A general framework for subexponential discrete logarithm algorithms in groups of unknown order

Andreas Enge
Lehrstuhl für Diskrete Mathematik,
Optimierung und Operations Research
Universität Augsburg
86135 Augsburg
Deutschland
enge@math.uni–augsburg.de

September 6, 2000

Abstract

We develop a generic framework for the computation of logarithms in finite class groups. The model allows to *formulate* a probabilistic algorithm based on collecting relations in an abstract way independently of the specific type of group to which it is applied, and to *prove* a subexponential running time if a certain smoothness assumption is verified. The algorithm proceeds in two steps: First, it determines the abstract group structure as a product of cyclic groups; second, it computes an explicit isomorphism, which can be used to extract discrete logarithms.

1 Motivation

For an additively written group G, an element $\alpha \in G$ and a second element $\beta \in \langle \alpha \rangle$, the *discrete logarithm problem* consists of determining an integer l, denoted by $\log_\alpha \beta$, such that $\beta = l\alpha$. The problem has increasingly received attention since it plays an important role in cryptography. Following Diffie and Hellman's and ElGamal's constructions ([DH76, ElG85]), it is possible to base a public key cryptosystem on a finite abelian group in a way that computing logarithms in the group amounts to breaking the system. It is widely believed (and in some instances proved) that conversely, if the discrete logarithm problem in the group is hard, then the cryptosystem so constructed is indeed secure. In a generic setting, in which no specific structural properties of the group representation can be exploited, the discrete logarithm problem is of exponential complexity ([Nec94, Sho97]). However, in multiplicative groups of finite fields, which were first suggested for cryptography, it can be solved by subexponential algorithms. As the name indicates, the growth of their running times is — in a sense which will be made precise in Section 3.4 — between polynomial and exponential, so that relatively large keys are needed to achieve a reasonable level of security.

To avoid subexponential attacks, different finite class groups have been proposed for cryptographic use; among them are elliptic curves ([Mil86, Kob87]), which constitute the state of the art of public key cryptography and which inherit the class group structure from the isomorphy with their Jacobians, Jacobians of hyperelliptic ([Kob89]) and more general ([GPS98, Ari99]) curves over finite fields, which are commonly seen as a possible replacement for elliptic curves in case the latter turn out to be insecure, and ideal class groups of number fields ([BW88]). Over the years, various algorithms have been developed which show that the discrete logarithm problem in some of these class groups can be solved in subexponential time, while others apparently resist this kind of attack. Precisely, the subexponential complexity has been proved for ideal class groups of imaginary quadratic number fields ([McC89, HM89]) and Jacobians of large genus hyperelliptic curves over finite fields ([Eng99]).

If nothing is known about the class group except its definition and arithmetic, then these algorithms all proceed in two stages: In the first step, the structure of the group as a direct product of cyclic groups of known orders is determined. In the second step, an explicit isomorphism is used to transfer the logarithm problem into the cyclic groups, in which it can be solved trivially by the extended Euclidean algorithm.

If the order of the base element α is known, then the first phase can be omitted, which results in a much faster algorithm. It is observed in [EG00] that all groups to which this kind of algorithm applies share a common structure, so that it is possible to *formulate* once and for all a generic algorithm for computing discrete logarithms in class groups of known order. Its subexponential running time can also be *proved* in general provided that the class group satisfies a certain smoothness assumption, which has to be verified separately for each type of group.

In this paper, we apply and extend the general framework of [EG00] to formulate a generic subexponential discrete logarithm algorithm for class groups of unknown order. It solves both computational tasks of determining the abstract group structure and of providing an explicit isomorphism of the group with its abstract model. For imaginary quadratic number fields, we recover the algorithm of [McC89, HM89], for hyperelliptic Jacobians that of [Eng99]. The subexponential complexity is again proved in complete generality under the same smoothness assumption; to show that logarithms in a further type of class groups can be computed in subexponential time, it is thus sufficient to verify this smoothness assumption without going over the details of the algorithm again.

2 The generic model

The setting in which the following algorithm can be applied is that of an *arithmetic formation*, introduced in [Kno75]. The basic idea is to have a structure in which each element admits non-unique decompositions into sums of primes, which allows to deduce a relation of logarithms

$$\sum (e_i - r_i) \log_\alpha \mathfrak{P}_i = 0$$

from a relation of elements of the form

$$\sum e_i \mathfrak{P}_i = \sum r_i \mathfrak{P}_i.$$

(Strictly speaking, this only holds if G is cyclic and generated by α.) In [EG00], further computational assumptions are introduced, and we need some more in our situation where the group order is unknown.

