The Discrete Logarithm Problem in Finite Fields of Small Characteristic

Habilitationsschrift

vorgelegt

der Fakultät Mathematik und Naturwissenschaften

der Technischen Universität Dresden

von

Dr. Jens Zumbrägel

geboren am 28. April 1980 in Vechta

Eingereicht am 2. Juni 2015

Wissenschaftlicher Vortrag und Aussprache,
sowie Probevorlesung am 28. Juni 2016

Die Habilitationsschrift wurde in der Zeit
von Okt. 2013 bis Mai 2015 im
Institut für Algebra angefertigt.

Contents

Abstract. The Discrete Logarithm Problem (DLP) in finite fields of small characteristic is currently a very active area of research, where some striking developments have taken place recently. In this introductory chapter we provide background on the DLP and its applications to cryptography, while our main concern is the DLP in finite fields. We give an overview of the state-of-art regarding algorithms for computing discrete logarithms in finite fields of any characteristic. In particular, we focus on the case of small characteristic and summarise the recent advancements as well as the contributions presented in this thesis.

22 pages

Abstract. In this paper we propose a binary field variant of the Joux-Lercier medium-sized Function Field Sieve, which results not only in complexities as low as $L_{q^n}(1/3, (4/9)^{1/3})$ for computing arbitrary logarithms, but also in an heuristic *polynomial time* algorithm for finding the discrete logarithms of degree one and two elements when the field has a subfield of an appropriate size. To illustrate the efficiency of the method, we have successfully solved the DLP in the finite fields with 2^{1971} and 2^{3164} elements, setting a record for binary fields.

20 pages

Abstract. In this paper we show how some recent ideas regarding the discrete logarithm problem (DLP) in finite fields of small characteristic may be applied to compute logarithms in some very large fields extremely efficiently. By combining the polynomial time relation generation from the authors' CRYPTO 2013 paper, an improved degree two elimination technique, and an analogue of Joux's recent small-degree elimination method, we solved a DLP in the record-sized finite field of 2^{6120} elements, using just a single core-month. Relative to the previous record set by Joux in the field of 2^{4080} elements, this represents a 50% increase in the bitlength, using just 5% of the core-hours. We also show that for the fields considered, the parameters for Joux's $L_Q(1/4 + o(1))$ algorithm may be optimised to produce an $L_Q(1/4)$ algorithm.

18 pages

Abstract. In late 2012 and early 2013 the discrete logarithm problem (DLP) in finite fields of small characteristic underwent a dramatic series of breakthroughs, culminating in a heuristic quasi-polynomial time algorithm, due to Barbulescu, Gaudry, Joux and Thomé. Using these developments, Adj, Menezes, Oliveira and Rodríguez-Henríquez analysed the concrete security of the DLP, as it arises from pairings on (the Jacobians of) various genus one and two supersingular curves in the literature, which were originally thought to be 128-bit secure. In particular, they suggested that the new algorithms have no impact on the security of a genus one curve over $\mathbb{F}_{2^{1223}}$, and reduce the security of a genus two curve over $\mathbb{F}_{2^{367}}$ to 94.6 bits. In this paper we propose a new field representation and efficient general descent principles which together make the new techniques far more practical. Indeed, at the '128-bit security level' our analysis shows that the aforementioned genus one curve has approximately 59 bits of security, and we report a total break of the genus two curve.

20 pages

Abstract. For q a prime power, the discrete logarithm problem (DLP) in $\mathbb{F}_q^\times$ consists in finding, for any $g \in \mathbb{F}_q^\times$ and $h \in \langle g \rangle$, an integer x such that $g^x = h$. For each prime p we exhibit infinitely many extension fields $\mathbb{F}_{p^n}$ for which the DLP in $\mathbb{F}_{p^n}^\times$ can be solved in expected quasi-polynomial time.

26 pages

The Discrete Logarithm Problem in Finite Fields of Small Characteristic

Background and Summary

Jens Zumbrägel

Institute of Algebra, TU Dresden, Germany

Abstract. The Discrete Logarithm Problem (DLP) in finite fields of small characteristic is currently a very active area of research, where some striking developments have taken place recently. In this introductory chapter we provide background on the DLP and its applications to cryptography, while our main concern is the DLP in finite fields. We give an overview of the state-of-art regarding algorithms for computing discrete logarithms in finite fields of any characteristic. In particular, we focus on the case of small characteristic and summarise the recent advancements as well as the contributions presented in this thesis.

1 Introduction

Given a finite cyclic group $(G, \cdot)$, a generator $g \in G$ and another group element $h \in G$, the Discrete Logarithm Problem (DLP) is the computational problem to find an integer x satisfying $g^x = h$. The integer x is uniquely determined modulo the group order and is called the *discrete logarithm* $\log_g h$ of the element h to the base g. Most prominently, the group G considered is the multiplicative group $\mathbb{F}_p^*$ of the field $\mathbb{F}_p$ of integers modulo a prime p, which is a cyclic group with difficult DLP. Besides this classical case, several other groups have been extensively studied, including the multiplicative group $\mathbb{F}_q^*$ of any finite field $\mathbb{F}_q$ of prime power order $q = p^n$, the group $E(\mathbb{F}_q)$ of $\mathbb{F}_q$-rational points on an elliptic curve E, or the Jacobian $J_C(\mathbb{F}_q)$ of a (hyperelliptic) curve C.

The study of discrete logarithms can be traced back to at least two centuries ago, when they appeared in Gauß' *Disquisitiones Arithmeticae* under the name of *indices*. The importance of the DLP became even more pronounced with the advent of public-key cryptography in 1976, as the hardness of the DLP (originally in $\mathbb{F}_p^*$) forms the basis of the famous Diffie-Hellman protocol [11] and other cryptographic primitives.

To describe the hardness of the DLP in a group of order N one usually considers the asymptotic complexity in the input size of the problem, which is proportional to $\log N$. To indicate the complexity, it has become customary to use the notation

$$L_N(\alpha, c) := \exp\left((c + o(1))(\log N)^{\alpha}(\log\log N)^{1-\alpha}\right),$$

where $\alpha \in [0,1]$, $c > 0$ and log denotes the natural logarithm; we may omit the subscript N when there is no ambiguity and denote $L(\alpha)$ to mean $L(\alpha, c)$ for some $c > 0$. Note that $L(0) = (\log N)^{c+o(1)}$, which corresponds to polynomial time, while $L(1) = N^{c+o(1)} = \exp(c + o(1))^{\log N}$ denotes exponential time. An algorithm with a running time of $L(\alpha)$ for some $0 < \alpha < 1$ is said to be of *subexponential* complexity.

Algorithms for solving the DLP can be broadly classified into two families. One class are the generic algorithms, which do not exploit a particular group representation and thus apply to any group. Generic algorithms, like Pollard's rho method [32], have a running time of $O(\sqrt{N})$ (unless N has only small prime factors, in which case the Pohlig-Hellman algorithm [31] applies). The other family are the so-called index calculus methods, in which for all elements in a certain specified subset (called factor base) the discrete logarithms are obtained by means of linear algebra. A basic version of the index calculus has been analysed by Adleman [1], resulting in a subexponential complexity of $L(\frac{1}{2})$. Subsequently, more advanced index calculus methods with lower complexity have been developed.

In particular, the first $L(\frac{1}{3})$-algorithm for computing discrete logarithms, published in 1984 by Coppersmith [8], targeted at binary finite fields. Later, the number field sieve, originally devised for the integer factoring problem [28], was adapted for the DLP in prime fields [34, 19] and resulted again in an $L(\frac{1}{3})$-algorithm. Inspired by the number field sieve, the function field sieve was developed for computing discrete logarithms in small (fixed) characteristic [2, 3, 18]. Finally, in 2006 two papers [20, 21] appeared that generalise the function field sieve and the number field sieve, respectively, to work also in the medium prime case, thereby obtaining $L(\frac{1}{3})$-algorithms for all families of finite field DLPs. Some improvements of the medium number field sieve have been reported recently, e.g., see [4].

Regarding the case of small characteristic some dramatic progress has taken place only recently and these developments will be the major subject of the present thesis. Indeed, almost 30 years after Coppersmith's algorithm the $L(\frac{1}{3})$-barrier was broken in a series of remarkable results [A,16, 5] that culminated in a "quasi-polynomial" running time $\exp(O((\log\log N)^2)) = (\log N)^{O(\log\log N)}$, which is in $L(o(1))$.

In this introductory chapter we illustrate the key ideas concerning the DLP in finite fields and we provide an overview of the state-of-art regarding the fastest algorithms for computing discrete logarithms in the various kinds of fields. All necessary mathematical prerequisites will be briefly introduced along the way. As we focus on finite fields we will be very short on other aspects of the DLP, like generic algorithms or the DLP on algebraic curves. We refer to Odlyzko's paper [30] or the recent survey [22] for an overview on these and other general aspects of the DLP, which are not covered here.

A remark on terminology. The cardinality of any finite field F is a prime power $|F| = p^n$, where the prime p is called the *characteristic* of F, which is the smallest positive integer m such that $m1_F = 0$. Conversely, given any prime power $q = p^n$ there exists a finite field of size q, which is unique up to isomorphism and is denoted by $\mathbb{F}_q$.

When considering a family of finite fields of order $q = p^n$, where $p = L_q(\alpha)$, then different DLP algorithms apply dependant on the range of α. Accordingly, in the case $\alpha > \frac{2}{3}$ we speak of *large* characteristic, in the case $\alpha \in (\frac{1}{3}, \frac{2}{3})$ of *medium* characteristic, and if $\alpha \in (0, \frac{1}{3})$ we say that the characteristic is *medium-small*, while the boundary cases $\alpha = \frac{2}{3}$ and $\alpha = \frac{1}{3}$ are special cases to be treated extra. Finally, in the case $\alpha = 0$, i.e., if p is of polynomial size in $\log q$, we speak of finite fields of *small* characteristic, which are the main topic of this thesis. In particular, if $q = p$ is a prime or $q = p^n$ with n fixed, then we are in (very) large characteristic $p = L_q(1)$, whereas if the characteristic p is fixed, then we have small characteristic $p = L_q(0)$; note however that $p = L_q(1)$ (or $p = L_q(0)$) does not imply that n (or p) has to be fixed.

