Some experiments investigating a possible $L(1/4)$ algorithm for the discrete logarithm problem in algebraic curves

Maike Massierer[*]

LORIA, Campus Scientifique, BP 239, 54506 Vandœuvre-lès-Nancy Cedex, France

maike.massierer@inria.fr

Abstract. The function field sieve, a subexponential algorithm of complexity $L(1/3)$ that computes discrete logarithms in finite fields, has recently been improved to an algorithm of complexity $L(1/4)$ and subsequently to a quasi-polynomial time algorithm. We investigate whether the new ideas also apply to index calculus algorithms for computing discrete logarithms in Jacobians of algebraic curves. While we do not give a final answer to the question, we discuss a number of ideas, experiments, and possible conclusions.

1 Introduction

The computation of discrete logarithms in certain classes of finite fields has recently been revolutionized by a number of developments building on the well-known function field sieve algorithm. As a result, pairing-based cryptosystems in small characteristic are no longer considered secure (see e.g. [11]), to name just one implication of these spectacular results.

The $L(1/3)$ subexponential complexity of the function field sieve was first improved by Joux [18] to $L(1/4 + o(1))$, and then by Göloğlu, Granger, McGuire, and Zumbrägel [10] to $L(1/4)$. Shortly thereafter, Bărbulescu, Gaudry, Joux, and Thomé [4] presented the first quasi-polynomial algorithm for computing discrete logarithms in finite fields. Granger, Kleinjung, and Zumbrägel [12] took some important steps towards provability of the heuristic complexity results by presenting an alternative descent method.

The computation of discrete logarithms in Jacobians of algebraic curves has developed essentially in parallel to finite fields. The $L(1/3)$ index calculus algorithm due to Enge, Gaudry, and Thomé [8] for computing discrete logarithms in Jacobians of low degree curves has very much in common with the function field sieve. In particular, many of the results that the function field sieve is based on hold analogously for algebraic curves, such as the splitting probability of polynomials and divisors, respectively.

The recent developments therefore raise the question of whether analogous improvements can be made to the index calculus algorithm for curves, thus producing an $L(1/4)$ or even quasi-polynomial algorithm. In this article, we report some thoughts on this questions, focusing particularly on the relation generation phase of the algorithm. While at this point, we are not able to answer the question completely, we discuss some possible approaches, the primary goal being to provide a basis for further discussion of this question in the scientific community.

We start by reviewing the concept of index calculus in general in Section 2, followed by a more detailed discussion of the function field sieve and its successors in Section 3 and index calculus in algebraic curves in Section 4. We then present some ideas for adaptation to curves

[*] The author was supported by the Swiss National Science Foundation under grant no. 151884.

in Section 5 and report on our experiments in Section 6. Finally, we discuss some possible theoretical conclusions including a conjecture in Section 7.

Acknowledgements. We thank Pierrick Gaudry for numerous helpful discussions on the content of this article. We thank Claus Diem for his comments on our conjecture and for making us aware of some of the results cited in Section 7.

2 Index calculus algorithms

One of the most prominent problems on which public key algorithms base their security is the discrete logarithm problem.

Definition 1. Let G be a multiplicative group. Given $g \in G$ and $h \in \langle g \rangle$, the *discrete logarithm problem (DLP)* is to compute a number $d \in \mathbb{Z}/\operatorname{ord}(g)\mathbb{Z}$ such that $g^d = h$. We call $d = \log_g h$ the *base-g discrete logarithm* of h.

For simplicity, it is often assumed that $G = \langle g \rangle$, i.e. that G is a cyclic group. In this paper, G is either a subgroup of the multiplicative group of a finite field of small characteristic or the Jacobian of an algebraic curve of large genus defined over a finite field. In both cases, the most efficient known attacks on the DLP are variants and further developments of a basic index calculus algorithm, which we describe below.

Suppose we want to compute a discrete logarithm $\log_g h$. The *main phase* of index calculus computes the discrete logarithms of all small primes of G. These are all primes of size below a certain bound, the *smoothness bound* B, and the set of such small primes is called the *factor base* $\mathcal{F}_B$. This phase can again be divided into two parts: First, one collects a sufficient number (more precisely, $|\mathcal{F}_B|$) of relations between the factor base elements, then one solves a sparse linear system in order to obtain the discrete logarithms of the factor base elements. Finally, the *individual logarithm phase* of the algorithm computes $\log_g h$ by rewriting this value as a sum of discrete logarithms of the factor base elements, which were computed earlier. For a detailed description of this index calculus method, see Algorithm 1.

Notice that both types of groups we are interested in are quotient groups. Therefore we write $G = G_1/G_2$, and for $\hat{g} \in G_1$, we denote by $g = \hat{g} \cdot G_2$ the class of $\hat{g}$ in G.

Algorithm 1 General outline of an index calculus algorithm

```
Input: $h \in G = G_1/G_2 = \langle g \rangle$, smoothness bound $B$
Output: $d = \log_g h$
1: Factor base: Construct factor base $\mathcal{F}_B = \{\hat{g}_1, \dots, \hat{g}_r\} \subseteq G_1$.
2: Relation collection: Construct relations of the form $g^{\alpha_i} = \prod_{j=1}^{r} g_j^{m_{i(j+1)}}$ for $i = 1, \dots, k > r$.
3: Linear algebra: Given the matrix $M = (m_{ij}) \in (\mathbb{Z}/|G|\mathbb{Z})^{k \times (r+1)}$, where $m_{i1} = -\alpha_i$ for all $i = 1, \dots, k$, compute a non-zero column vector $\gamma = (\gamma_1, \dots, \gamma_{r+1})^\top$ such that $M\gamma = 0$ and $\gamma_1 = 1$. Then we have $\gamma_{j+1} = \log_g g_j$ for all $j = 1, \dots, r$.
4: Individual logarithm: Search for $\beta$ such that $g^\beta h = \prod_{j=1}^{r} g_j^{\beta_j}$ for some $\beta_j$, output $d = -\beta + \sum_{j=1}^{r} \beta_j \gamma_{j+1}$.
```

Remark 1. Notice that the following slightly modified version of Algorithm 1 is equivalent. It will be useful later on. In step 2, we collect relations of the form $\prod_{j=1}^{r} g_j^{m_{ij}} = 1$. In step 3, the matrix M consists simply of the m_{ij}.

In order to discuss the complexity of this algorithm, let us denote by

$$L_x(\alpha, c) = \exp\left(c(1+o(1))(\log x)^{\alpha}(\log\log x)^{1-\alpha}\right)$$

the *subexponential complexity* in x, where x is usually the input size $|G|$, for $0 < \alpha < 1$ and a constant $c > 0$. When we do not want to specify the constant, we often write only $L_x(\alpha)$, or $L(\alpha)$ when the context is clear.

In both the case where G is a subgroup of the multiplicative group of a finite field and where G is the subgroup of the Jacobian of an algebraic curve of large genus, Algorithm 1 has heuristic complexity $L(1/2)$ when steps 2 and 4 are implemented in a simple way: In step 2, we pick random $\alpha_i \in \mathbb{Z}/|G|\mathbb{Z}$ and check whether g^{α_i} splits over the factor base. In step 4, we pick random $\beta \in \mathbb{Z}/|G|\mathbb{Z}$ until we find one such that $g^{\beta}h$ splits over the factor base.

The crucial question in the complexity analysis of the above algorithm is with which probability an element of G splits over the factor base. If the factor base is defined in terms of the smoothness bound B, we call such elements *B-smooth*.