Let $\mathbb{M}$ be the free abelian monoid over some countable (usually infinite) set of *primes* $\mathcal{P}$, and let

$$\deg : \mathbb{M} \to \mathbb{R}^+$$

be a homomorphism of monoids, which to each element of $\mathbb{M}$ associates its *size*. We assume that all primes have a size of at least 1, so that an element of size d is decomposed into a sum of at most $\lfloor d \rfloor$ primes, counting multiplicities. Let $\sim$ be an equivalence relation on $\mathbb{M}$, compatible with addition. If $G = \mathbb{M}/\sim$ is a finite group, then it is clearly abelian, and is called an *arithmetical formation*. We assume that the discrete logarithm problem has to be solved in such a formation G, and denote by N the order of G. Since N is unknown, let N' be such that the input size of the problem is in $O(\log N')$. (It is conceivable that $N \ll N'$, cf. [EG00]). As polynomial time effort may be neglected compared to subexponential one, we measure all complexities up to polynomials in $\log N'$, i.e. in $O^\sim$-notation, where for a positive function f of N' we have

$$O^\sim(f) = \bigcup_{k=0}^{\infty} O(f(N') \log^k N');$$

thus, a (time or space) bound is polynomial if and only if it is contained in $O^\sim(1)$.

We suppose that each element of G is given by a canonic representative in $\mathbb{M}$ of polynomial bit size and denote by $\deg g$ the size of this representative. Let

$$d \geq \lfloor \max\{\deg g : g \in G\} \rfloor,$$

so that the maximum number of primes in a decomposition of a group element is bounded above by d, even counting multiplicities. Then we require that $d \in O^\sim(1)$ and that the arithmetic of G is realised by manipulating the canonic representatives in polynomial time.

For some *smoothness bound* S let the *factor base* $\mathcal{P}_S = \{\mathfrak{P}_1, \ldots, \mathfrak{P}_{n_S}\}$ be the set of primes of size at most S, and call an element of the group S*–smooth* if its canonic representative is a sum of primes in $\mathcal{P}_S$. Since a distinction into smooth and non-smooth elements arises only for $S \in O^\sim(1)$, we assume this case; on the other hand, S must be sufficiently large so that $\mathcal{P}_S$ generates G, i.e. for any $g \in G$ there are non-negative integers e_i such that

$$g \sim \sum_{i=1}^{n_S} e_i \mathfrak{P}_i.$$

(The generating property was implicit in the setting of cyclic groups treated in [EG00].) Let $E \geq 5\max\{N, N'\} + d$, so that in fact all e_i in the relation above can be assumed to be smaller than E. On the other hand, we demand $\log E \in O^\sim(1)$, so that scalar multiples $e\mathfrak{P}$ for $e < E$ and $\mathfrak{P} \in \mathcal{P}_S$ can be computed in polynomial time by a double-and-add algorithm. It is also assumed that $\log n_S \in O^\sim(1)$, which is fulfilled for n_S subexponential in $\log N'$.

We suppose that the factor base can be constructed in $O^\sim(n_S^2)$ and that a smoothness test decides in $O^\sim(n_S)$ whether a given group element is S–smooth and if so, determines the decomposition over $\mathcal{P}_S$ in the same time bound. (In a generic setting, one would expect that $\mathcal{P}_S$ can be constructed by enumerating all elements of size at most S and trial division by all smaller elements. Likewise, the smoothness test might use trial division. In practice, both phases are usually faster, but this is compensated by other parts of the algorithm.)

Examples.

