HAL
archives-ouvertes.fr

A heuristic quasi-polynomial algorithm for discrete logarithm in finite fields of small characteristic

Razvan Barbulescu, Pierrick Gaudry, Antoine Joux, Emmanuel Thomé

▶ **To cite this version:**

Razvan Barbulescu, Pierrick Gaudry, Antoine Joux, Emmanuel Thomé. A heuristic quasi-polynomial algorithm for discrete logarithm in finite fields of small characteristic. Eurocrypt 2014, May 2014, Copenhagen, Denmark. pp.1-16, 10.1007/978-3-642-55220-5_1 . hal-00835446v2

HAL Id: hal-00835446
https://hal.inria.fr/hal-00835446v2

Submitted on 25 Nov 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A quasi-polynomial algorithm for discrete logarithm in finite fields of small characteristic

Razvan Barbulescu[1], Pierrick Gaudry[1], Antoine Joux[2,3], and Emmanuel Thomé[1]

[1] Inria, CNRS, University of Lorraine, France
[2] Cryptology Chair, Foundation UPMC – LIP 6, CNRS UMR 7606, Paris, France
[3] CryptoExperts, Paris, France

Abstract The difficulty of computing discrete logarithms in fields $\mathbb{F}_{q^k}$ depends on the relative sizes of k and q. Until recently all the cases had a sub-exponential complexity of type $L(1/3)$, similar to the factorization problem. In 2013, Joux designed a new algorithm with a complexity of $L(1/4+\epsilon)$ in small characteristic. In the same spirit, we propose in this article another heuristic algorithm that provides a quasi-polynomial complexity when q is of size at most comparable with k. By quasi-polynomial, we mean a runtime of $n^{O(\log n)}$ where n is the bit-size of the input. For larger values of q that stay below the limit $L_{q^k}(1/3)$, our algorithm loses its quasi-polynomial nature, but still surpasses the Function Field Sieve.

1 Introduction

The discrete logarithm problem (DLP) was first proposed as a hard problem in cryptography in the seminal article of Diffie and Hellman [7]. Since then, together with factorization, it has become one of the two major pillars of public key cryptography. As a consequence, the problem of computing discrete logarithms has attracted a lot of attention. From an exponential algorithm in 1976, the fastest DLP algorithms have been greatly improved during the past 35 years. A first major progress was the realization that the DLP in finite fields can be solved in subexponential time, i.e. $L(1/2)$ where $L_N(\alpha) = \exp\left(O((\log N)^\alpha (\log\log N)^{1-\alpha})\right)$. The next step further reduced this to a heuristic $L(1/3)$ running time in the full range of finite fields, from fixed characteristic finite fields to prime fields [2,6,11,3,17,18].

Recently, practical and theoretical advances have been made [15,10,16] with an emphasis on small to medium characteristic finite fields and composite degree extensions. The most general and efficient algorithm [16] gives a complexity of $L(1/4+o(1))$ when the characteristic is smaller than the square root of the extension degree. Among the ingredients of this approach, we find the use of a very particular representation of the finite field; the use of the so-called *systematic*

equation[1]; and the use of algebraic resolution of bilinear polynomial systems in the individual logarithm phase.

In this work, we present a new discrete logarithm algorithm, in the same vein as in [16] that uses an asymptotically more efficient descent approach. The main result gives a *quasi-polynomial* heuristic complexity for the DLP in finite fields of small characteristic. By quasi-polynomial, we mean a complexity of type $n^{O(\log n)}$ where n is the bit-size of the cardinality of the finite field. Such a complexity is smaller than any $L(\epsilon)$ for $\epsilon > 0$. It remains super-polynomial in the size of the input, but offers a major asymptotic improvement compared to $L(1/4 + o(1))$.

The key features of our algorithm are the following.

- We keep the field representation and the systematic equations of [16].
- The algorithmic building blocks are elementary. In particular, we avoid the use of Gröbner basis algorithms.
- The complexity result relies on three key heuristics: the existence of a polynomial representation of the appropriate form; the fact that the smoothness probabilities of some non-uniformly distributed polynomials are similar to the probabilities for uniformly random polynomials of the same degree; and the linear independence of some finite field elements related to the action of $\mathrm{PGL}_2(\mathbb{F}_q)$.

The heuristics are very close to the ones used in [16]. In addition to the arguments in favor of these heuristics already given in [16], we performed some experiments to validate them on practical instances.

Although we insist on the case of finite fields of small characteristic, where quasi-polynomial complexity is obtained, our new algorithm improves the complexity of discrete logarithm computations in a much larger range of finite fields.

More precisely, in finite fields of the form $\mathbb{F}_{q^k}$, where q grows as $L_{q^k}(\alpha)$, the complexity becomes $L_{q^k}(\alpha + o(1))$. As a consequence, our algorithm is asymptotically faster than the Function Field Sieve algorithm in almost all the range previously covered by this algorithm. Whenever $\alpha < 1/3$, our new algorithm offers the smallest complexity. For the limiting case $L(1/3, c)$, the Function Field Sieve remains more efficient for small values of c, and the Number Field Sieve is better for large values of c (see [18]).

This article is organized as follows. In Section 2, we state the main result, and discuss how it can be used to design a complete discrete logarithm algorithm. In Section 3, we analyze how this result can be interpreted for various types of finite fields, including the important case of fields of small characteristic. Section 4 is devoted to the description of our new algorithm. It relies on heuristics that are discussed in Section 5, from a theoretical and a practical point of view. Before getting to the conclusion, in Section 6, we propose a few variants of the algorithm.

[1] While the terminology is similar, no parallel is to be made with the systematic equations as defined in early works related to the computation discrete logarithms in $\mathbb{F}_{2^n}$, as [4].

2 Main result

We start by describing the setting in which our algorithm applies. It is basically the same as in [16]: we need a large enough subfield, and we assume that a sparse representation can be found. This is formalized in the following definition.