For the sake of concreteness, let us first assume that $G \subseteq \mathbb{F}_{q^n}^{\times}$, where $\mathbb{F}_{q^n}$ is a finite field of small characteristic, meaning that $q < L_{q^n}(1/2, 1/\sqrt{2})$. Hence all elements of $\mathbb{F}_{q^n}$ can be represented uniquely by polynomials of degree at most n in $\mathbb{F}_q[x]$. In this case, the *small primes* are defined to be the monic irreducible polynomials of small degree, i.e. for a given smoothness bound B we have the factor base

$$\mathcal{F}_B = \{f \in \mathbb{F}_q[x] \mid f \text{ monic, irreducible, } \deg f \leq B\}.$$

Hence an element of $\mathbb{F}_q[x]$ is B-smooth if all its irreducible factors have degree at most B. The probability of this happening is given by the following result.

Theorem 1 ([20]). *A polynomial over a finite field $\mathbb{F}_q$ of degree n is B-smooth with probability $u^{-u(1+o(1))}$, where $u = n/B$.*

Using this result, a rough analysis of the index calculus algorithm in $G \subseteq \mathbb{F}_{p^n}^{\times}$ is as follows. We choose $B = \log_q L_{q^n}(1/2, 1/\sqrt{2})$, so that $|\mathcal{F}_B| = L_{q^n}(1/2, 1/\sqrt{2})$. Then the probability that a given element of $\mathbb{F}_{q^n}$, represented by a polynomial over $\mathbb{F}_q$ of degree less than n, is B-smooth, is

$$u^{-u(1+o(1))} = \exp(-u(1+o(1))\log u) = L_{q^n}(1/2, 1/\sqrt{2})^{-1}$$

for $u = \frac{n}{B} = \sqrt{2}\left(\frac{\log q^n}{\log\log q^n}\right)^{1/2}$, according to Theorem 1. Since we need to collect about $L_{q^n}(1/2, 1/\sqrt{2})$ relations, step 2 takes time $L_{q^n}(1/2, 1/\sqrt{2})^2 = L_{q^n}(1/2, \sqrt{2})$ (notice that smoothness tests and polynomial factorization over $\mathbb{F}_q[x]$ can be done in time polynomial in q). The linear system to be solved in step 3 is of size $L_{q^n}(1/2, 1/\sqrt{2}) \times L_{q^n}(1/2, 1/\sqrt{2})$ and sparse, since there are at most n entries per row. Hence it can be solved with Wiedemann's or Lanczos' algorithm in time $L_{q^n}(1/2, 1/\sqrt{2})^2 = L_{q^n}(1/2, \sqrt{2})$. In step 4, the expected number of tries until we find a value for β such that $g^{\beta}h$ is smooth is $u^{u(1+o(1))} = L_{q^n}(1/2, 1/\sqrt{2})$, which is the time needed for the individual logarithm phase. Finally, since the factor base can clearly be enumerated (step 1) in time $L_{q^n}(1/2, 1/\sqrt{2})$, the total time of Algorithm 1 in the case of finite fields is

$$L_{q^n}(1/2, \sqrt{2}).$$

An analogous result can be proven for Jacobians of algebraic curves, since there is a smoothness result for divisors similar to Theorem 1. Let $\mathcal{C}$ be a projective algebraic curve of genus g given by an absolutely irreducible plane affine model $\mathcal{C} : C(x, y)$, where $C \in \mathbb{F}_q[x, y]$ and $\mathbb{F}_q$ is the exact constant field of the function field of $\mathcal{C}$. The arithmetic in the Jacobian of such curves is detailed in [17], and in particular, splitting a divisor into a sum of places can be performed in polynomial time. The factor base consists of divisor classes represented by prime divisors of degree bounded by B, therefore a divisor is B-smooth if it has only places of degree at most B in its support. The smoothness probability of divisors is analogous to that of polynomials (and, in fact, also that of integers or, more generally, elements of arithmetic semigroups):

Theorem 2 ([16, Theorem 13])**.** *Let $0 < \varepsilon < 1, \gamma = \frac{3}{1-\varepsilon}$, and n, B with $u = \frac{n}{B}$ be given such that $3\log_q(14g+4) \leq B \leq n^{\varepsilon}$ and $u \geq 2\log(g+1)$. Then for n and B sufficiently large (with an explicit bound depending only on ε but not on q or g), the probability that a given effective divisor on $\mathcal{C}$ of degree n is B-smooth is at least $u^{-u(1+o(1))}$.*

Using this theorem, we can show as above that index calculus in $\mathrm{Jac}_{\mathcal{C}}(\mathbb{F}_q)$, which is a group of size approximately q^g, has heuristic complexity $L_{q^g}(1/2, \sqrt{2})$ for $q, g \to \infty$.

The complexity results discussed in this section as well as the following section are always of heuristic nature, since the complexity analysis relies on heuristic assumptions, for example that the polynomials (respectively divisors) that are constructed in the computation have the same smoothness probability as random polynomials (respectively, divisors) of the same degree and that the linear system to be solved has full rank.

In the following we discuss the variants of Algorithm 1 that lead to first an $L(1/3)$ and then even $L(1/4)$ and quasi-polynomial algorithms for finite fields of small characteristic, and an $L(1/3)$ algorithm for a certain type of algebraic curves. In our exposition, we concentrate mainly on the relation collection phase of the algorithm, since it is the focus of this work.

3 Finite fields of small characteristic

The function field sieve, due to Adleman [1], and its successors are the best algorithms for computing discrete logarithms in finite fields of small characteristic. The function field sieve gets its name from the fact that relations are produced with the help of two different function fields, a strategy originally developed in the number field sieve (which is good for factoring integers and computing discrete logarithms in finite fields of large characteristic) and subsequently adapted to finite fields of small characteristic. By searching for half-relations in each of the function fields and then combining them into full relations afterwards, one is able to reduce the degree of the polynomials that are required to be smooth, thus increasing the smoothness probability. This leads to a relation collection phase of complexity $L(1/3)$, as opposed to $L(1/2)$ above. The individual logarithm phase is also modified so that it has complexity $L(1/3)$. The individual logarithm is computed with a so-called *descent strategy*, where $\log_g h$ is first written as a sum of logarithms of elements of moderate degree, and then one proceeds recursively, writing each summand as a sum of logarithms of elements of smaller degree, until one finally arrives at a sum of logarithms of elements of small enough degree (i.e. all lying in the factor base). Combining these two speed-ups, one gets an algorithm of overall complexity $L(1/3)$. We now give some more details of the relation collection phase, which can best be explained with the help of the commutative diagram given in Figure 1, where $K = \mathbb{F}_q$ and the field $L = \mathbb{F}_{q^n}$ on the bottom is the field where the discrete logarithm is to be computed.

