) $2^{(1-i)/2}\lambda_i \le \|b_i\| \le 2^{(n-1)/2}\lambda_i$ for $1 \le i \le n$, where $\lambda_1, \dots, \lambda_n$ are the successive minima for $\| \, \|$ on L.

The notion of "reduced" presented here is not the same as the more usual definition found, for example, in [19]. However, any basis which is reduced in the sense of [19] is reduced in our sense. Given a basis consisting of elements in $\mathbb{Z}^n$ with Euclidean norm bounded by B, the Lenstra-Lenstra-Lovász algorithm (LLL) finds a reduced basis in time $O(n^4 \log B)$ (see [19]).

Assume now that $t_1, \dots, t_n$ are elements in $\mathcal{O}_K$ which form a module basis for $\mathcal{O}_K$ over $\mathbb{Z}$. Then the map that sends the element $\sum a_i t_i \in \mathcal{O}_K$ to the vector $(a_1, \dots, a_n) \in \mathbb{Z}^n$ is an isomorphism from the additive group of $\mathcal{O}_K$ to $\mathbb{Z}^n$. The image of an ideal $\mathfrak{a}$ under this map is an n-dimensional lattice with determinant equal to $N(\mathfrak{a})$, the ideal norm of $\mathfrak{a}$. Applying LLL to this lattice, therefore, produces a vector of Euclidean norm $\le 2^{(n-1)/4}N(\mathfrak{a})^{1/n}$ and hence an element in $\mathfrak{a}$ with height bounded by $kn2^{(n-1)/4}N(\mathfrak{a})^{1/n}$, where $k = \max\{h(t_i)\}$.

2.2 Smoothness. Let B be a positive real number. We say that an integer is B-smooth if each of its prime factors is at most B. We say that an element $\gamma \in \mathcal{O}_K$ is B-smooth if its norm to $\mathbb{Q}$ is B-smooth in $\mathbb{Z}$. The algorithm we describe in §3 requires finding many smooth elements in two different number fields. In our analysis of it, we use the following theorem from [7] and corollary from [6]. Let $\psi(M, B)$ denote the number of positive integers $\le M$ which are B-smooth.

Theorem 2.2.1. *Let ϵ be a positive constant. Then*

$$\frac{\psi(M,B)}{M} = u^{-u(1+o(1))}$$

uniformly in the region $M \ge 10$ and $B \ge (\log M)^{1+\epsilon}$, where $u = (\log M)/\log B$ and the limit implicit in the $o(1)$ is for $u \to \infty$.

Corollary 2.2.2. *Suppose $g : \mathbb{R}_{\ge 2} \to \mathbb{R}_{\ge 1}$ is a function such that $g(y) = y^{1+o(1)}$ for $y \to \infty$. Then, as $x \to \infty$,*

$$\frac{xg(y)}{\psi(x,y)} \ge L_x[1/2; \sqrt{2} + o(1)]$$

uniformly for all $y \ge 2$.

2.3 Additive characters on $\mathcal{O}_K^*$. Let l be a rational prime which does not ramify in K. In §3, we are interested in constructing a product of smooth elements of K which is an l^eth power for some given positive integer e. To do so, we make use of a family of maps defined as follows. Let

$$\Gamma_1 = \{\gamma \in \mathcal{O}_K \bigm| N_{\mathbb{Q}}^K(\gamma) \not\equiv 0 \bmod l\}.$$

For each prime ideal ℓ in $\mathcal{O}_K$ dividing (l), let $\epsilon_\ell = |(\mathcal{O}_K/\ell)^*|$ and let ϵ be the least common multiple of the ϵ_ℓ. Then for all $\gamma \in \Gamma_1$,

$$\gamma^\epsilon \equiv 1 \bmod l.$$

Now define a map $\lambda_1 : \Gamma_1 \to l\mathcal{O}_K/l^2\mathcal{O}_K$ by

$$\lambda_1(\gamma) = (\gamma^\epsilon - 1) + l^2\mathcal{O}_K.$$

Furthermore, for $i > 1$, let $\Gamma_i = \{\gamma \in \Gamma_{i-1} | \lambda_{i-1}(\gamma) = 0\}$, and let $\lambda_i : \Gamma_i \to l^{2^{i-1}}\mathcal{O}_K/l^{2^i}\mathcal{O}_K$ be the function given by $\lambda_i(\gamma) = (\gamma^\epsilon - 1) + l^{2^i}\mathcal{O}_K$. For $1 \le j \le n$, let $\{b_j l^{2^{i-1}} + l^{2^i}\mathcal{O}_K\}$ be a module basis for $l^{2^{i-1}}\mathcal{O}_K/l^{2^i}\mathcal{O}_K$ over $\mathbb{Z}/l^{2^{i-1}}\mathbb{Z}$. Then λ_i is given by the maps

$$\lambda_{i,j} : \Gamma_i \to \mathbb{Z}/l^{2^{i-1}}\mathbb{Z}$$

defined by the congruence

$$\gamma^\epsilon - 1 \equiv \sum_{j=1}^{n} \lambda_{i,j}(\gamma) b_j l^{2^{i-1}} \bmod l^{2^i}.$$

Note that since $\lambda_i(\gamma\gamma') = \lambda_i(\gamma) + \lambda_i(\gamma')$ and $\lambda_{i,j}(\gamma\gamma') = \lambda_{i,j}(\gamma) + \lambda_{i,j}(\gamma')$, the maps λ_i and $\lambda_{i,j}$ are homomorphisms on the group of units of $\mathcal{O}_K$.

For any $\gamma \in K^*$ and any prime ideal $\mathfrak{p} \subseteq \mathcal{O}_K$, let $\mathrm{ord}_{\mathfrak{p}}(\gamma)$ be the exponent to which $\mathfrak{p}$ divides the fractional ideal generated by γ. The usefulness of the maps λ_i stems from the following result.

Proposition 2.3.1. *Let l be a prime that does not ramify in K, let e be a positive integer, and let ρ be the least integer such that $2^\rho > e$. Assume that the class number of K is not divisible by l and that the units in $\mathcal{O}_K$ which are congruent to $1 \bmod l^{e+1}$ are l^eth powers. Let $\gamma \in \Gamma_\rho$ be such that*

(i) $\mathrm{ord}_{\mathfrak{p}}(\gamma) \equiv 0 \bmod l^e$ *for all prime ideals $\mathfrak{p}$ in $\mathcal{O}_K$, and*
(ii) $\lambda_\rho(\gamma) = 0$.

Then γ is an l^eth power in $\mathcal{O}_K$.

For more about the maps λ_i, as well as a proof of Proposition 2.3.1, see [29, §3].

2.4. Model for $\mathbb{F}_q$. Let p be prime and let $q = p^n$. By a *model* for $\mathbb{F}_q$ we mean a set of cardinality q with an addition operation and a multiplication operation giving the set the structure of a field. Since it is possible to exhibit an isomophism between different models of $\mathbb{F}_q$ in polynomial time (see [22]), it suffices to be able to compute discrete logarithms in a model of our choosing.

Let r be the smallest prime congruent to $1 \bmod n$ such that n is prime to $(r-1)/f$ where f is the order of p in $(\mathbb{Z}/r\mathbb{Z})^*$. Let ζ_r be a primitive rth root of unity, F the unique subfield of $\mathbb{Q}(\zeta_r)$ of degree n over $\mathbb{Q}$, and $\mathcal{O}_F$ the ring of integers of F.