1) Ideal class groups of imaginary quadratic number fields

   Let $K = \mathbb{Q}(\sqrt{D})$ be a quadratic number field of discriminant $D < 0$ and $\mathcal{O} = \mathbb{Z}\left[\frac{D+\sqrt{D}}{2}\right]$ its ring of integers. The ideal class group G of K is defined as $\mathbb{M}/\sim$, where $\mathbb{M}$ is the set of integral ideals of $\mathcal{O}$, the free abelian monoid over the set $\mathcal{P}$ of prime ideals, and $\sim$ is induced by the principal ideals. The size function is given by the dual logarithm of the norm, $\deg \mathfrak{P} = \mathrm{ld}(\mathrm{N}_{K/\mathbb{Q}}(\mathfrak{P}))$. Let $N' = |D|$; by a theorem of Siegel's, $\log N' \in (2 + o(1)) \log N$ ([Sie36]); the reduction theory of binary quadratic forms implies that $d = \mathrm{ld}\left(\sqrt{\frac{N'}{3}}\right)$ is a suitable choice, and with $E = 5N' + d$ all computational requirements are met (for details, see [EG00]).

   Assuming the validity of the generalised Riemann hypothesis, by Corollary 6.2 of [Sch82] the class group is generated by $\mathcal{P}_S$ for $n_s \in O(\log^2 N')$, i.e. $S \in O(\log\log N')$, with effectively computable constants. We will assume n_S of subexponential instead of polynomial size, so that the generating property will be fulfilled.

2) Jacobians of hyperelliptic curves over finite fields

   Let $K = \mathbb{F}_q$ be the finite field with q elements. We consider hyperelliptic curves over K for which the prime divisor at infinity is ramified; denoting their genus by $g \geq 1$, these are smooth projective models of affine curves of the form

   $$H = Y^2 + vY - u$$

   with u, $v \in K[X]$, u monic of degree $2g+1$ and $v = 0$ if $\operatorname{char} K$ is odd resp. v of degree at most g if $\operatorname{char} K = 2$. The K–rational part G of the Jacobian of H is isomorphic to the ideal class group of $\mathcal{O} = K[X,Y]/(H)$, i.e., $G = \mathbb{M}/\sim$, where $\mathbb{M}$ is the set of integral ideals of $\mathcal{O}$ and $\sim$ is induced by the principal ideals. Interpreting the norm over $K(X)$ of an ideal of $\mathcal{O}$ as a polynomial in $K[X]$, the size of an ideal is given by the degree of its norm. We let $N' = q^g$, which by Weil's theorem is somehow close to N. By reduction theory, $d = g$, and a theorem of Artin's ([Art24], § 24, Formula (8)) implies that $E = 5(2g+1)q^g + g$ is a suitable choice so that all computational assumptions are fulfilled; for details, see [EG00]. Choosing a factor base with $S \geq \lceil 2\log_q(4g-2) \rceil$, the generating property also holds, see [Eng99].

Clearly, Example 2 can be extended to Jacobians of more general curves over finite fields. For instance, the generating property for $S \geq \lceil 2\log_q(4g-2) \rceil$ has been proved for arbitrary curves in [Eng00], Theorem 6.2. Since to date the arithmetic has been worked out for only few further classes of curves ([GPS98, Ari99]) and the smoothness result needed for proving the subexponential running time has been verified only for hyperelliptic curves, we renounce at presenting details.

3 The algorithm

As explained in Section 1, the algorithm proceeds in two stages. The first phase determines the group structure as an abstract product of cyclic groups of known order (with, as a byproduct, the group order). The second phase is in fact an algorithm for computing the image of any given group element under the isomorphism. Both phases are probabilistic. If a bound Θ on the group order N such that $\Theta/2 < N \leq \Theta$ is known, then they are of Las Vegas type, i.e. always output the correct answer. If only the input data α, $\beta \in G$ of the discrete logarithm problem are known, then the first phase may terminate with a wrong group structure. In case this structure is compatible with the discrete logarithm problem nevertheless, the second phase halts with the correct logarithm; otherwise, it reports failure, and the algorithm has to be run again. So for computing discrete logarithms, the algorithm is always of Las Vegas type, for computing the group order and structure it can only be relied upon if the additional information Θ is available. Notice that Θ can be computed in polynomial time for imaginary quadratic number fields by evaluating a truncated Euler product of the analytic class number formula ([McC89], p. 468; [Sch82], Theorem 6.3). A similar approach should also be successful for hyperelliptic Jacobians.

The algorithm is based on the observation that the group homomorphism

$$\mathbb{Z}^{n_S} \to G, \quad (e_1, \ldots, e_{n_S}) \mapsto \sum_{i=1}^{n_S} e_i \mathfrak{P}_i$$

is surjective because $\mathcal{P}_S$ is a generating set. Thus, its kernel Γ is a full dimensional lattice of determinant n_S, and $G \simeq \mathbb{Z}^{n_S}/\Gamma$. We call the elements of Γ *relations*.