In order to produce relations, one starts with a polynomial $\phi \in \mathbb{F}_q[x, y]$, typically of shape $\phi(x, y) = a(x)y + b(x)$, and maps it via ψ_1 and ψ_2 into $\mathcal{O}_1$ and $\mathcal{O}_2$, respectively. If $\psi_1(\phi) \in \mathcal{O}_1$ and $\psi_2(\phi) \in \mathcal{O}_2$ are both B-smooth, then one maps both of these elements into $\mathbb{F}_{q^n}$ via η_1 and η_2, where they produce a relation of the form

$$\eta_1(\psi_1(\phi)) = \eta_2(\psi_2(\phi)),$$

Figure 1: Commutative diagram illustrating relation collection phase of FFS and its successors

$$\begin{tikzcd}
& K[x,y] \arrow[dl, "\psi_1"'] \arrow[dr, "\psi_2"] & \\
\mathcal{O}_1 = K[x,y]/f_1(x,y) \arrow[dr, "\eta_1"'] & & \mathcal{O}_2 = K[x,y]/f_2(x,y) \arrow[dl, "\eta_2"] \\
& L = K[x]/k(x) &
\end{tikzcd}$$

due to the commutativity of the diagram. The function fields involved are $F_i = \mathrm{Quot}(\mathcal{O}_i) \supseteq \mathcal{O}_i, i = 1, 2$.

3.1 The original function field sieve

Applicability. The function field sieve computes discrete logarithms in a field $\mathbb{F}_{q^n} = \mathbb{F}_q[x]/k(x)$ (with k monic and irreducible of degree n) of small characteristic, meaning that $q \leq L_{q^n}(1/3)$. Write $K = \mathbb{F}_q$ and $L = \mathbb{F}_{q^n}$.

Function fields. The function field sieve usually chooses f_1 monic in y of degree d (a parameter to be optimized in the complexity analysis) and of degree $O(1)$ in x, and $f_2(x,y) = y - h(x)$ where h has degree n/d. It is further required that $f_1(x, h(x)) \equiv 0 \bmod k$, since this allows to define the maps in the way specified below.

Thus the function fields involved are $F_1 = \mathrm{Quot}(\mathcal{O}_1) = \mathbb{F}_q(x)[y]/f_1(x,y)$, an extension of $\mathbb{F}_q(x)$ of degree d, and $F_2 = \mathrm{Quot}(\mathcal{O}_2) = \mathbb{F}_q(x)[y]/f_2(x,y)$, which is in fact the rational function field $\mathbb{F}_q(x)$, since it is a degree 1 extension of $\mathbb{F}_q(x)$. Let us write $\mathcal{O}_i = \mathbb{F}_q[x, \alpha_i(x)]$ with $f_i(x, \alpha_i(x)) = 0$ for $i = 1, 2$, and notice that $\alpha_2 = h$.

Maps. We have $\psi_1 : y \mapsto \alpha_1(x), \psi_2 : y \mapsto h(x)$ and $\eta_i : P(x, \alpha_i(x)) \mapsto P(x, h(x))$. Notice that the η_i are well-defined since we have $f_i(x, h(x)) \equiv 0 \bmod k$.

Factor base. Since the $\mathcal{O}_i$ are in general not unique factorization domains, one considers ideals, which factor uniquely. Hence one defines the factor base in terms of ideals in O_i:

$$\mathcal{F}_B^{(i)} = \{\langle \ell(x), \alpha_i(x) - r(x)\rangle \mid \ell \text{ irreducible, } \deg \ell \leq B, f_i(x, r(x)) \equiv 0 \bmod \ell(x)\}$$

and $\mathcal{F}_B = \mathcal{F}_B^{(1)} \cup \mathcal{F}_B^{(2)}$.

Relations. In order to produce relations, one picks $\phi = a(x)y - b(x) \in \mathbb{F}_q[x,y]$ with $\deg a, \deg b \leq e$ for some sieving parameter e (to be optimized in the complexity analysis). One imposes further that a and b are coprime in order to avoid pairs (a,b) which are multiples of each other. We have $\psi_i(\phi) = a(x)\alpha_i(x) - b(x)$, and it is easy to see that the ideal $(a(x)\alpha_i(x) - b(x))O_i$ is smooth with respect to the factor base $\mathcal{F}_B^{(i)}$ if and only if the norm $N_{F_i|\mathbb{F}_q(x)}(a(x)\alpha_i(x) - b(x))$ is B-smooth as a polynomial (note that this is a polynomial, since $a\alpha_i - b$ is a polynomial). Furthermore, we have

$$N_{F_i|\mathbb{F}_q(x)}(a(x)\alpha_i(x) - b(x)) = \mathrm{Res}_y(a(x)y - b(x), f_i) = f_i^{(h)}(a, b)$$

where $f_i^{(h)}(a,b) = f_i(a/b)b^{\deg f_i}$ is the homogenization of f_i, regarded as a polynomial in y.

By sieving, we find enough pairs $(a,b) \in \mathbb{F}_q[x]^2$ such that $f_i^{(h)}(a,b)$ are B-smooth for both $i = 1, 2$; such pairs are called *doubly B-smooth*. By raising the decompositions to the respective class numbers, we obtain factorizations into principal ideals, or equivalently by looking at the generators of the ideals, half-relations in $\mathcal{O}_i$, which can then be mapped via η_i into $\mathbb{F}_{q^n}$, thus producing full relations there. For the many technical details we are skipping here, see e.g. [3].

Descent. In a first step, the descent rewrites the sought after logarithm as a sum of logarithms of $L(2/3)$-smooth elements. Then it recursively descends the elements in the sum to elements of $\mathcal{F}_B$ by sieving elements of a lattice.

Complexity. Choose $e = B = \log_q L_{q^n}(1/3)$ such that $|\mathcal{F}_B| = L_{q^n}(1/3)$, and choose $d = \left(\frac{n \log q}{\log n}\right)^{1/3}$. Working out the constants as shown e.g. in [3, Chapter 7.5], the function field sieve has a total complexity of $L_{q^n}(1/3, (32/9)^{1/3})$.

3.2 The Joux–Lercier variant of the function field sieve

Joux and Lercier [19] give a simplified version of FFS, where the function fields are no longer visible as such, though they are still present "in the background" as described above, and one no longer has to deal with ideals.

Applicability. As before, this variant of the function field sieve applies to fields $\mathbb{F}_{q^n} = \mathbb{F}_q[x]/k(x)$, and we write $K = \mathbb{F}_q, L = \mathbb{F}_{q^n}$.

Function fields. Let $f_1(x,y) = \gamma_1(y) - x$ and $f_2(x,y) = y - \gamma_2(x)$ for γ_i of degree d_i with $d_1 d_2 \geq n$, and assume that $k \mid \gamma_1(\gamma_2(x)) - x$.

The function fields involved are $F_1 = \mathbb{F}_q(x)[y]/(\gamma_1(y) - x)$, an extension of $\mathbb{F}_q(x)$ of degree d_1, and $F_2 = \mathbb{F}_q(x)[y]/(y - \gamma_2(x))$, which is in fact the rational function field (since it is a degree 1 extension of $\mathbb{F}_q(x)$). Hence as above, there is one rational side.

Interpreting F_1 as $\mathbb{F}_q(y)[x]/(\gamma_1(y) - x)$, which is the rational function field in y, one might even say that this algorithm has two rational sides.

Maps. We have $\psi_1 : y \mapsto \gamma_2(x), \psi_2 : x \mapsto \gamma_1(y)$ and $\eta_1 : x \mapsto \gamma_1(\gamma_2(x)) = \gamma_1(\eta_2(y)), \eta_2 : y \mapsto \gamma_2(x)$.

Factor base. Define

$$\mathcal{F}_B = \{\ell(x), \ell(y) \mid \ell \in \mathbb{F}_q[x] \text{ monic, irreducible, } \deg \ell \leq B\}.$$

Relations. By sieving (as above), find $\phi(x,y) = a(x)y - b(x) \in \mathbb{F}_q[x,y]$ such that $\phi(x, \gamma_2(x))$ and $\phi(\gamma_1(y), y)$ are both B-smooth. Such ϕ give relations since $\phi(x, \gamma_2(x)) = \phi(x,y) = \phi(\gamma_1(y), y)$.