Furthermore, for complexity considerations we make use of the notation $f \approx g$ to indicate that $f/g \to 1$.

Outline. We discuss the general DLP in a group in Section 2 and present briefly the most common cryptographic applications and generic algorithms. The index calculus method serves as a framework for all advanced DLP algorithms for finite fields and will be described in Section 3, where we first present it abstractly in a general group and then give some basic concrete instances. Section 4 is devoted to the number field sieve, which is currently the fastest method for DLP in both large and medium characteristic. Then Section 5 deals with the DLP in finite fields of small or medium-small characteristic and presents the recent dramatic developments in this area. Finally, Section 6 summarises own contributions to the analysis of the DLP in fields of small characteristic and provides an overview of the present thesis.

2 The DLP in a general group

The Discrete Logarithm Problem can be formulated for any group and we may assume without loss of generality that the group is cyclic. Most cryptographic protocols using the DLP can be formulated in this abstract setting. We state in this section the most important cryptographic applications and the common generic attacks. Since our main focus is the DLP in finite fields, we will here be rather short in our presentation.

Let $(G, \cdot)$ be a finite cyclic group of order N and let $g \in G$ be a generator. We assume that the group operation can be computed efficiently, i.e., in polynomial time, and that the group order is known. The surjective group

homomorphism $\mathbb{Z} \to G$, $x \mapsto g^x$ induces a group isomorphism

$$\varphi : \mathbb{Z}_N \to G\,, \quad [x] \mapsto g^x\,,$$

with inverse map $\varphi^{-1} = \log_g h : G \to \mathbb{Z}_N$, $h \mapsto \log_g h$. The map φ can be computed efficiently by using a square-and-multiply method, whereas the computation of φ^{-1} is in general a difficult problem, in fact this is the DLP. We note that this difficulty depends on the concrete representation of the group and is not formally proven.

Cryptographic applications. The difficulty of the DLP is nowadays widely used, e.g., for secure communication over the Internet. Virtually all public-key cryptosystems in use today are based on either the integer factorisation problem or the discrete logarithm problem. Some common cryptographic protocols using the DLP are briefly presented below. In each case the group $(G, \cdot)$ and a generator $g \in G$ are assumed to be publicly known.

In the Diffie-Hellman key-agreement protocol [11] two parties, usually referred to as Alice and Bob, choose random integers a and b, respectively, and exchange the group elements g^a and g^b over the public channel, hence they both can compute a common session key $(g^b)^a = g^{ab} = (g^a)^b$. Clearly, if the DLP in G is feasible, then the key can be computed from the public information, so we require the DLP to be hard.

The Diffie-Hellman protocol can be transformed into a public-key encryption scheme as showed by ElGamal [12]. Indeed, if Bob has announced his public key g^b and Alice has a secret message $m \in G$ for Bob, she chooses a random integer a and sends the pair $(g^a, m(g^b)^a)$ to Bob, who can decrypt by computing $(g^a)^b = (g^b)^a$. Moreover, there are digital signature schemes based on the DLP, e.g., the ElGamal [12] and Schnorr [35] signature schemes.

Generic algorithms. A generic algorithm uses only the group operation and thus applies to any group. Suppose that we want to find the discrete logarithm $\log_g h$ for a target element $h \in G$. Recall that $N = |G|$ is the group order.

In the *Baby-Step-Giant-Step* method, attributed to D. Shanks, one lets $M := \lceil \sqrt{N} \rceil$. We compute a table $\{(j, g^j) \mid j \in \{0 \dots M-1\}\}$ (baby steps), which we sort by the second component. Then we compute $k := g^{-M}$, as well as $h, hk, hk^2, \dots$ (giant steps) until a collision $hk^i = g^j$ is detected, in which case we output $\log_g h = iM + j$. The method requires $O(\sqrt{N})$ database entries and $O(\sqrt{N} \log N)$ group operations (or $O(\sqrt{N})$ if hash tables are used).

The *Pollard's rho* method [32] reduces the memory requirement to some negligible amount while preserving the square-root running time. Therefore, it is the preferred method in practice, however due to the randomised nature the analysis is more difficult. The idea is to recursively define pseudorandom sequences (k_i) in G and (a_i), (b_i) in $\mathbb{Z}_N$ such that $g^{a_i} h^{b_i} = k_i$ holds for any i. If $k_j = k_{j+\ell}$ holds for some $j, \ell > 0$, then there exists also i with $k_i = k_{2i}$, and

such collisions can be easily detected. In this case, $\log_g h = \frac{a_{2i}-a_i}{b_i-b_{2i}}$ is found, provided that the denominator is invertible modulo N.

Finally, there is the *Pohlig-Hellman* method [31], the efficiency of which depends on the particular group order. Let $N = p_1^{e_1} \cdot \ldots \cdot p_r^{e_r}$ be its prime factorisation. Then we have $\mathbb{Z}_N \cong \mathbb{Z}_{p_1^{e_1}} \times \cdots \times \mathbb{Z}_{p_r^{e_r}}$ by the Chinese Remainder Theorem, hence $x = \log_g h$ corresponds to a tuple $(x_1, \ldots, x_r)$ and we can consider the DLP for each factor $x_i \in \mathbb{Z}_{p_i^{e_i}}$ separately. Furthermore, if $e_i > 1$, then $x_i \in \mathbb{Z}_{p_i^{e_i}}$ can be found by obtaining the "p-ary" digits of x_i, one at a time starting with the least significant digit, by solving a DLP of order p_i. Therefore, the Pohlig-Hellman algorithm essentially reduces the DLP complexity in a group of order N to a group of order the largest prime factor of N, for which either the Baby-Step-Giant-Step or Pollard's rho method may be applied.

We remark that, although we are mainly interested in the DLP in finite fields for which more efficient index calculus methods apply, it is often required to use a combination of Pollard's rho method and the Pohlig-Hellman algorithm for dealing with the small prime power factors of the group order.

3 Index calculus methods

The index calculus methods are often more efficient than the generic algorithms, but they apply only to certain groups. We may describe the framework for any group, while details will depend on the concrete representation of the group elements.

Again, let $\log_g h$ to be found in a cyclic group G of order N. We choose a subset $F \subseteq G$ such that $\langle F \rangle = G$ (often we have $g \in F$), called the *factor base*. The idea is to first obtain $\log_g f$ for all $f \in F$. Consider the surjective group homomorphism

$$\varphi : \mathbb{Z}_N^F \to G\,, \quad (e_f)_{f\in F} \mapsto \prod_{f\in F} f^{e_f}\,.$$

The index calculus method consists of three phases.

1. **Relation generation.** Find vectors $(e_f)_{f\in F}$ in $\ker\varphi$, called *relations*, thus generating a subset $R \subseteq \ker\varphi$.
2. **Linear algebra.** Compute an element $0 \neq (x_f)_{f\in F} \in R^\perp$, i.e., satisfying $\sum_{f\in F} x_f e_f = 0$ for all $(e_f)_{f\in F} \in R$.
3. **Individual logarithm.** Find a preimage $(e_f)_{f\in F} \in \varphi^{-1}(h)$, for then we have found $\log_g h = \sum_{f\in F} e_f \log_g f$.

As the next result shows, provided that enough relations have been found, the discrete logarithms of the factor base elements are determined up to a scalar multiple. In practice, the condition given in the lemma is satisfied, if a few more relations than factor base elements have been obtained.

Lemma 1. *Suppose that* $\operatorname{span} R = \ker \varphi$ *and* $(x_f)_{f \in F} \in R^\perp$*, then there exists* $\lambda \in \mathbb{Z}_N$ *such that* $x_f = \lambda \log_g f$ *for all* $f \in F$*.*

Proof. It holds $R^\perp = (\operatorname{span} R)^\perp = (\ker \varphi)^\perp \cong \mathbb{Z}_N^F / \ker \varphi \cong \operatorname{im} \varphi \cong \mathbb{Z}_N$. On the other hand, we have $(\log_g f)_{f \in F} \in R^\perp$; indeed, since $\log_g : G \to \mathbb{Z}_N$ is a group homomorphism, it holds $\sum_{f \in F} e_f \log_g f = \log_g(\prod_{f \in F} f^{e_f}) = 0$ for all $(e_f)_{f \in F} \in R \subseteq \ker \varphi$.

Sparse linear algebra. After the relation generation step an $r \times m$-matrix A over $\mathbb{Z}_N$ has been found, where m is the size of the factor base F and $r \approx m$ is the number of relations. In order to obtain the factor base logarithms we need a solution $0 \neq x \in \mathbb{Z}_N^m$ of $Ax = 0$. Due to the relation generation method the matrix is usually of low average row weight w. For such sparse matrices iterative algorithms are available, most commonly used are the Lanczos [24] method or the Wiedmann [9] algorithm. Their cost is dominated by repeated computations of matrix-vector products Av, and the running is in $O(m^2 w)$ operations in $\mathbb{Z}_N$. Provided that $\log w = o(\log m)$ (and $\log \log N = o(\log m)$), which is usually the case, this is of complexity $m^{2+o(1)}$.

We note that, as there are $O(m)$ divisions in $\mathbb{Z}_N$ necessary, the group order N should avoid small prime factors, therefore the Pohlig-Hellman algorithm should be used for the small prime power factors. In practice, for high-scale computations the linear algebra step poses some challanges, as the iterative algorithms are not easily parallelisable. We also remark that a so-called structured Gaussian elimination (cf. [25, 19]) can be used to decrease the matrix dimension m, while increasing the weight w only moderately.