3.1 The group structure

The first phase of the algorithm consists of finding a basis for Γ and bringing it into a special form.

Algorithm 1

1) Choose a smoothness bound S and construct the factor base $\mathcal{P}_S = \{\mathfrak{P}_1, \ldots, \mathfrak{P}_n\}$ with $n = n_S$.

2) Find $14n\left(1 + \operatorname{ld} E + \frac{1}{2}\operatorname{ld} n\right)$ relations by repeatedly selecting random vectors $e = (e_1, \ldots, e_n)^T \in \{0, \ldots, E-1\}^n$ until the canonic representative of $\sum_{i=1}^n e_i \mathfrak{P}_i$ is S–smooth and decomposes as $\sum_{i=1}^n r_i \mathfrak{P}_i$, so that $r - e \in \Gamma$. Write these vectors as columns into a matrix A.

3) Compute the rank of A. If it is less than n, return to Step 2).

4) There are a matrix $\Delta = \operatorname{diag}(d_1, \ldots, d_n)$ with $d_i \in \mathbb{N}$, $1 \leq d_1 | \cdots | d_n$ and unimodular integral square matrices S and T of the appropriate dimensions such that $SAT = (\Delta | 0)$; Δ is called the Smith normal form *or* elementary divisor form *of A. Compute Δ and S.*

5) If an approximation $\Theta/2 < N \leq \Theta$ is known and $d_1 \cdots d_n > \Theta$, then return to Step 2). If $d_1 \cdots d_n \leq \Theta$, then G is proven to be of the type $\mathbb{Z}/d_1\mathbb{Z} \times \cdots \times \mathbb{Z}/d_n\mathbb{Z}$. Otherwise, this is very likely to be the correct group structure (see Section 3.3).

The correctness of Step 5) follows from the observations that $\operatorname{Im} A \subseteq \Gamma$, whence $\det \Gamma = N$ divides $d_1 \cdots d_n = |\det A|$, and that $\operatorname{Im} A = \Gamma$ if and only if $|\det A| = N$.

3.2 The isomorphism and discrete logarithms

Assume that Algorithm 1 has determined the correct group structure. Then the following algorithm computes $\varphi(\gamma)$ for $\gamma \in G$, where

$$\varphi : G \to \mathbb{Z}/d_1\mathbb{Z} \times \cdots \times \mathbb{Z}/d_n\mathbb{Z}$$

is an isomorphism. It expresses γ over the factor base by an approach similar to the relation collection phase of Algorithm 1 and exploits the knowledge of the matrix S.

Algorithm 2

1) Select random vectors $e = (e_1, \ldots, e_n)^T \in \{0, \ldots, E-1\}^n$ until the canonic representative of $\gamma + \sum_{i=1}^n e_i \mathfrak{P}_i$ is S–smooth and decomposes as $\sum_{i=1}^n r_i \mathfrak{P}_i$. Let $c = r - e$; then $\gamma = \sum_{i=1}^n c_i \mathfrak{P}_i$ in G.

2) Let $\varphi(\gamma) = Sc \bmod \Delta$, where by $(v_1, \ldots, v_n)^T \bmod \Delta$ we understand $(v_1 + d_1\mathbb{Z}, \ldots, v_n + d_n\mathbb{Z})^T$.

Notice that φ is well-defined: If $\gamma = \sum c_i \mathfrak{P}_i = \sum c_i' \mathfrak{P}_i$, then $c - c' \in \Gamma = \operatorname{Im} A$, so that $S(c - c') \in \operatorname{Im} \Delta$, i.e, $Sc - Sc' \equiv 0 \pmod{\Delta}$.

If $\beta = l\alpha$ is the element whose logarithm is sought, then we compute $\varphi(\alpha)$ and $\varphi(\beta)$ and try to solve

$$\varphi(\beta) \equiv l\varphi(\alpha) \pmod{\Delta}.$$

If this is possible, then we have found the correct discrete logarithm; otherwise, Algorithm 1 did not determine the correct group structure, and we have to restart the algorithm. (Beware that if no approximation Θ is known, then even in the first case the group structure determined by Algorithm 1 may be wrong.)