Descent. This is the same as for the function field sieve.

Complexity. This is also the same as for the function field sieve.

3.3 The L(1/4) algorithm

A series of papers [9, 18, 10, 4, 11, 12] with very similar ideas, often building on each other, drastically improved the efficiency of discrete logarithm computations in certain classes of finite fields. While Göloğlu, Granger, McGuire, and Zumbrägel [9] were the first to point out that relation collection can be achieved in polynomial time, Joux [18] gave the first algorithm of complexity

$L(1/4+o(1))$, which was subsequently improved to $L(1/4)$ in [10]. Finally, Granger, Kleinjung, and Zumbrägel took some steps in the direction of a provable complexity result by suggesting an alternative descent method in [12]. Most of these papers also present record breaking computations. The differences between these algorithms lie mostly in the individual logarithm (descent) phase, while they share very similar strategies for relation collection. Therefore, we present the basic ideas of all of them together in this section. We point out though that [9, 10, 11, 12] limit themselves to fields of characteristic two.

Applicability. The algorithms apply to fields $\mathbb{F}_{q^{mn}}$, where $q \approx n$ and m is small and fixed ([18, 4] consider only $m = 2$). Hence a given field must be embedded into a finite field of this shape as a first step. Writing $K = \mathbb{F}_{q^m}$ and $L = \mathbb{F}_{q^{mn}}$, the relation collection phase can also be described via the diagram in Figure 1.

Function fields. Let $f_1(x, y) = y - x^q$ and $f_2(x, y) = h_1(x)y - h_0(x)$ (respectively $f_2(x, y) = h_1(y)x - h_0(y)$) for polynomials $h_0, h_1 \in \mathbb{F}_q[x]$ of small, constant degree. Experiments suggest that suitable h_0, h_1 of degree at most 2 always exist, see [18].

Writing $\mathbb{F}_{q^{mn}} = \mathbb{F}_{q^m}[x]/k(x)$, [18, 4] require that $k \mid h_1(x)x^q - h_0(x)$, while [11, 12] require that $k \mid h_1(x^q)x - h_0(x^q)$, where the latter allows a slightly larger class of fields to be embedded into $\mathbb{F}_{q^{mn}}$ than the former.

We have $\mathcal{O}_1 \cong \mathbb{F}_{q^m}[x]$ and $\mathcal{O}_2 \cong \mathbb{F}_{q^m}[x, 1/h_1(x)]$ (respectively $\mathcal{O}_2 \cong \mathbb{F}_{q^m}[y, 1/h_1(y)]$), and the function fields are $F_i = \mathrm{Quot}(\mathcal{O}_i)$, which are both degree one extensions of $\mathbb{F}_{q^m}(x)$ (respectively F_2 is a degree one extension of $\mathbb{F}_{q^m}(y)$), i.e. they are the rational function field themselves.

Maps. Let $\psi_1 : y \mapsto x^q$ and $\psi_2 : y \mapsto h_0(x)/h_1(x)$ (respectively $x \mapsto h_0(y)/h_1(y)$), and let η_1 and η_2 be reduction modulo k.

Factor base. Set $B = 1$, the factor base consists of all monic linear polynomials over $\mathbb{F}_{q^m}$. Notice that [18] includes also quadratic polynomials for technical reasons concerning the descent, we disregard this here.

Relations. The approach of [18, 4] can be interpreted as follows. Start with a polynomial of the shape $\phi = (a^q y + b^q)(cx + d) - (ax + b)(c^q y + d^q)$ for $(a, b, c, d) \in \mathbb{F}_{q^m}^4 \setminus {\mathbb{F}_q}^4$. Then on the left (meaning $\psi_1(\phi) \in \mathcal{O}_1$), this splits completely: set $y \mapsto x^q$ and use the identity $F(x)^q G(x) - F(x)G(x)^q = G(x)\prod_{\alpha\in\mathbb{F}_q}(F(x) - \alpha G(x))$ for $F, G \in \mathbb{F}_{q^m}[x]$. On the right (meaning $\psi_2(\phi) \in \mathcal{O}_2$), this becomes a polynomial of small, constant degree and therefore has high splitting probability (for this purpose, h_1 is included in $\mathcal{F}_B$ by definition). We get a relation as soon as the right-hand-side is smooth.

The other papers start with a polynomial $\phi = xy + ay + bx + c \in \mathbb{F}_{q^m}[x, y]$ and map it to both sides. On the left, the resulting polynomial is of the form $\psi_1(\phi) = x^{q+1} + ax^q + bx + c$, and it can be shown (using a transformation resulting in a polynomial of shape $x^{q+1} + Dx + D$ and results of Bluher [2] and Helleseth–Kholosha [15] on the splitting probability of polynomials of this shape, see [9]) that such polynomials split with probability q^{-3}, which is much higher than the splitting probability given by Theorem 1 for random polynomials of the same degree. A parametrization of such polynomials which split completely is also given. On the right, again, the polynomial $\psi_2(\phi)$ splits with large constant probability.

Descent. We skip this, since it is different in all the papers mentioned above.

Complexity. Since $n \approx q$ and m is small and fixed, we have that q^m is polynomial in the input size q^{mn}. Then the relation generation takes only polynomial time, since the splitting probability is constant. The linear algebra is of size $q^m \times q^m$.

3.4 Comparison

A schematic comparison of the relation collection phase for all algorithms discussed in this section is given in Figure 2.

4 Algebraic curves

The algorithm of Enge, Gaudry, and Thomé [7, 8] is the most efficient currently known algorithm to compute discrete logarithms in Jacobians of algebraic curves. It has complexity $L(1/3)$, but it is important that this improvement over the $L(1/2)$ algorithm presented in Section 2 is not due to the search of relations in two different function fields, but rather to the bounds on the degrees of the curve equation, which are due to the fact that the algorithm applies only to C_{ab}-curves.

Applicability. The algorithm applies to so-called $\mathcal{C}_{ab}$-curves defined over $\mathbb{F}_q$ by an equation

$$\mathcal{C}_{ab} : y^a + x^b + f(x, y) = 0,$$

such that the affine part of the curve is smooth, and where $\gcd(a, b) = 1$ and $ai + bj \leq ab$ for all monomials $x^i y^j$ of f. Such curves have genus $g = (a-1)(b-1)/2$. The DLP is defined in its Jacobian $\mathrm{Jac}_{\mathcal{C}_{ab}}(\mathbb{F}_q)$, which has size about q^g. The elements of the Jacobian are degree zero divisor classes, which we denote by $[D]$ for a divisor D.

Function field. There is only one function field involved in the relation search, namely the function field $\mathbb{F}_q(\mathcal{C}_{ab}) = \mathbb{F}_q(x)[y]/(y^a + x^b + f(x, y))$ of the curve. It is a degree a extension of $\mathbb{F}_q(x)$.

Factor base. The factor base consists of all prime divisors on the curve of degree at most B, where $B = \log_q L_{q^g}(1/3)$ is chosen such that the factor base has size $L_{q^g}(1/3)$.

