Algorithmic Number Theory
MSRI Publications
Volume **44**, 2008

# The impact of the number field sieve on the discrete logarithm problem in finite fields

OLIVER SCHIROKAUER

## 1. Introduction

Let $p$ be a prime number and $n$ a positive integer, and let $q = p^n$. Let $\mathbb{F}_q$ be the field of $q$ elements and denote by $\mathbb{F}_q^*$ the multiplicative subgroup of $\mathbb{F}_q$. Assume $t$ and $u$ are elements in $\mathbb{F}_q^*$ with the property that $u$ is in the subgroup generated by $t$. The discrete logarithm of $u$ with respect to the base $t$, written $\log_t u$, is the least non-negative integer $x$ such that $t^x = u$.

In this paper we describe two methods to compute discrete logarithms, both of which derive from the number field sieve (NFS) factoring algorithm described in [Stevenhagen 2008] and [Lenstra and Lenstra 1993]. When factoring an integer $N$ with the NFS, we first choose a number ring $R$ as in [Stevenhagen 2008] for which there is a ring homomorphism $\phi : R \to \mathbb{Z}/N\mathbb{Z}$. Then we combine smooth elements in $R$ and in $\mathbb{Z}$ to obtain squares $\alpha_1 = a_1^2 \in R$ and $\alpha_2 = a_2^2 \in \mathbb{Z} \subseteq R$, such that $\phi(\alpha_1) = \phi(\alpha_2)$. If $\phi(a_1) \neq \pm\phi(a_2)$, then the gcd of $N$ and $a_1' - a_2$, where $a_1'$ is a representative in $\mathbb{Z}$ for $\phi(a_1)$, is a non-trivial factor of $N$. When using the strategy to compute discrete logarithms in $\mathbb{F}_q$, we choose either two number rings or two polynomial rings, call them $R_1$ and $R_2$, such that there are ring homomorphisms $\phi_1 : R_1 \to \mathbb{F}_q$ and $\phi_2 : R_2 \to \mathbb{F}_q$. We then construct two $(q-1)$-st powers $\alpha_1 \in R_1$ and $\alpha_2 \in R_2$ such that $\phi_1(\alpha_1) = t^x u \cdot \phi_2(\alpha_2)$ for some $x$. It follows that $x \equiv -\log_t u \bmod (q-1)$. When the chosen rings are number rings, the algorithm retains the title of number field sieve. In this case, if $q$ is prime, one of the rings can be taken to be $\mathbb{Z}$ as is done for factoring. However, it is often advantageous to use two non-trivial extensions of $\mathbb{Z}$ instead (see Section 2.2). When the chosen rings are polynomial rings, the algorithm is known as the function field sieve (FFS).

One difficulty encountered in the NFS and FFS discrete logarithm algorithms is that in order to combine smooth elements in $R_1$ into a $(q-1)$-st power which is mapped by $\phi_1$ to a multiple of $t^x u$, it is necessary to find pre-images of $t$ and $u$ under $\phi_1$ which are smooth. This problem is the subject of Section 4 of the paper. Sections 2 and 3 are devoted to descriptions of the NFS and FFS, respectively, in the case that $t$ and $u$ come with smooth pre-images under $\phi_1$.

The importance of the number field and function field sieves lies in the fact that they are faster than other algorithms for computing discrete logarithms, both asymptotically and in practice. The expected running time of the NFS of Section 2 is conjectured to be

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

where

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

and the $o(1)$ is for $q \to \infty$ subject to the constraint that $n$ does not grow too fast (see Section 2.8 for a precise formulation). The conjectured expected running time of the FFS of Section 3 is

$$L_q[1/3; (32/9)^{1/3} + o(1)], \tag{1.1}$$

where again the $o(1)$ is for $q \to \infty$, this time with the restriction that $p$ does not grow too fast (see Section 3.4). Taken together, the algorithms have a conjectured running time of $L_q[1/3; O(1)]$ for all finite fields. By contrast, no other discrete logarithm algorithm has a proven or conjectural running time faster than $L_q[1/2; O(1)]$. Coppersmith's method for fields of characteristic two [Coppersmith 1984], which predates the FFS by a decade and which also runs in time (1.1), is considered here as a special case of the FFS (see Section 3.8).

In practice, the current record for computing logarithms in a prime field is held by Kleinjung [Kleinjung 2007], who used the NFS as described in [Joux and Lercier 2003] to compute logarithms in a field whose cardinality is a prime of 160 digits. In characteristic two, Joux and Lercier's computation of logarithms in the field of size $2^{613} \approx 3.399 \cdot 10^{184}$ using the FFS [Joux and Lercier 2005a] is the present record. Interest in computations of discrete logarithms in fields that are neither prime nor of characteristic two is a relatively recent phenomenon, spurred on in part by cryptographic applications. In particular, the fact that the discrete logarithm problem on an elliptic curve over a prime field $\mathbb{F}_p$ can be transported to the logarithm problem in an extension of $\mathbb{F}_p$ [Menezes et al. 1993], [Frey et al. 1999], has focused attention on these fields. Joux and Lercier's use of the FFS to compute logarithms in the field of size $370801^{30} \approx 1.186 \cdot 10^{167}$ [Joux and Lercier 2005b] is an indication that at current levels, the difficulty of computing logarithms in these "intermediate" fields is comparable to that of doing so in the prime and characteristic two cases.

It often occurs in practice that more than one logarithm in a given field is sought. When this is the case, the NFS and FFS can be split into a precomputation stage in which the logarithms of the ”small” elements in the field are computed, and a fast reduction stage in which the desired logarithm is expressed in terms of the precomputed values. We refer the reader to [Schirokauer 2005] and [Joux and Lercier 2002] for descriptions of such versions of the NFS and FFS and note that they use the the same basic structure and techniques and have the same conjectural running times as the methods described in the present paper.

## 2. The number field sieve (NFS)

**2.1.** Let $p$ be an odd prime number. We adopt as a model for the finite field $\mathbb{F}_p$ the set of non-negative integers less than $p$, with addition and multiplication taken modulo $p$. Let $B$ be some positive real number, and recall that an integer is said to be $B$-smooth if each of its prime factors is at most $B$. We say that an element in $\mathbb{F}_p$ is $B$-smooth if it is $B$-smooth as an integer. Let $t$ and $u$ be elements in $\mathbb{F}_p^*$ which are $B$-smooth and for which $u \in \langle t \rangle$. In this section, we describe how to use the NFS to compute, not the residue of $\log_t u$ modulo $(p-1)$ as suggested in the introduction, but the residue of $\log_t u$ modulo an odd prime divisor $l$ of $p-1$. The reason for the restriction is given in Step 3 below. The algorithm we present can be modified to compute the residue of $\log_t u$ modulo any prime power divisor of $p-1$ [Schirokauer 1993]. Once these residues are known, $\log_t u$ is easily determined by means of the Chinese remainder theorem. We note that to compute the residue of $\log_t u$ modulo a power of a small prime, the exponential-time methods described in [Pomerance 2008] are preferable to the NFS. In what follows, therefore, we think of $l$ as being large.

**2.2. The NFS for prime fields.** In addition to a prime $p > 5$, an odd prime divisor $l$ of $p-1$, smoothness bound $B$, and elements $t$ and $u$ as described above, the algorithm takes as input a parameter $C \geq 1$ and an integral parameter $d$ satisfying $\log_2 p > d \geq 1$. It outputs an integer $x$ which is likely to be congruent to $\log_t u$ modulo $l$.

**Step 1. Constructing the number rings.** The idea of the NFS is to produce a relation in $\mathbb{F}_p$ involving $t$ and $u$ by choosing two number rings which come with maps to $\mathbb{F}_p$ and then building $l$-th powers in these rings in such a way that they have the same image in $\mathbb{F}_p$. The challenge is to find rings in which the construction of suitable $l$-th powers requires as little work as possible.

One approach, presented in [Joux and Lercier 2003], is to choose an irreducible polynomial $f_1$ of degree $d$ with small, integral coefficients and a root modulo $p$, call it $m$. The set of vectors $(a_0, \ldots, a_{d-1}) \in \mathbb{Z}^d$ having the property that the polynomial $\sum a_i X^i$ has $m$ as a root mod $p$ is a lattice. Lattice reduction techniques can, therefore, be used to obtain a polynomial $f_2$ of degree $d-1$, having integral coefficients of size approximately $p^{1/d}$ and $m$ as a root mod $p$. We now proceed with the rings $R_1 = \mathbb{Z}[\alpha_1]$ and $R_2 = \mathbb{Z}[\alpha_2]$, where $\alpha_1$ and $\alpha_2$ are roots in $\mathbb{C}$ of $f_1$ and $f_2$ respectively. Note that for $j = 1, 2$ the map $\phi_j : R_j \to \mathbb{F}_p$ that sends the element $\sum b_i {\alpha_j}^i$, with $b_i \in \mathbb{Z}$, to the element in $\{0, \dots, p-1\}$ congruent to $\sum b_i m^i$ mod $p$ is a ring homomorphism.