Definition 1 *A finite field K admits a* sparse medium subfield representation *if*

- *it has a subfield of q^2 elements for a prime power q, i.e. K is isomorphic to $\mathbb{F}_{q^{2k}}$ with $k \geq 1$;*
- *there exist two polynomials h_0 and h_1 over $\mathbb{F}_{q^2}$ of small degree, such that $h_1 X^q - h_0$ has a degree k irreducible factor.*

In what follows, we will assume that all the fields under consideration admit a sparse medium subfield representation. Furthermore, we assume that the degrees of the polynomials h_0 and h_1 are uniformly bounded by a constant δ. Later, we will provide heuristic arguments for the fact that any finite field of the form $\mathbb{F}_{q^{2k}}$ with $k \leq q+2$ admits a sparse medium subfield representation with polynomials h_0 and h_1 of degree at most 2. But in fact, for our result to hold, allowing the degrees of h_0 and h_1 to be bounded by any constant δ independent of q and k or even allowing δ to grow slower than $O(\log q)$ would be sufficient.

In a field in sparse medium subfield representation, elements will always be represented as polynomials of degree less than k with coefficients in $\mathbb{F}_{q^2}$. When we talk about the discrete logarithm of such an element, we implicitly assume that a basis for this discrete logarithm has been chosen, and that we work in a subgroup whose order has no small irreducible factor (we refer to the Pohlig-Hellman algorithm [20] to limit ourselves to this case).

Proposition 2 *Let $K = \mathbb{F}_{q^{2k}}$ be a finite field that admits a sparse medium subfield representation. Under the heuristics explained below, there exists an algorithm whose complexity is polynomial in q and k and which can be used for the following two tasks.*

1. *Given an element of K represented by a polynomial $P \in \mathbb{F}_{q^2}[X]$ with $2 \leq \deg P \leq k-1$, the algorithm returns an expression of $\log P(X)$ as a linear combination of at most $O(kq^2)$ logarithms $\log P_i(X)$ with $\deg P_i \leq \lceil \frac{1}{2} \deg P \rceil$ and of $\log h_1(X)$.*
2. *The algorithm returns the logarithm of $h_1(X)$ and the logarithms of all the elements of K of the form $X + a$, for a in $\mathbb{F}_{q^2}$.*

Before the presentation of the algorithm, which is made in Section 4, we explain how to use it as a building block for a complete discrete logarithm algorithm.

Let $P(X)$ be an element of K for which we want to compute the discrete logarithm. Here P is a polynomial of degree at most $k-1$ and with coefficients

in $\mathbb{F}_{q^2}$. We start by applying the algorithm of Proposition 2 to P. We obtain a relation of the form

$$\log P = e_0 \log h_1 + \sum e_i \log P_i,$$

where the sum has at most $\kappa q^2 k$ terms for a constant κ and the P_i's have degree at most $\lceil \frac{1}{2} \deg P \rceil$. Then, we apply recursively the algorithm to the P_i's, thus creating a descent procedure where at each step, a given element P is expressed as a product of elements, whose degree is at most half the degree of P (rounded up) and the arity of the descent tree is in $O(q^2 k)$.

At the end of the process, the logarithm of P is expressed as a linear combination of the logarithms of h_1 and of the linear polynomials, for which the logarithms are computed with the algorithm in Proposition 2 in its second form.

We are left with the complexity analysis of the descent process. Each internal node of the descent tree corresponds to one application of the algorithm of Proposition 2, therefore each internal node has a cost which is bounded by a polynomial in q and k. The total cost of the descent is therefore bounded by the number of nodes in the descent tree times a polynomial in q and k. The depth of the descent tree is in $O(\log k)$. The number of nodes of the tree is then less than or equal to its arity raised to the power of its depth, which is $(q^2 k)^{O(\log k)}$. Since any polynomial in q and k is absorbed in the $O()$ notation in the exponent, we obtain the following result.

Theorem 3 *Let $K = \mathbb{F}_{q^{2k}}$ be a finite field that admits a sparse medium subfield representation. Assuming the same heuristics as in Proposition 2, any discrete logarithm in K can be computed in a time bounded by*

$$\max(q, k)^{O(\log k)}.$$

3 Consequences for various ranges of parameters

We now discuss the implications of Theorem 3 depending on the properties of the finite field $\mathbb{F}_Q$ where we want to compute discrete logarithms in the first place. The complexities will be expressed in terms of $\log Q$, which is the size of the input.

Three cases are considered. In the first one, the finite field admits a sparse medium subfield representation, where q and k are almost equal. This is the optimal case. Then we consider the case where the finite field has small (maybe constant) characteristic. And finally, we consider the case where the characteristic is getting larger so that the only available subfield is a bit too large for the algorithm to have an optimal complexity.

In the following, we always assume that for any field of the form $\mathbb{F}_{q^{2k}}$, we can find a sparse medium subfield representation.

3.1 Case where the field is $\mathbb{F}_{q^{2k}}$, with $q \approx k$

The finite fields $\mathbb{F}_Q = \mathbb{F}_{q^{2k}}$ for which q and k are almost equal are tailored for our algorithm. In that case, the complexity of Theorem 3 becomes $q^{O(\log q)}$. Since $Q \approx q^{2q}$, we have $q = (\log Q)^{O(1)}$. This gives an expression of the form $2^{O\left((\log\log Q)^2\right)}$, which is sometimes called quasi-polynomial in complexity theory.

Corollary 4 *For finite fields of cardinality $Q = q^{2k}$ with $q + O(1) \geq k$ and $q = (\log Q)^{O(1)}$, there exists a heuristic algorithm for computing discrete logarithms in quasi-polynomial time*

$$2^{O\left((\log\log Q)^2\right)}.$$

We mention a few cases which are almost directly covered by Corollary 4. First, we consider the case where $Q = p^n$ with p a prime bounded by $(\log Q)^{O(1)}$, and yet large enough so that $n \leq (p + \delta)$. In this case $\mathbb{F}_Q$, or possibly $\mathbb{F}_{Q^2}$ if n is odd, can be represented in such a way that Corollary 4 applies.

Much the same can be said in the case where n is composite and factors nicely, so that $\mathbb{F}_Q$ admits a large enough subfield $\mathbb{F}_q$ with $q = p^m$. This can be used to solve certain discrete logarithms in, say, $\mathbb{F}_{2^n}$ for adequately chosen n (much similar to records tackled by [12,8,13,9,14]).