3.3 Success probability and running time analysis

The analysis of the algorithms is very similar to the special case treated in [Eng99]; we therefore do not give all the details.

Let N_S denote the number of S–smooth elements of G. Heuristically, one would expect that the probability of finding a relation in Step 2) of Algorithm 1 is about $\frac{N_S}{N}$, and since E is chosen sufficiently large, the deviation from this estimate can be shown to be small. Observe that the vectors $-e$ are contained in the cube $\{1-E,\dots,0\}^n$ and that the coefficient vectors r of smooth elements in the cube $\{0,\dots,d\}^n$. Since each obtained relation is of the form $c=r-e$, only relations inside the Minkowsky sum $W^+=\{1-E,\dots,d\}^n$ of these two cubes may be reached; moreover, the number of pairs $(e,r)=(r-c,r)$ yielding a fixed relation c is at most N_S. Hence, the probability of obtaining a relation is bounded above by $\frac{|\Gamma\cap W^+|N_S}{E^n}$, which can be shown to be no larger than $\frac{6\sqrt[40]{e}N_S}{5N}\in\frac{N_S}{N}\Theta(1)$ by estimating the cardinality of the intersection of a cube and a lattice. Similarly, there are exactly N_S pairs (e,r) yielding a fixed relation $c\in W^-=\{d+1-E,\dots,0\}^n$, so that the probability of finding a relation is bounded below by $\frac{|\Gamma\cap W^-|N_S}{E^n}\geq\frac{4N_S}{5\sqrt[40]{e}N}\in\frac{N_S}{N}\Theta(1)$. The same analysis applies to Step 1) of Algorithm 2, which amounts to finding one additional relation.

Our next aim is to determine the probability with which one run of Steps 1) to 4) of Algorithm 1 determines the correct group structure, i.e. that Step 2) determines the full lattice Γ. Assume that after finding i relations some sublattice $\Gamma_i\subseteq\Gamma$ is reached. We call the $(i+1)$-th relation *useful* if it increases the dimension of Γ_i by 1 or, after the maximal dimension is reached, it decreases its determinant. In the first case, let Γ_i' be a full dimensional lattice such that $\Gamma_i\subsetneq\Gamma_i'\subsetneq\Gamma$, in the second case, let $\Gamma_i'=\Gamma_i$. Then all relations in $\Gamma\backslash\Gamma_i'$ are useful, and the probability of finding a useless relation is bounded above by $\frac{|\Gamma_i'\cap W^+|N_S}{E^n}\leq\frac{7\sqrt[40]{e}N_S}{10N}$. Hence the conditional probability that a found relation is useless is at most $\frac{7\sqrt[40]{e}N_S}{N}/\frac{4N_S}{5\sqrt[40]{e}N}<\frac{12}{13}$.

To reach a full dimensional lattice, exactly n useful relations are needed. After that, Hadamard's bound shows that the determinant of the obtained lattice is at most $(\sqrt{n}E)^n$ since the coefficients of the relations are bounded by E in absolute value. As each further useful relation reduces the determinant by a factor of at least 2, at most $n\left(\operatorname{ld}E+\frac{1}{2}\operatorname{ld}n\right)$ additional useful relations are needed. Modelling the usefulness of relations by a Binomial $\left(14n\left(1+\operatorname{ld}E+\frac{1}{2}\operatorname{ld}n\right),1/13\right)$-distributed random variable X, we see that one run of Algorithm 1 is successful with probability at least

$$\begin{aligned} P\left(X \geq n\left(1+\operatorname{ld} E+\frac{1}{2}\operatorname{ld} n\right)\right) &\geq 1-P\left(|X-E(X)| \geq \frac{1}{13}E(X)\right) \\ &\geq 1-\frac{\operatorname{Var}(X)}{\frac{1}{169}E(X)^2} \text{ by Tschebyschew's inequality} \\ &= 1-\frac{168}{n\left(1+\operatorname{ld} E+\frac{1}{2}\operatorname{ld} n\right)} \to 1 \text{ for } n \to \infty. \end{aligned}$$

Since n will be chosen such that $n \to \infty$ for $N' \to \infty$, the success probability is in $1-o(1)$, and only $1+o(1)$ executions of the algorithm will be needed on average.