A second method to choose rings for the NFS is to define $m$ to be $\lfloor p^{1/d} \rfloor$ and let $f = \sum a_i X^i$ where the coefficients $a_0, \dots, a_d$ are obtained by writing $p$ in the base $m$. In other words, $0 \le a_i < m$ and

$$p = \sum_{i=0}^{d} a_i m^i.$$

Then $f$ is irreducible [Brillhart et al. 1981] and we choose as our two rings $\mathbb{Z}$ and $R = \mathbb{Z}[\alpha]$, where $\alpha \in \mathbb{C}$ satisfies $f(\alpha) = 0$. The required maps to $\mathbb{F}_p$ are the canonical projection from $\mathbb{Z}$ and its extension $\phi : R \to \mathbb{F}_p$ which sends $\sum b_i \alpha^i$, with $b_i \in \mathbb{Z}$, to the element in $\{0, \dots, p-1\}$ congruent to $\sum b_i m^i$ mod $p$.

Despite the fact that of these two methods, the former is frequently the better one in practice, we continue with the latter approach for the remainder of the section. Not only does the resulting presentation more closely parallel the usual formulation of the NFS algorithm for factoring, but the exposition is made less cumbersome by having only one non-trivial extension of $\mathbb{Z}$ to handle. All the techniques described for this one ring can be applied to the two non-trivial extensions produced by the first method.

Proceeding, therefore, with $f$ and $R$ as above, we observe that

(i) the coefficients of $f$ are bounded by $p^{1/d}$;
(ii) $f$ has a root mod $p$ of size at most $p^{1/d}$;
(iii) $f$ is monic.

A fourth property, which $f$ is not guaranteed to satisfy but which we assume holds, is

(iv) $l$ does not divide the discriminant of $f$.

The bounds in (i) and (ii) are critical to the running time analysis of the algorithm as they determine a bound for the numbers tested for smoothness in the next step. Properties (iii) and (iv) are not necessary to the algorithm but simplify our exposition. See [Buhler et al. 1993] and [Schirokauer et al. 1996] for a version of the algorithm that does not require that (iii) hold and [Schirokauer 1993] for the modifications necessary if (iv) does not hold.

**Step 2. Sieving.** Let $\mathbb{O}$ be the ring of integers of the number field $\mathbb{Q}(\alpha)$ and $N : \mathbb{Q}(\alpha) \to \mathbb{Q}$ the norm map. An element $\gamma \in \mathbb{O}$ is said to be $B$-smooth if $N(\gamma)$ is $B$-smooth in $\mathbb{Z}$, or equivalently, if each prime ideal dividing the ideal generated by $\gamma$ lies over a rational prime $\leq B$. In this step, we use sieving techniques to find the set $S$ of elements $(a, b) \in \mathbb{Z} \times \mathbb{Z}$ such that $|a|, |b| \leq C$ and both $a - b\alpha$ and $a - bm$ are $B$-smooth. This stage is exactly like the sieving step in the NFS for factoring. We refer the reader to [Buhler et al. 1993] and [Lenstra et al. 1993b] for details.

For $a, b \in \mathbb{Z}$ satisfying $|a|, |b| \leq C$, we have

$$\begin{aligned} |a - bm| &\leq C(p^{1/d} + 1), \\ |N(a - b\alpha)(a - bm)| = |b^d f(a/b)(a - bm)| &\leq 2^d (d + 1) C^d p^{1/d}. \end{aligned} \tag{2.3}$$

The product of the bounds in (2.3) then is a bound on the size of the integer that must be $B$-smooth for a pair $(a, b)$ to be in $S$. When the values $C$ and $d$ are chosen optimally, this product is significantly smaller than the size of the candidates for smoothness in other index calculus algorithms, such as those described in [McCurley 1990]and [Schirokauer et al. 1996]. It is for this reason that the NFS is faster than these methods.

**Step 3. Computing exponent vectors.** Let $\pi(B)$ be the number of rational primes $\leq B$ and let $q_1, \dots, q_{\pi(B)}$ be a list of these primes. Similarly, let $\varPi(B)$ be the number of prime ideals of $\mathbb{O}$ which are either of norm $\leq B$ and of degree 1 or lie over a rational prime $\leq B$ dividing $[\mathbb{O} : R]$, and let $\mathfrak{q}_1, \dots, \mathfrak{q}_{\varPi(B)}$ be a list of these ideals. As explained in [Buhler et al. 1993], for each $(a, b) \in S$, the prime ideal factors of $(a - b\alpha)$ are contained in this list. For each rational prime $q$ and integer $g$, let $v_q(g)$ be the exponent to which $q$ divides $g$. Similarly, for each prime ideal $\mathfrak{q} \subseteq \mathbb{O}$ and element $\gamma \in \mathbb{O}$, let $v_{\mathfrak{q}}(\gamma)$ be the exponent to which $\mathfrak{q}$ divides the ideal generated by $\gamma$.

For $(a, b) \in S$, compute the vector $V_{a,b}$ of length $\pi(B) + \varPi(B) + d$ whose first $\pi(B)$ entries are

$$v_{q_1}(a - bm), \dots, v_{q_{\pi(B)}}(a - bm),$$

whose next $\varPi(B)$ entries are

$$v_{\mathfrak{q}_1}(a - b\alpha), \dots, v_{\mathfrak{q}_{\varPi(B)}}(a - b\alpha),$$

and whose last $d$ entries are the images of $a - b\alpha$ under the character maps defined in the next paragraph. The values $v_{q_i}(a - bm)$ can be read off of the prime factorization of $a - bm$ and are easily obtained from the sieve in Step 2. The values $v_{\mathfrak{q}_i}(a - b\alpha)$ can be read off of the prime factorization of $N(a - b\alpha)$ for all $\mathfrak{q}_i$ for which the localization of $R$ at $R \cap \mathfrak{q}_i$ is integrally closed and hence is a discrete valuation ring. In this case, the sieve in Step 2 again produces the needed entries. For the remaining prime ideals, each of which lies over a prime dividing $[\mathbb{O} : R]$, the desired values can be efficiently computed using the method sketched in [Lenstra 1992] and described in detail in [Cohen 1993]. See [Stevenhagen 2008, formula (7.4)] and [Buhler et al. 1993] for a discussion of the relationship between the factorization of $(a - b\alpha)$ and $N(a - b\alpha)$.

Let

$$\Gamma = \{\gamma \in \mathbb{O} \mid N(\gamma) \not\equiv 0 \bmod l\}.$$

Let $\varepsilon$ be the least common multiple of the orders of the multiplicative groups $(\mathbb{O}/\ell)^*$, where $\ell$ ranges over the prime ideals lying above $l$. Since $l$ does not divide the discriminant of $f$ and is therefore unramified in $\mathbb{Q}(\alpha)$, we have for all $\gamma \in \Gamma$,

$$\gamma^\varepsilon \equiv 1 \bmod l. \tag{2.4}$$

Let $\lambda : \Gamma \to l\mathbb{O}/l^2\mathbb{O}$ be the map sending $\gamma$ to $(\gamma^\varepsilon - 1) + l^2\mathbb{O}$. We obtain $d$ maps

$$\lambda_j : \Gamma \to \mathbb{Z}/l\mathbb{Z}$$

by fixing a module basis $\{b_j l + l^2\mathbb{O}\}_{j=1,\dots,d}$ for $l\mathbb{O}/l^2\mathbb{O}$ over $\mathbb{Z}/l\mathbb{Z}$ and projecting $\lambda$ onto each coordinate. In other words, the $\lambda_j$ are given by the congruence

$$\gamma^\varepsilon - 1 \equiv \sum_{j=1}^{d} \lambda_j(\gamma) b_j l \bmod l^2.$$

Since $\lambda(\gamma\gamma') = \lambda(\gamma) + \lambda(\gamma')$ and $\lambda_j(\gamma\gamma') = \lambda_j(\gamma) + \lambda_j(\gamma')$, the maps $\lambda$ and $\lambda_j$ are homomorphisms on $\mathbb{O}^*$. We include in the vector $V_{a,b}$ the values $\lambda_1(a - b\alpha), \dots, \lambda_d(a - b\alpha)$.

The role of the maps $\lambda_j$ is to enable us to construct elements in $\mathbb{O}$ which are $l$-th powers. In the next step, we produce an element $\gamma \in \Gamma$ such that $v_\mathfrak{q}(\gamma) \equiv 0 \bmod l$ for all prime ideals $\mathfrak{q} \in \mathbb{O}$ and such that $\lambda(\gamma) = 0$. If the class number of $K$ is prime to $l$, as we expect for large $l$, then $\gamma$ is certain to generate the $l$-th power of a principal ideal and hence is the product of an $l$-th power and a unit $\omega$. Since any $l$-th power is mapped to 0 by $\lambda$, we find that $\omega$ is mapped to 0 as well. We claim that it is likely in this case that $\omega$, and in turn $\gamma$, is an $l$-th power. For more on the $\lambda_j$, including a precise formulation of the above claim and a heuristic argument supporting it, see [Schirokauer 1993]. Finally, note that it is in order to calculate the $\lambda_j$ that we need to work with a single prime divisor of $p - 1$. If analogous, logarithmic maps with values in $\mathbb{Z}/(p-1)\mathbb{Z}$ could be computed without factoring $p - 1$, then $\log_t u$ could be obtained without knowing the factorization of $p - 1$.

**Step 4. Linear algebra.** Let $V_t$ be the vector of length $\pi(B) + \Pi(B) + d$ whose first $\pi(B)$ entries are $v_{q_1}(t), \dots, v_{q_{\pi(B)}}(t)$ and whose last $\Pi(B) + d$ coordinates are all 0. Define $V_u$ similarly, with $u$ in place of $t$. Let $A$ be the matrix whose first column is $V_t$ and remaining columns are the vectors $V_{a,b}$. Now solve the congruence

$$AX \equiv -V_u \bmod l. \tag{2.5}$$

If $V_u$ is not in the column space of $A$, increase the parameter $C$ in order to enlarge the set $S$ and, one expects, the rank of $A$.