3.2 Case where the characteristic is polynomial in the input size

Let now $\mathbb{F}_Q$ be a finite field whose characteristic p is bounded by $(\log Q)^{O(1)}$, and let $n = \log Q / \log p$, so that $Q = p^n$. While we have seen that Corollary 4 can be used to treat some cases, its applicability might be hindered by the absence of an appropriately sized subfield: p might be as small as 2, and n might not factor adequately. In those cases, we use the same strategy as in [16] and embed the discrete logarithm problem in $\mathbb{F}_Q$ into a discrete logarithm problem in a larger field.

Let k be n if n is odd and $n/2$ if n is even. Then, we set $q = p^{\lceil \log_p k \rceil}$, and we work in the field $\mathbb{F}_{q^{2k}}$. By construction this field contains $\mathbb{F}_Q$ (because $p|q$ and $n|2k$) and it is in the range of applicability of Theorem 3. Therefore, one can solve a discrete logarithm problem in $\mathbb{F}_Q$ in time $\max(q,k)^{O(\log k)}$. Rewriting this complexity in terms of Q, we get $\log_p(Q)^{O(\log\log Q)}$. And finally, we get a similar complexity result as in the previous case. Of course, since we had to embed in a larger field, the constant hidden in the $O()$ is larger than for Corollary 4.

Corollary 5 *For finite fields of cardinality Q and characteristic bounded by $\log(Q)^{O(1)}$, there exists a heuristic algorithm for computing discrete logarithms in quasi-polynomial time*

$$2^{O\left((\log\log Q)^2\right)}.$$

We emphasize that the case $\mathbb{F}_{2^n}$ for a prime n corresponds to this case. A direct consequence of Corollary 5 is that discrete logarithms in $\mathbb{F}_{2^n}$ can be computed in quasi-polynomial time $2^{O((\log n)^2)}$.

3.3 Case where $q = L_{q^{2k}}(\alpha)$

If the characteristic of the base field is not so small compared to the extension degree, the complexity of our algorithm does not keep its nice quasi-polynomial form. However, in almost the whole range of applicability of the Function Field Sieve algorithm, our algorithm is asymptotically better than FFS.

We consider here finite fields that can be put into the form $\mathbb{F}_Q = \mathbb{F}_{q^{2k}}$, where q grows not faster than an expression of the form $L_Q(\alpha)$. In the following, we assume that there is equality, which is of course the worst case. The condition can then be rewritten as $\log q = O((\log Q)^\alpha (\log\log Q)^{1-\alpha})$ and therefore $k = \log Q/\log q = O((\log Q/\log\log Q)^{1-\alpha})$. In particular we have $k \leq q + \delta$, so that Theorem 3 can be applied and gives a complexity of $q^{O(\log k)}$. This yields the following result.

Corollary 6 *For finite fields of the form $\mathbb{F}_Q = \mathbb{F}_{q^{2k}}$ where q is bounded by $L_Q(\alpha)$, there exists a heuristic algorithm for computing discrete logarithms in subexponential time*

$$L_Q(\alpha)^{O(\log\log Q)}.$$

This complexity is smaller than $L_Q(\alpha')$ for any $\alpha' > \alpha$. Hence, for any $\alpha < 1/3$, our algorithm is faster than the best previously known algorithm, namely FFS and its variants.

4 Main algorithm: proof of Proposition 2

The algorithm is essentially the same for proving the two points of Proposition 2. The strategy is to find relations between the given polynomial $P(X)$ and its translates by a constant in $\mathbb{F}_{q^2}$. Let D be the degree of $P(X)$, that we assume to be at least 1 and at most $k-1$.

The key to find relations is the *systematic equation*:

$$X^q - X = \prod_{a\in\mathbb{F}_q} (X - a). \tag{1}$$

We like to view Equation (1) as involving the projective line $\mathbb{P}^1(\mathbb{F}_q)$. Let $\mathcal{S} = \{(\alpha,\beta)\}$ be a set of representatives of the $q+1$ points $(\alpha:\beta) \in \mathbb{P}^1(\mathbb{F}_q)$, chosen adequately so that the following equality holds.

$$X^qY - XY^q = \prod_{(\alpha,\beta)\in\mathcal{S}} (\beta X - \alpha Y). \tag{2}$$

To make translates of $P(X)$ appear, we consider the action of *homographies*. Any matrix $m = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$ acts on $P(X)$ with the following formula:

$$m \cdot P = \frac{aP + b}{cP + d}.$$

In the following, this action will become trivial if the matrix m has entries that are defined over $\mathbb{F}_q$. This is also the case if m is non-invertible. Finally, it is clear that multiplying all the entries of m by a non-zero constant does not change its action on $P(X)$. Therefore the matrices of the homographies that we consider are going to be taken in the following set of cosets:

$$\mathcal{P}_q = \mathrm{PGL}(\mathbb{F}_{q^2})/\,\mathrm{PGL}(\mathbb{F}_q).$$

(Note that in general $\mathrm{PGL}_2(\mathbb{F}_q)$ is not a normal subgroup of $\mathrm{PGL}_2(\mathbb{F}_{q^2})$, so that $\mathcal{P}_q$ is not a quotient group.)

To each element $m = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \in \mathcal{P}_q$, we associate the equation (E_m) obtained by substituting $aP + b$ and $cP + d$ in place of X and Y in Equation (2).

$$\begin{aligned}(aP+b)^q(cP+d) - (aP+b)(cP+d)^q &= \prod_{(\alpha,\beta)\in\mathcal{S}} \beta(aP+b) - \alpha(cP+d) \qquad (E_m)\\ &= \prod_{(\alpha,\beta)\in\mathcal{S}} (-c\alpha + a\beta)P - (d\alpha - b\beta)\\ &= \lambda \prod_{(\alpha,\beta)\in\mathcal{S}} P - \mathrm{x}(m^{-1}\cdot(\alpha:\beta)).\end{aligned}$$

This sequence of formulae calls for a short comment because of an abuse of notation in the last expression. First, λ is the constant in $\mathbb{F}_{q^2}$ which makes the leading terms of the two sides match. Then, the term $P - \mathrm{x}(m^{-1}\cdot(\alpha:\beta))$ denotes $P - u$ when $m^{-1}\cdot(\alpha:\beta) = (u:1)$ (whence we have $u = \frac{d\alpha - b\beta}{-c\alpha + a\beta}$), or 1 if $m^{-1}\cdot(\alpha:\beta) = \infty$. The latter may occur since when a/c is in $\mathbb{F}_q$, the expression $-c\alpha + a\beta$ vanishes for a point $(\alpha:\beta) \in \mathbb{P}^1(\mathbb{F}_q)$ so that one of the factors of the product contains no term in $P(X)$.