Taking into account that one linear combination of n primes can be computed in $O(n \log E)$, the analysis above shows that the expected running time for Step 2) of Algorithm 1 is in

$$O\left(n \log E \cdot n(\log E+\log n) \cdot \frac{N}{N_S}\right) \subseteq O^\sim\left(n^2 \frac{N}{N_S}\right)$$

and that the expected running time of Algorithm 2 is in $O^\sim\left(n\frac{N}{N_S}\right)$. By assumption, Step 1) of Algorithm 1 can be accomplished in $O^\sim(n^2)$, which is dominated by Step 2).

Concerning the linear algebra in Steps 3) and 4) of Algorithm 1, we follow [HM91]. Let $m = 14n\left(1+\operatorname{ld} E+\frac{1}{2}\operatorname{ld} n\right)$, $f(k) = k\log^2 k \log\log k$, and let $|A|$ denote the maximal absolute value among the coefficients of A. Using fast multiplication techniques, the rank of A and its Hermite normal form B, which is an intermediate step for computing the elementary divisor form Δ, are determined in

$$O\left(mn^2 f(n\log(n|A|))\right) \subseteq O^\sim\left(n^4\right),$$

and Δ together with the companion matrix S are computed in

$$O\left(n^3 \log|B| f(\log|B|)\right) \subseteq O^\sim\left(n^3\right)$$

since $|B| \leq N$ and $\log N \in O^\sim(1)$. Thus, the total expected running time of Algorithm 1 is

$$O^\sim\left(n^4 + n^2\frac{N}{N_S}\right),$$

which dominates the expected running time of

$$O^\sim\left(n\frac{N}{N_S}\right)$$

of Algorithm 2.

3.4 Subexponentiality

Clearly, $n = n_S$ grows with S, while $\frac{N}{N_S}$ decreases with S, so that the optimal algorithmic complexity is obtained for some intermediate value of S. It turns out that the best choice for n is subexponential. Recall the definition

$$L_{N'}(\rho) = e^{\rho\sqrt{\log N' \log\log N'}}$$

of the subexponential function with respect to the input size $\log N'$ and a parameter $\rho > 0$, which interpolates between the polynomial $e^{\rho\sqrt{(\log\log N')^2}} = (\log N')^\rho$ and the exponential function $e^{\rho\sqrt{(\log N')^2}} = N'^\rho$. Notice that $O^{\sim}(1) \subseteq L_{N'}(o(1))$.

Assume that a *smoothness result* of the following form is available: S can be chosen such that

$$n \in O\big(L_{N'}(\rho + o(1))\big)$$

and

$$\frac{N}{N_S} \in O\big(L_{N'}(\sigma + o(1))\big)$$

for some constants ρ, $\sigma > 0$. Then the expected running time of Algorithm 1 becomes

$$O\big(L_{N'}(\max\{4\rho, 2\rho + \sigma\} + o(1))\big).$$

Examples.

1) Ideal class groups of imaginary quadratic number fields of discriminant D

   Proposition 4.4 of [Sey87] shows that under the generalised Riemann hypothesis, the choice $S = \lceil \log L_N(\rho) \rceil$ yields $n \in O(L_N(\rho))$ and $\sigma = \frac{1}{2\rho}$. Letting $\rho = \frac{1}{2}$, which turns out to be optimal, we recover the running time of

   $$O\big(L_N(2 + o(1))\big) = O\big(L_{|D|}(\sqrt{2} + o(1))\big)$$

   of [McC89, HM89].

2) Jacobians of hyperelliptic curves over finite fields

   Theorem 6 of [ES00] shows that unconditionally, the choice $S = \lceil \log_q L_{q^g}(\rho) \rceil$ yields $\frac{N}{N_S} \in O\big(L_{q^g}(\sigma + o(1))\big)$ for $\sigma = \frac{1}{2\rho}$. Unfortunately, the need for rounding up S implies that no tighter bound than $n \in O(qL_{q^g}(\rho))$ can be shown. Thus, q must be subexponential in $g \log q$, which is only possible for g large compared to q. Precisely, it is shown in [Eng99], Theorem 8, that $n \in O\left(L_{q^g}\left(\rho + \frac{1}{\sqrt{\vartheta}}\right)\right)$ if all instances under consideration satisfy $g \geq \vartheta \log q$. Letting $\rho = \frac{1}{2}\left(\sqrt{1 + \frac{1}{\vartheta}} - \sqrt{\frac{1}{\vartheta}}\right)$, we have proved the following result:

Theorem 3 *Algorithms 1 and 2 compute discrete logarithms in Jacobians of hyperelliptic curves of genus g over finite fields $\mathbb{F}_q$ satisfying $g \geq \vartheta \log q$ for some positive constant ϑ in expected time in*

$$O\left(L_{q^g}\left(2\left(\sqrt{1+\frac{1}{\vartheta}}+\sqrt{\frac{1}{\vartheta}}\right)+o(1)\right)\right).$$

If $g/\log q$ tends to infinity for the instances under consideration, e.g. if q is constant and $g \to \infty$, then the complexity is

$$O\big(L_{q^g}(2+o(1))\big).$$

This is slightly better than the complexity proved in [Eng99] of

$$O\left(L_{q^g}\left(\frac{5}{\sqrt{6}}\left(\sqrt{1+\frac{3}{\vartheta}}+\sqrt{\frac{3}{\vartheta}}\right)+o(1)\right)\right)$$

since we based our analysis on faster algorithms for the linear algebra. For $\vartheta = 1$, for instance, the constant in the subexponential function drops from 7.61... to 4.82...

3.5 Outlook and remarks

Algorithms 1 and 2 generalise the well-known subexponential algorithms for computing the group structure and discrete logarithms in imaginary quadratic ideal class groups and hyperelliptic Jacobians to arbitrary arithmetical formations satisfying a few reasonable computational assumptions. This general framework is intended to simplify the development and the analysis of discrete logarithm algorithms for further class groups of unknown order. Specially interesting examples might be provided by more general Jacobians over finite fields. To apply the generic algorithms to a new class of arithmetical formations, it suffices to provide a description of their arithmetic meeting the requirements of Section 2. The proof of the subexponential running time then boils down to verifying the smoothness assumption of Section 3.4.

We remark that applying the results of Section 3.3 on the complexity of the linear algebra steps one may show that the discrete logarithm problem described in [MST99], which is set in some structure different from, but close to a group, can be solved slightly faster. Repeating the analysis of [MST99], which is in fact only valid for $g/\log q \to \infty$, and extending it for arbitrary values of ϑ, we obtain the following result.

Theorem 4 *There is a subexponential algorithm computing the regulator and discrete logarithms in the infrastructure of real quadratic hyperelliptic curves (i.e. curves for which the infinite prime divisor is splitting) of genus g over finite fields $\mathbb{F}_q$ of odd characteristic satisfying $g \geq \vartheta \log q$ for some positive constant ϑ. Its expected running time is in*

$$O\left(L_{q^g}\left(2\left(\sqrt{1+\frac{1}{\vartheta}}+\sqrt{\frac{1}{\vartheta}}\right)+o(1)\right)\right) = O\left(L_{q^{2g+2}}\left(\sqrt{2}\left(\sqrt{1+\frac{1}{\vartheta}}+\sqrt{\frac{1}{\vartheta}}\right)+o(1)\right)\right).$$

If $g/\log q$ tends to infinity for the instances under consideration, then the complexity is

$$O\big(L_{q^g}(2 + o(1))\big) = O\big(L_{q^{2g+2}}(\sqrt{2} + o(1))\big).$$

Using the theory of real quadratic hyperelliptic function fields of characteristic 2 developed in [MVZ98], it should be possible to extend the theorem to curves over arbitrary finite fields.

References

[Ari99] S. Arita. Algorithms for computations in Jacobians of C_{ab} curve and their application to discrete-log-based public key cryptosystems. In *Proceedings of Conference on The Mathematics of Public Key Cryptography, Toronto, June 12–17*, 1999.

[Art24] E. Artin. Quadratische Körper im Gebiete der höheren Kongruenzen II. *Mathematische Zeitschrift*, 19:207–246, 1924.

[BW88] Johannes Buchmann and H. C. Williams. A key-exchange system based on imaginary quadratic fields. *Journal of Cryptology*, 1:107–118, 1988.

[DH76] Whitfield Diffie and Martin E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6):644–655, November 1976.

[EG00] Andreas Enge and Pierrick Gaudry. A general framework for subexponential discrete logarithm algorithms. Research Report LIX/RR/00/04, LIX, June 2000. Available at http://www.math.uni-augsburg.de/~enge/vorabdrucke/subexp.ps.gz.

