SIAM J. DISC. MATH
Vol. 6, No. 1, pp. 124–138, February 1993

010

DISCRETE LOGARITHMS IN *GF(P)* USING THE NUMBER FIELD SIEVE*

DANIEL M. GORDON†

Abstract. Recently, several algorithms using number field sieves have been given to factor a number n in heuristic expected time $L_n[1/3; c]$, where

$$L_n[v; c] = \exp\{(c + o(1))(\log n)^v(\log\log n)^{1-v}\}$$

for $n \to \infty$.

This paper presents an algorithm to solve the discrete logarithm problem for $GF(p)$ with heuristic expected running time $L_p[1/3; 3^{2/3}]$. For numbers of a special form, there is an asymptotically slower but more practical version of the algorithm.

Key words. discrete logarithms, number field sieve

AMS(MOS) subject classification. 11Y16

1. Introduction. Given a prime p and integers a and b, the discrete logarithm problem in $GF(p)$ is to find an integer x (if any exists) such that

$$a^x \equiv b \pmod{p}. \tag{1}$$

The difficulty of computing discrete logarithms has been used in the construction of several cryptographic systems (see, for example, [15]). The most successful implementation of a discrete logarithm algorithm for $GF(p)$ to date is by LaMacchia and Odlyzko [11], who solved the discrete logarithm problem modulo primes of 58 and 67 digits using the Gaussian integers method. This method, introduced by Coppersmith, Odlyzko, and Schroeppel in [8], uses a complex quadratic field to aid the sieving process.

Define

$$L_x[v; c] = \exp\{(c + o(1))(\log x)^v(\log\log x)^{1-v}\}, \tag{2}$$

for $x \to \infty$. The Gaussian integers method, as well as several other methods described in [8], find discrete logarithms for $GF(p)$ in expected time $L_p[1/2; 1]$.

The idea of using number field sieves has been used recently for factoring. Lenstra et al. [13] have used a number field sieve to obtain rapid factorizations of numbers of the form $r^e \pm s$, for small r and s. Buhler, Lenstra, and Pomerance [5] have generalized this method to factor general numbers n in time $L_n[1/3; c]$. Adleman [1] and Coppersmith [7] have suggested further improvements.

Some necessary facts and heuristic assumptions about algebraic number theory and linear algebra computations are discussed in §2. In §3 an overview of an algorithm for computing discrete logarithms in $GF(p)$ using the number field sieve is given. Using these results and assumptions, §4 shows that the algorithm works in expected time $L_p[1/3; 3^{2/3}]$. Another version for special numbers, which is asymptotically slower but more practical, is given in §5.

2. Computational background. There are a number of specialized algorithms and heuristic assumptions that are needed to give a good running time for finding discrete logarithms with the number field sieve. Similar assumptions are used in [13] for estimating the time needed to factor with the number field sieve.

*Received by the editors April 9, 1990; accepted for publication (in revised form) February 26, 1992.
†Department of Computer Science, University of Georgia, Athens, Georgia 30602.

2.1. Smoothness. Call an integer y-smooth if all of its prime factors are at most y. Let $\psi(x, y)$ be the number of integers $\leq x$ that are y-smooth. We need results on the probabilities of various rational and algebraic integers being smooth. The following special case of a theorem of Canfield, Erdős, and Pomerance [6] gives an estimate for the probability of a number in a given range being smooth.

THEOREM 1. *Suppose that* $0 < w < v \leq 1, \gamma > 0$, *and* $\delta > 0$ *are fixed. Let* x *and* y *be functions of* p *such that* $x = L_p[v; \gamma]$ *and* $y = L_p[w; \delta]$ *for* $p \to \infty$. *Then*

$$\frac{\psi(x, y)}{x} = L_p[v - w; -\frac{\gamma}{\delta}(v - w)] \qquad \text{for } p \to \infty.$$

The ratio $\psi(x, y)/x$ is the probability that a random number in $(0, x]$ is y-smooth. In this paper, we deal with numbers near x that are not random, but we use the heuristic assumption that their probability of being smooth is also given by Theorem 1. For example, we assume that numbers of the form $c + dm$, for c and d running through a narrow range and m fixed, are smooth as often as random numbers of the same size.

The elliptic curve method (ECM) for factoring an integer n depends on finding an elliptic curve for which the order of the curve modulo a prime divisor of n is smooth (see [14]). The following conjecture implies that enough such curves exist so that the ECM can expect to find one in reasonable time.

CONJECTURE 1. *Given the conditions of Theorem* 1, *the probability that a random number in* $(x - \sqrt{x}, x + \sqrt{x})$ *is* y*-smooth is* $L_p[v - w; -\gamma/\delta(v - w)]$ *for* $p \to \infty$.

This conjecture implies the following special case of Conjecture 2.10 of [14].

CONJECTURE 2. *The expected time for the* ECM *to factor an* $L_p[v; c]$*-smooth integer in* $[0, p]$ *is* $L_p[v/2; \sqrt{2vc}]$ *for* $p \to \infty$.

2.2. Linear algebra. Another operation that will take a large part of the computation time is dealing with matrix equations over $\mathbb{Q}$. Given an $S \times T$ sparse integer matrix A, where $S > T$ and the entries in A are all at most T in absolute value, must find a linear relation over $\mathbb{Q}$ for the rows of A. This may be done by the following algorithm, due to Pomerance [17] (see [10] for an alternative algorithm).

ALGORITHM M. Let A be a $(T + 1) \times T$ matrix over $\mathbb{Z}$, with each row having at most E nonzero entries, each of absolute value at most T. This probabilistic algorithm returns a linear relation for the rows of A.

Step 1. Attempt to compute the rank r of A.

Choose a random prime $q_0 \leq ET \log T$. By using Gaussian elimination mod q_0, find the rank r_0 of A mod q_0. Rearrange the rows so that the first r_0 rows are linearly independent mod q_0. Call the rearranged rows $v_1, v_2, \cdots, v_{T+1}$. The result of the Gaussian elimination determines an $r_0 \times r_0$ submatrix $\hat{A}$ of the first r_0 rows of A such that $\hat{A}$ is nonsingular mod q_0.

Step 2. Attempt to express v_{r_0+1} as a linear combination of $v_1, \cdots, v_{r_0}$ mod q for each prime $q \leq ET \log T$.

We attempt this via Wiedemann's coordinate recurrence method [21]. Let $\mathbf{P}$ denote the product of the primes q for which we are successful, and let $\mathbf{P}'$ denote the product of the remaining primes up to $ET \log T$. If $\mathbf{P}' > (E^{1/2}T)^T$, then return to Step 1 and begin again.

Step 3. Attempt to compute the determinant D of $\hat{A}$.