When the parameters $B, C$, and $d$ are chosen in order to minimize the time required for the sieving in Step 2, subject to the constraint that $C$ is large enough that (2.5) can be solved, one finds that the time required for Step 2 is equal to $r^{2+o(1)}$, where $r$ is the column length of $A$ and the $o(1)$ is for $p \to \infty$. Since one would like to be able to solve (2.5) within the same amount of time, Gaussian elimination, which requires $O(r^3)$ steps, is not a good choice. Moreover, Gaussian elimination is not practical for matrices of the size arising in current implementations. Instead, it is best to use a method which takes advantage of the fact that almost all the entries of $A$ are 0. The most useful of these from a theoretical standpoint is the coordinate recurrence method which is described in [Wiedemann 1986] and which can be shown to solve (2.5) in time $r^{2+o(1)}$, as desired. Combinations and adaptations of three other methods, the conjugate gradient method, the Lanczos algorithm, and structured Gaussian elimination, have had success in practice (see [Odlyzko 2000] for discussion and references). Nevertheless, the linear algebra continues to be a significant practical concern and accounts in part for the fact that computing discrete logarithms with the NFS is more difficult than factoring, in which case the linear algebra is done modulo 2.

The entries in a solution to (2.5), excluding the first which corresponds to the element $t$, can be indexed by the pairs $(a, b) \in S$. Let $(x, \dots, x_{a,b}, \dots)$ be one such solution. Let

$$\delta = t^x u \prod (a - bm)^{x_{a,b}} \quad \text{and} \quad \gamma = \prod (a - b\alpha)^{x_{a,b}},$$

where $t$ and $u$ are thought of as integers. Then $v_q(\delta) \equiv 0 \bmod l$ for all primes $q$, and therefore $\delta$ is an $l$-th power. Similarly $v_{\mathfrak{q}}(\gamma) \equiv 0 \bmod l$ for all prime ideals $\mathfrak{q} \subseteq \mathbb{O}$. In addition, $\lambda_i(\gamma) = 0$ for $i = 1, \dots, d$. As remarked earlier, it is likely that $\gamma$ is then an $l$-th power in $\mathbb{O}$. We assume that this is the case; see [Schirokauer 1993] for comments on how to weaken this assumption. Clearly, $\delta$ is the $l$-th power of an element in $R$. The same may not be true of $\gamma$. However, since $f'(\alpha)\mathbb{O} \subseteq R$, we see that both $f'(\alpha)^l \delta$ and $f'(\alpha)^l \gamma$ are $l$-th powers of an element in $R$. Recall that $\phi : R \to \mathbb{F}_p$ is the ring homomorphism that satisfies $\phi(\alpha) = \phi(m)$. Since $\phi(f'(\alpha)^l \delta)$ and $\phi(f'(\alpha)^l \gamma)$ are $l$-th powers and $\phi(f'(\alpha)^l \delta) = t^x u \phi(f'(\alpha)^l \gamma)$, we find that $t^x u$ is an $l$-th power in $\mathbb{F}_p^*$ and conclude that $x \equiv -\log_t u \bmod l$.

We observe that the smoothness of $t$ and $u$ is used in the algorithm to ensure that these two elements appear in the relations constructed in Step 4. However, it suffices to know two elements $\tau$ and $\upsilon$ in $R$ such that $\phi(\tau) = t$ and $\phi(\upsilon) = u$ and such that the ideals $(\tau)$ and $(\upsilon)$ in $\mathbb{O}$ factor over the set $\mathfrak{q}_1, \dots, \mathfrak{q}_{\Pi(B)}$ introduced in Step 3. In this case, the vectors $V_t$ and $V_u$ are replaced by vectors $V_\tau$ and $V_\upsilon$ containing the exponents appearing in the prime ideal factorizations of $(\tau)$ and $(\upsilon)$ in $\mathbb{O}$, as well as the values $\lambda_j(\tau)$ and $\lambda_j(\upsilon)$, and the linear algebra yields products

$$\prod (a - bm)^{x_{a,b}} \quad \text{and} \quad \tau^x \upsilon \prod (a - b\alpha)^{x_{a,b}},$$

which are expected to be $l$-th powers.

EXAMPLE 2.6. Let $p$ be the Mersenne prime $2^{127} - 1$ discovered by Lucas in 1876. Since $p$ is of the form $r^e - s$ with $r$ and $s$ small in absolute value, we can use the techniques of the special number field sieve described in [Lenstra et al. 1993b]. The analysis given there reveals that the optimal value of $d$ in the present example is 3. We proceed to construct a cubic extension of $\mathbb{Q}$ by noting the $p$ divides $2^{129} - 4 = (2^{43})^3 - 4$. Hence the polynomial $f = X^3 - 4$ has a root $m$ mod $p$ which is close to $p^{1/3}$ and has extremely small coefficients. It is, therefore, ideally suited to our purpose, and we let $R = \mathbb{Z}[\sqrt[3]{4}]$ and $\phi : R \to \mathbb{F}_p$ be the map which sends $\sqrt[3]{4}$ to $2^{43}$. In Step 2, we look for pairs $a, b$ such that $a - 2^{43}b$ and $N(a - b\sqrt[3]{4}) = a^3 - 4b^3$ are both smooth. In Step 3, we encounter a potential complication due to the fact that $R$ is not the full ring of integers $\mathbb{O}$ of $\mathbb{Q}(\sqrt[3]{4})$, as demonstrated by the fact that $(\sqrt[3]{4})^2/2$ is equal to $\sqrt[3]{2}$ and so is integral. This difficulty is readily handled, however, by observing that $\mathbb{O} = \mathbb{Z}[\sqrt[3]{2}]$ and that the only prime dividing $[\mathbb{O} : R]$ is 2; see [Marcus 1977, Chapter 2]. Since $2 = (\sqrt[3]{2})^3$, there is only one prime ideal of $\mathbb{O}$ lying above 2 and its residue degree is one. The exponent to which this ideal divides $(a - b\sqrt[3]{4})$ is therefore equal to $v_2(N(a - b\sqrt[3]{4}))$. Thus all the entries in the exponent vectors $V_{a,b}$ in Step 3 can be obtained from the factorizations of $a - 2^{43}b$ and $N(a - b\sqrt[3]{4})$.

We consider briefly the computation of the residue of $\log_t v$ modulo the largest prime divisor of $p - 1$, a prime of eleven digits which we denote by $l$ and which divides $p - 1$ only once. Since $x^3 - 2$ splits completely over $\mathbb{F}_l$, the value of $\varepsilon$ in (2.4) is $l - 1$. We note that $\mathbb{O}$ is a principal ideal domain, as is easily seen by computing the Minkowski constant, and that $\lambda$ induces an injective $\mathbb{F}_l$-linear map from $\mathbb{O}^*/(\mathbb{O}^*)^l$ to $l\mathbb{O}/l^2\mathbb{O}$, a fact which can be proved by showing that $(\sqrt[3]{2} - 1)^{-1}$ is the fundamental unit of $\mathbb{O}$ (see [Marcus 1977, Exercise 5.36]) and checking that $\lambda(\sqrt[3]{2} - 1) \neq 0$. Thus any element in the kernel of $\lambda$ is an

$l$-th power. We conclude that the product $\prod(a - b\sqrt[3]{4})^{x_{a,b}}$ obtained in Step 4 is itself an $l$-th power and that the algorithm produces the desired residue.

One consequence of the special nature of $f$ in this example is that we could have computed generators for the prime ideals and unit group of $\mathbb{O}$ and explicitly factored the smooth elements found in Step 2 into a product of powers of these generators. In this case, we would not have needed the additive characters, but would have constructed $(p-1)$-st powers by finding a dependency modulo $p-1$ among the vectors containing the exponents appearing in these factorizations. In particular, we could have avoided factoring $p-1$. For more on this approach, which may be particularly attractive for number fields of small discriminant, see [Lenstra et al. 1993b; 1993a].

**2.7. The NFS for general fields.** Let $p$ be a prime and $n > 1$, and let $q = p^n$. In this section, we briefly describe two methods for computing logarithms in $\mathbb{F}_q$. In the first, we build the NFS on top of a number field $F$ of small discriminant having $\mathbb{F}_q$ as a residue field. To find a suitable field, we look for a prime $r$ having the property that $\mathbb{Q}(\zeta)$, where $\zeta$ is a primitive $r$th root of unity, has a subfield of degree $n$ over $\mathbb{Q}$ in which $p$ is inert. This subfield serves as $F$. If the Extended Riemann Hypothesis holds, then we are guaranteed to find such a prime $r$ satisfying $r < (\log q)^{O(1)}$ [Shoup 1992]. One attractive feature of our choice of $F$ is that its ring of integers $\mathbb{O}_F$ has an integral basis consisting of the conjugates of the trace of $\zeta$ in $F$. See [Schirokauer 2000] for details. With $F$ in hand, we construct an extension of $F$ of degree $d$ by adjoining to it a root $\alpha$ of the polynomial obtained by expanding $p$ in base $m$, where $m = \lfloor p^{1/d} \rfloor$. Next, we look for pairs $a, b \in \mathbb{O}_F \times \mathbb{O}_F$ such that $a - b\alpha$ and $a - bm$ are both $B$-smooth. As before, we consider approximately $C^2$ many pairs, chosen so that for any candidate, the coefficients appearing in the expressions $a = \sum a_j t_j$ and $b = \sum b_j t_j$, where $\{t_1, \dots, t_n\}$ is the integral basis specified above, are at most $C^{1/n}$ in absolute value. The $B$-smooth pairs can be identified using sieving techniques or the elliptic curve factoring method [Lenstra 1987]. For each such pair, as well as for $t$ and $u$, we construct exponent vectors, including this time the values of the additive characters for both $F$ and $F(\alpha)$. Finally, we perform linear algebra mod $l$ to obtain an integer which is likely to be the residue of $\log_t u \bmod l$.