Hence the right-hand side of Equation (E_m) is, up to a multiplicative constant, a product of $q+1$ or q translates of the target $P(X)$ by elements of $\mathbb{F}_{q^2}$. The equation obtained is actually related to the set of points $m^{-1}\cdot\mathbb{P}^1(\mathbb{F}_q) \subset \mathbb{P}^1(\mathbb{F}_{q^2})$.

The polynomial on the left-hand side of (E_m) can be rewritten as a smaller degree equivalent. For this, we use the special form of the defining polynomial: in K we have $X^q \equiv \frac{h_0(X)}{h_1(X)}$. Let us denote by $\tilde{a}$ the element a^q when a is any element of $\mathbb{F}_{q^2}$. Furthermore, we write $\tilde{P}(X)$ the polynomial $P(X)$ with all its coefficients raised to the power q. The left-hand side of (E_m) is

$$(\tilde{a}\tilde{P}(X^q) + \tilde{b})(cP(X)+d) - (aP(X)+b)(\tilde{c}\tilde{P}(X^q) + \tilde{d}),$$

and using the defining equation for the field K, it is congruent to

$$\mathcal{L}_m := \left(\tilde{a}\tilde{P}\left(\frac{h_0(X)}{h_1(X)}\right) + \tilde{b}\right)(cP(X)+d) - (aP(X)+b)\left(\tilde{c}\tilde{P}\left(\frac{h_0(X)}{h_1(X)}\right) + \tilde{d}\right).$$

The denominator of $\mathcal{L}_m$ is a power of h_1 and its numerator has degree at most $(1+\delta)D$ where $\delta = \max(\deg h_0, \deg h_1)$. We say that $m \in \mathcal{P}_q$ yields a relation if this numerator of $\mathcal{L}_m$ is $\lceil D/2 \rceil$-smooth.

To any $m \in \mathcal{P}_q$, we associate a row vector $v(m)$ of dimension q^2+1 in the following way. Coordinates are indexed by $\mu \in \mathbb{P}^1(\mathbb{F}_{q^2})$, and the value associated to $\mu \in \mathbb{F}_{q^2}$ is 1 or 0 depending on whether $P - \mathrm{x}(\mu)$ appears in the right-hand side of Equation (E_m). Note that exactly $q+1$ coordinates are 1 for each m. Equivalently, we may write

$$v(m)_{\mu \in \mathbb{P}^1(\mathbb{F}_{q^2})} = \begin{cases} 1 \text{ if } \mu = m^{-1} \cdot (\alpha : \beta) \text{ with } (\alpha : \beta) \in \mathbb{P}^1(\mathbb{F}_q), \\ 0 \text{ otherwise.} \end{cases} \tag{3}$$

We associate to the polynomial P a matrix $H(P)$ whose rows are the vectors $v(m)$ for which m yields a relation, taking at most one matrix m in each coset of $\mathcal{P}_q$. The validity of Proposition 2 crucially relies on the following heuristic.

Heuristic 7 *For any $P(X)$, the set of rows $v(m)$ for cosets $m \in \mathcal{P}_q$ that yield a relation form a matrix which has full rank q^2+1.*

As we will note in Section 5, the matrix $H(P)$ is heuristically expected to have $\Theta(q^3)$ rows, where the implicit constant depends on δ. This means that for our decomposition procedure to work, we rely on the fact that q is large enough (otherwise $H(P)$ may have less than q^2+1 rows, which precludes the possibility that it have rank q^2+1).

The first point of Proposition 2, where we descend a polynomial $P(X)$ of degree D at least 2, follows by linear algebra on this matrix. Since we assume that the matrix has full rank, then the vector $(\ldots, 0, 1, 0, \ldots)$ with 1 corresponding to $P(X)$ can be written as a linear combination of the rows. When doing this linear combination on the equations (E_m) corresponding to P we write $\log P(X)$ as a linear combination of $\log P_i$ where $P_i(x)$ are the elements occurring in the left-hand sides of the equations. Since there are $O(q^2)$ columns, the elimination process involves at most $O(q^2)$ rows, and since each row corresponds to an equation (E_m), it involves at most $\deg \mathcal{L}_m \leq (1+\delta)D$ polynomials in the left-hand-side[2]. In total, the polynomial D is expressed by a linear combination of at most $O(q^2 D)$ polynomials of degree less than $\lceil D/2 \rceil$. The logarithm of $h_1(X)$ is also involved, as a denominator of $\mathcal{L}_m$. We have not made precise the constant in $\mathbb{F}_{q^2}^*$ which occurs to take care of the leading coefficients. Since discrete logarithms in $\mathbb{F}_{q^2}^*$ can certainly be computed in polynomial time in q, this is not a problem.

Since the order of $\mathrm{PGL}_2(\mathbb{F}_{q^i})$ is $q^{3i} - q^i$, the set of cosets $\mathcal{P}_q$ has $q^3 + q$ elements. For each $m \in \mathcal{P}_q$, testing whether (E_m) yields a relation amounts to some polynomial manipulations and a smoothness test. All of them can be done

[2] This estimate of the number of irreducible factors is a pessimistic upper bound. In practice, one expects to have only $O(\log D)$ factors on average. Since the crude estimate does not change the overall complexity, we keep it that way to avoid adding another heuristic.

in polynomial time in q and the degree of $P(X)$ which is bounded by k. Finally, the linear algebra step can be done in $O(q^{2\omega})$ using asymptotically fast matrix multiplication algorithms, or alternatively $O(q^5)$ operations using sparse matrix techniques. Indeed, we have $q+1$ non-zero entries per row and a size of q^2+1. Therefore, the overall cost is polynomial in q and k as claimed.