For each prime $q|\mathbf{P}$, use Wiedemann's probabilistic determinant algorithm [21] to compute an integer $D_q \in \{0, 1, \cdots, q - 1\}$, which is the determinant of $\hat{A}$ mod q with

probability at least $1 - (ET)^{-2}$. Use the Chinese remainder theorem to compute the integer D_0 closest to zero with $D_0 \equiv D_q \bmod q$ for each prime $q|\mathbf{P}$. Repeat this step until a value of D_0 is found with $0 < |D_0| \leq (E^{1/2}T)^T$.

Step 4. Attempt to produce a linear relation among the rows of A.

With the Chinese remainder theorem and the results of Steps 2 and 3, compute the integers $c_1, \cdots, c_{r_0}$ closest to zero such that

$$D_0 v_{r_0+1} \equiv \sum_{i=1}^{r_0} c_i v_i \pmod{\mathbf{P}}.$$

If any c_i has absolute value exceeding $(E^{1/2}T)^T$, return to Step 3. Otherwise, we have found the relation

$$D_0 v_{r_0+1} = \sum_{i=1}^{r_0} c_i v_i. \tag{3}$$

THEOREM 2. *Suppose that* $T \geq E \geq 12$. *If Algorithm* M *terminates, then* (3) *is a correct equation. The expected running time of Algorithm* M *is* $O(E^2T^3 \log^3 T)$.

Proof. By the assumptions on A, we have that $\| v_i \| \leq E^{1/2}T$ for each row v_i of A. Thus, by Hadamard's inequality, the absolute value of the determinant of any submatrix of A is at most $(E^{1/2}T)^T$. From results of Rosser and Schoenfeld [18], it follows that the number of distinct prime factors of any such nonzero determinant is less than $2T$. However, from the same reference, the number $\pi(ET \log T)$ of primes $q \leq ET \log T$ exceeds $ET/3$. We can thus conclude that for at least half of the primes $q \leq ET \log T$, the rank of A mod q is equal to the rank r of A over $\mathbb{Q}$. Thus, with probability at least $1/2$, the number r_0 returned in Step 1 is equal to r. The running time for one iteration of Step 1 is $O(T^3 \log^2 T)$ bit operations.

If $r_0 = r$, then v_{r_0+1} is a linear combination of $v_1, \cdots, v_{r_0}$ over $\mathbb{Q}$, and the least common denominator of the rational scalars involved divides the determinant D of $\hat{A}$. Thus, if $r_0 = r$, then $\mathbf{P}' \leq (E^{1/2}T)^T$. If v_{r_0+1} is a linear combination of $v_1, \cdots, v_{r_0} \bmod q$, then Wiedemann's coordinate recurrence method will be able to express v_{r_0+1} as such a linear combination in $O(ET^2)$ operations mod q. Thus the running time for one iteration of Step 2 is $O(E^2T^3 \log^2 T)$ bit operations.

Wiedemann's determinant-finding algorithm can calculate the correct determinant with probability at least $1 - (ET)^{-2}$ in $O(ET^2 \log T)$ operations mod q. Among all the numbers D_q computed in Step 3, the probability that at least one such D_q is not congruent to D mod q is at most $\pi(ET \log T)(ET)^{-2}$. From [18] we have $\pi(ET \log T) < 2ET$. Thus the probability that the number D_0 computed in Step 3 is not D is at most $2(ET)^{-1}$. The time for the Chinese remainder theorem is $O(\log^2 \mathbf{P})$, which is $O((ET \log T)^2)$ by [18]. The total time for Step 3 is $O(E^2T^3 \log^3 T)$ bit operations.

If $D_0 = D$, then $D_0 v_{r_0+1}$ is an integral combination of $v_1, \cdots, v_{r_0}$, and the integer scalars $c_1, \cdots, c_{r_0}$ are all at most $(E^{1/2}T)^T$ in absolute value. Since $\mathbf{P} > 2(E^{1/2}T)^T$, knowing those scalars mod $\mathbf{P}$ is enough to determine them. Thus, if $D_0 = D$, then Step 4 will be successful; that is, we will not need to return to Step 3. Furthermore, (3) is a correct equation. The running time of Step 4 is $O(E^2T^3 \log^2 T)$. □

For the special number field sieve, we need only solve matrix equations modulo $p - 1$. This may be done using Wiedemann's algorithm in $O(ET^2 \log^2 T)$ bit operations for matrices satisfying the conditions specified in Algorithm M. If the factorization of $p - 1$ is known, a solution can be found modulo each prime factor, and a solution mod

$p-1$ can be obtained using the Chinese remainder theorem and Hensel's lemma. If not, then Wiedemann's algorithm may be used modulo $p-1$. Either the algorithm will work or it will discover a factor of $p-1$, and the algorithm may be repeated on each factor.

2.3. Algebraic number theory. Throughout this paper, p will be a prime for which we wish to solve the discrete logarithm problem in $GF(p)$. We represent $GF(p)$ by $\mathbb{Z}/p\mathbb{Z}$, where elements are identified with their least nonnegative residues.

We choose an integer m and $f(x) \in \mathbb{Z}[x]$ of degree k such that f is monic, irreducible over $\mathbb{Q}$, and $f(m) \equiv 0 \pmod{p}$. Such an f may be found by choosing an m of suitable size and finding the base m representation of p, say $p = \sum_{i=0}^{k} a_i m^i$. Then $f(x) = \sum_{i=0}^{k} a_i x^i$ satisfies $f(m) = p$ and is irreducible by a theorem of Brillhart, Filaseta, and Odlyzko [4].

We also require that p does not divide Δ_f, the discriminant of f. If this happens for a particular m, we may choose a different m, or alter f by adding m to some a_i and subtracting 1 from a_{i+1}. The irreducibility of the new f may be checked quickly; see [12]. Note that $\Delta_f = (-1)^{k(k-1)/2} R(f, f')$ may be calculated efficiently. $R(f, g)$ here denotes the resultant of f and g.

Let $\alpha \in \mathbb{C}$ denote a root of f, $K = \mathbb{Q}(\alpha)$, and $\mathcal{O}_K$ denote the ring of integers in K. If s is a prime number not dividing the index $[\mathcal{O}_K : \mathbb{Z}[\alpha]]$, then its factorization in $\mathcal{O}_K$ is given by the following proposition (see, for example, [22]).

PROPOSITION 1. *For a prime number s not dividing the index, suppose that f factors in $GF(s)[x]$ as*

$$f(x) \equiv \prod_i g_i(x)^{e_i} \bmod s, \tag{4}$$

with each g_i monic and irreducible mod s, *and $g_i \not\equiv g_j$ for $i \neq j$. Then $(s) = \prod_i \mathfrak{s}_i^{e_i}$, for different prime ideals $\mathfrak{s}_i = (s, g_i(\alpha))$ and $N(\mathfrak{s}_i) = s^{\deg(g_i)}$.*

In particular, since $(p, \Delta_f) = 1$, $\mathfrak{p} = (p, \alpha - m)$ is a first-degree prime factor of (p) in $\mathcal{O}_K$, and we have $\mathcal{O}_K/\mathfrak{p} \cong GF(p)$. We may define a homomorphism φ from $\mathbb{Z}[\alpha]$ to $\mathbb{Z}/p\mathbb{Z}$ as in other number field sieve algorithms, by sending α to $m \bmod p$.