Then $\mathcal{O}_F/p\mathcal{O}_F \cong \mathbb{F}_q$. Let t_1 be the trace of ζ_r in F and let $\{t_j\}_{1\leq j\leq n}$ be the set of conjugates of t_1 in $\mathcal{O}_F$. Then $\{t_j\}$ is a basis for $\mathcal{O}_F$ over $\mathbb{Z}$, and

$$R_q = \left\{ \sum_{j=1}^{n} a_j t_j \,\middle|\, 0 \leq a_j \leq p-1 \right\}$$

is a set of representatives for $\mathcal{O}_F/p\mathcal{O}_F$. Thus R_q, with addition and multiplication given by addition and multiplication in $\mathcal{O}_F$ taken modulo p, is a model for $\mathbb{F}_q$. We adopt this model, and for the remainder of the paper denote by $\mathbb{F}_q$ the field of q elements with underlying set R_q. In addition, we let $\phi : \mathcal{O}_F \to \mathbb{F}_q$ be the projection induced by sending γ to the unique element in R_q congruent to γ mod p.

Not surprisingly, the running time of our discrete logarithm algorithm depends on the size of r. It follows from Theorem 3 in [4] that if the extended Riemann hypothesis (ERH) is true, then there exists a constant c such that for all p prime and n prime to p, we have $r < cn^6(\log(np))^2$. We assume, therefore, that as $p \to \infty$ with n constant, r is less than a constant multiple of $(\log p)^2$, and that the discriminant of F, which we denote for the remainder of the paper by Δ_q and which equals r^{n-1}, is bounded by a constant multiple of $(\log p)^{2n-2}$.

3. The number field sieve for $n \geq 1$

We describe an algorithm for computing the residue of a discrete logarithm in $\mathbb{F}_q$ modulo a prime power l^e dividing $q-1$. In §5, this method is incorporated into a general discrete logarithm algorithm. We adopt the model for $\mathbb{F}_q$ given in subsection 2.4 and continue with all the notation introduced in that subsection.

Algorithm 3.1. This algorithm takes as input a prime power $q = p^n$, two integers d and B such that $d \geq 2$ and $B \geq 2^{n(n-1)/4}(r-1)^n$, an element $\sigma \in \mathcal{O}_F$ of height $\leq p^{\frac{1}{d+1}}$ such that $\phi(\sigma) \in \mathbb{F}_q^*$, a B-smooth element $\tau \in \mathcal{O}_F$ such that $\phi(\tau)$ is primitive in $\mathbb{F}_q$, and a prime power l^e dividing $q-1$ with $l > \max\{r, B\}$. It outputs the least positive residue of $\log_{\phi(\tau)} \phi(\sigma)$ modulo l^e. Note that this algorithm can be used to compute $\log_{\phi(\tau)} \phi(\sigma) \bmod l^e$ for any pair σ, τ so long as $h(\sigma) \leq p^{1/3}$ and l is sufficiently large.

The computations in the algorithm take place in two different number rings. One is the ring $\mathcal{O}_F$, and the second is an extension of $\mathcal{O}_F$ obtained by adjoining to $\mathcal{O}_F$ a root of a polynomial in $\mathcal{O}_F[X]$. The first two steps of our algorithm are concerned with the construction of a suitable polynomial.

Step 1. Let c be the smallest nonnegative integer such that

$$|N_{\mathbb{Q}}^F(2^c\sigma)| > (p^{\frac{1}{d+1}} + 1)^n (r-1)^n.$$

If

$$h(2^c\sigma) \leq 2^n(p^{\frac{1}{d+1}} + 1)(r-1)^2,$$

let $m = 2^c\sigma$ and proceed to Step 2. Otherwise, apply LLL reduction to the lattice obtained by embedding the ideal (σ) into $\mathbb{Z}^n$ as described in subsection 2.1 and in such a way produce an element $\gamma\sigma \in (\sigma)$ with height bounded by $2^{(n-1)/4}|N_{\mathbb{Q}}^F(\sigma)|^{1/n}(r-1)$. Let $m = 2^c\gamma\sigma$, where c is the smallest nonnegative integer such that

$$|N_{\mathbb{Q}}^F(m)| > (p^{\frac{1}{d+1}} + 1)^n (r-1)^n.$$

Notice that γ is B-smooth, and that

$$h(m) = 2^c h(\gamma\sigma) \leq 2^c 2^{\frac{n-1}{4}} |N_{\mathbb{Q}}^F(\sigma)|^{\frac{1}{n}}(r-1) = 2^{\frac{n-1}{4}} |N_{\mathbb{Q}}^F(2^c\sigma)|^{\frac{1}{n}}(r-1).$$

In the case that $c > 0$, we have

$$|N_{\mathbb{Q}}^F(2^c\sigma)| \leq |N_{\mathbb{Q}}^F(m)| \leq 2^n(p^{\frac{1}{d+1}}+1)^n(r-1)^n$$

and consequently,

$$h(m) \leq 2^n(p^{\frac{1}{d+1}}+1)(r-1)^2. \tag{3.2}$$

If $c = 0$, then (3.2) follows from the fact that $|N_{\mathbb{Q}}^F(\sigma)|^{1/n} \leq h(\sigma) \leq p^{\frac{1}{d+1}}$.

Step 2. Let

$$\left\{(a_{ij})_{\substack{i=0,\dots,d\\ j=1,\dots,n}} \in \mathbb{Z}^{n(d+1)} \middle| \sum_{i,j} a_{ij}t_j m^i \equiv 0 \bmod p\right\}.$$

Then L is a $n(d+1)$-dimensional lattice. Let α_{uvj} be given by the equation

$$t_v m^u = \sum_{j=1}^{n} \alpha_{uvj} t_j,$$

and let $\overline{\alpha_{uvj}}$ be the least nonnegative residue of α_{uvj} modulo p. Let $b(u,v) = (b(u,v)_{ij})$ be the vector in L which for $u = 1,\dots,d$ and $v = 1,\dots,n$ is given by

$$b(u,v)_{ij} = \begin{cases} -1 & \text{if } (i,j) = (u,v), \\ \overline{\alpha_{uvj}} & \text{if } i = 0, \\ 0 & \text{otherwise,} \end{cases}$$

and which for $u = 0$ and $v = 1,\dots,n$ is given by

$$b(u,v)_{ij} = \begin{cases} p & \text{if } (i,j) = (u,v), \\ 0 & \text{otherwise.} \end{cases}$$

Then $\{b(u,v)\}$ is a basis for L. Apply LLL to this basis and let $b = (b_{ij})$ be a vector in the resulting reduced basis which is of minimum Euclidean norm lying outside of the sublattice

$$L_0 = \{(a_{ij}) | \sum a_{ij}t_j m^i = 0\}.$$

For $i = 0,\dots,d$, let $\beta_i = \sum_{j=1}^n b_{ij}t_j$. Now find integers y_0 and y_1 of minimal absolute value such that $\beta_d + y_0$ and $\beta_0 - y_1 m$ are B-smooth. Since $N_{\mathbb{Q}}^F(\beta_d + y)$ and $N_{\mathbb{Q}}^F(\beta_0 - ym)$ are polynomials in $\mathbb{Z}[y]$, a sieve can be used to test for smoothness (see [27] and [28]), though the running time results of §4 are not affected if the elliptic curve factoring method of [21] is used to test candidates instead. Next let

$$f_0 = (\beta_d + y_0)X^d + (\beta_{d-1} - y_0 m)X^{d-1} + \sum_{i=2}^{d-2} \beta_i X^i + (\beta_1 + y_1)X + (\beta_0 - y_1 m)$$

and find y_2 of minimal absolute value such that $f_0 + (y_2X^2 - y_2mX)$ has discriminant prime to l and has smooth leading coefficient, this second condition being automatically satisfied for $d > 2$. Let $f = f_0 + (y_2X^2 - y_2mX)$ and, to simplify notation, write $f = \sum_{i=0}^d c_iX^i$. Finally, let $g = X^d + \sum_{i=0}^{d-1}(c_d)^{d-i-1}c_iX^i$.