[ElG85] Taher ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*, 31(4):469–472, July 1985.

[Eng99] Andreas Enge. Computing discrete logarithms in high-genus hyperelliptic Jacobians in provably subexponential time. Combinatorics and Optimization Research Report CORR 99-04, University of Waterloo, February 1999. Available at http://-cacr.math.uwaterloo.ca/techreports/1999/corr99-04.ps; to appear in *Mathematics of Computation*.

[Eng00] Andreas Enge. *Hyperelliptic Cryptosystems: Efficiency and Subexponential Attacks*. PhD thesis, Universität Augsburg, 2000. To appear.

[ES00] Andreas Enge and Andreas Stein. Smooth ideals in hyperelliptic function fields. Combinatorics and Optimization Research Report CORR 2000-08, University of Waterloo, January 2000. Available at http://cacr.math.uwaterloo.ca/techreports/-2000/corr2000-08.ps.

[Fum97] Walter Fumy, editor. *Advances in Cryptology — EUROCRYPT '97*, volume 1233 of *Lecture Notes in Computer Science*, Berlin, 1997. Springer-Verlag.

[GPS98] S. D. Galbraith, S. Paulus, and N. P. Smart. Arithmetic on superelliptic curves. Preprint, 1998.

[HM89] James L. Hafner and Kevin S. McCurley. A rigorous subexponential algorithm for computation of class groups. *Journal of the American Mathematical Society*, 2(4):837–850, 1989.

[HM91] James L. Hafner and Kevin S. McCurley. Asymptotically fast triangularization of matrices over rings. *SIAM Journal on Computing*, 20(6):1068–1083, 1991.

[Kno75] John Knopfmacher. *Abstract Analytic Number Theory*, volume 12 of *North-Holland Mathematical Library*. North-Holland Publishing Company, Amsterdam, 1975.

[Kob87] Neal Koblitz. Elliptic curve cryptosystems. *Mathematics of Computation*, 48(177):203–209, January 1987.

[Kob89] Neal Koblitz. Hyperelliptic cryptosystems. *Journal of Cryptology*, 1:139–150, 1989.

[LT82] H. W. Lenstra Jr. and R. Tijdeman, editors. *Computational Methods in Number Theory. Part II*, Amsterdam, 1982. Mathematisch Centrum.

[McC89] Kevin S. McCurley. Cryptographic key distribution and computation in class groups. In *[Mol89]*, pages 459–479, 1989.

[Mil86] Victor S. Miller. Use of elliptic curves in cryptography. In *[Wil86]*, pages 417–426, 1986.

[Mol89] Richard A. Mollin, editor. *Number Theory and Applications*, volume 265 of *NATO ASI Series C: Mathematical and Physical Sciences*, Dordrecht, 1989. Kluwer Academic Publishers.

[MST99] Volker Müller, Andreas Stein, and Christoph Thiel. Computing discrete logarithms in real quadratic congruence function fields of large genus. *Mathematics of Computation*, 68(226):807–822, 1999.

[MVZ98] Volker Müller, Scott Vanstone, and Robert Zuccherato. Discrete logarithm based cryptosystems in quadratic function fields of characteristic 2. *Designs, Codes and Cryptography*, 14(2):159–178, May 1998.

[Nec94] V. I. Nechaev. Complexity of a determinate algorithm for the discrete logarithm. *Mathematical Notes*, 55(2):165–172, 1994.

[Sch82] R. J. Schoof. Quadratic fields and factorization. In *[LT82]*, pages 235–286, 1982.

[Sey87] Martin Seysen. A probabilistic factorization algorithm with quadratic forms of negative discriminant. *Mathematics of Computation*, 48(178):757–780, 1987.

[Sho97] Victor Shoup. Lower bounds for discrete logarithms and related problems. In *[Fum97]*, pages 256–266, 1997.

[Sie36] Carl Ludwig Siegel. Über die Classenzahl quadratischer Zahlkörper. *Acta Arithmetica*, 1:83–86, 1936.

[Wil86] Hugh C. Williams, editor. *Advances in Cryptology — CRYPTO '85*, volume 218 of *Lecture Notes in Computer Science*, Berlin, 1986. Springer-Verlag.