3.1 A variant for rigorous analysis

The following variant of the index calculus method, proposed by Enge and Gaudry [13], and subsequently refined by Diem [10], is valuable for theoretical analysis. By this variant, one can compute discrete logarithms, provided only that it is feasible to express group elements as products over the factor base, as in the individual logarithm step of the classical index calculus method.

As before, let $(G, \cdot)$ be a cyclic group of order N, let $g \in G$ be a generator and let $h \in G$ be the target element for the DLP. Suppose that $F \subseteq G$ is a factor base of cardinality $|F| = m$. We choose $a, b \in \mathbb{Z}_N$ uniformly and independently at random and try to express $g^a h^b$ as a product $g^a h^b = \prod_{f \in F} f^{e_f}$. Once more than m such expressions have been found, we consider the matrix consisting of the collected rows $(e_f)_{f \in F}$ over $\mathbb{Z}_N$, and compute using invertible row transformations a row echelon form, which contains a vanishing row. Contrary to the classical index calculus method we do not require a rank condition for this matrix. Applying the invertible row transformations also to the numbers a and b, then considering the vanishing row we obtain an identity $g^{a'} h^{b'} = 1_G$. One can show that $b' \in \mathbb{Z}_N$ is uniformly distributed, so that b' is

coprime to N with high enough probability, in which case $\log_g h = \frac{a'}{b'} \in \mathbb{Z}_N$ has been found.

Instead of computing a row echelon form by a variant of the Gauß algorithm one may use sparse linear algebra techniques, which have an improved running time, however their analysis is more difficult and the above algorithm has to be modified [13]. In particular, it is then necessary to fulfill rank conditions for the generated matrix following a technique from Pomerance [33].

3.2 Basic concrete versions

The class of groups for which the index calculus method is applicable includes the multiplicative groups of prime fields and of fields of small fixed characteristic. We describe for these cases a simple index calculus method and provide a running time analysis, which also serves as a basis for the more advanced index calculus methods.

Suppose that G is $\mathbb{Z}_p^*$, the multiplicative group of a prime field $\mathbb{F}_p = \mathbb{Z}_p$, of order $N = p - 1$, with a given generator $g \in G$. As factor base we choose

$$F := \{f \mid f \leq B,\ f \text{ prime}\} \subseteq G$$

for some bound B (by slight abuse of notation, for $f \in \mathbb{Z}$ we denote the class $[f] \in \mathbb{Z}_p$ also by f). For simplicity, we assume that $g \in F$ (otherwise, we include it into the factor base). To generate relations, for random $e \in \mathbb{Z}_N$ we compute $g^e \in \mathbb{Z}_p$, lift it to an element in $\mathbb{N}$, and check by trial division whether it has only prime divisors $\leq B$. If successful, we obtain a relation $g^e \equiv \prod_{f \in F} f^{e_f} \bmod p$ in G. Once enough relations (more than $|F|$) have been found, we compute $\log_g f$ for all $f \in F$ by solving a linear system over $\mathbb{Z}_N$. Finally, given a target element $h \in G$ we similarly obtain one more relation of the form $hg^e = \prod_{f \in F} f^{e_f}$ to obtain $\log_g h$.

Considering a finite field $\mathbb{F}_q = \mathbb{F}_{p^n}$ of fixed characteristic p, note that $\mathbb{F}_{p^n}$ is usually represented as a quotient ring $\mathbb{F}_p[X]/\langle I \rangle$, where $I \in \mathbb{F}_p[X]$ is an irreducible polynomial of degree n. For $G = \mathbb{F}_q^*$, it is then straightforward to adapt the basic index calculus method for $\mathbb{Z}_p^*$ described above to the present situation. In particular, as factor base we choose all irreducible polynomials in $\mathbb{F}_p[X]$ of some bounded degree b, i.e.,

$$F := \{f \mid f \in \mathbb{F}_p[X],\ \deg f \leq b,\ f \text{ irreducible}\} \subseteq G$$

(where we employ a similar abuse of notation). It suffices in practice to include only the monic polynomials into the factor base. In fact, one may perform the discrete logarithm computation in $\mathbb{F}_q^*/\mathbb{F}_p^*$, i.e., ignoring constants in $\mathbb{F}_p^*$, to obtain $\log_g h$ modulo $\frac{N}{p-1}$. Using the Pohlig-Hellman algorithm with the fact that $p - 1$ divides the product of the small prime power divisors of the group order $N = p^n - 1$, the remaining information of $\log_g h$ is deduced easily.

3.3 Complexity analysis

A positive integer is called *B-smooth* if all its prime divisors are $\leq B$. The (heuristic) running time analysis for the basic index calculus method in $\mathbb{Z}_p^*$, as well as for the more advanced algorithms presented in Section 4, is based on the following result on the asymptotic density of smooth numbers among the integers.

Theorem 2 (Canfield, Erdős, Pomerance [7]). *A random integer in $\{1, \ldots, M\}$ is B-smooth with probability*

$$P = u^{-u(1+o(1))}, \quad \textit{where} \quad u = \tfrac{\log M}{\log B}.$$

Corollary 3. *Let $M = L_N(2\alpha, \mu)$ and $B = L_N(\alpha, \beta)$, then the expected number of trials until a random number in $\{1, \ldots, M\}$ is B-smooth is $L_N(\alpha, \frac{\alpha\mu}{\beta})$.*

For analysing the basic version of the index calculus method in $\mathbb{Z}_p^*$, we set the smoothness bound $B = L(\frac{1}{2}, \beta)$ and we have $M = N = L(1, 1)$. As we need about $|F| \approx B/\log B \leq B$ relations, our estimated running time equals

$$L(\tfrac{1}{2}, \beta) \cdot L(\tfrac{1}{2}, \tfrac{1}{2\beta}) = L(\tfrac{1}{2}, \beta + \tfrac{1}{2\beta}),$$

and the optimal choice $\beta := \frac{1}{\sqrt{2}}$ results in a running time of $L(\frac{1}{2}, \sqrt{2})$ for the relation generation. The linear algebra running time (using iterative techniques for sparse matrices) is about $B^2 = L(\frac{1}{2}, 2\beta) = L(\frac{1}{2}, \sqrt{2})$ as well, while the individual logarithm phase is of lower complexity.

Similarly, a polynomial is called *b-smooth* if all its irreducible factors are of degree $\leq b$; hence, the 1-smooth polynomials are precisely those that split into linear factors.

Theorem 4 (Odlyzko, Lovorn, cf. [29]). *A random polynomial $f \in \mathbb{F}_q[X]$ of degree m is b-smooth with probability $P = u^{-u(1+o(1))}$, where $u = \frac{m}{b}$.*

For the DLP in $G = \mathbb{F}_{p^n}^*$, where p is fixed, we obtain quite analogously a running time of $L(\frac{1}{2}, \sqrt{2})$.

4 The number field sieve

The number field sieve is an advanced index calculus method with $L(\frac{1}{3})$-complexity. It was originally devised for the integer factorisation problem, but the method can be adapted to apply for the DLP in prime fields and more generally fields of large or medium characteristic. The principle of these $L(\frac{1}{3})$-algorithms is to generate relations in a way that, although the elements on both sides have to be simultaneously smooth, they are of a considerable smaller "size", when compared to the basic version, so that the smoothness probability is increased.

The setup for computing discrete logarithms in $\mathbb{Z}_p^*$ is as follows. Let $m \in \mathbb{Z}$ and $f \in \mathbb{Z}[X]$ be an irreducible polynomial such that $f(m) = p$, and let $\alpha \in \mathbb{C}$ be a root of f. An application of Gauß' Lemma shows that the map $\mathrm{ev}_\alpha : \mathbb{Z}[X] \to \mathbb{C}$, $h \mapsto h(\alpha)$ has kernel $\ker \mathrm{ev}_\alpha = \langle f \rangle$, and hence $\mathbb{Z}[\alpha] \cong \mathbb{Z}[X]/\langle f \rangle$. Since $f(m) \bmod p = 0$ we deduce that there is a homomorphism $\varphi : \mathbb{Z}[\alpha] \to \mathbb{Z}_p$ such that $\varphi(h(\alpha)) = h(m) \bmod p$ for any $h \in \mathbb{Z}[X]$, i.e., we have the following commutative diagram:

$$\begin{tikzcd}
 & \mathbb{Z}[X] \arrow[dl, "\mathrm{ev}_\alpha"'] \arrow[dr, "\mathrm{ev}_m"] & \\
\mathbb{Z}[\alpha] \arrow[dr, "\varphi"'] & \circlearrowright & \mathbb{Z} \arrow[dl, "\pi"] \\
 & \mathbb{Z}_p &
\end{tikzcd}$$

For applying the index calculus method we need a way to factor $h(m) \in \mathbb{Z}$ and $h(\alpha) \in \mathbb{Z}[\alpha]$ over a smoothness base, however for $\mathbb{Z}[\alpha]$ this is a more intricate issue, requiring some concepts from algebraic number theory.

Suppose that f is monic of degree d. Then α is an algebraic integer, $K := \mathbb{Q}(\alpha)$ is a number field of degree d and $\mathbb{Z}[\alpha]$ is an order in $\mathcal{O}_K$, the ring of integers in K, which is a Dedekind domain. We make use of the norm $N : \mathbb{Q}(\alpha) \to \mathbb{Q}$ satisfying $N(\mathbb{Z}[\alpha]) \subseteq \mathbb{Z}$ in order to generate smooth elements. In $\mathcal{O}_K$ we have a unique prime ideal decomposition, so we may consider the factorisation $\langle h(\alpha) \rangle = \prod_i \mathfrak{p}_i^{e_i}$, where $\mathfrak{p}_i \subseteq \mathcal{O}_K$ are prime ideals. In this case, for the norms we obtain $N(h(\alpha)) = \prod_i N(\mathfrak{p}_i^{e_i})$. We choose $\{\mathfrak{p} \subseteq \mathcal{O}_K \text{ prime} \mid N(\mathfrak{p}) \leq B\}$ as the factor base for the left hand side, for which we obtain "virtual" logarithms using so-called Schirokauer maps, cf. [34, 21].