We say a prime ideal of $\mathcal{O}_K$ is *bad* if its norm divides the index. All other prime ideals will be called *good*.

Prime numbers dividing the index can be recognized efficiently using a theorem of Dedekind (see [22]): Suppose that f factors mod s as in (4). Then the prime number s divides the index if and only if there is some j for which $e_j \geq 2$ and

$$(g_j \bmod s) \left| \left(s^{-1}(f - \prod_i g_i^{e_i}) \bmod s \right) \right.$$

as elements of $GF(s)[x]$.

For any $y \in \mathbb{Z}$, call an algebraic integer in $\mathbb{Z}[\alpha]$ y-smooth if it is divisible only by good prime ideals of $\mathcal{O}_K$ of norm at most y. We must find smooth numbers of the form $c + d\alpha$, for c and d rational, coprime integers of moderate size.

To do so, we start by attempting to factor

$$\begin{aligned} |N(c+d\alpha)| &= |(-d)^k f(-c/d)| \\ &= |c^k - a_{k-1}c^{k-1}d + \cdots + a_1 c(-d)^{k-1} + a_0(-d)^k| \\ &\leq (k+1) \cdot \max\{|c|, |d|\}^k \cdot \max_i\{|a_i|\}. \end{aligned} \tag{5}$$

PROPOSITION 2. *Suppose that $c, d \in \mathbb{Z}$ are coprime and $N(c+d\alpha)$ is relatively prime to the index $[\mathcal{O}_K : \mathbb{Z}[\alpha]]$. Then $(c + d\alpha)$ factors completely into good first-degree prime ideals in $\mathcal{O}_K$.*

Proof. For each rational prime s dividing $|N(c + d\alpha)|$, there is a unique ideal of norm s dividing $(c + d\alpha)$. This is because, if a prime ideal dividing s divides $(c + d\alpha)$, then $\alpha \equiv -c/d$ modulo the ideal, and since the right side is rational, the congruence holds mod s. Thus $c_s \equiv -c/d \pmod{s}$ is a root of f mod s and, by Proposition 1, determines the unique ideal $\mathfrak{s} = (s, \alpha - c_s)$ dividing $c + d\alpha$.

The norm $N(\mathfrak{s}) = |\mathcal{O}_K/\mathfrak{s}|$ is clearly a power of s. We have $|\mathbb{Z}[\alpha]/(\mathfrak{s} \cap \mathbb{Z}[\alpha])| = s$, since the representatives of classes in $\mathbb{Z}[\alpha]/(\mathfrak{s} \cap \mathbb{Z}[\alpha])$ are just $\alpha, \alpha + 1, \cdots, \alpha + (s - 1)$. Since $|\mathcal{O}_K/\mathbb{Z}[\alpha]|$ is relatively prime to s, $\mathcal{O}_K/\mathbb{Z}[\alpha]$ maps to the identity under reduction mod $\mathfrak{s}$, so $|\mathcal{O}_K/\mathfrak{s}| = s$ as well. Therefore the power of $\mathfrak{s}$ dividing $(c + d\alpha)$ is the same as the power of s dividing the norm. □

For the number fields K we deal with here, the discriminant will be huge, so most operations in K will be impractical. One operation we must be able to do is take a small set of units, given as products of a large number of algebraic integers, and find a multiplicative dependency among them.

Let r_1 be the number of real embeddings of K, let $2r_2$ be the number of complex embeddings, and let $r = r_1 + r_2$. Let $\sigma_1, \cdots, \sigma_{r_1}$ denote the real embeddings, and $\sigma_{r_1+1}, \overline{\sigma_{r_1+1}}, \cdots, \sigma_r, \overline{\sigma_r}$ the others. We define a mapping $l : K \to \mathbb{C}^{r_1+r_2}$ in the usual way, by

$$l(x) = (\log|\sigma_1(x)|, \cdots, \log|\sigma_{r_1}(x)|, 2\log|\sigma_{r_1+1}(x)|, \cdots, 2\log|\sigma_r(x)|).$$

This mapping sends the units in $\mathcal{O}_K$ into a lattice $\mathcal{L} \in \mathbb{R}^r$, with roots of unity mapped to the origin. The following theorem of Dobrowolski [9] shows that other units cannot be too close to the origin.

LEMMA 1. *Let γ be a nonzero algebraic integer in K, and denote by $\overline{|\gamma|}$ the maximal modulus of its conjugates. Then*

$$\overline{|\gamma|} < 1 + \frac{\log k}{6k^2}$$

only if γ is a root of unity.

This implies that, for any unit u that is not a root of unity, $\| l(u) \| > \log(1 + ((\log k)/6k^2)) > 1/(10k^2)$ for $k > 1$.

THEOREM 3. *Suppose that $M > 80rk^2$, and let $u_1, \cdots u_{2r}$ be units in $\mathcal{O}_K$, with $\| l(u_i) \| < M$ for $i = 1, \cdots, 2r$. Then there is a nontrivial linear relation*

$$\sum_{i=1}^{2r} c_i \cdot l(u_i) = \mathbf{0} \tag{6}$$

with each c_i an integer with $|c_i| < M^2$.

Proof. Consider the set S of all sums $\sum_{i=1}^{2r} c_i \cdot l(u_i)$ with $0 \le c_i < M^2$. There are formally M^{4r} such sums, and it suffices to show that two of them are equal.

For all vectors $s \in S$, we have $\| s \| < 2rM^3$. Therefore all $s \in S$ are in an r-dimensional sphere of radius $2rM^3$, and, by the lemma, no two members of $\mathcal{L}$ are closer than $1/(10k^2)$ to each other. Let $V_r(x)$ denote the volume of an r-dimensional sphere of radius x. Then the number of lattice points in the sphere is at most

$$\frac{V_r(2rM^3 + 1/(20k^2))}{V_r(1/(20k^2))} < (80rk^2M^3)^r = M^{3r}(80rk^2)^r.$$

This is less than M^{4r}, however, and so, by the pigeonhole principle, there must be two equal vectors in S. □

This dependence does not cancel out the units completely, since the resulting unit $\prod u_i^{c_i}$ could be a root of unity. If an lth root of unity is in a field of degree $k \geq \phi(l)$, then we have $l < 6k \log\log k$ by [18]. The root of unity that it is can be determined by calculating the arguments of each $\sigma_r(u_i)$.