In the second approach, proposed in [Joux et al. 2006], two number rings with maps to $\mathbb{F}_q$ are obtained by letting $f_1$ be any polynomial of degree $n$ with small coefficients which is irreducible mod $p$ and letting $f_2 = f_1 + p$. Then for $i = 1, 2$, we set $R_i = \mathbb{Z}[\alpha_i]$, where $\alpha_i \in \mathbb{C}$ is a root of $f_i$. Because $f_1$ is irreducible mod $p$, there exists $v \in \mathbb{F}_q$ such that $\mathbb{F}_q = \mathbb{F}_p(v)$ and $v$ is a root of the polynomial in $\mathbb{F}_p[X]$ obtained by reducing the coefficients of $f_1$ mod $p$. For $i = 1, 2$, we let $\phi_i : R_i \to \mathbb{F}_q$ be the map that sends $\alpha_i$ to $v$. The natural next step in the NFS is to look for pairs $a, b \in \mathbb{Z}\times\mathbb{Z}$ such that $a-b\alpha_1$ and $a-b\alpha_2$ are $B$-smooth. We might expect this search to go more quickly than the corresponding step in the previous approach, since $\alpha_1$ and $\alpha_2$ are of degree $n$ over $\mathbb{Q}$, whereas $\alpha$ and $m$ above are of degrees $dn$ and $n$ respectively. However, this advantage is countered by the fact that instead of having $2n$ coefficients $a_1, \dots, a_n, b_1, \dots, b_n$ available as was the case before, there are only two parameters $a$ and $b$ to vary. In order to increase the pool of smoothness candidates, it is necessary to expand the search to include higher degree polynomials in $\alpha_1$ and $\alpha_2$. A bound on this degree enters as a new parameter in the analysis of the method. Once enough pairs are found, linear algebra is performed on the associated exponent vectors in order to obtain the desired residue of $\log_t u \bmod l$.

**2.8. Running time.** We consider first the running time of the sieving stage in the case that the field in which we compute is prime. As we have seen, the candidates for smoothness in this case are bounded by

$$C(p^{1/d}+1)\cdot(d+1)C^d p^{1/d} \le 2dC^{d+1}p^{2/d}. \tag{2.9}$$

Let $x$ denote the right hand side of (2.9). In order to be able to solve (2.5), we expect that the number of double-smooth pairs which are needed, call it $N$, is slightly larger than the length of the vectors $V_{a,b}$ appearing in Step 3. This length is equal to $\pi(B)+\Pi(B)+d$, which is bounded by $(d+1)B+d$. Thus, if $d = B^{o(1)}$ for $p \to \infty$, we have

$$N = B^{1+o(1)}. \tag{2.10}$$

Let $\psi(x, B)$ be the number of positive integers $\le x$ which are $B$-smooth. We adopt the critical assumption that the integers which are tested for smoothness in Step 2 behave like random numbers with regard to the property of being $B$-smooth. Then we can interpret the quotient $xN/\psi(x, B)$ as the number of pairs that need to be tested. The following theorem, which is copied verbatim from [Buhler et al. 1993], tells us the optimal value for this quantity in the case that (2.10) holds.

THEOREM 2.11. *Suppose $g$ is a function defined for all $y \ge 2$ that satisfies $g(y) \ge 1$ and $g(y) = y^{1+o(1)}$ for $y \to \infty$. Then as $x \to \infty$,*

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

*uniformly for all $y \ge 2$. In addition,*

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

*for $x \to \infty$ if and only if $y = L_x[1/2; \sqrt{2}/2+o(1)]$ for $x \to \infty$.*

Theorem 2.11 reveals that if (2.10) is valid, then

$$C^2 \geq L_x[1/2; \sqrt{2} + o(1)]. \tag{2.12}$$

In the best case that equality holds in (2.12), the values of $C$ and $d$ which minimize $C$ satisfy

$$\begin{aligned} C &= L_p[1/3; (8/9)^{1/3} + o(1)], \\ d &= \big((3 + o(1)) \log p / \log\log p\big)^{1/3}, \end{aligned} \tag{2.13}$$

where the limit implicit in the $o(1)$ is for $p \to \infty$. We thus arrive at the conjecture that the number of pairs that need to be tested in Step 2 for the algorithm to succeed is equal to

$$L_p[1/3; (64/9)^{1/3} + o(1)]. \tag{2.14}$$

In [Buhler et al. 1993] and [Schirokauer 2000], the reader will find a much more careful analysis in support of this claim. A quick calculation shows that when (2.13) holds, $x = L_p[2/3; (64/3)^{1/3} + o(1)]$, and a second calculation using Theorem 2.11 reveals that the optimal value of $B$ satisfies

$$B = L_p[1/3; (8/9)^{1/3} + o(1)]. \tag{2.15}$$

Finally, investigation of the entire method in the case that (2.13) and (2.15) hold, leads to the conjecture that (2.14) not only represents the sieving time in Step 2 but is also a bound for the running time of the other steps of the algorithm. We note that in the case of the special number field sieve exhibited in Example 2.6, the numbers being tested for smoothness are bounded in absolute value by $2dC^{d+1}p^{1/d}$. The fact that the power of $p$ is smaller in this expression than in (2.9) results in a reduced running time of $L_p[1/3; (32/9)^{1/3} + o(1)]$. The gain from $(64/9)^{1/3}$ to $(32/9)^{1/3}$ means that for large $p$, logarithms can be computed in $\mathbb{F}_p$ when $p$ is special in nearly the same amount of time as is needed to handle a general prime field of size $\sqrt{p}$.

We now turn to the complexity of the methods described for non-prime fields. To analyze the first approach, in which the NFS is built on top of a field $F$, we define the height of $\gamma \in \mathbb{O}_F$ by the formula $h(\gamma) = \max\{|\sigma\gamma|\}$, where $\sigma$ ranges over the embeddings of $F$ into $\mathbb{R}$. Then the norm of $\gamma$ is at most $h(\gamma)^n$ in absolute value. The elements in $\mathbb{O}_F$ that are tested for smoothness in this version of the NFS are of the form $b^d f(a/b)(a - bm)$ and are of height at most

$$2dC^{(d+1)/n} r^{d+1} p^{2/d}.$$

Hence, the integers being tested for $B$-smoothness are bounded in absolute value by

$$(2d)^n C^{d+1} q^{2/d} r^{(d+1)n} \leq d^n C^{d+1} q^{2/d} (\log q)^{O(dn)}.$$

We see immediately that the way in which $n$ grows as $q \to \infty$ has an effect on the running time of the algorithm. Indeed, if

$$n \leq o\Big(\frac{\log q}{\log\log q}\Big)^{1/3}, \tag{2.16}$$

then, when $d = (\log q/\log\log q)^{1/3}$, the factor $d^n(\log q)^{O(dn)}$ is at most $L_q[2/3; o(1)]$. In this case, it is conjectured that the algorithm runs in time $L_q[1/3; (64/9)^{1/3} + o(1)]$ as in the prime case. Moreover, if the little-oh in (2.16) is replaced by a big-oh, the secondary constant $(64/9)^{1/3}$ is lost but the primary constant $1/3$ is retained in the running time. We leave it as an exercise to show more generally that if $q \to \infty$ subject to the restriction that $n \leq O(\log q/\log\log q)^{e_n}$ for some constant $e_n$, then the running time is conjecturally equal to $L_q[\max\{1/3, (1+e_n)/4\}; O(1)]$.

To analyze the second approach given, in which $R_1$ is generated by a polynomial $f_1$ with small coefficients and $R_2$ by the polynomial $f_2 = f_1 + p$, we again seek a bound on the integers being tested for smoothness. Considering $R_1$ first, we note that the norm of an element of the form $\sum_{i=0}^{e} a_i {\alpha_1}^i$ is equal to the resultant of $f_1$ and the polynomial $\sum a_i X^i$. This resultant, in turn, is equal to the determinant of the associated Sylvester matrix and is therefore bounded in absolute value by $(n+e)^{n+e} M^n D_1^e$, where $M$ is a bound on the absolute value of the coefficients $a_i$ and $D_1$ is a bound, assumed to be small, on the absolute value of the coefficients of $f_1$. Combining this quantity with the corresponding value for $f_2$, we obtain a bound on the smoothness candidates of

$$(n+e)^{2(n+e)} M^{2n} (pD_1 + D_1^2)^e. \tag{2.17}$$

In order to compare this bound with the one arising in the prime field case, we assume that the number of candidates tested is $C^2$. It follows that $M < C^{2/e}$, and we can replace (2.17) with

$$x = (n+e)^{2(n+e)} C^{4n/e} (pD_1 + D_1^2)^e. \tag{2.18}$$

The constraint on $C$ is then given, as before, by (2.12). It is now straightforward to verify that (2.12) and (2.18) can be simultaneously satisfied with $C = L_q[1/3; O(1)]$ if and only if $n$ is bounded by $O(\log q/\log\log q)^{2/3}$ but not bounded by $O(\log q/\log\log q)^k$ for any $k < 1/3$. Indeed, for a given $n = (\log q/\log\log q)^{e_n}$ with $1/3 \leq e_n \leq 2/3$, the bound $e$ on the degree of the polynomials in $\alpha_1$ and $\alpha_2$ that are tested for smoothness should be set equal to $\lceil(\log q/\log\log q)^{e_n - 1/3})\rceil$. We refer the reader to [Joux et al. 2006] for more details and conclude by remarking that the two NFS methods for general fields which we have presented, taken in conjunction, conjecturally run in time $L[1/3; O(1)]$ so long as $q \to \infty$ with $n \leq O(\log q/\log\log q)^{2/3}$.