Let ω be a root of f, in which case $\alpha = c_d\omega$ is a root of g. The number fields we proceed to compute in are F and $K = F(\omega) = F(\alpha)$. Note that the polynomial

f has three special properties. The first is that $f(m) \in p\mathcal{O}_F$. As a result, we can extend the ring homomorphism $\phi : \mathcal{O}_F \to \mathbb{F}_q$ introduced in subsection 2.4 to a ring homomorphism from $\mathcal{O}_F[\alpha]$ to $\mathbb{F}_q$ by sending α to $\phi(c_d m)$. The second is that the leading and constant terms of f are B-smooth. Consequently, α is B-smooth. This fact allows us to include α, and in turn σ, into the relation we construct in Step 4. The requirement that f has a smooth leading coefficient, however, is only a matter of convenience. It allows us to work in $\mathcal{O}_F[\alpha]$. For an alternative approach, in which no constraint is put on the leading coefficient of f and the computations take place in the ring $\mathcal{O}_F[\omega] \cap \mathcal{O}_F[\omega^{-1}]$, see [6, §12]. The third property of f is that it has small coefficients. By this we mean that the height of these coefficients is close to that of m. To show that this is the case for $\beta_0, \ldots, \beta_d$, let

$$M = \left\{ (b_{ij}) \in \mathbb{Z}^{n(d+1)} \,\middle|\, |b_{ij}| < \frac{p^{\frac{1}{d+1}} + 1}{2} \right\}.$$

Since $|M| > p^n$, there exist two distinct vectors (c_{ij}) and (d_{ij}) in M such that $\sum (c_{ij} - d_{ij}) t_j m^i \equiv 0 \bmod p$. Moreover, $(c_{ij} - d_{ij}) \notin L_0$, since if it were, then for some i, the sum

$$\sum_{j=1}^{n} (c_{ij} - d_{ij}) t_j$$

would be both nonzero and congruent to 0 mod m, contradicting the fact that

$$|N_{\mathbb{Q}}^{F}\big(\sum_{j=1}^{n} (c_{ij} - d_{ij}) t_j\big)| < (p^{\frac{1}{d+1}} + 1)^n (r-1)^n < |N_{\mathbb{Q}}^{F}(m)|.$$

According to the definition of reduced given in subsection 2.1, the existence of a vector in $L - L_0$ with Euclidean norm less than $\sqrt{n(d+1)}(p^{\frac{1}{d+1}} + 1)$ implies that a reduced basis contains a vector outside of L_0 of Euclidean norm less than

$$\sqrt{2^{n(d+1)-1} n(d+1)}(p^{\frac{1}{d+1}} + 1).$$

We conclude that, for each i,

$$h(\beta_i) \le \sqrt{2^{n(d+1)-1} n(d+1)} \left(p^{\frac{1}{d+1}} + 1\right)(r-1). \tag{3.3}$$

In §4, we see that in the cases that concern us y_0, y_1, and y_2, are negligible compared to the bound in (3.3) and thus the height of the coefficients of f does not significantly exceed this quantity.

Step 3. Our goal in this step is to collect pairs $(a, b) \in \mathcal{O}_F \times \mathcal{O}_F$ such that the polynomial

$$N_{\mathbb{Q}}^{F}\left((-b)^d f(-a/b)(a + bm)\right) \tag{3.4}$$

is B-smooth. We indicate below how many pairs are needed. Writing $a = \sum a_j t_j$ and $b = \sum b_j t_j$, we see that (3.4) is a polynomial over $\mathbb{Z}$ in the $2n$ coefficients $a_1, \ldots, a_n, b_1, \ldots, b_n$. Therefore, a sieve can be used to test for smoothness, though as before, the elliptic curve factoring method can be used instead without affecting the running time analysis of §4. Whichever test is used, the coefficients a_j, b_j should be taken as small as possible. Note that since

$$N_F^K(ac_d + b\alpha) = N_F^K(c_d(a + b\omega)) = {c_d}^{d-1}(-b)^d f(-a/b)$$

and c_d is B-smooth, each pair (a, b) found has the property that $ac_d + b\alpha$ and $a + bm$ are both B-smooth.

Let T_F be the set of prime ideals $\mathfrak{p}$ of F such that $\mathfrak{p} \cap \mathbb{Z}$ is generated by a prime $\leq B$, and let T_K be the set of prime ideals of K satisfying the same condition and of degree 1 over F. For each (a, b) obtained, compute an exponent vector $w_{a,b}$ of length $|T_F| + |T_K| + n + nd$ as follows. Index the coordinates of $w_{a,b}$ by the primes in T_F, the primes in T_K, and j ranging from 0 to $n + nd - 1$. Let ρ be the least integer such that $2^\rho > e$. Now let

$$\begin{aligned} w_{a,b}(\mathfrak{p}) &= \mathrm{ord}_{\mathfrak{p}}(c_d(a + bm)) && \text{for} \quad \mathfrak{p} \in T_F, \\ w_{a,b}(\mathfrak{q}) &= \mathrm{ord}_{\mathfrak{q}}(ac_d + b\alpha) && \text{for} \quad \mathfrak{q} \in T_K, \\ w_{a,b}(j) &= \lambda^F_{\rho,j}(c_d(a + bm)) && \text{for} \quad j = 0, \dots, n - 1, \\ w_{a,b}(j) &= \lambda^K_{\rho,j-n}(ac_d + b\alpha) && \text{for} \quad j = n, \dots, n + nd - 1, \end{aligned}$$

where $\lambda^F_{\rho,j}$ and $\lambda^K_{\rho,j-n}$ are the maps from subsection 2.3 for the fields F and K respectively. In addition, compute the vector w_τ given by

$$\begin{aligned} w_\tau(\mathfrak{p}) &= \mathrm{ord}_{\mathfrak{p}}(\tau) && \text{for} \quad \mathfrak{p} \in T_F, \\ w_\tau(\mathfrak{q}) &= 0 && \text{for} \quad \mathfrak{q} \in T_K, \\ w_\tau(j) &= \lambda^F_{\rho,j}(\tau) && \text{for} \quad j = 0, \dots, n - 1, \\ w_\tau(j) &= 0 && \text{for} \quad j = n, \dots, n + nd - 1. \end{aligned}$$