If the root of unity is not one, we will look at other vectors $\mathbf{c}'$ until one is found for which $\prod u_i^{c_i'} = 1$. In practice, an lth root of unity could be eliminated by raising the equation to the lth power. We will not do that here, to avoid dealing with the possibility of losing information when l and $p-1$ have a common divisor.

By the above, if $M > 80rk^2$ and we are given $2r$ units $u_1, \cdots, u_{2r}$ with $\| l(u_i) \| < M$ for $i = 1, \cdots, 2r$, then there is a nontrivial relation $\prod_{i=1}^{2r} u_i^{c_i} = 1$ with each c_i an integer with $|c_i| < 6k(\log\log k)M^2$.

Of course, existence is not enough. For the algorithm, we must find such a nontrivial relation. This can be done using an application of the Lenstra–Lenstra–Lovász (LLL) algorithm due to Babai [2]. For a lattice $\mathcal{L}$, let $\lambda(\mathcal{L})$ be the length of the shortest nonzero vector in $\mathcal{L}$.

THEOREM 4. *Let $b_1, \cdots, b_n$ be vectors in $\mathbb{Z}^n$ with Euclidean length less than N, and let $\mathcal{L}$ denote the lattice generated by $b_1, \cdots, b_n$. We can find a vector $v \in \mathcal{L}$ such that*

$$\| v \| \leq (3/\sqrt{2})^n \lambda(\mathcal{L})$$

in time $O\left(n^{5+\epsilon}(\log N)^{2+\epsilon}\right)$, for any $\epsilon > 0$.

This algorithm will be used to find the dependency of Theorem 3. The time estimate is the same as for the LLL algorithm [12], using fast multiplication.

THEOREM 5. *Suppose that $M > 80rk^2$, and let $u_1, \cdots, u_{2r}$ be units in $\mathcal{O}_K$, with $\| l(u_i) \| < M$ for $i = 1, \cdots, 2r$. A nontrivial relation $\prod_{i=1}^{2r} u_i^{c_i} = 1$ can be found in time $O(r^{5+\epsilon}(\log M)^{2+\epsilon})$, for any $\epsilon > 0$.*

Proof. Let $l_m(x)$ denote $l(x)$ with each coordinate l_i replaced by $\lfloor 2^m l_i \rfloor$, and let $\mathcal{L}_m$ be the lattice generated by $l_m(u_1), \cdots, l_m(u_{2r})$.

For $\mathbf{c} = (c_1, c_2, \cdots, c_{2r})$ as in Theorem 3,

$$\| \sum_{i=1}^{2r} c_i \cdot l_m(u_i) \| = \| \sum_{i=1}^{2r} c_i \cdot (2^m l(u_i) + \epsilon_i) \| = \| \mathbf{0} + \sum_{i=1}^{2r} c_i \cdot \epsilon_i \| < 2r^{3/2} M^2,$$

where each ϵ_i is a vector with all coordinates less than 1 in absolute value. We will show that such vectors $\mathbf{c}$ are short vectors in $\mathcal{L}_m$ and that they are sufficiently shorter than other vectors to guarantee that the algorithm of Theorem 4 will find one.

There is a (highly unlikely) possibility that $\sum_{i=1}^{2r} c_i \cdot l_m(u_i) = \mathbf{0}$ for all choices of $c_1, \cdots, c_{2r}$ in Theorem 3, so that the shortest nonzero vector could be longer than $2r^{3/2}M^2$. If the algorithm ever failed because of this, we could repeat it with a lattice $\mathcal{L}_m'$ where one coordinate l_j is replaced by $\lceil 2^m l_j \rceil$ instead of $\lfloor 2^m l_j \rfloor$. By the Gelfond–Schneider theorem (see, for example, [3]) the lattices are different, since $2^m l_j$ cannot be an integer. Therefore no vector $\mathbf{c}$ that is not a root of unity with $c_j \neq 0$ could be zero in both $\mathcal{L}_m$ and $\mathcal{L}_m'$, and at least one lattice (say $\mathcal{L}_m$) has $\lambda(\mathcal{L}_m) < 2r^{3/2}M^2$.

Any vector $\sum_{i=1}^{2r} c_i \cdot l_m(u_i)$ not corresponding to a relation of the form (6) will have one coordinate at least $\lfloor 2^m/10k^2 \rfloor$ in absolute value, by Lemma 1. Taking $2^m > 20k^2r^2 5^r M^2$, this implies that the vector has length greater than $2r^2 5^r M^2$.

By Theorem 4, we can find a vector in $\mathcal{L}_m$ of length at most $(3/\sqrt{2})^{2r}\lambda(\mathcal{L}_m)$. However, $2r^2 5^r M^2 > (3/\sqrt{2})^{2r}\lambda(\mathcal{L}_m)$, so the vector found must correspond to a relation (6). □

3. Discrete logarithms in $GF(p)$. The algorithm consists of two main parts. The first is finding the discrete logarithms of a factor base of small rational primes, which only must be done once for a given p. The second actually finds the logarithm of an individual $b \in GF(p)$ by finding the logs of a number of "medium-sized" primes and combining these to find the log of b. In addition, for each number field used (one for the precomputation and several for the individual logarithm calculations), the good degree-one prime ideals of small norm in that field must be determined using the method discussed in §2.

We will assume that a, the base for the discrete logarithm, is B-smooth, where B is a bound for the size of primes in the factor base. If a is not smooth, then we may choose a random number that is smooth over the factor base, call it a', and use it as the base for logarithms instead of a. Then find $\log_{a'} a$, and use the identity

$$\log_a b \equiv \log_{a'} b / \log_{a'} a \pmod{p-1}.$$

If a' is not a generator for $GF(p)^*$, then $\log_{a'} a$ and $\log_{a'} b$ may not exist. If this happens, we just choose another value of a' until we find one for which $\log_{a'} a$ exists. Alternatively, we could factor $p-1$ using the number field sieve factoring algorithm and then test if an a' is a generator by checking that $(a')^{(p-1)/q} \not\equiv 1 \pmod{p}$ for each prime q dividing $p-1$. There is no guarantee that a small generator exists, but Shoup [20] has shown that the extended Riemann hypothesis implies that there is a constant c such that for all primes p, $GF(p)^*$ has a generator less than $c\,\omega(p-1)^4(\log(\omega(p-1))+1)^4 \log^2 p$. Here $\omega(n)$ is the number of distinct prime factors of n.