## 3. The function field sieve (FFS)

Adleman [1994] describes a function field analogue of the number field sieve which he calls the function field sieve. In order to compute logarithms in a finite field of characteristic $p$, the algorithm makes use of an algebraic extension of $\mathbb{F}_p(X)$ in the same way that the number field sieve makes use of an algebraic extension of $\mathbb{Q}$. Like the number field sieve, it is conjectured to run in time $L_q[1/3; O(1)]$, provided that the cardinality $q$ of the finite field tends to $\infty$ in a restricted fashion. In this section, we give a sketch of the FFS and a conjecture as to its complexity. We define a notion of smoothness for elements in $\mathbb{F}_q$ and as we did in the preceding section, restrict ourselves to the special case that the logarithm base and the element whose logarithm we seek are smooth. The algorithm we describe below is not the one found in [Adleman 1994] but instead is a modification of the simpler and improved version which is presented in [Adleman and Huang 1999] and which incorporates some of the techniques found in the algorithm of Coppersmith for fields of characteristic two [Coppersmith 1984]. The relationship between the FFS and Coppersmith's method is explored in Section 3.8.

**3.1. The FFS.** Let $p$ be a prime and $q = p^n$. We begin by choosing a model for the finite field $\mathbb{F}_q$. Let $g$ be a polynomial of minimal degree such that $X^n + g$ is irreducible and at least one root of $g$ in some algebraic closure of $\mathbb{F}_p$ is of multiplicity one. Let $f = X^n + g$ and fix as a model for $\mathbb{F}_q$ the set of polynomials in $\mathbb{F}_p[X]$ of degree $< n$, with addition and multiplication taken modulo $f$.

Recall that a polynomial in $\mathbb{F}_p[X]$ is said to be $B$-smooth if it factors into irreducibles all of which are of degree at most $B$. Thinking of the elements of $\mathbb{F}_q$ as polynomials, we can apply the notion of smoothness to elements in $\mathbb{F}_q$. The version of the FFS we now describe takes as input a smoothness bound $B \geq 1$, two $B$-smooth elements $t, u \in \mathbb{F}_q^*$ satisfying $u \in \langle t \rangle$, and two integral parameters $C \geq 0$ and $d \geq 1$. It outputs $\log_t u$.

**Step 1. Constructing an extension field.** Let $F = \mathbb{F}_p(X)$. We build an extension of $F$ by adjoining to it a root of a polynomial of degree $d$. The special form of $f$ allows us to find a suitable polynomial with particularly small coefficients. Let $k$ be the smallest multiple of $d$ greater than or equal to $n$ and let $H = Y^d + X^{k-n} g \in \mathbb{F}_p[X, Y]$. The fact that $g$ has a root of multiplicity prime to $d$ implies by Eisenstein's criterion that $H$ is absolutely irreducible. Let $\overline{F}$ be an algebraic closure of $F$ and let $\alpha \in \overline{F}$ be a root of $H$, considered as a polynomial in $Y$ over $\mathbb{F}_p[X]$. Let $R = \mathbb{F}_p[X][\alpha]$ and denote by $K$ the field of fractions of $R$. Let $\phi : R \to \mathbb{F}_q$ be the map which sends an element $h(X, \alpha)$ to the polynomial of degree $< n$ congruent to $h(X, X^{k/d})$ mod $f$. Since $f$ divides $(X^{k/d})^d + X^{k-n} g$, we see that $\phi$ is a ring homomorphism.

**Step 2. Sieving.** Let $N : K \to F$ be the norm map. We say that an element in $R$ is $B$-smooth if its image under $N$ is $B$-smooth. Let $S$ be the set of pairs $(a, b) \in \mathbb{F}_p[X] \times \mathbb{F}_p[X]$ such that $\deg(a)$ and $\deg(b)$ are at most $C$ and both $a - b\alpha$ and $a - bm$ are $B$-smooth. In this step, we use a sieve to identify the elements in $S$. We refer the reader to [Gao and Howell 1999; Gordon 1993b; Joux and Lercier 2002; Thomé 2001] for details and note that testing candidates for smoothness is of no consequence to the complexity analysis of the algorithm as it is possible to factor polynomials in polynomial time using the algorithm of [Berlekamp 1970].

**Step 3. Computing valuation vectors.** Let $M_F$ be the set of discrete valuations of $F$ of degree $\leq B$ (see [Stichtenoth 1993] for background information on discrete valuations and places in function fields). Let $M_K$ be the set of discrete valuations of $K$ which are extensions of the valuations in $M_F$. For each $(a, b) \in S$, we construct a vector $V_{a,b}$ containing the values $v(a - bm)$ for all $v \in M_F$, and $v(a - b\alpha)$ for all $v \in M_K$.

The valuations in $M_F$, excluding the valuation at $\infty$ which we denote by $v_\infty$, are in one-to-one correspondence with the irreducible polynomials in $F$. Indeed, for each such polynomial $h$, we obtain a valuation $v_h$ whose value at an element $\gamma$ is the exponent to which $h$ divides $\gamma$. Thus, to determine $v_h(a-bm)$ for some pair $(a, b) \in S$, it suffices to factor $(a - bm)$ into irreducibles, a task already accomplished in Step 2. Since $v_h(a-bm) = 0$ for all $h$ of degree $> B$, and since for all $\gamma \in F$,

$$\sum_{h \text{ irreducible}} \deg(h) v_h(\gamma) = -v_\infty(\gamma),$$

we see that $v_\infty(a - bm)$ is obtained immediately.

Let $v$ be an element in $M_K$ which does not lie over $v_\infty$. Let $Q = \{\gamma \in K \mid v(\gamma) > 0\}$ be the associated place. As was the case with the NFS, if the localization of $R$ at $R \cap Q$ is a discrete valuation ring, then $v(a - b\alpha)$ can be read off the factorization of $N(a - b\alpha)$. If not, $v(a - b\alpha)$ can be computed by a method that uses Newton polygons to construct a fractional power series containing the information needed to evaluate $v$. This technique is discussed in detail in [Adleman and Huang 1999].

In the case that $v$ is an extension of $v_\infty$, we proceed by homogenizing $H$ with respect to a new variable $Z$ and dehomogenizing with respect to either $X$ or $Y$. Doing so yields a new polynomial $H'$. Renaming variables, we obtain a domain $\mathbb{F}_p[U, V]/H'(U, V)$ whose field of fractions $K'$ is isomorphic to $K$ under a map which sends the places associated to the extensions of $v_\infty$ to finite places in $K'$. The associated valuations can now be computed with the Newton polygon method alluded to above.

**Step 4. Linear algebra.** Let $V_t$ be the vector of length $|M_F|+|M_K|$ whose first $|M_F|$ entries are the values $v(t)$ for $v \in M_F$ and whose remaining coordinates are 0's. Define $V_u$ in the same way but with $t$ replaced by $u$. Let $A$ be the matrix whose first column is $V_t$ and remaining columns are the vectors $V_{a,b}$, and solve the congruence

$$AX \equiv -V_u \bmod (q-1)/(p-1). \tag{3.2}$$

Unlike what we encountered with the NFS, the linear algebra this time is done modulo a number which may be composite. See [McCurley 1990; Schirokauer 1993] for some comments on this situation. We continue under the assumption that we are able to produce a solution $(x, \ldots, x_{a,b}, \ldots)_{(a,b)\in S}$ to (3.2). Let

$$\delta = t^x u \prod (a - bm)^{x_{a,b}} \quad \text{and} \quad \gamma = \prod (a - b\alpha)^{x_{a,b}},$$

where $t$ and $u$ are thought of here as elements in $\mathbb{F}_p[X]$. Since

$$v(\delta) \equiv 0 \bmod (q-1)/(p-1) \quad \text{for all } v \in M_F,$$

we know that $\delta$ is a product of an element in $\mathbb{F}_p^*$ and a $(q-1)/(p-1)$-st power. Since any such power in $\mathbb{F}_q^*$ is in $\mathbb{F}_p^*$, we find that $\phi(\delta) \in \mathbb{F}_p^*$. Let $h$ be the gcd of $(q-1)/(p-1)$ and the class number of $K$. If $h = 1$, then the fact that $v(\gamma) \equiv 0 \bmod (q-1)/(p-1)$ for all $v \in M_K$ implies that $\gamma$ is the product of an an element in $\mathbb{F}_p^*$ and a $(q-1)/(p-1)$-st power. Hence, $\phi(\gamma) \in \mathbb{F}_p^*$. In this case, $t^x u = \phi(\delta)\phi(\gamma)^{-1} = \mu$ for some $\mu \in \mathbb{F}_p^*$, and we have

$$x \equiv -\log_t u \bmod (q-1)/(p-1).$$

All that remains is the computation of a logarithm in $\mathbb{F}_p^*$, namely $\log_{t'} \mu$ where $t' = t^{(q-1)/(p-1)}$. This can be accomplished with the NFS or one of the methods described in [Pomerance 2008]. If $h > 1$, we adopt the modification presented in [Schirokauer 2002].