Finally, recall from Step 1 that $m = 2^c\gamma\sigma$ and let w_σ be the vector given by

$$\begin{aligned} w_\sigma(\mathfrak{p}) &= \mathrm{ord}_{\mathfrak{p}}(c_d 2^c \gamma) && \text{for} \quad \mathfrak{p} \in T_F, \\ w_\sigma(\mathfrak{q}) &= \mathrm{ord}_{\mathfrak{q}}(\alpha) && \text{for} \quad \mathfrak{q} \in T_K, \\ w_\sigma(j) &= \lambda^F_{\rho,j}(c_d 2^c \gamma) && \text{for} \quad j = 0, \dots, n - 1, \\ w_\sigma(j) &= \lambda^K_{\rho,j-n}(\alpha) && \text{for} \quad j = n, \dots, n + nd - 1. \end{aligned}$$

In Step 4 below, we need to solve the congruence

$$AX \equiv -w_\sigma \bmod l^e, \tag{3.5}$$

where A is the matrix whose first column is w_τ and whose remaining columns are the vectors $w_{a,b}$. It is this requirement that dictates how many (a, b) must be collected in the present step. As soon as w_σ is in the column space of A, no more pairs are needed. In particular, it suffices to find enough (a, b) so that A has rank $|T_F| + |T_K| + n + nd$ over $\mathbb{F}_l$.

To compute the entries in the exponent vectors corresponding to the primes in T_F, first find the prime ideal factorization in $\mathcal{O}_F$ of the rational primes $\leq B$ by means of Algorithm 6.2.9 in [8], using t_1 as a generator for F over $\mathbb{Q}$. Then employ Algorithm 4.8.17 in [8] to compute the order of $a + bm$ at each prime in T_F. For the entries corresponding to primes in T_K a different approach is more efficient. Let $\mathfrak{q}'$ be a prime ideal in $\mathcal{O}_K$ containing an element $ac_d + b\alpha$ and let $\mathfrak{q} = \mathfrak{q}' \cap \mathcal{O}_F$. Then $\mathfrak{q}' = (\mathfrak{q}, ac_d + b\alpha)$, and no other prime ideal above $\mathfrak{q}$ contains $ac_d + b\alpha$. Therefore $\mathrm{ord}_{\mathfrak{q}'}(ac_d + b\alpha) = \mathrm{ord}_{\mathfrak{q}}(N^K_F(ac_d + b\alpha))$, and the method for computing the order of an element at a prime of $\mathcal{O}_F$ can be used. The only trick is to be able to distinguish the primes in T_K lying above a given $\mathfrak{q} \in T_F$. However, this is easily done, since two representations $(\mathfrak{q}, ac_d + b\alpha)$ and $(\mathfrak{q}, a'c_d + b'\alpha)$ correspond to the same prime if and only if $c_d(b'a - ba') \in \mathfrak{q}$. Finally, to compute the values of λ^F_ρ and λ^K_ρ, it is enough to know the degrees of the primes in $\mathcal{O}_F$ and $\mathcal{O}_K$ lying above l. In the case of F, use Algorithm 6.2.9 in [8] to decompose l. For K, then, it suffices to determine, for each prime ideal ℓ in $\mathcal{O}_F$ lying above l, the factorization type over $\mathcal{O}_F/\ell$ of the polynomial g reduced mod ℓ. For a discussion of methods to do this, see [22].

Step 4. Using the linear algebra techniques discussed for instance in [10], [16], and [36], solve equation (3.5).

Step 5. Let $(x, \dots, x_{(a,b)}, \dots)$ be a solution to (3.5). Then the integers x and $x_{(a,b)}$ satisfy the following conditions:

(i) $\operatorname{ord}_{\mathfrak{p}}\left(\tau^x c_d 2^c \gamma \prod \left(c_d(a+bm)\right)^{x_{(a,b)}}\right) \equiv 0 \bmod l^e$ for all $\mathfrak{p} \in T_F$,

(ii) $\lambda_\rho^F(\tau^x c_d 2^c \gamma \prod \left(c_d(a+bm)\right)^{x_{(a,b)}}) = 0$,

(iii) $\operatorname{ord}_{\mathfrak{q}}\left(\alpha \prod (ac_d + b\alpha)^{x_{(a,b)}}\right) \equiv 0 \bmod l^e$ for all $\mathfrak{q} \in T_K$,

(iv) $\lambda_\rho^K\left(\alpha \prod \left(ac_d + b\alpha\right)^{x_{(a,b)}}\right) = 0$.

According to Proposition 2.3.1 and the heuristic arguments presented in [29, §3], there is good reason to believe that the products $\tau^x c_d 2^c \gamma \prod \left(c_d(a+bm)\right)^{x_{(a,b)}}$ and $\alpha \prod \left(ac_d + b\alpha\right)^{x_{(a,b)}}$ are both l^eth powers. If this is the case then, since ϕ is a ring homomorphism and $\phi(ac_d + b\alpha) = \phi(c_d(a+bm))$, we see that $\phi(\tau^x c_d 2^c \gamma)\phi(\alpha)^{-1}$ is an l^eth power in $\mathbb{F}_q$. Furthermore, since $\phi(\tau^x c_d 2^c \gamma)\phi(\alpha)^{-1} = \phi(\tau^x)\phi(\sigma)^{-1}$, we have $x \equiv \log_{\phi(\tau)} \phi(\sigma) \bmod l^e$. In other words, x is likely to be the number we seek, and all that remains is to check whether it is. If it is not, a number of options can be pursued. One is to run the entire algorithm again but with a different model for $\mathbb{F}_q$ obtained by replacing the number r by the next largest prime r' which is congruent to 1 mod n and for which p is inert in the degree n subfield of the r'th cyclotomic field. For other alternatives, we refer the reader to the description of Version 3.9 of the algorithm in [29, §3].

4. Optimizing algorithm 3.1

We minimize the running time of Algorithm 3.1, using as a guide the analysis in [6] of the number field sieve factoring algorithm. The result we obtain is used in §5, where we see that the general discrete logarithm problem can be solved by repeated application of Algorithm 3.1 with optimal input. Throughout this section, assume that n is constant and that all $o(1)$'s are for $p \to \infty$.

We begin by computing how many elements need to be tested for smoothness in Step 3 in order to find enough pairs (a, b) to construct the matrix A, and in Step 2 in order to find y_0 and y_1. To this end, for a prime p and integers $d \geq 2$ and $B \geq 2^{n(n-1)/4}(r-1)^n$, let $C_{p,d,B}$ be the least integer such that when Algorithm 3.1 is run with input $q = p^n, d, B$, any pair $\sigma, \tau \in \mathcal{O}_F$ which is suitable input for the algorithm, and any prime power l^e dividing $q-1$ with $l > \max\{r, B\}$, it is the case that for all pairs (a, b) found in Step 3, the coefficients a_j, b_j in the expressions $a = \sum a_j t_j$ and $b = \sum b_j t_j$ are $\leq C_{p,d,B}$ in absolute value. In addition, let $C'_{p,d,B}$ be the least integer such that when the algorithm is run with input as above, the elements y_0, y_1, and y_2 found in Step 2 are $\leq C'_{p,d,B}$ in absolute value. Note that $C_{p,d,B}$ and $C'_{p,d,B}$ are well defined, since neither depends on τ or l^e and since for a given d only finitely many σ can be input into the algorithm. Using (3.2), (3.3), the fact that $h(a)$ and $h(b)$ are $\leq C_{p,d,B}(r-1)$, and the assumption that r is bounded by a constant multiple of $(\log p)^2$, we see that expression (3.4), which represents the size of the numbers being tested for smoothness in Step 3, is bounded by

$$kd^{\frac{3n}{2}}2^{n^2d}(\log p)^{n(2d+9)}p^{\frac{2n}{d}}(C_{p,d,B})^{n(d+1)}(C'_{p,d,B})^n \tag{4.1}$$

for some constant k. Using (3.2) and (3.3) again, we see that the numbers being tested for smoothness to find y_0 and y_1 are bounded by

$$k'd^n2^{n^2d}(\log p)^{4n}p^{\frac{n}{d}}(C'_{p,d,B})^n \tag{4.2}$$

for some constant k'.