Relations. One searches for polynomial functions $\phi \in \mathbb{F}_q(\mathcal{C}_{ab})$ such that $\mathrm{div}(\phi)$ is B-smooth. If $\mathrm{div}(\phi) = \sum \nu_i P_i$ with $P_i \in \mathcal{F}_B$, then $\sum \nu_i [P_i] = 0$ gives a relation in $\mathrm{Jac}_{\mathcal{C}_{ab}}$. Smoothness of a divisor can be tested by checking if $\mathrm{N}_{\mathbb{F}_q(\mathcal{C}_{ab})|\mathbb{F}_q(x)}(\phi) \in \mathbb{F}_q(x)$, and actually $\in \mathbb{F}_q[x]$ since ϕ is a polynomial function, is B-smooth. Notice that $\mathrm{N}_{\mathbb{F}_q(\mathcal{C}_{ab})|\mathbb{F}_q(x)}(\phi) = \mathrm{Res}_y(\phi(x, y), y^a + x^b + f(x, y))$ and can therefore easily be computed as a resultant.

Descent. The algorithm recursively descends a place of degree $g^{1/3+\tau}, \tau \in [0, 2/3]$, to a sum of places of degree $g^{1/3+\tau/2}$ by sieving elements of a lattice. Notice that there is no initial smoothing.

Complexity. Choosing $a \approx g^\alpha, b \approx g^{1-\alpha}$ for $\alpha \in [1/3, 2/3]$, and $\deg_y \phi \approx g^{\alpha-1/3}, \deg_x \phi \approx g^{2/3-\alpha}$, one can estimate

$$\deg_x \left(\mathrm{N}_{\mathbb{F}_q(\mathcal{C}_{ab})|\mathbb{F}_q(x)}(\phi)\right) \leq g^{2/3},$$

hence the norm (and therefore $\mathrm{div}(\phi)$) is B-smooth with probability $L_{q^g}(1/3)^{-1}$. Hence one can collect $|\mathcal{F}_B| = L_{q^g}(1/3)$ relations in time $L_{q^g}(1/3)$, and the linear system can be solved in the same time. The descent phase has the same complexity, and therefore we get an overall complexity of $L_{q^g}(1/3)$, where q and g grow to infinity.

5 Ideas for a faster relation generation phase for curves

The first goal in an attempt to transport the ideas of the $L(1/4)$ algorithm to algebraic curves may be to devise a strategy to produce relations in a more efficient way. One may take Joux's

Figure 2: Comparison relation search of FFS and its successors

Function field sieve [1], Section 3.1

$$\begin{tikzcd}
& \mathbb{F}_q[x,y] \arrow[dl, "y \mapsto \alpha_1(x)"'] \arrow[dr, "y \mapsto h(x)"] & \\
\mathbb{F}_q[x,y]/f_1(x,y) \arrow[dr, "\bmod k(x)"'] & & \mathbb{F}_q[x,y]/(y-h(x)) \arrow[dl, "\bmod k(x)"] \\
& \mathbb{F}_{q^n} = \mathbb{F}_q[x]/k(x) &
\end{tikzcd}$$

where $k(x) \mid f_1(x, h(x))$

$\mathcal{F}_B = \{\langle \ell(x), \alpha_i(x) - r(x)\rangle \mid \ell$ irreducible,
$\deg \ell \leq B, f_i(x, r(x)) \equiv 0 \bmod \ell(x)\}$
with $B = \log_q L_{q^n}(1/3)$

$\phi = a(x)y - b(x) \in \mathbb{F}_q[x,y]$

Criterion for relation:
$f_1(x, a(x)/b(x))b(x)^d, a(x) - b(x)h(x)$ are B-smooth

Joux–Lercier variant of FFS [19], Section 3.2

$$\begin{tikzcd}
& \mathbb{F}_q[x,y] \arrow[dl, "y \mapsto \gamma_2(x)"'] \arrow[dr, "x \mapsto \gamma_1(y)"] & \\
\mathbb{F}_q[x,y]/(\gamma_1(y)-x) \arrow[dr, "x \mapsto \gamma_1(\gamma_2(x))"'] & & \mathbb{F}_q[x,y]/(y-\gamma_2(x)) \arrow[dl, "y \mapsto \gamma_2(x)"] \\
& \mathbb{F}_{q^n} = \mathbb{F}_q[x]/k(x) &
\end{tikzcd}$$

where $k(x) \mid \gamma_1(\gamma_2(x)) - x$

$\mathcal{F}_B = \{\ell(x), \ell(y) \mid \ell \in \mathbb{F}_q[x]$ monic, irreducible,
$\deg \ell \leq B\}$
with $B = \log_q L_{q^n}(1/3)$

$\phi = a(x)y - b(x) \in \mathbb{F}_q[x,y]$

Criterion for relation:
$\phi(x, \gamma_2(x)), \phi(\gamma_1(y), y)$ are B-smooth

French relation search [18, 4], Section 3.3

$$\begin{tikzcd}
& \mathbb{F}_{q^2}[x,y] \arrow[dl, "y \mapsto x^q"'] \arrow[dr, "y \mapsto h_0(x)/h_1(x)"] & \\
\mathbb{F}_{q^2}[x,y]/(y-x^q) \arrow[dr, "\bmod k(x)"'] & & \mathbb{F}_{q^2}[x,y]/(h_1(x)y-h_0(x)) \arrow[dl, "\bmod k(x)"] \\
& \mathbb{F}_{q^{2n}} = \mathbb{F}_{q^2}[x]/k(x) &
\end{tikzcd}$$

where $k(x) \mid h_1(x)x^q - h_0(x)$

$\mathcal{F}_B = \{\ell(x) \mid \ell \in \mathbb{F}_{q^2}[x]$ monic, linear$\}$
i.e. $B = 1$

$\phi = (a^q y + b^q)(cx + d) - (ax + b)(c^q y + d^q) \in \mathbb{F}_{q^2}[x,y]$

Criterion for relation:
$(a^q c - ac^q)xh_0(x) + (a^q d - bc^q)h_0(x) +$
$+ (b^q c - ad^q)xh_1(x) + (b^q d - bd^q)h_1(x)$
splits into linear factors

Irish–Swiss relation search [11, 12], Section 3.3

$$\begin{tikzcd}
& \mathbb{F}_{q^m}[x,y] \arrow[dl, "y \mapsto x^q"'] \arrow[dr, "x \mapsto h_0(y)/h_1(y)"] & \\
\mathbb{F}_{q^m}[x,y]/(y-x^q) \arrow[dr, "\bmod k(x)"'] & & \mathbb{F}_{q^m}[x,y]/(h_1(y)x-h_0(y)) \arrow[dl, "\bmod k(x)"] \\
& \mathbb{F}_{q^{mn}} = \mathbb{F}_{q^m}[x]/k(x) &
\end{tikzcd}$$

where $k(x) \mid h_1(x^q)x - h_0(x^q)$

$\mathcal{F}_B = \{\ell(x), \ell(y) \mid \ell \in \mathbb{F}_{q^m}[x]$ monic, linear$\}$
i.e. $B = 1$

$\phi = xy + ay + bx + c \in \mathbb{F}_{q^m}[x,y]$

Criterion for relation:
$x^{q+1} + ax^q + bx + c$ and
$yh_0(y) + ayh_1(y) + bh_0(y) + ch_1(y)$
split into linear factors

point of view and try to amplify relations (e.g. via homographies), or that of the Irish–Swiss team of finding families of functions that have higher-than-usual splitting probability. In both cases, the shape of the function field plays a crucial role, and the new algorithms profit precisely from this freedom to choose convenient function fields.