The reason for requiring a to be smooth is to have at least one inhomogeneous relation for the logs of the factor base, using the equation

$$\log_a a = 1 \equiv \sum_{q^t \| a} t \log_a q \pmod{p-1}. \tag{7}$$

3.1. Precomputation. Let p be a prime and a be a primitive element of $GF(p)$. As described in §2.3, choose an integer m and an irreducible monic polynomial $f(x) \in \mathbb{Z}[x]$ such that $(p, \Delta_f) = 1$ and $f(m) \equiv 0 \pmod{p}$. Let $\alpha \in \mathbb{C}$ denote a root of f, $K = \mathbb{Q}(\alpha)$, and $\mathcal{O}_K$ denote the ring of integers in K. Let $\mathfrak{p} = (p, \alpha - m)$, so we have $\mathcal{O}_K/\mathfrak{p} \cong GF(p)$.

The factor base $\mathcal{B}$ will consist of two parts: $\mathcal{B}_{\mathbb{Q}}$ will be rational primes $\le B$, and $\mathcal{B}_K$ will be good prime ideals in $\mathcal{O}_K$ of degree one and norm $\le B$. Let $\mathcal{B}'$ denote the subset of $\mathcal{B}_{\mathbb{Q}}$ consisting of the prime factors of a.

For the precomputation stage, we solve for the logarithms of the rational primes. We will do this by sieving through pairs of small integers c and d. A "hit" will be a coprime pair c, d for which $c + dm$ and $c + d\alpha$ are both smooth over $\mathcal{B}$. These can be searched for efficiently by sieving $c + dm$ and $N(c + d\alpha)$. Suppose that we find a c and d for which both are smooth, say

$$c + dm = \prod_{s \text{ prime}, s \le B} s^{w_s(c,d)} \tag{8}$$

and

$$|N(c + d\alpha)| = \prod_{s \text{ prime}, s \le B} s^{v_s(c,d)}, \tag{9}$$

for $v_s, w_s \in \mathbb{Z}_{\geq 0}$. By Proposition 2, for each s in (9) with $v_s > 0$ there is a unique ideal $\mathfrak{s}$ in $\mathcal{B}_K$ lying over s and dividing $c + d\alpha$. Let $v_{\mathfrak{s}}(c, d) = v_s(c, d)$ for this ideal, and be zero for other ideals in $\mathcal{B}_K$ of norm s. Thus we have

$$c + dm = \prod_{s \in \mathcal{B}_{\mathbb{Q}}} s^{w_s(c,d)} \tag{10}$$

and

$$(c + d\alpha) = \prod_{\mathfrak{s} \in \mathcal{B}_K} \mathfrak{s}^{v_{\mathfrak{s}}(c,d)}. \tag{11}$$

In the Gaussian integers method, where K is a complex quadratic field with class number one, the factorization into ideals in (11) can be rewritten as a product of algebraic integers in $\mathcal{O}_K$ and one of a few (at most six) units. Then the equations can be related using $\varphi(c + d\alpha) \equiv c + dm \pmod{p}$, and, from enough of these equations, a solution can be determined that gives the logs of every element of $\mathcal{B}$. A similar technique will be used for special p in §5. For the number fields K that we are dealing with here, we must use a different method.

We continue sieving through pairs (c, d) until we have collected more than $|\mathcal{B}|$ equations of the form (10) and (11). Then we form a matrix with the w_s's and $v_{\mathfrak{s}}$'s for each equation as its rows and apply Algorithm M to the submatrix of columns corresponding to elements of $\mathcal{B} - \mathcal{B}'$. In this way, we cancel out all those primes to find equations involving only primes in $\mathcal{B}'$ (the resulting equations could be trivial, but we will use the heuristic assumption that they will behave as if they were random equations). We then have a set $\mathcal{S}$ of pairs (c, d) and integers $x(c, d)$ for $(c, d) \in \mathcal{S}$ such that

$$\prod_{(c,d) \in \mathcal{S}} (c + dm)^{x(c,d)}$$

is divisible only by primes in $\mathcal{B}'$, and

$$\prod_{(c,d) \in \mathcal{S}} (c + d\alpha)^{x(c,d)} = U, \tag{12}$$

where U is a unit in $\mathcal{O}_K$.

After gathering $2r$ equations of the form (12), we may find a combination of these that cancels all the units, by Theorem 5. This results in an equation of the following form:

$$\prod_{c,d} (c + d\alpha)^{y(c,d)} = 1, \tag{13}$$

and so

$$\prod_{c,d} (c + dm)^{y(c,d)} \equiv \prod_{c,d} \varphi(c + d\alpha)^{y(c,d)} \equiv 1 \pmod{p}. \tag{14}$$

Using the factorizations in (10), this gives

$$\prod_{s \in \mathcal{B}'} s^{z_s} \equiv 1 \pmod{p}, \tag{15}$$

where $z_s = \sum_{c,d} w_s(c,d)y(c,d)$.

Taking logs, we have that

$$\sum_{s\in\mathcal{B}'} z_s \log_a s \equiv 0 \pmod{p-1}. \tag{16}$$

Once we have more than $|\mathcal{B}'|$ such equations, we can attempt to solve these homogeneous equations together with (7) and obtain the logs of every prime in $\mathcal{B}'$, using Gaussian elimination modulo $p-1$. If the matrix does not determine a unique solution, we may collect more equations until it does. Since $|\mathcal{B}'| < \log p$, the fact that we must have $|\mathcal{B}'|$ runs of Algorithm M will not affect the complexity analysis.

3.2. Finding individual logarithms. To compute the logarithm of b, we first convert the problem into finding logarithms of "medium-sized" primes. This is done by choosing random integers $l \in [1, p-1]$ until we find one for which

$$a^l b \equiv q_1 q_2 \cdots q_t \pmod{p}, \tag{17}$$

where each of the q_i are moderately sized (say $\leq p^{1/k}$). Then, by finding the discrete logarithms of each q_i, we will obtain the discrete logarithm of b.

For each i, take $m_i = q_i h_i$, where h_i is a number smooth over $\mathcal{B}$ chosen so that m_i is close to $p^{1/k}$. Let $f_i(x)$ be a monic polynomial of degree k such that $f_i(m_i) \equiv 0 \pmod{p}$ and define

$$f_{i,j}(x) = f_i(x) + j(m_i - x).$$

Then $f_{i,j}(m_i) \equiv 0 \pmod{p}$, and, if $f_{i,j}(x)$ is irreducible over $\mathbb{Q}$ and $\alpha_{i,j}$ is a root of $f_{i,j}(x)$, then in $\mathbb{Q}(\alpha_{i,j})$, $|N(\alpha_{i,j})| = |f_{i,j}(0)|$. We sieve through values of j to find ones for which $f_{i,j}(0)$ is B-smooth and continue until we find one with $f_{i,j}(x)$ irreducible, and $(pf_{i,j}(0), \Delta_{f_{i,j}}) = 1$. We will use this polynomial to find the logarithm of q_i.