Assume now that $d = d(p)$ and $B = B(p)$ are functions of p satisfying $d \geq 2$ and $B \geq 2^{n(n-1)/4}(r-1)^n$, and let $C = C(p) = C_{p,d(p),B(p)}$ and $C' = C'(p) = C'_{p,d(p),B(p)}$. In addition, let $\sigma(p)$ be an element in $\mathcal{O}_F$ with the property that when Algorithm 3.1 is run with input $p^n, d(p), B(p), \sigma(p)$, and any τ, the number of pairs in $\mathcal{O}_F \times \mathcal{O}_F$ which need to be tested in Step 3 is maximal. In other words, replacing $\sigma(p)$ by a different element does not increase the number of tests required. Similarly, let $\sigma'(p) \in \mathcal{O}_F$ be such that when the algorithm is run with input $p^n, d(p), B(p), \sigma'(p)$, and any τ, the number of tests required to find y_0 and y_1 in Step 2 is maximal.

To determine the asymptotic behavior of C and C', we depend on four assumptions. Two of these pertain to the performance of Algorithm 3.1 when $p^n, d(p), B(p)$, and $\sigma(p)$ are input. For this case, let $T(p)$ be the number of pairs (a, b) tested in Step 3, let $N(p)$ be the number of pairs (a, b) collected in this step, and let $L(p)$ be the length of the columns of the matrix A appearing in equation (3.5). Let $S(p)$ be the probability that a random integer bounded by (4.1) is B-smooth. Our assumptions then are that

$$N(p) = L(p)^{1+o(1)} \quad \text{and} \quad \frac{N(p)}{T(p)} = S(p)^{1+o(1)}. \tag{4.3}$$

The first equation reflects our expectation that A is close to square and the second our belief that integers arising as values of polynomials behave with respect to smoothness like random integers of the same size.

Our second pair of assumptions is concerned with how Algorithm 3.1 runs with input $p^n, d(p), B(p)$, and $\sigma'(p)$. For this case, let $Y(p)$ be the maximum absolute value of the elements y_0 and y_1 found in Step 2, and let $S'(p)$ be the probability that a random integer bounded by (4.2) is B-smooth. Our assumptions are that

$$C'(p) = Y(p)^{1+o(1)} \quad \text{and} \quad Y(p) = S'(p)^{1+o(1)}. \tag{4.4}$$

The first simply states that y_2 is small. The second is based again on our conviction that polynomial values are random with respect to smoothness.

Recall that the length of the vectors forming the columns of A is equal to $|T_F| + |T_K| + n + nd$ and hence lies between $B/\log B$ and a constant multiple of dB. This fact and assumptions (4.3) lead to the conclusion that

$$\left(\frac{xB/\log B}{\psi(x,B)}\right)^{1+o(1)} \leq C^{2n} \leq \left(\frac{xdB}{\psi(x,B)}\right)^{1+o(1)}, \tag{4.5}$$

where $x = kd^{3n/2}2^{n^2d}(\log p)^{n(2d+9)}p^{2n/d}C^{n(d+1)}C'^n$. Similarly, assumptions (4.4) imply that

$$C' = \left(\frac{x'}{\psi(x',B)}\right)^{1+o(1)}, \tag{4.6}$$

where $x' = k'd^n 2^{n^2 d}(\log p)^{4n} p^{n/d} C'^n$. We use the following result to convert (4.5) and (4.6) into a conjecture about the running time of Algorithm 3.1.

Proposition 4.7. *Assume d, B, C, and C' are functions of p satisfying (4.5) and (4.6). Assume also that $dB \geq 2$. Then*

$$C^{2n} \geq L_p[1/3; (64n/9)^{1/3} + o(1)]. \tag{4.8}$$

Furthermore, if

$$d = ((3n)^{1/3} + o(1))(\log p/\log\log p)^{1/3} \tag{4.9}$$

and

$$B = L_p[1/3; (8n/9)^{1/3} + o(1)], \tag{4.10}$$

then the equality holds in (4.8).

Proof. It follows from (4.5), Corollary 2.2.2, and the fact that $x \to \infty$ as $p \to \infty$ that

$$C^{2n} \geq \Big(\frac{xB/\log B}{\psi(x,B)}\Big)^{1+o(1)} \geq \Big(\frac{xB/\log B}{\psi(x,B)}\Big)^{1+o(1)} \geq L_x[1/2, \sqrt{2} + o(1)].$$

Following now the proof of Lemma 10.12 in [6], we square the logarithm of both sides of this inequality and then divide each side by its logarithm. As a result, we obtain

$$\begin{aligned}\frac{n^2(\log C)^2}{\log\log C} &\geq (1+o(1))\log x \\ &= (1+o(1))\Big(\frac{2n}{d+1}\log p + n(d+1)\log C + n\log C'\Big) \\ &\geq (1+o(1))\Big(\frac{2n}{d+1}\log p + n(d+1)\log C\Big).\end{aligned}$$

Dividing through by n^2, applying Lemma 10.9 from [6], and multiplying the resulting inequality through by n yields

$$2n\log(C(p)) \geq (1+o(1))\left(d\log d + \sqrt{(d\log d)^2 + 4n\log(p^{1/d})\log\log(p^{1/d})}\right).$$

We obtain the first part of the proposition by minimizing the right-hand side of this inequality. We note that $(d\log d)^2$ and $\log(p^{1/d})\log\log(p^{1/d})$ must be of the same order of magnitude, in which case d must be a multiple of $(\log p/\log\log p)^{1/3}$. With a little bit of calculus, we then see that the right-hand side attains its minimum of

$$\left(\left(\frac{8n}{9}\right)^{1/3} + o(1)\right)(\log p)^{1/3}(\log\log p)^{2/3}$$

when $d = ((3n)^{1/3} + o(1))(\log p/\log\log p)^{1/3}$.