EXAMPLE 3.3. Let $q = 2^{127}$ and assume we are trying to compute logarithms in $\mathbb{F}_q$. We adopt as our model for $\mathbb{F}_q$ the set of polynomials of degree at most 126, with addition and multiplication done modulo $f = X^{127} + X + 1$. The same optimization that was used in our NFS example indicates that we should let $d = 3$. Taking advantage of the special form of $f$, we let

$$H = Y^3 + X^2(X + 1).$$

The map $\phi$ in this case sends $\alpha$ to $X^{43}$ and the pairs of polynomials tested for smoothness in Step 2 are of the form $a^3 + b^3 X^2(X + 1)$ and $a + bX^{43}$.

We consider the problem of computing the valuation vectors described in Step 3. We do not give citations for the statements made below but refer the reader to [Stichtenoth 1993, Chapter III] for the theorems on which they are based.

Let $F = \mathbb{F}_2(X)$. Let $P_X$ and $P_{X+1}$ be the places of $F$ corresponding to the irreducible polynomials $X$ and $X+1$ respectively, and let $P_\infty$ be the place of $F$ at infinity. Let $Q$ be a place of $K$ lying over a place $P$ of $F$ other than $P_X$, $P_{X+1}$ or $P_\infty$, and let $v_Q$ be the associated valuation. Since $\alpha^3 = X^2(X+1)$ in $K$, we see that $v_Q(\alpha) = 0$. It follows that $R \cap Q$ is integrally closed, and therefore, the values $v_Q(a+b\alpha)$ can be read off of the factorization of $N(a+b\alpha)$. Since $K$ is a pure cubic extension of $F$ and $v_X(X^2(X+1))$ and $v_{X+1}(X^2(X+1))$ are both prime to $[K:F]$, we find that $P_X$ and $P_{X+1}$ are totally ramified in $K$. It follows that there is only one place, with residue degree one, lying above each of these places. We conclude, by the same reasoning given for the prime lying above 2 in Example 2.6, that $v_Q(a+b\alpha) = v_X(N(a+b\alpha))$, where $Q$ is the lone place above $P_X$ and $v_{Q'}(a+b\alpha) = v_{X+1}(N(a+b\alpha))$, where $Q'$ lies over $P_{X+1}$. We note that in the latter case, $R \cap Q'$ is a discrete valuation ring. Indeed, $H$ is non-singular at the point $(1,0)$. By contrast $H$ is singular at $(0,0)$ and the local ring $R \cap Q$ is not integrally closed, as is made explicit by the fact that $\alpha^2/X$ is a cube root of $X(X^2+1)$.

Finally, we consider $P_\infty$. Homogenizing the curve $Y^3 + X^2(X+1)$ with respect to a third variable $Z$ and dehomogenizing with respect to $X$ yields the curve $Y^3 + Z + 1$. Let $\beta$ be a root of $V^3 + U + 1$ in an algebraic closure of $\mathbb{F}_2(U)$. Let $E = \mathbb{F}_2(U)(\beta)$ and note that the map $\psi : K \to E$ given by

$$\psi(X) = 1/U \quad \text{and} \quad \psi(\alpha) = \beta/U$$

is an isomorphism. The place $P_\infty \subset F$ is mapped by $\psi$ to the place $P_U$ of $\mathbb{F}_2(U)$ corresponding to the polynomial $U$. To determine the splitting behavior of this place in $E$, we look at the splitting behavior of the image of the polynomial $V^3+U+1$ in the residue field of $P_U$. This image is $V^3+1$, which is the product of a linear factor and an irreducible quadratic factor over $\mathbb{F}_2$. We conclude that $P_U$ splits into a place of degree 1 and a place of degree 2 in $E$. Let $v_1$ and $v_2$ be the corresponding valuations and observe that any element of the form $a + b\alpha \in K$, where $a$ and $b$ are polynomials in $X$, is mapped to an element in $E$ of the form $U^\varepsilon(a' + b'\beta)$ for some $\varepsilon$, where $a'$ and $b'$ are polynomials in $U$. We can now use the fact that $v_1(a' + b'\beta)$ is the exponent to which $U$ appears in the norm of $(a' + b'\beta)$ in $\mathbb{F}_2[U]$ and that $v_2(a' + b'\beta) = 0$ to complete the computation of the valuation vector for the pair $(a, b)$.

**3.4. Running time.** In the analysis of the FFS, we adopt the same assumption as we did in the the case of the NFS, namely that the elements being tested for smoothness behave as random elements. In this case, the elements are polynomials in $\mathbb{F}_p[X]$ and so to make use of our assumption, we need results concerning the probability that a polynomial of degree $M$ picked at random is $B$-smooth. Such results can be found in [Bender and Pomerance 1998] and reveal that the probability of interest is, roughly speaking, equal to the probability that an integer of size at most $p^M$ is $p^B$-smooth. It follows that the size of the factor base and number of pairs $(a, b)$ tested in Step 2 should be asymptotically equal to the analogous quantities in the special number field sieve. Note that we use the special NFS here because it and the FFS both make use of a small field extension obtained by taking advantage of a special representation of $\mathbb{F}_q$. Since the linear algebra problem is the same in both the special NFS and FFS, we arrive at the conjecture that the running time of the FFS, like that of the special NFS, is

$$L_q[1/3; (32/9)^{1/3} + o(1)]. \tag{3.5}$$

In [Adleman and Huang 1999], the authors provide a heuristic argument in support of this conjecture which analyzes the FFS directly and does not proceed by analogy with the NFS.

The only obstruction to our conjecture is the requirement that the smoothness bound be at least one. As a consequence, we find that the factor base in the algorithm is at least size $p$ and the linear algebra in Step 4 requires at least $p^2$ many steps. As $q \to \infty$, it may be the case that $p^2$ is greater than (3.5) and that (3.5) is therefore not valid. The reader can easily check that this does not happen if

$$p \le n^{o(\sqrt{n})} \tag{3.6}$$

as $q \to \infty$. It may, however, happen if (3.6) is relaxed to

$$p \le n^{O(\sqrt{n})}. \tag{3.7}$$

In this case the consequences are not so dramatic as the primary constant of 1/3 in (3.5) is retained in the running time of the algorithm. Outside the range defined by (3.7), however, the FFS no longer runs in time $L_q[1/3; O(1)]$. For these fields, the time required by the algorithm is $p^{2+o(1)}$; see [Schirokauer 2002].

Since $p = L_q[s; c]$ if and only if $n = c^{-1}(\log q/\log\log q)^{1-s}$, we see from the above discussion that for $q \to \infty$ such that $n \ge (\log q/\log\log q)^e$, the FFS runs conjecturally in time $L_q[\max\{1/3, 1-e\}; O(1)]$. Combining this result with that for the NFS presented in Section 2.8 , we conjecture that the algorithm which chooses for a given $q$ the faster of the NFS and FFS runs in time $L_q[1/3; O(1)]$ for all fields.

**3.8. Coppersmith's algorithm.** Coppersmith [1984] presents a method for computing logarithms in fields of characteristic two which has a conjectural expected running time of $L_q[1/3; c + o(1)]$, where $q$ is the cardinality of the field, the $o(1)$ is for $q \to \infty$, and $c$ is a constant which is equal to $(32/9)^{1/3}$ in the case that a certain quantity appearing in the algorithm is close to a power of two, and which is slightly larger otherwise. Coppersmith includes in his article a description of how to use his method to compute logarithms in the finite field considered in Example 3.3. He uses the same model for $\mathbb{F}_q$ as we give and observes that for any pair $(a, b) \in \mathbb{F}_2[X] \times \mathbb{F}_2[X]$,

$$(a + bX^{33})^4 \equiv a^4 + b^4 X(X + 1) \bmod (X^{127} + X + 1). \tag{3.9}$$

Thus, for each $(a, b)$ for which both sides of (3.9) are smooth, we obtain a relation in $\mathbb{F}_q$ involving only polynomials of degree at most the smoothness bound. Each such relation yields a linear relation among the logarithms of the elements appearing in it. If $t$ and $u$ are smooth, then once enough relations are found, $\log_t u$ can be determined using linear algebra mod $(q-1)$.

Assume now that we decide to compute logarithms in $\mathbb{F}_q$ with the FFS with $d = 4$, instead of the optimal value of 3. Then $H = Y^4 + X(X + 1)$ and $m = X^{33}$. Let $\alpha, R, K$ and $\phi$ be as given in the description of the FFS in Section 3.1, and observe that $K$ is a purely inseparable extension of $\mathbb{F}_2(X)$. The norm map $N : K \to \mathbb{F}_2(X)$ in this case sends $\gamma \in K$ to $\gamma^4$ and is additive as well as multiplicative. Since the norm of any element in the kernel of $\phi$ is in $(X^{127} + X + 1)$, we see that for any pair $\delta, \gamma \in R$ such that $\phi(\delta) = \phi(\gamma)$,

$$N(\delta) \equiv N(\gamma) \bmod (X^{127} + X + 1).$$

In the case that $\delta = a + bX^{33}$ and $\gamma = a + b\alpha$, we obtain congruence (3.9). Thus, finding $a, b$ such that the norms of $a + bX^{33}$ and $a + b\alpha$ are smooth, as required by the FFS, is equivalent to finding $a, b$ such that both sides of (3.9) are smooth. In this way, we see that Coppersmith's algorithm is a special case of the FFS in which the extension field disappears and the relations can be realized in the base ring $\mathbb{F}_2[X]$.