Once a suitable value of j has been found, the factorization of α_i $(= \alpha_{i,j})$ in $K_i = \mathbb{Q}(\alpha_i)$ gives us the following equations:

$$m_i = q_i h_i \equiv \varphi(\alpha_i) \pmod{p} \tag{18}$$

and

$$(\alpha_i) = \prod_{\mathfrak{s}\in\mathcal{B}_{K_i}} \mathfrak{s}^{u_{\mathfrak{s}}}. \tag{19}$$

As in the precomputation stage, we will sieve through small c and d until we collect enough equations of the form (10) and (11) to cancel factors not in $\mathcal{B}'$ and obtain

$$q_i h_i \prod_{c,d} (c + dm_i)^{t(c,d)} \equiv \varphi(\alpha_i) \prod_{c,d} \varphi(c + d\alpha_i)^{t(c,d)} \equiv 1 \pmod{p}, \tag{20}$$

where the left product is divisible only by q_i and primes in $\mathcal{B}'$. Note that we only need one such equation, since the logs of primes in $\mathcal{B}'$ are known from the precomputation.

Thus we have

$$q_i \equiv \prod_{s\in\mathcal{B}'} s^{z'_s} \pmod{p},$$

and so

$$\log_a q_i \equiv \sum_{s \in \mathcal{B}'} z'_s \log_a s \pmod{p-1}. \tag{21}$$

We do this procedure once for each q_i and combine their logarithms to find $\log_a b$. The sieving and cancellation in this stage is the same as in the precomputation. The only difference is that we must keep (18) and (19) and find other equations with rank sufficient to cancel out the factors in those equations and the units that arise. It is a reasonable heuristic assumption that the equations will have full rank, and most discrete logarithm algorithms involve a similar assumption. An exception is the rigorous algorithm of Pomerance in [16], but we have no version of his Lemma 4.1 that works in this setting.

4. Runtime analysis. We will choose two parameters to optimize the performance: the size of B will be $L_p[1/3;\delta]$ and the size of m will be $L_p[2/3;\gamma]$, with δ and γ to be chosen later.

For the precomputation, take

$$k = \left\lceil \frac{1}{\gamma} \left(\frac{\log p}{\log\log p} \right)^{1/3} \right\rceil.$$

Then choose $m \in \mathbb{Z}$ less than $p^{1/k}$ and f irreducible of degree k as described earlier. Let α be a root of f, and $K = \mathbb{Q}(\alpha)$.

We will search through pairs of integers c, d that are relatively prime and at most $L_p[1/3;\lambda]$ in absolute value. There are thus $L_p[1/3, 2\lambda]$ pairs. We have

$$|c + dm| \le L_p[2/3;\gamma] \quad \text{and} \quad |N(c + d\alpha)| \le L_p[2/3;\gamma + \lambda/\gamma]$$

by (6).

Using the heuristic assumptions of §2.1, we expect to obtain enough hits to solve for the logs of $\mathcal{B}'$ after

$$L_p[1/3; \frac{\gamma}{3\delta} + \frac{\gamma + \lambda/\gamma}{3\delta} + \delta]$$

trials. Letting this equal $L_p[1/3;2\lambda]$, we obtain

$$\lambda = \frac{2\gamma^2 + 3\delta^2\gamma}{6\delta\gamma - 1}. \tag{22}$$

The time necessary to sieve through all these values is $L_p[1/3;2\lambda]$. Each use of Algorithm M to solve the matrix equations takes time $L_p[1/3;3\delta]$, taking $T = L_p[1/3;\delta]$ and $E = O(\log p)$. To cancel the units as described in §2.3 takes time $L_p[1/3;2\delta]$. This follows from Theorem 5, taking $M = \exp(L_p[1/3;\delta])$.

This is done $|\mathcal{B}'| < \log p$ times, so the total time is still $L_p[1/3;3\delta]$. Altogether, the precomputation takes time $L_p[1/3;3\delta]$.

To calculate the discrete log of a particular $b \in GF(p)$, we choose a random $l \in [1, p-1]$ and see if $a^l b \bmod p$ is $L_p[2/3;\gamma]$-smooth. Assuming Conjecture 2, the ECM can detect such smooth numbers with probability $1 - o(1)$ in time $L_p[1/3; 2\sqrt{\gamma/3}]$. If no factorization is found after that amount of time, another value of l can be tried. We expect to find an l for which $a^l \bmod p$ is smooth after $L_p[1/3;1/(3\gamma)]$ trials, by Theorem 1.

Once such a value has been found, we have $a^l b \equiv q_1 q_2 \cdots q_t \pmod{p}$, and it suffices to find the discrete logarithm of each q_i.

Then we choose $m_i = q_i h_i$ of size $L_p[2/3; \gamma]$ for each q_i, and find an irreducible monic polynomial f of degree k for which $f(m_i) \equiv 0 \pmod{p}$ and $f_i(0)$ is B-smooth. The constant term of f is $L_p[2/3; \gamma]$, so finding a smooth value should take time $L_p[1/3; \gamma/(3\delta)]$.

The next step is to collect equations as in the precomputation. The parameters are the same, and so the time will be the same, unlike most discrete logarithm algorithms, for which the precomputation takes more time than finding individual logarithms.

The total time is $L_p[1/3; M]$, where

$$M = \max\left\{2\lambda, 3\delta, \frac{1}{3\gamma} + 2\sqrt{\frac{\gamma}{3}}, \frac{\gamma}{3\delta}\right\}.$$

By choosing $\gamma = (\frac{3}{8})^{1/3}$, $\delta = 3^{-1/3}$, and $\lambda = (\frac{9}{8})^{1/3}$, we note that (22) is satisfied, and we achieve an optimal time of $L_p[1/3; 3^{2/3}]$.

5. Discrete logs for special p. As with the number field sieve factoring algorithm, it is possible to modify the discrete logarithm algorithm for numbers of a special form. The method we present here is a generalization of the Gaussian integer method to higher-degree fields. While asymptotically slower than the method of §3, it avoids the use of Algorithm M and so is more practical for numbers of a reasonable size.

In [15] McCurley offers \$100 for breaking a Diffie–Hellman scheme (which is no harder than, and may be equivalent to, finding discrete logarithms) with the prime $p = 2 \cdot 739 \cdot q + 1$, where $q = (7^{149} - 1)/6$. For this number, the scheme given below would be faster than the method of §3, although, since p has 128 digits, even this method would require an exorbitant amount of computer time.

Let

$$k = \left\lceil \frac{1}{\gamma} \left(\frac{\log p}{\log \log p} \right)^{1/5} \right\rceil$$

for some $\gamma > 0$ to be chosen later. The special method will apply to primes p for which there exists an irreducible monic polynomial f of degree k and integer m near $p^{1/k}$ for which $f(m) \equiv 0 \pmod{p}$, and all the coefficients of f are small. "Small" is a flexible term, but can be taken to mean that the resulting field $K = \mathbb{Q}(\alpha)$ for α a root of f has small enough discriminant that the class group and unit group can be dealt with.