To prove the second claim, we assume $d(p)$ and $B(p)$ are as given and let $c(p)$ be defined by the equation

$$2n\log(C(p)) = c(p)(\log p)^{1/3}(\log\log p)^{2/3}.$$

Using (4.6), (4.9), (4.10), and Theorem 2.2.1, we readily verify that $C'^n \leq L_p[2/3; o(1)]$. Thus we find that

$$\log x = (1+o(1))\big((8n^2/3)^{1/3} + c(p)(3n/8)^{1/3}\big)(\log p)^{2/3}(\log\log p)^{1/3}.$$

It follows from Theorem 2.2.1 that

$$\frac{x}{\psi(x,B)} = L_p[1/3; (n/9)^{1/3} + c(p)/4 + o(1)]$$

and consequently that

$$\Big(\frac{xB}{\psi(x,B)}\Big)^{1+o(1)} = L_p[1/3; (3n)^{1/3} + c(p)/4 + o(1)].$$

Using (4.5), we see that $c(p) = (64n/9)^{1/3} + o(1)$. This concludes the proof of the proposition. □

Proposition 4.7 leads us to believe that the number of tests required to find sufficiently many suitable pairs (a, b) in Step 3 of Algorithm 3.1 is at least

$$L_p[1/3; (64n/9)^{1/3} + o(1)], \tag{4.11}$$

with equality occurring if the parameters d and B satisfy (4.9) and (4.10). The same can be said of the running time for Step 3, since the factor representing the time needed to test for smoothness, whether with a sieve or with the elliptic curve factoring method, is swallowed up by the $o(1)$ in (4.11). We argue that in the case that $h(\tau)$ is sufficiently small, the running time of the entire algorithm is given by (4.11).

Using the result quoted in subsection 2.1, we determine that the time needed for the LLL basis reductions in Steps 1 and 2 does not exceed (4.11). It is a direct consequence of (4.6) and Theorem 2.2.1 that the time required to find y_0, y_1, and y_2 is also less than this bound. Since the running times of Algorithms 6.2.9 and 4.10.17 in [8] are polynomial in $\log B$ and $\log |\Delta_q|$ and since, under the ERH, $|\Delta_q| = O((\log p)^{2n-2})$, we have good reason to believe that each exponent vector $w_{a,b}$ in Step 3 can be computed in time

$$(|T_F| + |T_K| + n + nd)(\log p)^{O(1)} \leq dB(\log p)^{O(1)}(\log p)^{O(1)},$$

and consequently that the time needed to obtain all exponent vectors is bounded by $(dB)^{2+o(1)}(\log p)^{O(1)}$. This quantity is equal to (4.11) when d and B satisfy (4.9) and (4.10). Note that it is in the computation of w_τ in Step 3 that the size of τ comes into play. In order to guarantee that the time needed to compute w_τ is inconsequential, we adopt here the asumption that $h(\tau) \leq p$. The reader will see that such a restriction does not interfere with the general discrete logarithm algorithm given in the next section. In Step 4, the time required to solve (3.5) is bounded by the product of the maximum number of nonzero entries appearing in a column of the matrix A, the square of the maximum dimension of A, and a constant power of the logarithm of this dimension. Using the first assumption of (4.3), we readily see that this product is equal to (4.11) when (4.9) and (4.10) hold. Our last concern is the danger described in Step 5 that the fields F and K do not satisfy the conditions in Proposition 2.3.1 and that, consequently, the algorithm needs to be repeated. The heuristic analysis given in [29, §3] suggests that the probability that F and K meet the desired conditions is at least $1 - (l-1)^{-1}$. Relying on this evidence, we assume that the factor introduced into the running time by Step 5 is asymptotically negligible and thus arrive at the following conjecture.

Conjecture 4.12. *Let $d = d(p)$ and $B = B(p)$ be functions such that $d \geq 2$ and $B \geq 2^{n(n-1)/4}(r-1)^n$. Let $T(p)$ be the maximum running time of Algorithm 3.1 upon input of $q = p^n, d(p), B(p)$, any $\sigma \in \mathcal{O}_F$ such that $h(\sigma) \leq p^{\frac{1}{d+1}}$ and $\phi(\sigma) \in \mathbb{F}_q^*$,* *any B-smooth $\tau \in \mathcal{O}_F$ such that $h(\tau) \leq p$ and $\phi(\tau)$ is primitive in $\mathbb{F}_q$, and any prime power l^e dividing $q-1$ with $l > \max\{r, B\}$. Then $T(p)$ is bounded below by (4.11), with equality holding in the case that d and B satisfy (4.9) and (4.10).*

We conclude with two remarks concerning Algorithm 3.1 and the number field sieve factoring method. First, since

$$L_p[1/3; (64n/9)^{1/3} + o(1)] = L_q[1/3; (64/9)^{1/3} + o(1)],$$

we see that the conjectured running time of Algorithm 3.1 is the same as the conjectured time needed by the number field sieve to factor a number the size of q. Second, we note that, in contrast to the situation for factoring, the present algorithm does not lend itself easily to the case that there exists $r, s \in \mathcal{O}_F$ of small height such that p divides $r^e + s$. The reader familiar with the version of the number field sieve used to factor divisors of rational integers of this form will recall that the advantage gained is the smallness of the coefficients of the polynomial used to define the number field (see [20]). The analogous version of Algorithm 3.1 for special q, however, suffers because the polynomial f used to construct the field K in Step 2 depends on the element σ as well as q and only has small coefficients for very few inputs. However, as Gordon describes in [12] for the case of prime fields, one can adjust the algorithm so as to use the special field made available by the form of q. We leave it to the reader to generalize Gordon's modifications to the case $n > 1$. The resulting method is slower than Algorithm 3.1, achieving a running time of $L_q[2/5; c + o(1)]$ with c constant, but it is likely to be faster for values of q that can be handled in practice at this time. Indeed, the McCurley challenge [25], which asked for the solution to a discrete logarithm problem in a prime field of cardinality

$$p = \frac{739 \cdot 7^{149} - 736}{3},$$

was recently solved with the number field sieve, using the field afforded by the special form of p (see [35]).

5. General discrete logarithms

We incorporate Algorithm 3.1 into a solution to the general discrete logarithm problem. We continue with all the notation and the model for $\mathbb{F}_q$ introduced in subsection 2.4.

Algorithm 5.1. This algorithm takes as input a prime power $q = p^n$, a primitive element $t \in \mathbb{F}_q^*$, a second element $s \in \mathbb{F}_q^*$, and two integers d and B such that $d \geq 2$ and $B \geq 2^{n(n-1)/4}(r-1)^n$. It outputs $\log_t s$.

Step 1. Let $k = 2^{n(n-1)/4}(r-1)^n$. If $q^{\frac{1}{d+1}} < 2k$, compute $\log_t s$ using the Pohlig-Hellman-Silver method described, for instance, in [25, §4.2]. Otherwise, continue as follows.

Step 2. Factor $q-1$ into a product $\prod l^e$ of prime powers using the number field sieve.

Step 3. For each prime l dividing $q-1$, find an element $\gamma_l = \sum a_j t_j \in \mathcal{O}_F$ which is B-smooth and whose image under ϕ is not in the subgroup of $\mathbb{F}_q^*$ of order $(q-1)/l$. Do this by testing elements, choosing candidates in such a way that $\max\{|a_j|\}$ is as small as possible. Let

$$\tau = \prod_{l|q-1} \gamma_l.$$

Then τ is B-smooth and $\phi(\tau)$ is primitive in $\mathbb{F}_q^*$.