In the case of algebraic curves, the function field is given by the curve in question, and there is unfortunately no freedom to choose it. Therefore, we ask ourselves instead whether there exist curves (with corresponding function fields) for which we are easily able to produce relations. The curves should, of course, have no known weaknesses with respect to the DLP. In particular, they should not be supersingular, or more generally, not have a Jacobian of smooth order, they should not have small embedding degree, and they should obviously not have genus 0. In fact, in order to mimic the finite field algorithms, we would need a curve defined over $\mathbb{F}_q$ of genus about q, so that the Jacobian has order about q^q.

A more precise formulation of the question we wish to answer is the following. We write $\mathrm{poly}(q) = q^{\Theta(1)}$.

Question 1. *Let $\mathcal{C}$ be a projective curve defined over $\mathbb{F}_q$ of degree $\mathrm{poly}(q)$. Under which conditions is the number of polynomial functions of degree $O(1)$ defining divisors which split into factors of degree $O(1)$ unexpectedly large, say of the form $\mathrm{poly}(q)$?*

Unfortunately our impression is that there may not exist curves which satisfy all of these criteria. A precise formulation of this impression is given in Section 7, but before we get to it, we explain the reasoning and experiments that led to this conclusion.

A first observation is that the function fields chosen in the $L(1/4)$ algorithms for finite fields are not interesting in our context, since the corresponding curves $y - x^q$ and $h_1(x)y + h_0(x)$ both have genus zero.

We tried two approaches, corresponding to the point of view of Joux (and the "French team") and that of the "Irish–Swiss team", respectively.

The first approach would be to find a curve, together with one smooth function, which gives a relation, and to then amplify this into many relations using homographies. This works well for the specific curve $y - x^q$, since it is of particularly simple shape, as shown in Example 1. For curves with more complicated equations, this becomes more difficult, as the application of the homography is no longer compatible with taking the resultant. Example 2 shows a curve where this approach does not work, i.e. where we find exactly two polynomial functions that have smooth principal divisors but no more.

The second approach would be to find a curve, together with a (parametrized) family of smooth functions, such that the (parametrized) resultant has high splitting probability. Again, this works well for the curve $y - x^q$, since we have $\mathrm{Res}_y(\phi(x, y), y - x^q) = \nu\phi(x, x^q)$ for some constant ν, and therefore the norm of ϕ depends on ϕ in a very simple way. For this reason, simple ϕ's produce simple $\mathrm{N}(\phi)$'s, and it is easy to produce polynomials for which results on higher splitting probability are known, see Example 1. The difficulty about generalizing this approach is that there exist few known families of polynomials with high splitting probability, and it is difficult to produce these few very simple polynomials with more interesting curves.

Example 1. Let $\mathcal{C} : y - x^q = 0$ and $\phi(x, y) = y - x \in \mathbb{F}_q(\mathcal{C})$. Then

$$\mathrm{N}_{\mathbb{F}_q(\mathcal{C})|\mathbb{F}_q(x)}(\phi) = \mathrm{Res}_y(y - x, y - x^q) = \nu(x^q - x) = \nu \prod_{\alpha\in\mathbb{F}_q} (x - \alpha)$$

for some constant ν. Since $\mathrm{N}(\phi)$ splits completely into linear factors, $\mathrm{div}(\phi)$ is smooth and therefore gives a relation.

In order to amplify this one relation into many, Joux proposes to apply a homography $x \mapsto \frac{ax+b}{cx+d}$, where $a, b, c, d \in \mathbb{F}_{q^2}$. After multiplication by $(cx+d)^{q+1}$, this gives

$$(ax+b)^q(cx+d) - (ax+b)(cx+d)^q = (cx+d) \prod_{\alpha \in \mathbb{F}_q} ((a - \alpha c)x + (b - \alpha d)),$$

and one has found many polynomials that split into linear factors. How many? As many as there are matrices $\begin{pmatrix} a & b \\ c & d \end{pmatrix} \in \mathrm{PGL}_2(\mathbb{F}_{q^2}) / \mathrm{PGL}_2(\mathbb{F}_q)$, which has cardinality $q^3 + q$.

Interpreting this in terms of relations on the curve $\mathcal{C}$, we "apply" the homography to $\phi(x, y) = y - x$ as follows. Setting

$$\phi^{\mathrm{hom}}(x, y) = (a^q y + b^q)(cx + d) - (ax + b)(c^q y + d^q),$$

we get

$$\mathrm{Res}_y(\phi^{\mathrm{hom}}(x, y), y - x^q) = \nu(cx + d) \prod_{\alpha \in \mathbb{F}_q} ((a - \alpha c)x + (b - \alpha d))$$

for a constant ν, which gives about q^3 different relations in $\mathrm{Jac}_{\mathcal{C}}$.

Alternatively, we may define $\phi(x, y) = xy + ay + bx + c$ for $a, b, c, d \in \mathbb{F}_{q^m}$, which gives

$$\mathrm{Res}_y(\phi(x, y), y - x^q) = x^{q+1} + ax^q + bx + c.$$

If $c \neq ab$ and $b \neq a^q$, then the transformation $x \mapsto \frac{ab+c}{b+a^q}x + a$ maps the result to $x^{q+1} - Dx + D$ which, for $D \in \mathbb{F}_{q^m}$, splits into linear factors for about q^{m-3} values of D according to [2, Lemma 4.4]. In other words, ϕ has splitting probability q^{-3}, which is much higher than that of a random polynomial of degree $q + 1$, which is on the order of q^{-q} according to Theorem 1.

The connection between the two approaches, and an interesting observation, is that polynomials $(ax + b)^q(cx + d) - (ax + b)(cx + d)^q$ are of the form $Ax^{q+1} + Bx^q + Cx + D$, and such polynomials have splitting probability q^{-3} according to [2], which is higher than expected. It is possible that this phenomenon holds more generally. For example, $(ax^2 + bx + c)^q(dx^2 + ex + f) - (ax^2 + bx + c)(dx^2 + ex + f)^q$ splits completely into factors of degree 2 and has terms $x^{2q+2}, x^{2q+1}, x^{2q}, x^{q+2}, x^{q+1}, x^q, x^2, x, 1$, so one might expect that polynomials $Ax^{2q+2} + Bx^{2q+1} + Cx^{2q} + Dx^{q+2} + Ex^{q+1} + Fx^q + Gx^2 + Hx + I$ split into quadratic factors with higher probability than expected. Counting arguments as well as experiments for $m = 2$ and small fields suggest that this may be the case with probability q^{-7}, but such counting experiments are only possible for very small fields and therefore only give a vague idea, and we are not aware of any results similar to those of [2, 15] for such polynomials. If such a thing were true, we could search for pairs of $\mathcal{C}, \phi$ such that the resultant has such a shape. Notice that it is obviously possible to extend this idea, using e.g. polynomials of larger degrees and possibly products of polynomials of different degrees.

6 Experiments

We discuss the approaches that we tried using some exemplary curves. In this entire section, we give resultants only up to multiplication by a constant, since this simplifies notation and is sufficient for determining the splitting properties.