For the second part of Proposition 2 we replace P by X during the construction of the matrix. In that case, both sides of the equations (E_m) involve only linear polynomials. Hence we obtain a linear system whose unknowns are $\log(X+a)$ with $a \in \mathbb{F}_{q^2}$. Since Heuristic 7 would give us only the full rank of the system corresponding to the right-hand sides of the equations (E_m), we have to rely on a specific heuristic for this step:

Heuristic 8 *The linear system constructed from all the equations (E_m) for $P(X) = X$ has full rank.*

Assuming that this heuristic holds, we can solve the linear system and obtain the discrete logarithms of the linear polynomials and of $h_1(X)$.

5 Supporting the heuristic argument in the proof

For Heuristic 7, we propose two approaches to support this heuristic. Both allow to gain some confidence in the validity of the heuristic, but of course none affect the heuristic nature of this statement.

For the first line of justification, we denote by $\mathcal{H}$ the matrix of all the $\#\mathcal{P}_q = q^3+q$ vectors $v(m)$ defined as in Equation (3). Associated to a polynomial P, Section 4 defines the matrix $H(P)$ formed of the rows $v(m)$ such that the numerator of $\mathcal{L}_m$ is smooth. We will give heuristics that $H(P)$ has $\Theta(q^3)$ rows and then prove that $\mathcal{H}$ has rank q^2+1, which of course does not prove that its submatrix $H(P)$ has full rank.

In order to estimate the number of rows of $H(P)$ we assume that the numerator of $\mathcal{L}_m$ has the same probability to be $\lceil \frac{D}{2} \rceil$-smooth as a random polynomial of same degree. In this paragraph, we assume that the degrees of h_0 and h_1 are bounded by 2, merely to avoid awkward notations; the result holds for any constant bound δ. The degree of the numerator of $\mathcal{L}_m$ is then bounded by $3D$, so we have to estimate the probability that a polynomial in $\mathbb{F}_{q^2}[X]$ of degree $3D$ is $\lceil \frac{D}{2} \rceil$-smooth. For any prime power q and integers $1 \leq m \leq n$, we denote by $N_q(m,n)$ the number of m-smooth monic polynomials of degree n. Using analytic methods, Panario et al. gave a precise estimate of this quantity (Theorem 1 of [19]):

$$N_q(n,m) = q^n \rho\left(\frac{n}{m}\right)\left(1 + O\left(\frac{\log n}{m}\right)\right), \tag{4}$$

where ρ is Dickman's function defined as the unique continuous function such that $\rho(u) = 1$ on $[0,1]$ and $u\rho'(u) = \rho(u-1)$ for $u > 1$. We stress that the constant κ hidden in the $O()$ notation is independent of q. In our case, we are interested in the value of $N_{q^2}(3D, \lceil \frac{D}{2} \rceil)$. Let us call D_0 the least integer such

that $1 + \kappa\left(\frac{\log(3D)}{\lceil D/2 \rceil}\right)$ is at least $1/2$. For $D > D_0$, we will use the formula (4); and for $D \leq D_0$, we will use the crude estimate $N_q(n,m) \geq N_q(n,1) = q^n/n!$. Hence the smoothness probability of $\mathcal{L}_m$ is at least $\min\left(\frac{1}{2}\rho(6), 1/(3D_0)!\right)$.

More generally, if $\deg h_0$ and $\deg h_1$ are bounded by a constant δ then we have a smoothness probability of $\rho(2\delta+2)$ times an absolute constant. Since we have q^3+q candidates and a constant probability of success, $H(P)$ has $\Theta(q^3)$ rows.

Now, unless some theoretical obstruction occurs, we expect a matrix over $\mathbb{F}_\ell$ to have full rank with probability at least $1-\frac{1}{\ell}$. The matrix $\mathcal{H}$ is however peculiar, and does enjoy regularity properties which are worth noticing. For instance, we have the following proposition.

Proposition 9 *Let ℓ be a prime not dividing q^3-q. Then the matrix $\mathcal{H}$ over $\mathbb{F}_\ell$ has full rank q^2+1.*

Proof. We may obtain this result in two ways. First, $\mathcal{H}$ is the incidence matrix of a $3-(q^2+1, q+1, 1)$ combinatorial design called *inversive plane* (see e.g. [21, Theorem 9.27]). As such we obtain the identity

$$\mathcal{H}^T\mathcal{H} = (q+1)(J_{q^2+1} - (1-q)I_{q^2+1})$$

(see [21, Theorem 1.13 and Corollary 9.6]), where J_n is the $n \times n$ matrix with all entries equal to one, and I_n is the $n \times n$ identity matrix. This readily gives the result exactly as announced.

We also provide an elementary proof of the Proposition. We have a bijection between rows of $\mathcal{H}$ and the different possible image sets of the projective line $\mathbb{P}^1(\mathbb{F}_q)$ within $\mathbb{P}^1(\mathbb{F}_{q^2})$, under injections of the form $(\alpha:\beta) \mapsto m^{-1}\cdot(\alpha:\beta)$. All these q^3+q image sets have size $q+1$, and by symmetry all points of $\mathbb{P}^1(\mathbb{F}_{q^2})$ are reached equally often. Therefore, the sum of all rows of $\mathcal{H}$ is the vector whose coordinates are all equal to $\frac{1}{1+q^2}(q^3+q)(q+1) = q^2+q$.

Let us now consider the sum of the rows in $\mathcal{H}$ whose first coordinate is 1 (as we have just shown, we have q^2+q such rows). Those correspond to image sets of $\mathbb{P}^1(\mathbb{F}_q)$ which contain one particular point, say $(0:1)$. The value of the sum for any other coordinate indexed by e.g. $Q \in \mathbb{P}^1(\mathbb{F}_{q^2})$ is the number of image sets $m^{-1}\cdot\mathbb{P}^1(\mathbb{F}_q)$ which contain both $(0:1)$ and Q, which we prove is equal to $q+1$ as follows. Without loss of generality, we may assume $Q = \infty = (1:0)$. We need to count the relevant homographies $m^{-1} \in \mathrm{PGL}_2(\mathbb{F}_{q^2})$, modulo $\mathrm{PGL}_2(\mathbb{F}_q)$-equivalence $m \equiv hm$. By $\mathrm{PGL}_2(\mathbb{F}_q)$-equivalence, we may without loss of generality assume that m^{-1} fixes $(0:1)$ and $(1:0)$. Letting $m^{-1} = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$, we obtain $(b:d) = (0:1)$ and $(a:c) = (1:0)$, whence $b = c = 0$, and both $a, d \neq 0$. We may normalize to $d = 1$, and notice that multiplication of a by a scalar in $\mathbb{F}_q^*$ is absorbed in $\mathrm{PGL}_2(\mathbb{F}_q)$-equivalence. Therefore the number of suitable m is $\#\mathbb{F}_{q^2}^*/\mathbb{F}_q^* = q+1$.