For instance, if $r^e - s \equiv 0 \pmod{p}$, for a small positive integer r and a nonzero integer s of small absolute value, let l be the smallest integer for which $kl > e$. Then $r^{kl} \equiv s r^{kl-e} \pmod{p}$, and so if we pick $m = r^l$ and $f(x) = x^k - s r^{kl-e}$, we have $f(m) \equiv 0 \pmod{p}$.

For the number q, above, we could take $k = 6$, $m = 7^{25}$, and $f(x) = x^6 - 7$. The number p is more difficult; with the same k and m, we would need to take $f(x) = 739x^6 - 5152$. Using a nonmonic polynomial would not cause major difficulties, but the larger coefficients would increase the difficulty of operations in $\mathcal{O}_K$ and reduce the hit rate for the sieving.

Let α be a root of f, and $K = \mathbb{Q}(\alpha)$. For simplicity, we will assume that $\mathcal{O}_K = \mathbb{Z}[\alpha]$ is a unique factorization domain.

Choose $B = L_p[2/5; \delta]$, where $\delta > 0$ is another parameter to be chosen later. Our factor base $\mathcal{B}$ will consist of rational primes $< B$ ($\mathcal{B}_{\mathbb{Q}}$), first-degree primes (algebraic

integers, not ideals) in $\mathcal{O}_K$ with norm less than B and a fundamental set of units in $\mathcal{O}_K$ ($\mathcal{B}_K$). We will be dealing explicitly with the ideals and the units in K, and so it is necessary to calculate generators for the unit group and the ideals in $\mathcal{B}_K$. This may be done as in [13], by searching elements of the form $\sum_{i=0}^{k-1} a_i\alpha^i$, with a_i's of small absolute value, for ones of small norm, and combining these to obtain the necessary units and generators of the ideals.

The base for logarithms for algebraic numbers is not important; it may be a small prime that generates $(O_K/\mathfrak{p})^*$, for $\mathfrak{p}$ a prime ideal of norm p, or an algebraic number ρ with $a \equiv \varphi(\rho) \pmod{p}$.

The precomputation step will determine the discrete logs of the whole factor base, not just a subset of the rational part. As before, sieve through c and d less than $L_p[2/5; \lambda]$, looking for values with $c + dm$ and $N(c + d\alpha)$ both smooth. We have

$$c + dm = L_p[4/5; \gamma], \quad \text{and} \quad N(c + d\alpha) = L_p[3/5; \lambda/\gamma] = L_p[4/5; 0].$$

Therefore the probability of both being B-smooth is $L_p[2/5; -2\gamma/(5\delta)]$. Obtaining $L_p[2/5; \delta]$ hits will take expected time

$$L_p[2/5; 2\gamma/(5\delta) + \delta],$$

with $\lambda = \gamma/(5\delta) + \delta/2$.

Each hit gives us an equation involving logarithms of the factor base. Once we have more than $|\mathcal{B}| = L_p[2/5; \delta]$ hits, we solve the resulting matrix equation over $\mathbb{Z}/(p-1)\mathbb{Z}$ using Wiedemann's algorithm in time $L_p[2/5; 2\delta]$. Heuristically, we expect there to be a unique solution, which will give the logarithms of the factor base.

To find an individual logarithm, we again reduce the problem to finding the logs of medium-sized primes q_i by looking for $a^\ell b \pmod{p}$ smooth. Now it will be advantageous to take the q_i's much smaller than m, say of size $L_p[3/5; \theta]$. Assuming Conjecture 2, if $a^\ell b$ is this smooth, we expect the ECM to factor it with probability $1 - o(1)$ in time $L_p[3/10; \sqrt{6\theta/5}]$. We expect a smooth number to occur in about $L_p[2/5; 2/(5\theta)]$ trials, so the total time is $L_p[2/5; 2/(5\theta)]$.

For each q_i, we will sieve c and d for which $q_i | (c + dm)$, say fixing d and taking $c = c_0 + eq_i$, to find one value for which $(c + dm)/q_i$ and $N(c + d\alpha)$ are both B-smooth. Once this happens we are done, since, from the precomputation, we know the logs of the whole factor base.

We cannot change m as in the general method, since this would result in a field with large discriminant. Therefore at least one of c and d must be about as big as q_i, so $(c + dm)/q_i = L_p[4/5; \gamma]$, and $N(c + d\alpha) = L_p[4/5; \theta/\gamma]$. (Note that, for the general number field sieve method, $N(c + d\alpha)$ would be $L_p[1; 1]$, which is why multiple fields were needed.) The expected time to find both B- smooth is therefore

$$L_p\left[2/5; \frac{2(\gamma + \theta/\gamma)}{5\delta}\right].$$

Thus the time for the precomputation is $L_p[2/5; \mu]$, where

$$\mu = \max\left\{\frac{2\gamma}{5\delta} + \delta, 2\delta\right\}, \tag{23}$$

and the time for finding individual logarithms is $L_p[2/5; \nu]$, where

$$\nu = \max\left\{\frac{2}{5\theta}, \frac{2(\gamma + \theta/\gamma)}{5\delta}\right\}. \tag{24}$$

Since θ does not occur in the precomputation, we may choose it to make the two terms in (24) equal, as follows:

$$\theta = \frac{-\gamma^2 + \sqrt{\gamma^4 + 4\delta\gamma}}{2}.$$

The choices for γ and δ depend on how time is to be divided between the two stages. Enlarging δ reduces the time needed to find individual logarithms, but at the cost of increasing the precomputation time. If the times are to be equal (say if only one logarithm is desired for a given p), then the optimal values are

$$\gamma = 10^{-1/5} \quad \text{and} \quad \delta = \left(\frac{4}{125}\right)^{1/5},$$

giving a time of $L_p[2/5; \mu] = L_p[2/5; \nu]$, where

$$\mu = \nu = \left(\frac{128}{125}\right)^{1/5} \approx 1.00475.$$

If many instances are to be done for one p, more time could be spent on the precomputation. For $\mu \geq (128/125)^{1/5}$, if we spend $L_p[2/5; \mu]$ time on the precomputation, each logarithm can be found in time

$$L_p\left[2/5; \left(\frac{128}{125\mu^2}\right)^{1/3}\right].$$

For any $c \geq 1$, the Gaussian integer method can find logarithms in time $L_p[1/2; 1/(2c)]$ if $L_p[1/2; c]$ is spent on the precomputation. Where the above method becomes faster than the Gaussian integer method depends largely on the $o(1)$ terms and the choice of f, but for a good f it is well under 100 digits. More research is needed to say for which size primes and polynomials the special number field sieve algorithm is a practical improvement.