Example 2. Consider $\mathcal{C} : y^2 - x^{q+1} + 2x - 1 = 0$, which is a hyperelliptic, non-supersingular curve of genus $g = (q-1)/2$, and $\phi(x, y) = y - x + 1$. Then

$$\mathrm{Res}_y(\phi, y^2 - x^{q+1} + 2x - 1) = x(x^q - x)$$

and therefore splits into linear factors. However, exhaustive search shows that the only functions of the form $\psi = axy + by + cx + d$ with $a, b, c, d \in \mathbb{F}_{q^2}$ such that $\mathrm{Res}_y(\psi, y^2 - x^{q+1} + 2x - 1)$ splits into linear factors and produces a non-trivial relation are $\psi = y + x - 1, y - x + 1$, for $q = 7, 11, 19, 23$ (notice that these are all primes $\equiv 3 \bmod 4$). For these values of ψ, we have

$$\mathrm{Res}_y(\psi, y^2 - x^{q+1} + 2x - 1) = x(x^q - x).$$

Therefore, for this curve and the values of q mentioned above, there is no way of applying a homography appropriately to ϕ or of finding ψ with higher splitting probability.

Example 3. Let $\mathcal{C} : y^2 - x^{q-1} + 1 = 0$, which is a hyperelliptic, non-supersingular curve of genus $g = (q-3)/2$. In fact, it is a CM-curve and has $2q-2$ automorphisms. We look for a polynomial function $\phi \in \mathbb{F}_q(\mathcal{C})$ such that $\mathrm{Res}_y(\phi, y^2 - x^{q-1} + 1) = x^q - x$ or a small multiple thereof. Since the curve equation is quadratic in y, the function must be of shape $\phi(x, y) = \phi_1(x)y + \phi_0(x)$. Furthermore, the degrees of ϕ_0 and ϕ_1 should be very small. If we allow degree at most one, then by solving

$$\mathrm{Res}_y(\phi_1(x)y + \phi_0(x), y^2 - x^{q-1} + 1) = (x^q - x)(x + a)$$

for some a, we get $a = 0, \phi_0 = x, \phi_1 = 0$ and therefore $\phi(x, y) = xy$. This is not interesting, as ϕ is not irreducible.

If we allow ϕ_0, ϕ_1 of degree up to two, then with the same reasoning, we get $\phi(x, y) = (x^2 + ax)y$ for any choice of a. Again, this is not interesting as it is not irreducible.

Example 4. Let $\mathcal{C} : y^2 - x^{q-2} + x + 1 = 0$, which is a hyperelliptic, non-supersingular curve of genus $g = (q-3)/2$. As in Example 3, we try to find a function $\phi = \phi_1(x)y + \phi_0(x)$ such that $\mathrm{Res}(\phi, y^2 - x^{q-2} + x + 1)$ is a multiple of $x^q - x$ by some linear factors. Trying to solve the equation, we find that neither linear nor quadratic ϕ_0, ϕ_1 produce a resultant which is divisible by $x^q - x$. The same is true for $\mathcal{C} : y^2 - x^{q-1} - 1$ and many other curves.

Example 5. Let $\mathcal{C} : y^2 + y = x^n$ be a Koblitz–Buhler curve (see [5]), which is a hyperelliptic curve. Its equation can be transformed into $y^2 = x^n + 3/4$ via the transformation $y \mapsto y - 1/2$ in odd characteristic. For this curve, which functions split over $\mathbb{F}_q[x]$ depends on q. For some values of q (e.g. $q = 7$), we can show that there are no ϕ_0, ϕ_1 of degree 2 such that $\mathrm{Res}_y(\phi_1(x)y + \phi_0(x), y^2 + y = x^n) = x^q - x$ by solving the corresponding polynomial system with coefficients of the ϕ_i as indeterminates.

Example 6. Consider the Hermitian curve $\mathcal{C} : x^{q+1} + y^{q+1} - 1 = 0$. These curves have genus $g = q(q-1)/2$ and are known for the fact that they have many $\mathbb{F}_{q^2}$-rational points, more precisely, exactly $q^3 + 1$. In other words, the function field $\mathbb{F}_{q^2}(\mathcal{C})$ is maximal (in the sense that the upper Hasse–Weil bound on the number of places of degree one is attained). For this reason, such curves are of interest e.g. in coding theory. It is also known that Hermitian curves are supersingular, and therefore not interesting in the context of cryptography. For more on Hermitian curves, see e.g. [21, Chapter VI, VII].

Now let $\phi = xy + ay + bx^2 + cx + d$ with $a, b, c, d \in \mathbb{F}_{q^2}$. Experiments suggest that this function has higher-than-expected splitting probability. More precisely, for fields small enough so that we can enumerate all functions of this shape over $\mathbb{F}_{q^2}$, more than q^6 out of q^8 possible such functions ϕ split into factors of degree at most two. This is much more than expected for a random function of the same degree (here, the splitting probability behaves like q^{-q} for $q \to \infty$). However, experiments also show that special things are happening here: If the resultant is 2-smooth, then it is always automatically already 1-smooth (i.e. we did not find functions that produced a resultant which was 2-smooth but not 1-smooth).

Example 7. Let $\mathcal{C} : y^{q+1} + x^{q+1}y - 1 = 0$, which is a non-hyperelliptic, non-supersingular curve of genus $g = q(q+1)/2$. Experiments for $q = 5, 7, 11, 13, 17, 19, 23$ show that there is no irreducible function $\phi = xy + ay + bx + c \in \mathbb{F}_{q^2}[x, y]$ such that $\mathrm{Res}_y(\phi, y^{q+1} + x^{q+1}y - 1)$ splits completely into linear factors.

7 A possible obstruction

A possible conclusion from the experiments described in Section 6 is that curves for which principal divisors with higher-than-expected splitting probability exist are very special, and in fact, so special that the DLP is already known to be easy for these curves. A more precise formulation and possible answer to Question 1 is the following.

Conjecture 1. *Let $(\mathcal{C}_q)_q$ be a family of projective curves defined over $\mathbb{F}_q$ of genus* $\mathrm{poly}(q)$*, given by an equation of degree* $\mathrm{poly}(q)$*. Assume that there exist* $\mathrm{poly}(q)$ *monic, irreducible polynomial functions $\phi \in \mathbb{F}_{q^2}(\mathcal{C}_q)$ of degree $O(1)$ such that* $\mathrm{div}(\phi)$ *is $O(1)$-smooth. Then* $\mathrm{Jac}_{\mathcal{C}_q}$ *is isogenous to a product of elliptic curves, for large enough q.*

The main example supporting this conjecture are Hermitian curves, which are supersingular and therefore, in particular, have a Jacobian isogenous to a product of supersingular elliptic curves.

It is a priori not clear how to prove such a statement, since one would have to relate splitting probabilities of rational functions to the structure of the Jacobian of a curve. However, there are some results related to the special case of Question 1 where we set both constants equal to one:

Question 2. *Let $\mathcal{C}$ be a projective curve defined over $\mathbb{F}_q$ of degree* $\mathrm{poly}(q)$*. Under which conditions is the number of lines defining divisors which split completely unexpectedly large, say of the form* $\mathrm{poly}(q)$*?*

Heuristically, for a curve of degree d, we expect about $q^2/d!$ such divisors to split completely. Moreover, for reflexive curves of fixed degree $d \geq 3$, it has been proven by Diem that this count holds asymptotically:

Theorem 3 ([6, Theorem 3])**.** *Let $d \geq 3$ be fixed, and let $\mathcal{C}$ be a projective curve of genus at least one given by a plane model of degree d. If $d > 4$, then assume that the plane model is reflexive. Then the number of divisors on $\mathcal{C}$ that are given by lines in $\mathbb{P}^2$ and split completely into distinct points is in $\frac{1}{d!}q^2 + O(q^{3/2})$.*

For example, Hermitian curves are non-reflexive, and as explained in Example 6, they have many points. This corresponds to the fact that there are many divisors given by lines, and therefore also many that split.