These two facts show that the row span of $\mathcal{H}$ contains the vectors $(q^2+q, \ldots, q^2+q)$ and $(q^2+q, q+1, \ldots, q+1)$. The vector $(q^3-q, 0, \ldots, 0)$ is obtained

as a linear combination of these two vectors, which suffices to prove that $\mathcal{H}$ has full rank, since the same reasoning holds for any coordinate.

□

Proposition 9, while encouraging, is clearly not sufficient. We are, at the moment, unable to provide a proof of a more useful statement. On the experimental side, it is reasonably easy to sample arbitrary subsets of the rows of $\mathcal{H}$ and check for their rank. To this end, we propose the following experiment. We have considered small values of q in the range $[16, \ldots, 64]$, and made 50 random picks of subsets $S_i \subset \mathcal{P}_q$, all of size exactly $q^2 + 1$. For each we considered the matrix of the corresponding linear system, which is made of selected rows of the matrix $\mathcal{H}$, and computed its determinant δ_i. For all values of q considered, we have observed the following facts.

- First, all square matrices considered had full rank over $\mathbb{Z}$. Furthermore, their determinants had no common factor apart possibly from those appearing in the factorization of $q^3 - q$ as predicted by Proposition 9. In fact, experimentally it seems that only the factors of $q + 1$ are causing problems.
- We also explored the possibility that modulo some primes, the determinant could vanish with non-negligible probability. We thus computed the pairwise GCD of all 50 determinants computed, for each q. Again, the only prime factors appearing in the GCDs were either originating from the factorization of $q^3 - q$, or sporadically from the birthday paradox.

q	#trials	in $\gcd(\{\delta_i\})$	in $\gcd(\delta_i, \delta_j)$	q	#trials	in $\gcd(\{\delta_i\})$	in $\gcd(\delta_i, \delta_j)$
16	50	17	691	37	50	2, 19	2879
17	50	2, 3	431, 691	41	50	2, 3, 7	none above q^2
19	50	2, 5	none above q^2	43	50	2, 11	none above q^2
23	50	2, 3	none above q^2	47	50	2, 3	none above q^2
25	50	2, 13	none above q^2	49	50	2, 5	none above q^2
27	50	2, 7	1327	53	50	2, 3	none above q^2
29	50	2, 3, 5	none above q^2	59	50	2, 3, 5	none above q^2
31	50	2	1303, 3209	61	50	2, 31	none above q^2
32	50	3, 11	none above q^2	64	50	5, 13	none above q^2

Table1. Prime factors appearing in determinant of random square submatrices of $\mathcal{H}$ (for one given set of random trials)

These results are summarized in table 1, where the last column omits small prime factors below q^2. Of course, we remark that considering square submatrices is a more demanding check than what Heuristic 7 suggests, since our algorithm only needs a slightly larger matrix of size $\Theta(q^3) \times (q^2 + 1)$ to have full rank.

A second line of justification is more direct and natural, as it is possible to implement the algorithm outlined in Section 4, and verify that it does provide the desired result. A Magma implementation validates this claim, and has been

used to implement descent steps for an example field of degree 53 over $\mathbb{F}_{53^2}$. An example step in this context is given for applying our algorithm to a polynomial of degree 10, attempting to reduce it to polynomials of degree 6 or less. Among the 148,930 elements of $\mathcal{P}_q$, it sufficed to consider only 71,944 matrices m, of which about 3.9% led to relations, for a minimum sufficient number of relations equal to $q^2 + 1 = 2810$ (as more than half of the elements of $\mathcal{P}_q$ had not even been examined at this point, it is clear that getting more relations was easy—we did not have to). As the defining polynomial for the finite field considered was constructed with $\delta = \deg h_{0,1} = 1$, all left-hand sides involved had degree 20. The polynomials appearing in their factorizations had the following degrees (the number in brackets give the number of distinct polynomials found for each degree): 1(2098), 2(2652), 3(2552), 4(2463), 5(2546), 6(2683). Of course this tiny example size uses no optimization, and is only intended to check the validity of Proposition 2.

As for Heuristic 8, it is already present in [16] and [10], so this is not a new heuristic. Just like for Heuristic 7, it is based on the fact that the probability that a left-hand side is 1-smooth and yields a relation is constant. Therefore, we have a system with $\Theta(q^3)$ relations between $O(q^2)$ indeterminates, and it seems reasonable to expect that it has full rank. On the other hand, there is not as much algebraic structure in the linear system as in Heuristic 7, so that we see no way to support this heuristic apart from testing it on several inputs. This was already done (including for record computations) in [16] and [10], so we do not elaborate on our own experiments that confirm again that Heuristic 8 seems to be valid except for tiny values of q.

An obstruction to the heuristics. As noted by Cheng, Wan and Zhuang [5], the irreducible factors of $h_1 X^q - h_0$ other than the degree k factor that is used to define $\mathbb{F}_{q^{2k}}$ are problematic. Let P be such a problematic polynomial. The fact that it divides the defining equation implies that it also divides the $\mathcal{L}_m$ quantity that is involved when trying to build a relation that relates P to other polynomials. Therefore the first part of Proposition 2 can not hold for this P. Similarly, if P is linear, its presence will prevent the second part of Proposition 2 to hold since the logarithm of P can not be found with the technique of Section 4. We present here a technique to deal with the problematic polynomials. (The authors of [5] proposed another solution to keep the quasi-polynomial nature of algorithm.)