Parameter choices and complexity analysis. We let $h := h_1 X + h_0 \in \mathbb{Z}[X]$, where $|h_i| \leq E$, to be the sieving polynomial and we want both $N(h(\alpha))$ and $h(m)$ to be B-smooth. As we are aiming at a $L(\frac{1}{3})$-algorithm we set $B = L(\frac{1}{3}, \beta)$ and $E = L(\frac{1}{3}, \varepsilon)$, as well as $d = (\delta + o(1))(\frac{\log N}{\log \log N})^{1/3}$ with parameters $\beta, \delta, \varepsilon > 0$ to be determined.

From the Kalkbrener bound [23] we get $\log |N(h(\alpha))| \approx \log \|f\|_\infty + d \log E$, where $\|f\|_\infty$ denotes the maximum absolute value of the coefficients of f, while $\log |h(m)| \approx \log m$. We can choose $\|f\|_\infty \leq m$ and $m \approx N^{1/d}$, so that $\log m \approx \frac{1}{d} \log N$, which implies $m = L(\frac{2}{3}, \frac{1}{\delta})$. Furthermore, $d \log E = L(\frac{2}{3}, \delta\varepsilon)$, so we get

$$|N(h(\alpha))h(m)| = L(\tfrac{2}{3}, \delta\varepsilon + \tfrac{2}{\delta}).$$

Under the heuristic assumption that the quantity $|N(h(\alpha)h(m)|$ is uniformly distributed for a random polynomial h, we get from Corollary 3 for the probability P of this quantity being B-smooth that $\frac{1}{P} = L(\frac{1}{3}, \frac{\delta^2\varepsilon + 2}{3\beta\delta})$.

The sieving space size $\approx E^2$ should be equal to the linear algebra complexity $\approx B^2$, therefore $\beta = \varepsilon$, and as we want about B relations we set $\frac{1}{P} \approx B$. Therefore, we obtain the condition $\frac{\delta^2\beta+2}{3\beta\delta} = \beta$, or $\delta^2\beta + 2 = 3\beta^2\delta$. The optimal choice $\delta^2 = 2/\beta$ then yields $\beta = (\frac{8}{9})^{1/3}$, resulting in a total complexity of $L(\frac{1}{3}, (\frac{64}{9})^{1/3} \approx 1.923)$.

In the so-called special number field sieve we have small coefficients of f, namely $\log \|f\|_\infty = o(\log m)$, which leads to a faster algorithm. Indeed, we get $\delta^2\beta + 1 = 3\beta^2\delta$, and for $\delta^2 = 1/\beta$ we get $\beta = (\frac{4}{9})^{1/3}$, resulting in a complexity of $L(\frac{1}{3}, (\frac{32}{9})^{1/3} \approx 1.526)$.

The medium number field sieve. This variation of the number field sieve applies to the DLP in finite fields of large or medium characteristic [21].

The setup is as follows (cf. [4]). Choose irreducible polynomials $f, g \in \mathbb{Z}[X]$ such that $f \bmod p$ and $g \bmod p$ have a common irreducible factor $I \in \mathbb{F}_p[X]$ of degree n. (A simple choice is to let $g = f + p$, if $f \bmod p$ contains an irreducible degree n factor; more advanced selection methods are sketched below.) Let $\alpha, \beta \in \mathbb{C}$ be roots of f and g, respectively, so that we have the following diagram:

$$\begin{tikzcd}
 & \mathbb{Z}[X] \arrow[dl, "\mathrm{ev}_\alpha"'] \arrow[dr, "\mathrm{ev}_\beta"] & \\
\mathbb{Z}[X]/\langle f\rangle \cong \mathbb{Z}[\alpha] \arrow[dr, "\varphi"'] \arrow[rr, phantom, "\circlearrowright"] & & \mathbb{Z}[\beta] \cong \mathbb{Z}[X]/\langle g\rangle \arrow[dl, "\psi"] \\
 & \mathbb{F}_{p^n} &
\end{tikzcd}$$

As in the number field sieve for prime fields we obtain relations by finding polynomials $h \in \mathbb{Z}[X]$ (of some degree $\leq t$ and $\|h\|_\infty \leq E$) such that both $N(h(\alpha)) = \mathrm{Res}(h, f)$ and $N(h(\beta)) = \mathrm{Res}(h, g)$ are B-smooth, where Res denotes the resultant. The Kalkbrener bound [23] implies here that

$$\log |N(h(\alpha))N(h(\beta))| \approx t(\log \|f\|_\infty + \log \|g\|_\infty) + (d_f + d_g) \log E,$$

where d_f and d_g are the degrees of f and g, respectively. Therefore, the running time will crucially depend on the degree and the coefficient size of the selected polynomials f and g in the setup phase of the algorithm. The state-of-art work [4] achieves improvements by a clever polynomial selection.

Indeed, for large characteristic we first choose a polynomial $f \in \mathbb{Z}[X]$ of degree $d + 1$ with $\|f\|_\infty$ small such that $f \bmod p$ has an irreducible factor I of degree n. Then we choose a polynomial $g \in \mathbb{Z}[X]$ of degree d such that $I \mid g \bmod p$ and $\|g\|_\infty$ as small as possible; this can be achieved by LLL reduction [26], resulting in the estimation $\log \|g\|_\infty \approx \frac{n}{d+1} \log p$, while we have $d_f = d+1$ and $d_g = d \geq n$. With suitably chosen parameters we get a resulting running time of $L(\frac{1}{3}, (\frac{64}{9})^{1/3})$, the same as in the prime field case.

For medium characteristic the so-called Conjugation Method improves upon the original selection method. Here, we let $\mu := Y^2 + aY + b \in \mathbb{Z}[Y]$ be an irreducible polynomial with small coefficients such that $\mu \bmod p$ has a root $\lambda \in \mathbb{F}_p$. We then choose $g_0, g_1 \in \mathbb{Z}[X]$ with $\|g_i\|_\infty$ small and $\deg g_0 < n = \deg g_1$. With this we let $I := \lambda g_0 + g_1$, $g := ug_0 + g_1$, where $\lambda = \frac{u}{v} \bmod p$ with $\log \|g\|_\infty \approx \frac{1}{2} \log p$, as well as $f := \mathrm{Res}_Y(\mu, Yg_0 - g_1) = g_1^2 + ag_1g_0 + bg_0^2$, so that $\|f\|_\infty$ is small, while $d_f = 2n$ and $d_g = n$. The complexity analysis results in a running time of $L(\frac{1}{3}, (\frac{96}{9})^{1/3} \approx 2.201)$.

5 Fields of small characteristic

First we remark that if given two finite fields of same size, but represented by different polynomials, it is possible to map efficiently between the two representations [27]; therefore for attacking the DLP we can choose the field defining polynomial to our advantage.

Coppersmith's method. The Coppersmith algorithm [8] published in 1984 was the first $L(\frac{1}{3})$-algorithm for a finite field DLP. It is based on the identity $(u + v)^2 = u^2 + v^2$, which holds for any polynomials $u, v \in \mathbb{F}_2[X]$, and the method does not generalise to prime fields. This was a first hint that the DLP in small characteristic appears to be easier than in prime fields. In fact, it has been the fastest general DLP algorithm for binary (or fixed characteristic) finite fields until 2013.

This index calculus algorithm targets at a binary finite field $\mathbb{F}_{2^n}$, represented as $\mathbb{F}_2[X]/\langle I \rangle$, where $I \in \mathbb{F}_2[X]$ is an irreducible degree n polynomial of the form $I := X^n - J$, where $J \in \mathbb{F}_2[X]$ is of low degree (less than $n^{2/3}$). Let the factor base consist of the irreducible polynomials up to a degree bound $\leq b$, and choose positive integers h and ℓ such that $h2^\ell \geq n$. For relation generation, we consider $f := uX^h + v \in \mathbb{F}_2[X]$, where $u, v \in \mathbb{F}_2[X]$ are coprime polynomials of degree $\leq d$, where d is a sieving parameter, and compute

$$f^{2^\ell} = (uX^h + v)^{2^\ell} = u^{2^\ell} X^{h2^\ell} + v^{2^\ell} \equiv u^{2^\ell} X^{h2^\ell - n} J + v^{2^\ell} =: g \pmod{I}.$$

A relation is found if the polynomials $f, g \in \mathbb{F}_2[X]$ on both sides are b-smooth. Note that the corresponding degrees, namely $\deg f \leq h + d$ and $\deg g \leq r + 2^\ell d$ with $r := \deg X^{h2^\ell - n} J$, can be made rather small by suitably chosen parameters. Indeed, we let $d = (c + o(1))n^{1/3}(\log n)^{1/3}$ and suppose that $2^\ell \approx \sqrt{\frac{n}{d}}$, as well as $h = \lceil \frac{n}{2^\ell} \rceil$. Then $\deg f$ and $\deg g$ are about $\sqrt{nd} = (\sqrt{c} + o(1))n^{2/3}(\log n)^{1/3}$. Choosing $b = (c + o(1))n^{1/3}(\log n)^{1/3}$, by applying an analogue of Corollary 3, we get for the probability P of both polynomials being b-smooth that $\frac{1}{P} = L(\frac{1}{3}, \frac{2}{3\sqrt{c}})$. In order to generate enough relations we set $\frac{2}{3\sqrt{c}} = c$, so that $c = (\frac{4}{9})^{1/3}$, resulting in an overall complexity

of $L(\frac{1}{3}, (\frac{32}{9})^{1/3})$ for relation generation. This matches the linear algebra complexity using sparse matrix techniques, while the individual logarithm phase can be shown to have lower complexity.