Since we are interested in curves where there are rational functions with higher-than-expected splitting probability, this rules out reflexive curves as possible candidates. This leaves non-reflexive curves, which are relatively well-studied in the literature, see e.g. [13, 14].

A projective variety V is called *reflexive* if the conormal variety of V and its dual variety V' are equal up to the canonical identification of projective space with its double dual. Varieties defined over characteristic zero base fields are always reflexive, and even over positive characteristic fields, most varieties are reflexive. The main examples for non-reflexive varieties are strange curves, which have the property that all tangents pass through a common point.

In fact, non-reflexive curves are so special that they have been classified according to their degree by Ballico and Hefez [14]. Among other things, they show that for non-reflexive plane curves $\mathcal{C}$ of degree d with “moderate singularities” defined over fields of characteristic $p > 2$, we always have $p \mid d - 1$. Moreover, for such curves, they give the following result.

Theorem 4 ([14, Proposition 1]). *Any non-reflexive plane curve of degree $d = q+1 \geq 4$, for q a power of the characteristic of the field of definition, and of genus greater than zero is projectively equivalent to the curve*

$$x^{q+1} + y^{q+1} + z^{q+1} = 0. \tag{1}$$

These are the Hermitian curves, which are known to be supersingular. As discussed earlier, such curves are not interesting in our context. Summarizing, because of Diem’s theorem, the only curves which are candidates in response to Question 2 are non-reflexive curves, but such curves are not interesting in our context. This is in accordance with our experiments, in which we found only curves of shape (1).

The question that remains is whether a more general statement, perhaps along the lines of our conjecture, can be proven in response to the more general Question 1. One possible approach to this would be to use Cebotarev’s density theorem, which is the main argument in the proof of Diem’s theorem.

8 Conclusion

Our investigation of possible ways of adapting the relation collection phase of the new $L(1/4)$ and quasi-polynomial algorithms to algebraic curves suggests that this is a difficult problem. In particular, our experiments indicate that curves which permit rational functions whose principal divisors have higher-than-expected splitting probability are very special and therefore not interesting in our context, since the DLP in these curves is already known to be weak. We conjecture that this is always true and show that a special case of this conjecture can be proven using known results of Diem and Ballico–Hefez. If the more general statement could be proven, this would rule out a more general approach that allows polynomial functions of constant degree and the corresponding divisors to split into factors of constant degree. Clearly, this would still not mean that there is no $L(1/4)$ DLP algorithm for Jacobians of curves, it would mean simply that one needs to investigate more sophisticated ideas than the rather obvious analogy we have considered here.

References

[1] L. M. Adleman. The function field sieve. In *Algorithmic Number Theory (ANTS I)*, volume 877 of *LNCS*, pages 108–121. Springer, 1994.

[2] A. W. Bluher. On $x^{q+1} + ax + b$. *Finite Fields Appl.*, 10:285–305, 2004.

[3] R. Bărbulescu. *Algorithmes de logarithmes discrets dans les corps finis.* PhD thesis, Université de Lorraine, Available at `http://tel.archives-ouvertes.fr/tel-00925228/file/these_avec_resume.pdf`, 2013.

[4] R. Bărbulescu, P. Gaudry, A. Joux, and E. Thomé. A heuristic quasi-polynomial algorithm for discrete logarithm in finite fields of small characteristic. In P. Q. Nguyen and E. Oswald,

editors, *Advances in Cryptology: Proceedings of EUROCRYPT '14*, volume 8441 of *LNCS*, pages 1–16. Springer, 2014.

[5] J. Buhler and N. Koblitz. Lattice basis reduction, Jacobi sums and hyperelliptic cryptosystems. *Bull. Aust. Math. Soc.*, 58:147–154, 1998.

[6] C. Diem. On the discrete logarithm problem for plane curves. *J. Théor. Nombres Bordeaux*, 24:639–667, 2012.

[7] A. Enge and P. Gaudry. An $L(1/3+\varepsilon)$ algorithm for the discrete logarithm problem for low degree curves. In M. Naor, editor, *Advances in Cryptology: Proceedings of EUROCRYPT '07*, volume 4515 of *LNCS*, pages 379–393. Springer, 2007.

[8] A. Enge, P. Gaudry, and E. Thomé. An $L(1/3)$ discrete logarithm algorithm for low degree curves. *J. Cryptology*, 24:24–41, 2011.

[9] F. Göloğlu, R. Granger, G. McGuire, and J. Zumbrägel. On the function field sieve and the impact of higher splitting probabilities. In R. Canetti and J. A. Garay, editors, *Advances in Cryptology: Proceedings of CRYPTO '13*, volume 8043 of *LNCS*, pages 109–128. Springer, 2013.

[10] F. Göloğlu, R. Granger, G. McGuire, and J. Zumbrägel. Solving a 6120-bit DLP on a desktop computer. In T. Lange, K. Lauter, and P. Lisoněk, editors, *Proceedings of SAC '13*, LNCS, pages 136–152. Springer, 2013.

[11] R. Granger, T. Kleinjung, and J. Zumbrägel. Breaking '128-bit secure' supersingular binary curves. In J. A. Garay and R. Gennaro, editors, *Advances in Cryptology: Proceedings of CRYPTO '14*, volume 8617 of *LNCS*, pages 126–145. Springer, 2014.

[12] R. Granger, T. Kleinjung, and J. Zumbrägel. On the powers of 2. Available at http://eprint.iacr.org/2014/300, 2014.

[13] A. Hefez. Non-reflexive curves. *Compos. Math.*, 69:3–35, 1988.

[14] A. Hefez and E. Ballico. Non-reflexive projective curves of low degree. *Manuscripta Math.*, 70:385–396, 1991.

[15] T. Helleseth and A. Kholosha. $x^{2^l}+x+a$ and related affine polynomials over $\mathrm{GF}(2^k)$. *Cryptogr. Commun.*, 2(1):85–109, 2010.

[16] F. Heß. Computing relations in divisor class groups of algebraic curves over finite fields. Available at http://www.staff.uni-oldenburg.de/florian.hess/publications/dlog.pdf, 2002.

[17] F. Heß. Computing Riemann–Roch spaces in algebraic function fields and related topics. *J. Symbolic Comput.*, 33:425–445, 2002.

[18] A. Joux. A new index calculus algorithm with complexity $L(1/4+o(1))$ in small characteristic. In T. Lange, K. Lauter, and P. Lisoněk, editors, *Proceedings of SAC '13*, LNCS, pages 355–379. Springer, 2013.

[19] A. Joux and R. Lercier. The function field sieve in the medium prime case. In S. Vaudenay, editor, *Advances in Cryptology: Proceedings of EUROCRYPT '06*, volume 4004 of *LNCS*, pages 254–270. Springer, 2006.

[20] A. M. Odlyzko. Discrete logarithms in finite fields and their cryptographic significance. In T. Beth, N. Cot, and I. Ingemarsson, editors, *Advances in Cryptology: Proceedings of EUROCRYPT '84*, volume 209 of *LNCS*, pages 224–314. Springer, 1984.

[21] H. Stichtenoth. *Algebraic Function Fields and Codes.* Springer, Berlin–Heidelberg–New York, 1993.