Proposition 10 *For each problematic polynomial P of degree D, we can find a linear relation between $\log P$, $\log h_1$ and $O(D)$ logarithms of polynomials of degree at most $(\delta - 1)D$ which are not problematic.*

Proof. Let P be an irreducible factor of $h_1 X^q - h_0$ of degree D. Let us consider P^q; by reducing modulo $h_1 X^q - h_0$ and clearing denominators, there exists a polynomial $A(X)$ such that

$$h_1^D P^q = h_1^D \tilde{P}\left(\frac{h_0}{h_1}\right) + (h_1 X^q - h_0) A(X). \tag{5}$$

Since P divides two of the terms of this equality, it must also divide the third one, namely the polynomial $\mathcal{R} = h_1^D \tilde{P}(h_0/h_1)$. Let $v_P \geq 1$ be the valuation of P in $\mathcal{R}$. In the finite field $\mathbb{F}_{q^{2k}}$ we obtain the following equalities between logarithms:

$$(q - v_P) \log P = -D \log h_1 + \sum_i e_i \log Q_i,$$

where Q_i are the irreducible factors of $\mathcal{R}$ other than P and e_i their valuation in $\mathcal{R}$. A polynomial Q_i can not be problematic. Otherwise, it would divide the right-hand side of Equation (5), and therefore, also the left-hand side, which is impossible. Since $v_P \leq \frac{\deg \mathcal{R}}{\deg P} \leq \delta < q$, the quantity $q - v_P$ is invertible modulo ℓ (we assume, as usual that ℓ is larger than q) and we obtain a relation between $\log P$, $\log h_1$ and the logarithms of the non-problematic polynomials Q_i. The degree of $\mathcal{R}/P^{v_P}$ is at most $(\delta - 1)D$, which gives the claimed bound on the degrees of the Q_i. □

If $\delta \leq 2$, this proposition solves the issues raised by [5] about problematic polynomials. Indeed, for each problematic polynomial of degree $D > 1$, it will be possible to rewrite its logarithm in terms of logarithms of non-problematic polynomials of at most the same degree that can be descended in the usual way. Similarly, each problematic polynomial of degree 1 can have its logarithm rewritten in terms of the logarithms of other non-problematic linear polynomials. Adding these relations to the ones obtained in Section 4, we expect to have a full-rank linear system.

If $\delta > 2$, we need to rely on the additional heuristic. Indeed, when descending the Q_i that have a degree potentially larger than the degree of D, we could hit again the problematic polynomial we started with, and it could be that the coefficients in front of $\log P$ in the system vanishes. More generally, taking into account all the problematic polynomials, if when we apply Proposition 10 to them we get polynomials Q_i of higher degrees, it could be that descending those we creates loops so that the logarithms of some of the problematic polynomials could not be computed. We expect this event to be very unlikely. Since in all our experiments it was always possible to obtain $\delta = 2$, we did not investigate further.

Finding appropriate h_0 and h_1. One key fact about the algorithm is the existence of two polynomials h_0 and h_1 in $\mathbb{F}_{q^2}[X]$ such that $h_1(X)X^q - h_0(X)$ has an irreducible factor of degree k. A partial solution is due to Joux [16] who showed how to construct such polynomials when $k \in \{q-1, q, q+1\}$. No such deterministic construction is known in the general case, but experiments show that one can apparently choose h_0 and h_1 of degree at most 2. We performed an experiment for every odd prime power q in $[3, \ldots, 1000]$ and every $k \leq q$ and found that we could select $a \in \mathbb{F}_{q^2}$ such that $X^q + X^2 + a$ has an irreducible factor of degree k. Finally, note that the result is similar to a commonly made heuristic in discrete logarithm algorithms: for fixed $f \in \mathbb{F}_{q^2}[X, Y]$ and random $g \in \mathbb{F}_{q^2}[X, Y]$, the polynomial $\mathrm{Res}_Y(f, g)$ behaves as a random polynomial of same degree with respect to the degrees of its irreducible factors.

6 Some directions of improvement

The algorithm can be modified in several ways. On the one hand one can obtain a better complexity if one proves a stronger result on the smoothness probability. On the other hand, without changing the complexity, one can obtain a version which should behave better in practice.

6.1 Complexity improvement

Heuristic 7 tells that a rectangular matrix with $\Theta(q)$ times more rows than columns has full rank. It seems reasonable to expect that only a constant times more rows than columns would be enough to get the full rank properties (as is suggested by the experiments proposed in Section 5). Then, it means that we expect to have a lot of choices to select the best relations, in the sense that their left-hand sides split into irreducible factors of degrees as small as possible.

On average, we expect to be able to try $\Theta(q)$ relations for each row of the matrix. So, assuming that the numerators of $\mathcal{L}_m$ behave like random polynomials of similar degrees, we have to evaluate the expected smoothness that we can hope for after trying $\Theta(q)$ polynomials of degree $(1+\delta)D$ over $\mathbb{F}_{q^2}$. Set $u = \log q/\log\log q$, so that $u^u \approx q$. According to [19] it is then possible to replace $\lceil D/2 \rceil$ in Proposition 2 by the value $O(D \log\log q/\log q)$.

Then, the discussion leading to Theorem 3 can be changed to take this faster descent into account. We keep the same estimate for the arity of each node in the tree, but the depth is now only in $\log k/\log\log q$. Since this depth ends up in the exponent, the resulting complexity in Theorem 3 is then

$$\max(q,k)^{O(\log k/\log\log q)}.$$

6.2 Practical improvements

Because of the arity of the descent tree, the breadth eventually exceeds the number of polynomials below some degree bound. It makes no sense, therefore, to use the descent procedure beyond this point, as the recovery of discrete logarithms of all these polynomials is better achieved as a pre-computation. Note that this corresponds to the computations of the $L(1/4+\epsilon)$ algorithm which starts by pre-computing the logarithms of polynomials up to degree 2. In our case, we could in principle go up to degree $O(\log q)$ without changing the complexity.

We propose another practical improvement in the case where we would like to spend more time descending a given polynomial P in order to improve the quality of the descent tree rooted at P. The set of polynomials appearing in the right-hand side of Equation (E_m) in Section 4 is $\{P-\lambda\}$, because in the factorization of $X^q - X$, we substitute X with $m \cdot P$ for homographies m. In fact, we may apply m to $(P : P_1)$ for any polynomial P_1 whose degree does not exceed that of P. In the right-hand sides, we will have only factors of form $P - \lambda P_1$ for λ in $\mathbb{F}_{q^2}$. On the left-hand sides, we have polynomials of the same

degree as before, so that the smoothness probability is expected to be the same. Nevertheless, it is possible to test several P_1 polynomials, and to select the one that leads to the best tree.