This analysis supposes that $\sqrt{\frac{n}{d}}$ is close to some power 2^ℓ, which cannot be fulfilled for all n. For the general case we thus get a slightly worse complexity, namely $L(\frac{1}{3}, 4^{1/3})$, where $(\frac{32}{9})^{1/3} \approx 1.526$ and $4^{1/3} \approx 1.587$. We also remark that Coppersmith's algorithm can be easily adapted to the case of fields of fixed characteristic p, by using the identity $(u+v)^p = u^p + v^p$ for $u, v \in \mathbb{F}_p[X]$.

Function field sieve. An adaption of the number field sieve for discrete logarithms in prime fields led to the function field sieve devised by Adleman and Huang [2, 3] and further developed by Joux and Lercier [18]. This algorithm targets at finite fields $\mathbb{F}_{p^n}$ of small characteristic p and has (in the Joux-Lercier version) a complexity of $L(\frac{1}{3}, (\frac{32}{9})^{1/3})$, like the special number field sieve.

The basic idea is to define $\mathbb{F}_{p^n} = \mathbb{F}_p[X]/\langle I \rangle$, where $I = m^d + f_{d-1}m^{d-1} + \cdots + f_1 m + f_0 = f(m)$ with polynomials $f(Y) = \sum f_i Y^i \in (\mathbb{F}_p[X])[Y]$ and $m \in \mathbb{F}_q[X]$ suitably chosen. Then $f \in \mathbb{F}_p[X, Y]$ is a bivariate polynomial, which defines an algebraic curve C and its associated function field $\mathbb{F}_p(C) = \mathrm{Quot}\big(\mathbb{F}_q[X, Y]/f\big)$ (if f is irreducible). The technical details are quite intricate and go beyond the scope of this survey chapter.

Medium function field sieve. Joux and Lercier in 2006 proposed [20] the following simplified variant of the function field sieve, which employs just the rational function field of a univariate polynomial ring. The algorithm applies to the whole range of finite fields $\mathbb{F}_{p^n}$ of medium-small characteristic, i.e., $p = L_{p^n}(\alpha)$, where $\alpha \leq \frac{1}{3}$. We can as well apply the algorithm to extension fields $\mathbb{F}_{q^m}$, where q is any prime power.

The representation of the field $\mathbb{F}_{q^m}$ is as follows. Let $f, g \in \mathbb{F}_q[X]$ be polynomials such that $g(f(X)) - X$ has an irreducible factor $I \in \mathbb{F}_q[X]$ of degree m. Let x be a root of I in $\mathbb{F}_{q^m}$ and let $y := f(x)$, hence $x = g(y)$. Then we have the following diagram:

$$\begin{tikzcd}
 & \mathbb{F}_q[X,Y] \arrow[dl, "Y \mapsto f(X)"'] \arrow[dr, "X \mapsto g(Y)"] & \\
\mathbb{F}_q[X] \arrow[dr, "\mathrm{ev}_x"'] & \circlearrowleft & \mathbb{F}_q[Y] \arrow[dl, "\mathrm{ev}_y"] \\
 & \mathbb{F}_{q^m} &
\end{tikzcd}$$

Now if $q = L_{q^m}(\frac{1}{3})$ then for $a, b, c \in \mathbb{F}_q$ we consider $h := XY + aY + bX + c \in \mathbb{F}_q[X, Y]$, which leads in $\mathbb{F}_{q^m}$ to the following identity

$$xf(x) + af(x) + bx + c = g(y)y + ay + bg(y) + c.$$

If the corresponding polynomials on both sides, namely $h(X, f(X)) = Xf(X) + af(X) + bX + c$ and $h(g(Y), Y) = g(Y)Y + aY + bg(Y) + c$ are 1-smooth, then a relation has been found. We may choose the polynomials f and g such that $\deg f, \deg g \approx \sqrt{m}$, which leads to an algorithm with complexity $L(\frac{1}{3}, 3^{1/3} \approx 1.442)$.

In the general case, where $q = L_{q^m}(\alpha)$ with $\alpha \leq \frac{1}{3}$, in order to obtain an $L(\frac{1}{3})$-algorithm we set as the degree bound for the factor base $b = (c + o(1))(\frac{\log q}{\log\log q})^{1/3-\alpha}$ and consider polynomials of the form $h := h_1(X)Y + h_0(X)$ in order to generate relations. Note that for $\alpha = 0$ the case $\log q = o(\log\log q^m)$ has to be treated extra with a slightly modified analysis. If $q = L_{q^m}(0)$, i.e., the case of small characteristic, this results in an algorithm of complexity $L(\frac{1}{3}, (\frac{32}{9})^{1/3})$.

5.1 Towards a quasi-polynomial DLP algorithm

The dramatic recent progress concerning the discrete logarithm problem in finite fields of small characteristic is based on the following simple result, which can be easily deduced from the identity $\prod_{\gamma\in\mathbb{F}_q}(X - \gamma) = X^q - X$.

Lemma 5. *Let $u, v \in \mathbb{F}_q[X]$. Then there holds*

$$v \prod_{\gamma\in\mathbb{F}_q} (u - \gamma v) = u^q v - uv^q.$$

The new algorithms target at finite fields $\mathbb{F}_Q$ of the form $Q = q^{km}$, where $m \approx q$. Thus with $q = p^\ell$ we have $Q = p^{k\ell m}$, so that the extension degree $k\ell m$ is composite. In general, a given finite field $\mathbb{F}_{p^n}$ can be embedded into $\mathbb{F}_Q$, where $Q = p^{k\ell n}$, since then $\mathbb{F}_Q = \mathbb{F}_{(p^n)^{k\ell}}$ is an extension of $\mathbb{F}_{p^n}$ of degree $k\ell$, where $\ell \approx \log n/\log p$.

We consider the DLP in $\mathbb{F}_Q$, where $Q = (q^k)^m = q^{km}$, with $k \geq 2$ and q fixed. For any $a \in \mathbb{F}_{q^k}[X]$ we may write $a(X)^q = \tilde{a}(X^q)$ with $\deg \tilde{a} = \deg a$, where the coefficients of $\tilde{a}$ are the q-th power of the coefficients of a.

The setup can be seen in the context of the Joux-Lercier function field sieve [20], where however the degrees of the polynomials f and g in the setup are not balanced. In fact, we consider $f := X^q$ and $g := \frac{h_0(X)}{h_1(X)}$ for some $h_0, h_1 \in \mathbb{F}_{q^k}[X]$ of small degree, which leads to the following field representation [A]. We define $\mathbb{F}_{q^{km}}$ as $\mathbb{F}_{q^k}[X]/\langle I\rangle$, where $I \in \mathbb{F}_{q^k}[X]$ is an irreducible degree m polynomial dividing $h_1(X^q)X - h_0(X^q)$ for $h_0, h_1 \in \mathbb{F}_{q^k}[X]$ of low degree $\leq d_h$. Let x be a root of I in $\mathbb{F}_Q$, then we get $y = f(x) = x^q$ and $x = g(y) = \frac{h_0(y)}{h_1(y)}$, thus for any $u, v \in \mathbb{F}_{q^k}[X]$ we have

$$v(x) \prod_{\gamma\in\mathbb{F}_q} \big(u(x) - \gamma v(x)\big) = \tilde{u}(y)v(x) - u(x)\tilde{v}(y) = \tilde{u}(y)v(g(y)) - u(g(y))\tilde{v}(y).$$

Note that the field representation requires that $m \leq qd_h + 1$. Alternatively, in [17, 5] the field representation used is $\mathbb{F}_{q^2}[X]/\langle I\rangle$, where $I \mid X^q h_1(X) -$

$h_0(X)$, thus here we have $m \leq q + d_h$. We remark that our field representation may have practical advantages by allowing a larger field extension degree m.

With these field representations the relation generation can be achieved in polynomial time, which was a major breakthrough in the DLP analysis in small characteristic. Letting $u := \alpha X + \beta$ and $v := \gamma X + \delta$ in $\mathbb{F}_{q^k}[X]$ we obtain

$$u^q v - uv^q = (\alpha X + \beta)^q (\gamma X + \delta) - (\alpha X + \beta)(\gamma X + \delta)^q.$$

This is (up to a scalar) of the form $X^{q+1} + aX^q + bX + c$, and

$$x^{q+1} + ax^q + bx + c = \tfrac{1}{h_1(y)} \big(yh_0(y) + ayh_1(y) + bh_0(y) + ch_1(y) \big) ,$$

where on the right hand side we have a "random" polynomial of low degree $d_h + 1$, while the left hand side splits by Lemma 5. In fact, we have the following result stating exactly for which triples $(a, b, c) \in \mathbb{F}_{q^k}^3$ the polynomial splits, which turns out to be very useful for developing a new DLP algorithm with proven complexity.

Theorem 6. *[6] For $a, b, c \in \mathbb{F}_{q^k}$ consider the polynomial $X^{q+1} + aX^q + bX + c$. If $c \neq ab$ and $b \neq a^q$, then the polynomial splits if and only if*

$$\frac{(b - a^q)^{q+1}}{(c - ab)^q} \in \left\{ \frac{(u^{q^2} - u)^{q+1}}{(u^q - u)^{q^2+1}} \mid u \in \mathbb{F}_{q^k} \setminus \mathbb{F}_{q^2} \right\}.$$

In particular, $X^{q+1} + aX^q + bX + c$ splits with probability $\approx q^{-3}$.