## 4. General discrete logarithms

Let $p$ be a prime and $q = p^n$, and let $R_1$ and $R_2$ be rings, together with maps $\phi_1 : R_1 \to \mathbb{F}_q$ and $\phi_2 : R_2 \to \mathbb{F}_q$, chosen for use in the NFS or FFS discrete logarithm algorithm. As we have seen, these methods proceed in two stages. First, sieving techniques are used to find pairs of smooth elements $(\delta_1, \delta_2) \in R_1 \times R_2$ satisfying $\phi_1(\delta_1) = \phi_2(\delta_2)$, and then a linear algebra computation produces the desired logarithm. We note that the particular logarithm problem being tackled comes into play in the second stage but has no bearing on the search in the first stage. In fact, once sufficiently many smooth pairs are collected, the linear algebra can be tailored to produce $\log_t u$ for any $t$ and $u$ so long as they are the images in $\mathbb{F}_q$ of smooth elements in $R_1$ or $R_2$. In this section we discuss how to proceed if either $t$ or $u$ is an element for which we do not have a smooth pre-image under $\phi_1$ or $\phi_2$.

We begin with the case that $u$ fails to come with a smooth pre-image. One approach to this difficulty is to replace $R_1$ or $R_2$ by a ring $S$ that does contain a known smooth pre-image of $u$. The challenge is to do so while retaining all the desired features of the replaced ring. In the original adaptation of the number field sieve to the discrete logarithm problem, Gordon [1993a] provides a suitable method in the case that the field is prime and $u$ is represented by a moderately sized integer. To make use of the strategy, a reduction step is performed first in which a value $z$ is found such that $t^z u$ is represented by an integer with moderately sized factors. The algorithm runs in time $L_p[1/3; (64/9)^{13} + o(1)]$ for $p \to \infty$, but is not used in practice due to the fact that the entire NFS must be run multiple times in order to compute a single logarithm.

A second approach, which has its origins in Coppersmith's paper [1984] on computing discrete logarithms in characteristic two and is the method currently employed in practice, uses as its building block the following technique. Let $\gamma$ be an element in $R_1$ and assume $(\delta_1, \delta_2) \in R_1 \times R_2$ satisfies

(i) $\gamma | \delta_1$
(ii) $\delta_1/\gamma$ and $\delta_2$ are smooth.
(iii) $\phi_1(\delta_1) = \phi_2(\delta_2)$.

Then $\phi_1(\gamma) = \phi_2(\delta_2)/\phi_1(\delta_1/\gamma)$ in $\mathbb{F}_q$, and since $\delta_2$ and $\delta_1/\gamma$ are smooth, the logarithms of their images can be computed. Hence, the logarithm of $\phi_1(\gamma)$ can be determined.

At first glance, it might appear that this strategy is sufficient for computing $\log_t u$ for arbitrary $u$. Indeed, given the fact that only one pair of smooth elements is needed, the algorithm should be faster than those described earlier. The problem is that if we set the smoothness bound equal to the one used in the NFS or FFS, then for most $u$, any pre-image of $u$ in $R_1$ will be so large that, because of condition (i), the time needed to find $\delta_1$ and $\delta_2$ will greatly exceed the time need to compute the logarithms of $\phi_2(\delta_2)$ and $\phi_1(\delta_1/u)$ in $\mathbb{F}_q$.

The solution is to repeat the process multiple times, with more moderate expectations each time. In particular, we adopt a sequence of smoothness bounds $B_1 > \cdots > B_k = B$, where $B$ is the smoothness bound used in the NFS or FFS. Initially, we follow the above approach to reduce the problem of computing $\log_t u$ to the problem of computing the logarithms of a collection of images of elements in $R_1$ and $R_2$ of norm size at most $B_1$. Here, size refers to absolute value in the case of the NFS and degree in the case of the FFS. We then implement the procedure for each of these elements, using the bound $B_2$, thereby reducing the problem to the computation of logarithms of images of elements with norm size at most $B_2$. After $k$ rounds of descent, the original problem is reduced to the one which can be solved by the NFS or FFS. In fact, if we work back up the tree making all the necessary substitutions, we obtain $B$-smooth elements

$\sigma_1, \sigma_1' \in R_1$ and $\sigma_2, \sigma_2' \in R_2$ such that $u = \phi_1(\sigma_1)\phi_1(\sigma_1')^{-1}\phi_2(\sigma_2)\phi_2(\sigma_2')^{-1}$. Given this representation of $u$, the valuation vector $V_u$ is easily produced and $\log_t u$ can be computed with one running of the NFS or FFS.

One difficulty that arises in the analysis of this descent technique is that, because smoothness bounds are used that are larger than those in the NFS or FFS, the time required to sieve with all the primes of size up to such a bound dominates the entire computation. It thus becomes necessary to factor smoothness candidates individually. When using the reduction in conjunction with the NFS, for instance, these factorizations must be done with the elliptic curve factoring method and are costly. Nevertheless, Commeine and Semaev [2006] have shown that in the case of a prime field $\mathbb{F}_p$ the descent runs conjecturally in time $L_p[1/3; 3^{1/3} + o(1)]$ for $p \to \infty$. This is faster than the NFS of Section 2. On the FFS side, where polynomials can be factored quickly using polynomial gcd's, Joux and Lercier [2005a] provide an analysis showing that the running time of the reduction is bounded by that of the FFS of Section 3 so long as $p \leq n^{O(\sqrt{n})}$. Recall that this is the entire range in which the FFS runs in time $L_q[1/3; O(1)]$. Though it does not seem unlikely that the descent approach will lead to an $L[1/3; O(1)]$ reduction for all fields, the details in cases other than those cited have yet to be worked out.

In practice, the descent method appears to be very fast, as a look at any of the implementation announcements cited in the introduction will reveal. This may seem surprising, particularly in the NFS case, given our comments concerning the large smoothness bounds that are used and the apparent need to test smoothness candidates individually. However, an initial reduction step [Joux and Lercier 2003], which we do not describe here, and the use of sieving techniques despite the large bounds, speed the method up greatly. In this context, we note that condition (i) forces the search for smooth elements to take place in a lattice when working over $\mathbb{Z}$, or the analogous structure when working over $\mathbb{F}_p[X]$. For example, when the reduction is applied to $\gamma \in R_1$ in the case that $R_1 = \mathbb{Z}[\alpha]$ for some algebraic integer $\alpha$ and the elements tested for smoothness are linear in $\alpha$, the search is confined to the lattice

$$\{(a, b) \in \mathbb{Z}^2 \mid a - b\alpha \equiv 0 \bmod \gamma\}.$$

Sieving over a lattice structure is regularly done in implementations of the NFS and FFS and does not pose a difficulty. See [Pollard 1993] for an introduction to the subject.

We conclude this section by addressing the question of what to do if we know of no smooth element in $R_1$ or $R_2$ which maps to the base $t$. One way to proceed is to pick elements in $\mathbb{F}_q^*$ until a primitive element $t'$ with a known smooth preimage in $R_1$ or $R_2$ is found. The desired logarithm can then be determined by means of the identity

$$\log_t u \equiv \frac{\log_{t'} u}{\log_{t'} t} \bmod (q-1).$$

If $q = p$ is prime and the Extended Riemann Hypothesis (ERH) holds, then we are assured of finding such a $t'$ by testing the elements in $\mathbb{F}_p^*$ represented by the integers $\leq (\log\ p)^{O(1)}$ [Shoup 1992]. More generally, and again under the assumption that the ERH is true, a primitive element in $\mathbb{F}_q^*$ having a smooth pre-image in the ring $\mathbb{O}_F$ introduced in Section 2.7 can be obtained in time $(\log\ q)^{O(n)}$ [Buchmann and Shoup 1996]. If one represents $\mathbb{F}_q$ as a quotient of $\mathbb{F}_p[X]$ and searches for $t'$ among the elements in $\mathbb{F}_q^*$ represented by polynomials in $\mathbb{F}_p[X]$ of small degree, one is certain of succeeding within $(np)^{O(1)}$ trials [Shoup 1992]. Thus, we see that so long as $n \leq o(\log\ q/\log\log\ q)^{1/3}$ or $p \leq n^{o(\sqrt{n})}$, finding a primitive element with smooth pre-image can be accomplished without affecting the running time of the NFS or FFS.

In the event that we are unable to switch primitive elements, there is a second option. We can apply to $t$ the descent reduction just described for handling $u$. Doing so produces a representation of $t$ as the product of the images of smooth elements, or their inverses, from $R_1$ and $R_2$. Such a representation can then be used to obtain a valuation vector $V_t$, which in turn can be incorporated into the linear algebra computation. This approach highlights the fact that, although $t$ and $u$ have different positions in the logarithm problem, they are incorporated in the same way into the NFS and FFS methods described in this paper.

## References

[Adleman 1994] L. M. Adleman, "The function field sieve", pp. 108–121 in *Algorithmic number theory* (Ithaca, NY, 1994), edited by L. M. Adleman and M.-D. Huang, Lecture Notes in Comput. Sci. **877**, Springer, Berlin, 1994.