Step 4. Let σ be the element in R_q such that $\phi(\sigma) = s$. Use the elliptic curve factoring method to test randomly selected elements in $\{0, \ldots, q-2\}$ until an integer z is found such that the element in R_q congruent to $\tau^z\sigma \bmod p$ is $p^{\frac{n}{d+1}}k^{-1}$-smooth. Call this element η. Using Algorithm 6.2.9 in [8], find the prime ideals lying above the primes dividing $N_{\mathbb{Q}}^F(\eta)$. If the norm of any of these ideals is greater than $p^{\frac{n}{d+1}}k^{-1}$, begin Step 4 again. Otherwise use Algorithm 4.10.17 in [8] to factor (η) into a product $\prod \mathfrak{q}_i^{e_i}$ of powers of prime ideals. For each i, apply LLL reduction to the lattice obtained by mapping $\mathfrak{q}_i$ into $\mathbb{Z}^n$ as related in subsection 2.1, and in this way produce an element $v_i \in \mathfrak{q}_i$ such that

$$h(v_i) \leq 2^{\frac{n-1}{4}} N(\mathfrak{q}_i)^{\frac{1}{n}}(r-1) \leq 2^{\frac{n-1}{4}} \left(\frac{p^{\frac{n}{d+1}}}{k}\right)^{\frac{1}{n}} (r-1) = p^{\frac{1}{d+1}}.$$

Notice that $\prod v_i^{e_i} \in (\eta)$ and let v be the element in $\mathcal{O}_F$ such that $v\eta = \prod v_i^{e_i}$. Then v is k-smooth. Since $k \leq B$, it is also the case that v is B-smooth.

Step 5. For each prime power l^e dividing $q-1$ and for all i, compute the least positive residue mod l^e of $\log_{\phi(\tau)} \phi(v_i)$. If $l \leq \max\{r, B\}$, use the Pohlig-Hellman-Silver method. Otherwise, use Algorithm 3.1.

Step 6. For each prime power l^e dividing $q-1$, compute the least positive residue of $\log_{\phi(\tau)} \phi(v) \bmod l^e$. If $l \leq \max\{r, B\}$, use the Pohlig-Hellman-Silver method. Otherwise, proceed as follows. Let $\sigma \in \mathcal{O}_F$ be any element of height $\leq p^{\frac{1}{d+1}}$. Run Steps 1–3 of Algorithm 3.1 with input parameters d and B as if to compute $\log_{\phi(\tau)} \phi(\sigma)$. In particular, obtain the matrix A described in Step 3. Next solve the congruence

$$AX \equiv -w_v \bmod l^e, \tag{5.2}$$

where w_v is defined as w_τ was in Algorithm 3.1 except with τ now replaced by v. The first entry in the solution to (5.2) is likely to be congruent to $-\log_{\phi(\tau)} \phi(v) \bmod l^e$. If it is not, repeat the process using a new model for $\mathbb{F}_q$ as described in Step 5 of Algorithm 3.1.

Step 7. Use the Chinese remainder theorem and the fact that

$$z + \log_{\phi(\tau)} s \equiv \sum e_i \log_{\phi(\tau)} \phi(v_i) - \log_{\phi(\tau)} \phi(v) \mod (q-1),$$

to compute $\log_{\phi(\tau)} s$.

Step 8. Let θ be the element in R_q such that $\phi(\theta) = t$. Using Steps 4-7, with s replaced by t and σ replaced by θ, compute $\log_{\phi(\tau)} t$. Now determine $\log_t s$ by means of the identity

$$\log_t s \equiv \frac{\log_{\phi(\tau)} s}{\log_{\phi(\tau)} t} \mod (q-1).$$

This concludes our description of Algorithm 5.1.

In the conjecture below and the ensuing discussion, we assume that n is constant and that all $o(1)$'s are for $p \to \infty$.

Conjecture 5.3. *Let $d = d(p) \geq 2$ and $B = B(p) \geq 2^{n(n-1)/4}(r-1)^n$ be functions such that*

$$d = ((3n)^{1/3} + o(1))(\log p/\log\log p)^{1/3} \tag{5.4}$$

and

$$B = L_p[1/3; (8n/9)^{1/3} + o(1)].$$

For $s, t \in \mathbb{F}_q^*$, let $T_{s,t}(p)$ be the expected running time of Algorithm 5.1 upon input of p, n, the pair s, t, and parameters $d(p)$ and $B(p)$. Let $T(p)$ be the maximum value of $T_{s,t}(p)$ taken over all pairs s, t. Then

$$T(p) = L_q[1/3; (64/9)^{1/3} + o(1)]. \tag{5.5}$$

As indicated in subsection 2.4, we assume that $r = O((\log p)^2)$. Under this assumption, (5.4) implies Algorithm 5.1 stops after Step 1 for only finitely many primes p. Conjecture 5.3, therefore, is based on our analysis of the subsequent steps of the algorithm. Since the time needed by the number field sieve factoring method to factor a number the size of q is conjectured to be $L_q[1/3; (64/9)^{1/3} + o(1)]$ ([6]), we begin with Step 3.

In [5, §4], the authors show that if the ERH is true, then there exists a constant c such that, for all p and each l dividing $q-1$, the set

$$\left\{ \sum a_j t_j \in \mathcal{O}_F \;\middle|\; |a_j| \le c(\log p)^{\max\{n-1,2\}} \right\}$$

contains an element whose image under ϕ lies outside the subgroup of $\mathbb{F}_q^*$ of order $(q-1)/l$. We therefore expect the asymptotic running time of Step 3 to be polynomial in $\log p$. Turning to Step 4, we conjecture on the basis of Theorem 2.2.1 that the expected number of trials required to find a suitable z is $L_p[1/3; (n/9)^{1/3} + o(1)]$. The conjectured expected time needed for the elliptic curve method to determine whether a candidate is $p^{\frac{n}{d+1}} k^{-1}$-smooth is $L_p[1/3; (8n/9)^{1/3} + o(1)]$ (see [21]). We conclude that z can be found in time $L_p[1/3; (3n)^{1/3} + o(1)]$, and leave it to the reader to verify that the time required subsequently to factor (η) and obtain the elements v_i is polynomial in $\log p$. Next, using Conjecture 4.12 and the fact that the Pohlig-Hellman-Silver method requires time $O(e(\log p + l))$ to compute the residue of a logarithm mod l^e, we calculate that the expected time needed to compute all the residues in Steps 5 and 6 is $L_p[1/3; (64n/9)^{1/3} + o(1)]$. Finally, we observe that the running time of Step 7 is polynomial in $\log p$ and that Step 8 increases the running time of the algorithm by a constant factor. Considering all the steps together and using the fact that

$$L_p[1/3; (64n/9)^{1/3} + o(1)] = L_q[1/3; (64/9)^{1/3} + o(1)],$$

we obtain (5.5), and our argument in support of Conjecture 5.3 is complete.

References

[1] L.M. Adleman, *The function field sieve*, Algorithmic number theory, ANTS-I (eds. L.M. Adleman, M.-D. Huang), Lecture Notes in Comput. Sci., vol. 877, Springer-Verlag, Berlin, 1994, pp. 108–121. MR **96d:**11135

[2] L.M. Adleman, J. DeMarrais, *A subexponential algorithm for discrete logarithms over all finite fields*, Math. Comp. **161** (1993), 1–15. MR **94e:**11140

[3] L.M. Adleman, M.-D. Huang, *Function field method for discrete logarithms over finite fields*, preprint (1998).