In contrast to the $L(\frac{1}{3})$-algorithms presented previously, the complexity analysis of this relation generation does not depend on assuming that the elements occurring on both sides are uniformly distributed and then applying theorems on the smoothness density, i.e., Theorem 2 or Theorem 4. Instead we use (one one side) polynomials that are smooth by construction and therefore obtain much higher splitting probabilities. Hence, we may choose a very small factor base, which leads to a greatly improved complexity. Indeed, if k is constant and m is chosen such that $q = O(m)$, then the relation generation step and linear algebra complexity is polynomial in $\log Q = q^{1+o(1)}$.

Individual logarithm. Given that the relation generation and linear algebra steps are very efficient, the attention is now on the individual logarithm phase, which previously used to be a step with lower complexity. However, since the factor base is very small, this step has become more of an issue.

In order to obtain an individual logarithm $\log_g h$ for a target element $h \in \mathbb{F}_Q$, one usually applies a *descent* strategy. This means building a tree in which h is the root and the leaves consist of factor base elements. Furthermore, if $y_1, \ldots, y_r \in \mathbb{F}_Q$ are the children of some element $x \in \mathbb{F}_Q$, they are supposed to be of smaller "size", i.e., degree in the polynomial representation,

while a relation $x = \prod_i y_i^{e_i}$ has been obtained. Once a tree of this form is built, it is easy to compute an expression of h as a product $h = \prod f^{e_f}$ of factor base elements and thus to deduce the logarithm $\log_g h$.

For performing the descent step, i.e., rewriting some element $x \in \mathbb{F}_Q$ as a product $x = \prod_i y_i^{e_i}$ of smaller degree elements $y_i \in \mathbb{F}_Q$, there are different methods available, including some new strategies that use techniques for the fast relation generation. The methods are of different complexity, dependant on the degree of x, and in practice one uses a combination of several of these. We may build up the descent tree using

- degree two elimination [A,17],
- small degree Gröbner Basis descent [17],
- quasi-polynomial time descent [5],
- large degree classical descent,
- initial split.

At the beginning of the descent process, when elements are represented by polynomials of high degree, it is relatively easy to find an expression involving lower degree polynomials. This is achieved by an initial split using a continuous fraction method and then by a so-called special-q lattice descent. As these methods are somewhat classical (see, e.g., [20]) we will focus here rather on the new descent strategies for lower degree polynomials.

The idea of Joux's Gröbner basis descent [17] is the following. Suppose that $P(y)$ is to be eliminated. For $u, v \in \mathbb{F}_{q^k}[X]$ of degree $\leq D$ by Lemma 5 there holds

$$v(x) \prod_{\gamma \in \mathbb{F}_q} \big(u(x) - \gamma v(x)\big) = u(x)^q v(x) - u(x) v(x)^q = \tilde{u}(y) v(\tfrac{h_0(y)}{h_1(y)}) - u(\tfrac{h_0(y)}{h_1(y)}) v(y)$$

with the right hand side $R(y)$ of low degree $D + d_h D = (d_h + 1)D$. Considering now the equation $R(y) \equiv 0 \bmod P(y)$ in the $\mathbb{F}_q$-components of the coefficients of u and v, we get a bilinear quadratic system, where we have $2(D+1)k$ variables and $d_P k$ equations. Now if the cofactor is D-smooth, we have eliminated $P(y)$. The running time analysis of this method is based on the complexity of solving bilinear quadratic systems by Gröbner basis methods, which has been investigated by Spaenlehauer [36]; it depends on the degrees d_u and d_v of the polynomials u and v (roughly, the cost C satisfies $\log C \approx c \log d_u^{d_v}$ for some constant c, if $d_u \geq d_v$ and k is constant). This results in an overall descent complexity of $L(\frac{1}{4} + o(1))$ as analysed by Joux [17], while a slightly better balanced choice of degrees for u and v leads to an improved complexity of $L(\frac{1}{4})$, see [B].

The on-the-fly degree two elimination of [A] can be seen as a special case of the Gröbner basis elimination. For $d_P = 2$ we let $d_u = d_v = 1$, which heuristically works for $d_h \leq 2$ and $k > 3$. This method is a basic ingredient of our new descent strategy [D].

Alternatively, one can use the following method of solving degree two logs in batches [17]. Consider the identity used for relation generation

$$x^{q+1} + ax^q + bx + c = \tfrac{1}{h_1(y)}\big(yh_0(y) + ayh_1(y) + bh_0(y) + ch_1(y)\big),$$

and, for each $u \in \mathbb{F}_{q^k}$, substitute x by $P := x^2 + ux$ and then solve a linear system over the factor base $F_u := \{x^2 + ux + v \text{ irreducible} \mid v \in \mathbb{F}_{q^k}\}$.

Generalising this idea to polynomials P of any degree leads to the quasi-polynomial time descent algorithm of [5]; it can be viewed as a descent method by linear algebra. In a step of this descent one can rewrite any element of $\mathbb{F}_Q$ represented by a polynomial $P \in \mathbb{F}_{q^k}[X]$ of degree $< m$ as a product $P = \prod_i R_i^{e_i}$ of polynomials of degree $\leq \frac{\deg P}{2}$, in time polynomial in q^k and m, where the number of elements R_i is in $O(mq^k)$.

While this quasi-polynomial time algorithm is asymptotically the fastest, it appears however not (yet) to be used in record computations, while the Gröbner basis method has been widely employed in practice recently.

Complexity results. Consider a finite field $\mathbb{F}_Q$ with $Q = q^{km}$ represented as $\mathbb{F}_Q = \mathbb{F}_{q^k}[X]/\langle I\rangle$, where $I \in \mathbb{F}_{q^k}[X]$ is a degree m irreducible polynomial dividing $X^q h_1 - h_0$ for some small degree polynomials $h_0, h_1 \in \mathbb{F}_{q^k}[X]$, thus we have $m \leq q + d_h$. For k fixed we get from the polynomial time relation generation and the quasi-polynomial time descent that the DLP in $\mathbb{F}_Q$ can be solved in a heuristic running time (see [5, Th. 3]) of

$$q^{O(\log m)} = \exp(O(\log q \log m)). \qquad (*)$$

This algorithm has an impact on general finite fields of small and medium-small characteristic. Indeed, suppose that a DLP in a finite field $\mathbb{F}_{p^n}$ has to be found. We embed this field in $\mathbb{F}_Q$, where $Q = q^{kn} = (p^n)^{k\ell}$ and $q = p^\ell$.

In the case of small characteristic, i.e., $p = L_{p^n}(0)$, we have $\log p = O(\log n)$ and $\log\log p^n = \log n + \log\log p \approx \log n$. We let $q = p^\ell$ such that $q \geq n - d_h$ and $\log q = O(\log n)$, hence $(*)$ implies a running time of $\exp(O(\log\log p^n)^2))$, which is quasi-polynomial in the logarithm $\log p^n$ of the group order.

If the characteristic is medium-small, i.e., $p = L_{p^n}(\alpha)$, where $0 < \alpha < \frac{1}{3}$, we let $q = p$, so that $\log q = \log p = O((\log p^n)^\alpha (\log\log p^n)^{1-\alpha})$. By $(*)$ and observing that $\log n = O(\log\log p^n)$ we get a complexity of $L_{p^n}(\alpha)^{O(\log\log p^n)} = \exp(O((\log p^n)^\alpha (\log\log p^n)^{2-\alpha}) = L_{p^n}(\alpha + o(1))$, which improves on the function field sieve having $L(\frac{1}{3})$-complexity.

Kummer extensions and automorphisms. Kummer theory provides us with particularly useful polynomials for the field representation, as observed in [16]. Let $1 < n \mid q - 1$, so that $\mathbb{F}_q$ contains the n-th roots of unity, denoted by μ_n. Let $c \in \mathbb{F}_q$, let $x := \sqrt[n]{c}$ be a root of $X^n - c \in \mathbb{F}_q[X]$ and let m be the degree of the minimal polynomial of x over $\mathbb{F}_q$, so that $\mathbb{F}_q(x) = \mathbb{F}_{q^m}$. Letting $t := \frac{q-1}{n}$,

then for the q-th power Frobenius σ we have $\sigma(x) = x^q = (x^n)^t x = c^t x$, and hence $m = \mathrm{ord}(\sigma) = \mathrm{ord}(a^t)$. Therefore, we see that $X^n - c$ is irreducible, i.e., $m = n$, if and only if $\mathrm{ord}(a^t) = n$. In particular, for a generator $c \in \mathbb{F}_q^*$ we have that $I := X^{q-1} - c \in \mathbb{F}_q[X]$ is an irreducible polynomial dividing $X^q - cX = h_1 X^q - h_0$, with $h_1 = 1$, $h_0 = cX$ of degree $\leq d_h = 1$.

Similarly, one can show that $X^{q+1} - X - c \in \mathbb{F}_q[X]$ is irreducible if $c = -\lambda^{q+1}$ for a generator $\lambda \in \mathbb{F}_{q^2}^*$. Here, we have $X^{q+1} - X - c = h_1 X^q - h_0$, with $h_1 = X$, $h_0 = aX$.

Having degree at most one for the polynomials h_0 and h_1 in the field representation has also some practical advantages for the relation generation and especially for the individual logarithm phase. Furthermore, when defining $\mathbb{F}_{q^n}$ by $\mathbb{F}_q[X]/\langle I \rangle$ we can use factor base preserving automorphisms to reduce the complexity of the linear algebra step. In fact, for the q-th power Frobenius we have $(x + a)^q = x^q + a^q = c^t x + a^q = c^t(x + \frac{a^q}{c^t})$. The group generated by the Frobenius automorphism of order n acts on the factor base, effectively reducing the variables of the linear algebra problem by a factor of about n.

6 Overview of this thesis

This habilitation thesis deals with the DLP in finite fields of small characteristic. We are concerned with the following two principal aspects of the problem.