[Adleman and Huang 1999] L. M. Adleman and M.-D. A. Huang, "Function field sieve method for discrete logarithms over finite fields", *Inform. and Comput.* **151**:1-2 (1999), 5–16.

[Bender and Pomerance 1998] R. L. Bender and C. Pomerance, "Rigorous discrete logarithm computations in finite fields via smooth polynomials", pp. 221–232 in *Computational perspectives on number theory* (Chicago, 1995), edited by D. A. Buell and J. T. Teitelbaum, AMS/IP Stud. Adv. Math. **7**, Amer. Math. Soc., Providence, RI, 1998.

[Berlekamp 1970] E. R. Berlekamp, "Factoring polynomials over large finite fields", *Math. Comp.* **24** (1970), 713–735.

[Brillhart et al. 1981] J. Brillhart, M. Filaseta, and A. Odlyzko, "On an irreducibility theorem of A. Cohn", *Canad. J. Math.* **33**:5 (1981), 1055–1059.

[Buchmann and Shoup 1996] J. Buchmann and V. Shoup, "Constructing nonresidues in finite fields and the extended Riemann hypothesis", *Math. Comp.* **65**:215 (1996), 1311–1326.

[Buhler et al. 1993] J. P. Buhler, H. W. Lenstra, Jr., and C. Pomerance, "Factoring integers with the number field sieve", pp. 50–94 in *The development of the number field sieve*, edited by A. K. Lenstra and H. W. Lenstra, Jr., Lecture Notes in Math. **1554**, Springer, Berlin, 1993.

[Cohen 1993] H. Cohen, *A course in computational algebraic number theory*, Graduate Texts in Mathematics **138**, Springer, Berlin, 1993.

[Commeine and Semaev 2006] A. Commeine and I. Semaev, "An algorithm to solve the discrete logarithm problem with the number field sieve", pp. 174–190 in *Public key cryptography*, edited by M. Yung et al., Lecture Notes in Comput. Sci. **3958**, 2006.

[Coppersmith 1984] D. Coppersmith, "Fast evaluation of logarithms in fields of characteristic two", *IEEE Trans. Inform. Theory* **30**:4 (1984), 587–594.

[Frey et al. 1999] G. Frey, M. Müller, and H.-G. Rück, "The Tate pairing and the discrete logarithm applied to elliptic curve cryptosystems", *IEEE Trans. Inform. Theory* **45**:5 (1999), 1717–1719.

[Gao and Howell 1999] S. Gao and J. Howell, "A general polynomial sieve: Designs and codes — a memorial tribute to Ed Assmus", *Des. Codes Cryptogr.* **18**:1-3 (1999), 149–157.

[Gordon 1993a] D. M. Gordon, "Discrete logarithms in GF($p$) using the number field sieve", *SIAM J. Discrete Math.* **6**:1 (1993), 124–138.

[Gordon 1993b] K. Gordon, D. andMcCurley, "Massively parallel computation of discrete logarithms", pp. 312–324 in *Advances in Cryptology — Crypto '92*, edited by E. F. Brickell, Lecture Notes in Comput. Sci. **740**, Springer, Berlin, 1993.

[Joux and Lercier 2002] A. Joux and R. Lercier, "The function field sieve is quite special", pp. 431–445 in *Algorithmic number theory* (Sydney, 2002), Lecture Notes in Comput. Sci. **2369**, Springer, Berlin, 2002.

[Joux and Lercier 2003] A. Joux and R. Lercier, "Improvements to the general number field sieve for discrete logarithms in prime fields. A comparison with the Gaussian integer method", *Math. Comp.* **72**:242 (2003), 953–967.

[Joux and Lercier 2005a] A. Joux and R. Lercier, "Discrete logarithms in GF($2^{607}$) and GF($2^{613}$)", email to the NMBRTHRY mailing list, 23 September 2005.

[Joux and Lercier 2005b] A. Joux and R. Lercier, "Discrete logarithms in GF($370801^{30}$) — 168 digits — 556 bits", email to the NMBRTHRY mailing list, 9 November 2005.

[Joux et al. 2006] A. Joux, R. Lercier, N. P. Smart, and F. Vercauteren, "The function field sieve in the medium prime case", pp. 326–344 in *Advances in Cryptology – CRYPTO 2006*, edited by C. Dwork, Lecture Notes in Comput. Sci. **4117**, 2006.

[Kleinjung 2007] T. Kleinjung, "Discrete logarithms in GF($p$) — 160 digits", email to the NMBRTHRY mailing list, 5 February 2007.

[Lenstra 1987] H. W. Lenstra, Jr., "Factoring integers with elliptic curves", *Ann. of Math.* (2) **126**:3 (1987), 649–673.

[Lenstra 1992] H. W. Lenstra, Jr., "Algorithms in algebraic number theory", *Bull. Amer. Math. Soc.* (*N.S.*) **26**:2 (1992), 211–244.

[Lenstra and Lenstra 1993] A. K. Lenstra and H. W. Lenstra, Jr. (editors), *The development of the number field sieve*, Lecture Notes in Mathematics **1554**, Springer, Berlin, 1993.

[Lenstra et al. 1993a] A. K. Lenstra, H. W. Lenstra, Jr., M. S. Manasse, and J. M. Pollard, "The factorization of the ninth Fermat number", *Math. Comp.* **61**:203 (1993), 319–349.

[Lenstra et al. 1993b] A. K. Lenstra, H. W. Lenstra, Jr., M. S. Manasse, and J. M. Pollard, "The number field sieve", pp. 11–42 in *The development of the number field sieve*, edited by A. K. Lenstra and H. W. Lenstra, Jr., Lecture Notes in Math. **1554**, Springer, Berlin, 1993.

[Marcus 1977] D. A. Marcus, *Number fields*, Springer, New York, 1977.

[McCurley 1990] K. S. McCurley, "The discrete logarithm problem", pp. 49–74 in *Cryptology and computational number theory* (Boulder, CO, 1989), edited by C. Pomerance, Proc. Sympos. Appl. Math. **42**, Amer. Math. Soc., Providence, RI, 1990.

[Menezes et al. 1993] A. J. Menezes, T. Okamoto, and S. A. Vanstone, "Reducing elliptic curve logarithms to logarithms in a finite field", *IEEE Trans. Inform. Theory* **39**:5 (1993), 1639–1646.

[Odlyzko 2000] A. Odlyzko, "Discrete logarithms: the past and the future", *Des. Codes Cryptogr.* **19**:2-3 (2000), 129–145.

[Pollard 1993] J. M. Pollard, "The lattice sieve", pp. 43–49 in *The development of the number field sieve*, edited by A. K. Lenstra and H. W. Lenstra, Jr., Lecture Notes in Math. **1554**, Springer, Berlin, 1993.

[Pomerance 2008] C. Pomerance, "Elementary thoughts on discrete logarithms", pp. 385–396 in *Surveys in algorithmic number theory*, edited by J. P. Buhler and P. Stevenhagen, Math. Sci. Res. Inst. Publ. **44**, Cambridge University Press, New York, 2008.

[Schirokauer 1993] O. Schirokauer, "Discrete logarithms and local units", *Philos. Trans. Roy. Soc. London Ser. A* **345**:1676 (1993), 409–423.

[Schirokauer 2000] O. Schirokauer, "Using number fields to compute logarithms in finite fields", *Math. Comp.* **69**:231 (2000), 1267–1283.

[Schirokauer 2002] O. Schirokauer, "The special function field sieve", *SIAM J. Discrete Math.* **16**:1 (2002), 81–98.

[Schirokauer 2005] O. Schirokauer, "Virtual logarithms", *J. Algorithms* **57**:2 (2005), 140–147.

[Schirokauer et al. 1996] O. Schirokauer, D. Weber, and T. Denny, "Discrete logarithms: the effectiveness of the index calculus method", pp. 337–361 in *Algorithmic* *number theory* (Talence, 1996), edited by H. Cohen, Lecture Notes in Comput. Sci. **1122**, Springer, Berlin, 1996.

[Shoup 1992] V. Shoup, "Searching for primitive roots in finite fields", *Math. Comp.* **58**:197 (1992), 369–380.

[Stevenhagen 2008] P. Stevenhagen, "The number field sieve", pp. 83–100 in *Surveys in algorithmic number theory*, edited by J. P. Buhler and P. Stevenhagen, Math. Sci. Res. Inst. Publ. **44**, Cambridge University Press, New York, 2008.

[Stichtenoth 1993] H. Stichtenoth, *Algebraic function fields and codes*, Springer, Berlin, 1993.

[Thomé 2001] E. Thomé, "Computation of discrete logarithms in $\mathbb{F}_{2^{607}}$", pp. 107–124 in *Advances in cryptology — ASIACRYPT 2001* (Gold Coast), edited by C. Boyd, Lecture Notes in Comput. Sci. **2248**, Springer, Berlin, 2001.

[Wiedemann 1986] D. H. Wiedemann, "Solving sparse linear equations over finite fields", *IEEE Trans. Inform. Theory* **32**:1 (1986), 54–62.

OLIVER SCHIROKAUER
DEPARTMENT OF MATHEMATICS
OBERLIN COLLEGE
10 NORTH PROFESSOR STREET
OBERLIN, OH 44074-1019
UNITED STATES
oliver.schirokauer@oberlin.edu