[4] E. Bach, J. Shallit, *Factoring with cyclotomic polynomials*, Math. Comp. **52** (1989), 201–219. MR **89k:**11127

[5] J.A. Buchmann, V. Shoup, *Constructing nonresidues in finite fields and the extended Riemann hypothesis*, Math. Comp. **65** (1996), 1311–1326. MR **96j:**11169

[6] J.P. Buhler, H.W. Lenstra, Jr., C. Pomerance, *Factoring integers with the number field sieve*, The development of the number field sieve (eds. A.K. Lenstra, H.W. Lenstra, Jr.), Lecture Notes in Math., vol. 1554, Springer-Verlag, Berlin, 1993, pp. 50–94. MR **96m:**11116

[7] E.R. Canfield, P. Erdös, C. Pomerance, *On a problem of Oppenheim concerning "factorisatio numerorum"*, J. Number Theory **17** (1983), 1–28. MR **85j:**11012

[8] H. Cohen, *A course in computational algebraic number theory*, Springer-Verlag, Berlin, 1993. MR **94i:**11105

[9] D. Coppersmith, *Modifications to the number field sieve*, J. Cryptology **6** (1993), 169–180. MR **94h:**11111

[10] T. Denny,, *A Lanczos implementation for $GF(p)$*, Universität des Saarlandes, to appear.

[11] T. ElGamal, *A subexponential-time algorithm for computing discrete logarithms over $GF(p^2)$*, IEEE Trans. Inform. Theory **31** (1985), 473–481. MR **86j:**11130

[12] D. Gordon, *Discrete logarithms using the number field sieve*, Siam J. Discrete Math. **6** (1993), 124–138. MR **94d:**11104

[13] D. Gordon, K. McCurley, *Massively parallel computation of discrete logarithms*, Advances in cryptology – Crypto '92 (ed. E.F. Brickell), Lecture Notes in Comput. Sci., vol. 740, Springer-Verlag, Berlin, 1993, pp. 312–323.

[14] M. Kraitchik, *Théorie des nombres, Vol. 1*, Gauthier-Villars, Paris, 1922.

[15] M. Kraitchik, *Recherches sur la théorie des nombres*, Gauthier-Villars, Paris, 1924.

[16] M. LaMacchia, A. Odlyzko, *Solving large sparse linear systems over finite fields*, Advances in cryptology – Crypto '90 (eds. A.J. Menezes, S.A. Vanstone), Lecture Notes Comput. Sci., vol. 537, 1991, pp. 109–133.

[17] S. Lang, *Algebraic number theory*, Addison-Wesley, Reading, MA, 1970. MR **44:**181

[18] A.K. Lenstra, H.W. Lenstra, Jr., (eds.), *The development of the number field sieve, Lecture Notes in Math.*, vol. 1554, Springer-Verlag, Berlin, 1993. MR **96m:**11116

[19] A.K. Lenstra, H.W. Lenstra, Jr., L. Lovász, *Factoring polynomials with rational coefficients*, Math. Ann. **261** (1982), 515–534. MR **84a:**12002

[20] A.K Lenstra, H.W. Lenstra, Jr., M.S. Manasse, J.M. Pollard, *The number field sieve*, The development of the number field sieve (eds. A.K. Lenstra, H.W. Lenstra, Jr.), Lecture Notes in Math., vol. 1554, Springer-Verlag, Berlin, 1993, pp. 11–42. MR **96m:**11116

[21] H.W. Lenstra, Jr., *Factoring integers with elliptic curves*, Ann. of Math. **126** (1987), 649–673. MR **89g:**11129

[22] H.W. Lenstra, Jr., *Algorithms for finite fields*, Number theory and cryptography (ed. J.H. Loxton), London Math. Soc. Lecture Note Ser., vol. 154, Cambridge Univ. Press, Cambridge, 1990, pp. 76–85. MR **92b:**11091

[23] H.W. Lenstra, Jr., *Algorithms in algebraic number theory*, Bull. Amer. Math. Soc. **26** (1992), 211–244. MR **93g:**11131

[24] R. Lovorn-Bender, *Rigorous, subexponential discrete logarithms in $GF(p^2)$*, Siam J. Discrete Math. (to appear).

[25] K. McCurley, *The discrete logarithm problem*, Cryptology and computational number theory (ed. C. Pomerance), Proc. Sympos. Appl. Math., vol. 42, 1990, pp. 49–74. MR **92d:**11133

[26] A. Menezes, T. Okamoto, S. Vanstone, *Reducing elliptic curve logarithms to logarithms in a finite field*, IEEE Trans. Inform. Theory **39** (1993), 1639–1646. MR **95e:**94038

[27] J.M. Pollard, *The lattice sieve*, The development of the number field sieve (eds. A.K. Lenstra, H.W. Lenstra, Jr.), Lecture Notes in Math., vol. 1554, Springer-Verlag, Berlin, 1993, pp. 43–49. MR **96m:**11116

[28] C. Pomerance, *The role of smooth numbers in number-theoretic algorithms*, Proceedings of the International Congress of Mathematicians (Zurich, 1994) (ed. S.D. Chatterji), Birkhäuser, Basel, 1995, pp. 411–422. MR **97m:**11156

[29] O. Schirokauer, *Discrete logarithms and local units*, Theory and applications of numbers without large prime factors (ed. R.C. Vaughan), Philos. Trans. Roy. Soc. London Ser. A, vol. 345, The Royal Society, London, 1993, pp. 409–423. MR **95c:**11156

[30] O. Schirokauer, *Determing the gap between the function field sieve and the number field sieve*, in preparation.

[31] O. Schirokauer, D. Weber, T. Denny, *Discrete logarithms - the effectiveness of the index calculus method*, Algorithmic number theory, ANTS-II (ed. H. Cohen), Lecture Notes in Comput. Sci., vol. 1122, Springer-Verlag, Berlin, 1996, pp. 337–361. MR **98i:**11109

[32] V. Shoup, *Searching for primitive roots in finite fields*, Math. Comp. **58** (1992), 369-380. MR **92e:**11140

[33] D. Weber, *An implementation of the number field sieve to compute discrete logarithms mod p*, Advances in cryptology – Eurocrypt '95 (eds. L.C. Guillou, J.-J. Quisquater), Lecture Notes in Comput. Sci., vol. 921, Springer-Verlag, Berlin, 1995, pp. 95–105.

[34] D. Weber, *Computing discrete logarithms with the number field sieve*, Algorithmic number theory, ANTS-II (ed. H. Cohen), Lecture Notes in Comput. Sci., vol. 1122, Springer-Verlag, Berlin, 1996, pp. 391–403. MR **98k:**11186

[35] D. Weber, T. Denny, *The solution of McCurley's discrete log challenge*, Advances in cryptology – Crypto '98 (ed. H. Krawczyk), Lecture Notes in Comput. Sci., vol. 1462, Springer-Verlag, Berlin, 1998.

[36] D. H. Wiedemann, *Solving sparse linear equations over finite fields*, IEEE Trans. Inform. Theory **32** (1986), 54–62. MR **87g:**11166

Department of Mathematics, Oberlin College, Oberlin, OH 44074
E-mail address: `oliver@occs.oberlin.edu`