1. **Large-scale computations of discrete logarithms**
   This aspect is particularly important for the security assessment of contemporary cryptosystems, which are often based on the hardness of the DLP. It includes all algorithmic improvements, whether being practical speed-ups or even affecting the asymptotic running time, as well as a running time analysis, which however is based on heuristic assumptions.
2. **Rigorous proofs of the running time of DLP algorithms**
   While providing algorithms with heuristic running time analysis is common practice and a useful tool for concrete security estimations, from a mathematical point of view this is unsatisfactory. This part focuses on developing new algorithms which avoid any heuristic assumptions and whose running time can be proven rigorously.

We remark that, while these goals seem to be largely independent, progress from either aspect often feeds into the other, as illustrated below.

Results. During the last two years some striking advancement in solving the DLP in finite fields of small characteristic have been made, which have had a considerable impact on cryptology research, especially in the vivid area of identity-based cryptography. I am fortunate to have been part of these recent developments and to have co-authored a number of relevant papers on this topic. This habilitation collects my major articles on the subject, and a short summary of these is given below.

Table 1. Discrete logarithm record computations in finite fields. We list the cases where either the asymptotic complexity or the field size has been improved.

bitlength	charact.	who/when	running time
127	2	Coppersmith 1984 [8]	$L(1/3\,,\,1.526..1.587)$
401	2	Gordon, McCurley 1992	$L(1/3\,,\,1.526..1.587)$
n/a	small	Adleman 1994 [2]	$L(1/3\,,\,1.923)$
427	large	Weber, Denny 1998	$L(1/3\,,\,1.526)$
521	2	Joux, Lercier 2001	$L(1/3\,,\,1.526)$
607	2	Thomé 2001	$L(1/3\,,\,1.526..1.587)$
613	2	Joux, Lercier 2005	$L(1/3\,,\,1.526)$
556	medium	Joux, Lercier 2006	$L(1/3\,,\,1.442)$
676	3	Hayashi et al. 2010	$L(1/3\,,\,1.442)$
923	3	Hayashi et al. 2012	$L(1/3\,,\,1.442)$
1175	medium	Joux 24 Dec 2012 [16]	$L(1/3\,,\,1.260)$
1425	medium	Joux 6 Jan 2013 [16]	$L(1/3\,,\,1.260)$
1778	2	Joux 11 Feb 2013 [17]	$L(1/4+o(1))$
1971	2	GGMZ 19 Feb 2013 [A]	$L(1/3\,,\,0.763)$
4080	2	Joux 22 Mar 2013 [17]	$L(1/4+o(1))$
6120	2	GGMZ 11 Apr 2013 [B]	$L(1/4)$
6168	2	Joux 21 May 2013	$L(1/4+o(1))$
n/a	small	BGJT 18 Jun 2013 [5]	$L(0+o(1))$
9234	2	GKZ 31 Jan 2014 [15]	$L(1/4+o(1))$

[A] Our first paper on the DLP in finite fields of small characteristic is an adaption of the medium function field sieve for the use in binary fields. It features a heuristic polynomial time algorithm for finding the discrete logarithm of degree one and two elements, and presents discrete logarithm computations in finite fields with 2^{1971} and 2^{3164} elements, setting a record for binary fields. This article has won the Best Paper Award at the highly prestigious cryptology conference CRYPTO in August 2013.

[B] We have shown how to combine the polynomial time relation generation of [A] and an analogue of Joux's small-degree elimination method using Gröbner bases [17] for solving a DLP in the record-sized finite field with 2^{6120} elements, using the equivalent of just one week on a four-core desktop computer. We also show how to optimise the parameters of the Gröbner basis descent to produce an $L(\frac{1}{4})$-algorithm.

[C] We investigate the relevance of the new DLP algorithms on practically proposed cryptosystems in the context of pairing-based cryptography on supersingular curves. Along the way, we improved and extended the new methods to make the attacks far more effective and more widely applicable. In particular, at the '128-bit security' level our analysis shows that a common genus one curve offers only 59 bits of security, while we report a total break of a genus two curve.

[D] One of these aforementioned practical improvements has led to a mainly theoretical work, which outlines a novel descent strategy (see also [14]). The resulting algorithm is a new quasi-polynomial algorithm for the DLP in small characteristic, which is based on less heuristic assumptions and features a rigorous complexity analysis, which is based on the irreducibility of a certain algebraic curve. The demonstration that our descent has always the stated complexity distinguishes it from all the previous recent work. Besides, the new descent method appears to also have practical advantages and has been used lately in record computations.

We have also set a new discrete logarithm record in a finite field with 2^{9234} elements, announced on 31 Jan 2014 [15]. Indeed, the Irish Centre for High-End Computing (ICHEC) has offered our group early access to its new HPC cluster Fionn in November 2013. Our work used an involved and highly optimised large-scale computation of about 400 k core hours. At present this record is still valid, cf. Table 1.

Statement of own contributions. The papers [A] and [B] have been a collaboration with F. Göloğlu, R. Granger and G. McGuire, then at University College Dublin, while the papers [C] and [D] are joint work work with R. Granger and T. Kleinjung at EPFL in Lausanne, Switzerland. My own contributions in this collaborative efforts can be outlined as follows.

As is common in the area of computational algebraic number theory, our progress and achievements are largely due to a close interplay between theoretical aspects, algorithmic considerations and feedback from implementations of the algorithms. In the aforementioned research teams I have been the major responsible for all algorithmic aspects and for implementing mathematical algorithms on different platforms, including computer algebra systems (Magma), the number theory library NTL for C++, parallel computing tools (MPI and openMPI) and large-scale computation maintenance techniques, as well as documentation of the results.

The ability of "testing" rapidly many various ideas by implementing them for small cases often provides interesting observations from these experiments. A very good example for this is the higher splitting probability of polynomials of the form $X^{q+1} + aX^q + bX + c \in \mathbb{F}_{q^k}[X]$. This family of polynomials, which forms the fundamental ingredient of our renowned work [A] that led to polynomial time relation generation and the further improvements [17, 5], has actually been found by some of my computer experiments.

The article [B] deals mainly with algorithmic improvements and clever optimisations for a huge discrete logarithm problem. I have written large parts of this article and act as its corresponding author. Furthermore I made a considerable contribution in carrying out details of the improved complexity analysis of the approach in [17], leading to an $L(\frac{1}{4})$-algorithm.

The work [C] is again of mainly algorithmic character and contains many remarkable improvements. It deals with concrete security estimations, which involves an extensive running time analysis, as well as implementing key parts of the algorithm and performing simulations. I have observed that earlier published work on the security of supersingular curves can be improved by large margins. My contribution also includes being the main responsible for the analysis of a common '128-bit secure' supersingular binary elliptic curve.

Finally, in our recent theoretical work [D] on a new quasi-polynomial time algorithm with provable running time, I did a major contribution on the proof of its main result, i.e., the correctness of the algorithm, which employs the on-the-fly degree two elimination of [A]. The argument of the proof is based on showing absolute irreducibility of an algebraic curve, which may in general be a very hard problem; our approach is based on tools from invariant theory to tackle the issue. Besides, I am interested in beneficial mathematical modelling of problems, and in this regard I have been carrying out details of Enge-Gaudry-Diem's work on rigorous versions of the index calculus method, which enabled in our setting the elimination of one of our earlier heuristic assumption.

References

[A] F. Göloğlu, R. Granger, G. McGuire, J. Zumbrägel, "On the function field sieve and the impact of higher splitting probabilities," in: Advances in Cryptology–CRYPTO 2013, LNCS 8043, pp. 109–128, Springer (2013)

[B] F. Göloğlu, R. Granger, G. McGuire, J. Zumbrägel, "Solving a 6120-bit DLP on a desktop computer," in: Selected Areas in Cryptography—SAC 2013, LNCS 8282, pp. 136–152, Springer (2014)

[C] R. Granger, T. Kleinjung, J. Zumbrägel, "Breaking 128-bit securesupersingular binary curves," in: Advances in Cryptology—CRYPTO 2014, LNCS 8617, pp. 126–145, Springer (2014)

[D] R. Granger, T. Kleinjung, J. Zumbrägel, "On the discrete logarithm problem in finite fields of fixed characteristic," Preprint (2015), submitted to a mathematics journal

1. L. M. Adleman, "A subexponential algorithm for the discrete logarithm problem with applications to cryptography," in: 20th Annual Symposium on Foundations of Computer Science, pp. 55–60, IEEE (1979)
2. L. M. Adleman, "The function field sieve," in: Algorithmic number theory, pp. 108–121, Springer (1994)
3. L. M. Adleman, M.-D. A. Huang, "Function field sieve method for discrete logarithms over finite fields," *Inform. and Comput.*, vol. 151, no. 1 (1999), pp. 5–16
4. R. Barbulescu, P. Gaudry, A. Guillevic, F. Morain, "Improving NFS for the discrete logarithm problem in non-prime finite fields," in: Advances in Cryptology—EUROCRYPT 2015, pp. 129–155, Springer (2015)
5. R. Barbulescu, P. Gaudry, A. Joux, E. Thomé, "A heuristic quasi-polynomial algorithm for discrete logarithm in finite fields of small characteristic," in: Advances in Cryptology–Eurocrypt, LNCS 8441, pp. 1–16, Springer (2014)
6. A. W. Bluher, "On $x^{q+1}+ax+b$," *Finite Fields Appl.*, vol. 10, no. 3 (2004), pp. 285–305
7. E. R. Canfield, P. Erdős, C. Pomerance, "On a problem of Oppenheim concerning 'factorisatio numerorum'," *J. Number Theory*, vol. 17, no. 1 (1983), pp. 1–28
8. D. Coppersmith, "Fast evaluation of logarithms in fields of characteristic two," *IEEE Trans. Inform. Theory*, vol. 30, no. 4 (1984), pp. 587–594