This strategy can also be useful in the following context (which will not occur for large enough q): it can happen that for some triples (q, D, D') one has $N_{q^2}(3D, D')/q^n \approx 1/q$. In this case we have no certainty that we can descend a degree-D polynomial to degree D', but we can hope that at least one of the P_1 allows to descend.

Finally, if one decides to use several auxiliary P_1 polynomials to descend a polynomial P, it might be interesting to take a set of polynomials P_1 with an arithmetic structure, so that the smoothness tests on the left-hand sides can benefit from a sieving technique.

7 Conclusion

The algorithm presented in this article achieves a significant improvement of the asymptotic complexity of discrete logarithm in finite fields, in almost the whole range of parameters where the Function Field Sieve was presently the most competitive algorithm. Compared to existing approaches, and in particular to the line of recent works [15,10], the practical relevance of our algorithm is not clear, and will be explored by further work.

We note that the analysis of the algorithm presented here is heuristic, as discussed in Section 5. Some of the heuristics we stated, related to the properties of matrices $H(P)$ extracted from the matrix $\mathcal{H}$, seem accessible to more solid justification. It seems plausible to have the validity of algorithm rely on the sole heuristic of the validity of the smoothness estimates.

The crossing point between the $L(1/4)$ algorithm and our quasi-polynomial one is not determined yet. One of the key factors which hinders the practical efficiency of this algorithm is the $O(q^2 D)$ arity of the descent tree, compared to the $O(q)$ arity achieved by techniques based on Gröbner bases [15] at the expense of a $L(1/4 + \epsilon)$ complexity. Adj et al. [1] proposed to mix the two algorithms and deduced that the new descent technique must be used for cryptographic sizes. Indeed, by estimating the time required to compute discrete logarithms in $\mathbb{F}_{3^{6 \cdot 509}}$, they showed the weakness of some pairing-based cryptosystems.

Acknowledgements

The authors would like to thank Daniel J. Bernstein for his comments on an earlier version of this work, and for pointing out to us the possible use of asymptotically fast linear algebra for solving the linear systems encountered.

References

1. Adj, G., Menezes, A., Oliveira, T., Rodríguez-Henríquez, F.: Weakness of $\mathbb{F}_{3^{6 \cdot 509}}$ for discrete logarithm cryptography. Cryptology ePrint Archive, Report 2013/446 (2013), `http://eprint.iacr.org/2013/446/`

2. Adleman, L.: A subexponential algorithm for the discrete logarithm problem with applications to cryptography. In: Foundations of Computer Science, 1979., 20th Annual Symposium on. pp. 55–60. IEEE (1979)
3. Adleman, L.: The function field sieve. In: Algorithmic number theory – ANTS I. Lecture Notes in Comput. Sci., vol. 877, pp. 108–121. Springer (1994)
4. Blake, I.F., Fuji-Hara, R., Mullin, R.C., Vanstone, S.A.: Computing logarithms in finite fields of characteristic two. SIAM J. Alg. Disc. Meth. 5(2), 276–285 (Jun 1984)
5. Cheng, Q., Wan, D., Zhuang, J.: Traps to the BGJT-algorithm for discrete logarithms. Cryptology ePrint Archive, Report 2013/673 (2013), `http://eprint.iacr.org/2013/673/`
6. Coppersmith, D.: Fast evaluation of logarithms in fields of characteristic two. IEEE Transactions on Information Theory 30(4), 587–594 (1984)
7. Diffie, W., Hellman, M.: New directions in cryptography. IEEE Transactions on Information Theory 22(6), 644–654 (1976)
8. Göloglu, F., Granger, R., McGuire, G., Zumbrägel, J.: Discrete logarithm in GF(2^{1971}) (Feb 2013), announcement to the NMBRTHRY list
9. Göloglu, F., Granger, R., McGuire, G., Zumbrägel, J.: Discrete logarithm in GF(2^{6120}) (Apr 2013), announcement to the NMBRTHRY list
10. Göloglu, F., Granger, R., McGuire, G., Zumbrägel, J.: On the Function Field Sieve and the impact of higher splitting probabilities. In: Advances in Cryptology – CRYPTO 2013. Lecture Notes in Comput. Sci., vol. 8043, pp. 109–128. Springer (2013)
11. Gordon, D.M.: Discrete logarithms in GF(p) using the number field sieve. SIAM Journal on Discrete Mathematics 6(1), 124–138 (1993)
12. Joux, A.: Discrete logarithm in GF(2^{1778}) (Feb 2013), announcement to the NMBRTHRY list
13. Joux, A.: Discrete logarithm in GF(2^{4080}) (Mar 2013), announcement to the NMBRTHRY list
14. Joux, A.: Discrete logarithm in GF(2^{6168}) (May 2013), announcement to the NMBRTHRY list
15. Joux, A.: Faster index calculus for the medium prime case. Application to 1175-bit and 1425-bit finite fields. In: Advances in Cryptology – EUROCRYPT 2013, Lecture Notes in Comput. Sci., vol. 7881, pp. 177–193. Springer (2013)
16. Joux, A.: A new index calculus algorithm with complexity L($1/4 + o(1)$) in very small characteristic. Cryptology ePrint Archive, Report 2013/095 (2013)
17. Joux, A., Lercier, R.: The function field sieve in the medium prime case. In: Advances in Cryptology – EUROCRYPT 2006. Lecture Notes in Comput. Sci., vol. 4005, pp. 254–270. Springer (2006)
18. Joux, A., Lercier, R., Smart, N., Vercauteren, F.: The number field sieve in the medium prime case. In: Advances in Cryptology – CRYPTO 2006, Lecture Notes in Comput. Sci., vol. 4117, pp. 326–344. Springer (2006)
19. Panario, D., Gourdon, X., Flajolet, P.: An analytic approach to smooth polynomials over finite fields. In: Algorithmic number theory – ANTS III, Lecture Notes in Comput. Sci., vol. 1423, pp. 226–236. Springer (1998)
20. Pohlig, S., Hellman, M.: An improved algorithm for computing logarithms over $GF(p)$ and its cryptographic signifiance. IEEE Transactions on Information Theory 24(1), 106–110 (1978)
21. Stinson, D.R.: Combinatorial designs : constructions and analysis. Springer (2003)