The general number field sieve algorithm is definitely not practical for any reasonable numbers. The crossover point for $L_p[1/2; 1]$ and $L_p[1/3; 3^{2/3}]$ (the times for the Gaussian integer method and the general number field sieve) is 218 digits. The crossover point for $L_p[2/5; 1.00475]$ and $L_p[1/3; 3^{2/3}]$ (the times for the special and general number field sieves) is above 320,000 digits.

If $\mathcal{O}_K$ has class number $h > 1$, then we must cancel the nonprincipal ideals that occur in (11). If we have calculated h, then the algorithm may proceed as in §3, with Algorithm M replaced by Wiedemann's algorithm modulo h, to obtain an equation involving only principal ideals.

Finally, it should be noted that the special number field sieve can also be applied to primes that are values of homogeneous forms in two variables, as well as polynomials. Let f be a polynomial of degree k, and X and Y be integers near $p^{1/k}$, such that

$$Y^k f(X/Y) = X^k + a_{k-1}X^{k-1}Y + \cdots + a_0 Y^k \equiv 0 \pmod{p}.$$

Then the above method may still be used, with the homomorphism $\varphi(c + d\alpha) = c + dX/Y$. Then the sieving phase searches for values of c and d for which $c + d\alpha$ and $cY + dX$ are both smooth. The analysis is the same as given above.

6. Recent developments. The general number field sieve algorithm is still impractical for large numbers, largely because of the need for Gaussian elimination over $\mathbb{Q}$. Methods to avoid this problem have been suggested by Adleman [1] for number field sieve factoring and by Schirokauer [19] for discrete logarithms over $GF(p)$. Coppersmith very recently has suggested using multiple fields to factor n in time $L_n[1/3; c]$ with $c \approx 1.902$, an improvement over $c \approx 2.08$ for the original algorithm of Buhler, Lenstra, and Pomerance, and $c \approx 1.92$ for the methods of Lenstra and Adleman. The resulting algorithms, while faster, are still impractical for numbers within reach of modern computers. Use of the number field sieve in number-theoretic algorithms is a rapidly-developing area. These developments, and the improvements of the constants above, are likely to continue.

The practicality of the special number field sieve is of interest for discrete log-based cryptosystems. By choosing a prime p with a good f and m (as in §5) as the base for such a system, its security would be weakened. A person with knowledge of f might be able to use it as a "trapdoor" to break the system. More study is needed to say how much of an advantage this would actually be.

Acknowledgments. The author thanks Carl Pomerance for allowing the presentation of his Algorithm M here and for many suggestions that greatly improved the form and content of this paper. Thanks also to Andrew Odlyzko for several e-mail discussions about discrete logarithms, and to Hendrik Lenstra for helpful comments.

REFERENCES

[1] L. M. ADLEMAN, *Factoring numbers using singular integers*, in Proc. 23rd ACM Symposium on Theory of Computing, New Orleans, LA, 1991, pp. 64–71.

[2] L. BABAI, *On Lovász's lattice reduction and the nearest lattice point problem*, in Proc. 2nd Annual Symposium on the Theoretical Aspects of Computing, Paris, France, K. Mehlhorn, ed., Springer, Berlin, pp. 13–20.

[3] A. BAKER, *Transcendental Number Theory*, Cambridge University Press, Cambridge, UK, 1975.

[4] J. BRILLHART, M. FILASETA, AND A. ODYLZKO, *On an irreducibility theorem of A. Cohn*, Canad. J. Math, 33 (1981), pp. 1055–1059.

[5] J. BUHLER, H. W. LENSTRA, JR., AND C. POMERANCE, *Factoring integers with the number field sieve*, preprint.

[6] E. R. CANFIELD, P. ERDŐS, AND C. POMERANCE, *On a problem of Oppenheim concerning "Factorisatio Numerorum,"* J. Number Theory, 17 (1983), pp. 1–28.

[7] D. COPPERSMITH, *Modifications to the number field sieve*, J. Cryptology, to appear.

[8] D. COPPERSMITH, A. M. ODLYZKO, AND R. SCHROEPPEL, *Discrete logarithms in GF(p)*, Algorithmica, 1 (1986), pp. 1–15.

[9] E. DOBROWOLSKI, *On the maximal modulus of conjugates of an algebraic integer*, Bull. Acad. Polon. Sci. Ser. Sci. Math. Astronom. Phys., 26 (1978), pp. 291–292.

[10] E. KALTOFEN AND B. D. SAUNDERS, *On Wiedemann's method for solving sparse linear systems*, Proceedings AAECC-5 SLNCS 536 (1991), pp. 29–38.

[11] B. LAMACCHIA AND A. M. ODLYZKO, *Computation of discrete logarithms in prime fields*, Designs, Codes and Cryptography, 1 (1991), pp. 47–62.

[12] A. K. LENSTRA, H. W. LENSTRA, JR., AND L. LOVÁSZ, *Factoring polynomials with rational coefficients*, Math. Ann., 261 (1982), pp. 515–534.

[13] A. K. LENSTRA, H. W. LENSTRA, JR., M. S. MANASSE, AND J. M. POLLARD, *The number field sieve*, in Proc. 22nd ACM Symposium on Theory of Computing, Baltimore, MD, 1990, pp. 564–572.

[14] H. W. LENSTRA, JR., *Factoring integers with elliptic curves*, Ann. Math., 126 (1987), pp. 649–673.

[15] K. MCCURLEY, *The discrete logarithm problem*, in Cryptology and Computational Number Theory, Proceedings of Symposia in Applied Mathematics, American Mathematical Society, Providence, RI, 1990.

[16] C. POMERANCE, *Fast, rigorous factorization and discrete logarithm algorithms*, in Discrete Algorithms and Complexity, D. S. Johnson et al., eds., Academic Press, Orlando, 1987, pp. 119–143.
[17] ———, personal communication, 1990.
[18] J. B. ROSSER AND L. SCHOENFELD, *Approximate formulas for some functions of prime numbers*, Illinois J. Math., 6 (1962), pp. 64–94.
[19] O. SCHIROKAUER, *On Pro-Finite Groups and on Discrete Logarithms*, Ph.D. thesis, Univ. of California, Berkeley, CA, May 1992.
[20] V. SHOUP, *Searching for primitive roots in finite fields*, Math. Comput., 58 (1992), pp. 369–380.
[21] D. H. WIEDEMANN, *Solving sparse linear equations over finite fields*, IEEE Trans. Inform. Theory, 32 (1986), pp. 54–62.
[22] H. ZANTEMA, *Class numbers and units*, in Computational Methods in Number Theory, Vol. II, H. W. Lenstra, Jr. and R. Tijdeman, eds., Mathematisch Centrum, Amsterdam, 1982, pp. 213–234.