9. D. Coppersmith, "Solving homogeneous linear equations over GF(2) via block Wiedemann Algorithm," *Math. Comp.*, vol. 62, no. 205 (1994), pp. 333–350
10. C. Diem, "On the discrete logarithm problem in elliptic curves," *Compositio Math.*, vol. 147, pp. 75–104.
11. W. Diffie, M. E. Hellman, "New Directions in Cryptography," *IEEE Trans. Inform. Theory*, vol. 22, no. 6 (1976), pp. 644–654
12. T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," in: Advances in Cryptology—CRYPTO '84, LNCS 196, pp. 10–18, Springer (1985)
13. A. Enge, P. Gaudry, "A general framework for subexponential discrete logarithm algorithms," *Acta Arithmetica*, vol. 102 (2002), pp. 83–103
14. R. Granger, T. Kleinjung, J. Zumbrägel, "On the Powers of 2," IACR Cryptology ePrint Archive (2014), eprint.iacr.org/2014/300
15. R. Granger, T. Kleinjung, J. Zumbrägel, "Discrete logarithms in GF(2ˆ9234)," E-mail to the NMBRTHRY mailing list (2014)
16. A. Joux, "Faster index calculus for the medium prime case; application to 1175-bit and 1425-bit finite fields," in: Advances in Cryptology—EUROCRYPT 2013, LNCS 7881, pp. 177–193, Springer (2013)
17. A. Joux, "A new index calculus algorithm with complexity L (1/4+ o (1)) in small characteristic," in: Selected Areas in Cryptography—SAC 2013, LNCS 8282, pp. 355–379, Springer (2014)
18. A. Joux, R. Lercier, "The function field sieve is quite special," in: Algorithmic Number Theory, pp. 431–445, Springer (2002)
19. A. Joux, R. Lercier, "Improvements to the general number field sieve for discrete logarithms in prime fields," *Math. Comp.*, vol. 72, no. 242 (2003), pp. 953–967
20. A. Joux, R. Lercier, "The function field sieve in the medium prime case," in: Advances in Cryptology—EUROCRYPT 2006, LNCS 4117, pp. 254–270, Springer (2006)
21. A. Joux, R. Lercier, N. Smart, F. Vercauteren, "The number field sieve in the medium prime case," in: Advances in Cryptology—CRYPTO 2006, pp. 326–344, Springer (2006)
22. A. Joux, A. Odlyzko, C. Pierrot, "The Past, evolving Present and Future of Discrete Logarithm," in: Open Problems in Mathematical and Computational Science, Springer (2014), 23 pages
23. M. Kalkbrener, "An upper bound on the number of monomials in determinants of sparse matrices with symbolic entries," *Mathematica Pannonica*, vol. 73 (1997), pp. 73–82
24. C. Lanczos, "An iteration method for the solution of the eigenvalue problem of linear differential and integral operators," *J. Research Nat. Bur. Standards*, vol. 45 (1950), pp. 255–282
25. B. A. LaMacchia, A. M. Odlyzko, "Solving Large Sparse Linear Systems over Finite Fields," in: Advances in Cryptology—CRYPTO '90, LNCS 537, pp. 109–133, Springer (1991)
26. A. K. Lenstra, H. W. Lenstra, L. Lovász, "Factoring polynomials with rational coefficients," *Math. Ann.*, vol. 261, no. 4 (1982), 515–534
27. H. W. Lenstra, Jr., "Finding isomorphisms between finite fields," *Math. Comp.*, vol. 56, no. 193 (1991), pp. 329–347
28. A. K. Lenstra, H. W. Lenstra (eds), *The number field sieve*, Springer, 1993
29. A. M. Odlyzko, "Discrete logarithms in finite fields and their cryptographic significance," in: Advances in Cryptology—CRYPTO '85, LNCS 209, pp. 224–314, Springer (1985)
30. A. Odlyzko, "Discrete logarithms: The past and the future," in: Towards a Quarter-Century of Public Key Cryptography, pp. 59–75, Springer (2000)
31. S. C. Pohlig, M. E. Hellman, "An improved algorithm for computing logarithms over GF(p) and its cryptographic significance (Corresp.)," *IEEE Trans. Inform. Theory*, vol. 24, no. 1 (1978), pp. 106–110
32. J. M. Pollard, "Monte Carlo methods for index computation (mod p)," *Math. Comp.*, vol. 32, no. 143 (1978), pp. 918–924

33. C. Pomerance, "Fast, rigorous factorization and discrete logarithm algorithms," in: Discrete algorithms and complexity, Academic Press, NY (1987), pp. 119-143.
34. O. Schirokauer, "Using number fields to compute logarithms in finite fields," *Math. Comp.*, vol. 69, no. 231 (2000), pp. 1267–1283
35. C.-P. Schnorr, "Efficient signature generation by smart cards," *J. Cryptology*, vol. 4, no. 3 (1991), pp. 161–174
36. P. J. Spaenlehauer, *Solving multi-homogeneous and determinantal systems: algorithms, complexity, applications*, Ph. D. dissertation, Université Pierre et Marie Curie (Univ. Paris 6), 2012

On the Function Field Sieve and the Impact of Higher Splitting Probabilities*

Application to Discrete Logarithms in $\mathbb{F}_{2^{1971}}$ and $\mathbb{F}_{2^{3164}}$

Faruk Göloğlu, Robert Granger, Gary McGuire, and Jens Zumbrägel

Complex & Adaptive Systems Laboratory and
School of Mathematical Sciences
University College Dublin, Ireland
{farukgologlu,robbiegranger}@gmail.com, {gary.mcguire,jens.zumbragel}@ucd.ie

Abstract. In this paper we propose a binary field variant of the Joux-Lercier medium-sized Function Field Sieve, which results not only in complexities as low as $L_{q^n}(1/3, (4/9)^{1/3})$ for computing arbitrary logarithms, but also in an heuristic *polynomial time* algorithm for finding the discrete logarithms of degree one and two elements when the field has a subfield of an appropriate size. To illustrate the efficiency of the method, we have successfully solved the DLP in the finite fields with 2^{1971} and 2^{3164} elements, setting a record for binary fields.

Keywords: Discrete logarithm problem, function field sieve.

1 Introduction

When it comes to selecting appropriate parameters for public-key cryptosystems, one invariably observes a trade-off between security and efficiency. At a most basic level, for example, larger keys usually mean higher security, but worse performance.

A related rule of thumb which one does well to keep in mind, is that a specialised parameter which improves efficiency, typically (or potentially) weakens security. Examples abound of such specialisations and consequent attacks: discrete logarithms modulo Mersenne (or Crandall) primes and the Special Number Field Sieve [19]; Optimal Extension Fields [2] and Weil descent for elliptic curves [8]; high-compression algebraic tori [23] and specialised index calculus [10]; quasi-cyclic or dyadic McEliece variants [21] and Gröbner basis attacks [6], and more recently elliptic curves over binary fields [7], to name just a few. In practice therefore, one should be wary of any additional structure, which may potentially weaken a system.

* Research supported by the Claude Shannon Institute, Science Foundation Ireland Grant 06/MI/006. The fourth author was in addition supported by SFI Grant 08/IN.1/I1950.

In this paper we give a fairly extreme example of this principle in the case of binary (or in general small characteristic) fields which possess a small to medium-sized intermediate field. In 2006 Joux and Lercier designed a particularly efficient variation of the Function Field Sieve (FFS) algorithm for computing discrete logarithms [16], which at the time possessed the fastest asymptotic complexity of all known discrete logarithm algorithms for appropriately balanced q and n, namely $L_{q^n}(1/3, 3^{1/3}) \approx L_{q^n}(1/3, 1.442)$, where

$$L_{q^n}(a, c) = \exp\left((c + o(1))\,(\log q^n)^a (\log\log q^n)^{1-a}\right),$$

and q^n is the cardinality of the finite field.

In 2012, Joux proposed a more efficient method of obtaining relations, dubbed 'pinpointing', which applies to a specialisation of the function field setup of [16]. In this approach, each relation found via classical sieving can be amplified into many more [13], which is advantageous when sieving is the dominant phase, rather than the linear algebra (or individual logarithm phase). The overall complexity of this technique for solving the DLP can be as low as $L_{q^n}(1/3, (8/9)^{1/3}) \approx L_{q^n}(1/3, 0.961)$. To demonstrate the practicality of the approach, Joux solved the DLP in two cases: in a 1175-bit field and in a 1425-bit field, setting records for medium-sized base fields, in this case prime fields.

In this work we demonstrate that a basic assumption used in the analysis of virtually all fast index calculus algorithms can be very wrong indeed; in the case of binary fields possessing a subfield of an appropriate size, this leads to the dramatic conclusion that the logarithms of degree one elements over this subfield can be solved in *polynomial time*. As far as we are aware, no other algorithm for the collecting of relations and the linear algebra step has beaten the $L_{q^n}(1/3)$ barrier. Our fundamental observation is that the splitting probabilities in Joux-Lercier's variation of the FFS can be *cubic* in the reciprocal of the degree – rather than exponential. The reason for this is the richer structure of binary extension fields relative to prime fields, which lends weight to the argument that such fields should be avoided in practice. We also exploit our basic observation to efficiently compute the logarithms of degree two elements — which until now were the bottleneck of the individual logarithm descent phase — which for a range of binary fields results in the fastest $L_{q^n}(1/3)$ algorithm to date, namely $L_{q^n}(1/3, (4/9)^{1/3}) \approx L_{q^n}(1/3, 0.763)$, which is precisely the square root of the complexity of the ordinary FFS, for which $c = (32/9)^{1/3}$.

We emphasise that our relation generation method arises purely as a specialisation of [16], and is thus completely independent of [13]. However, at a high level, our relation generation method *may* be viewed as a form of one-sided pinpointing, but with two central differences to that of [13]. Firstly, we do not need to search for an initial splitting polynomial, since we have an explicit description of *all* such polynomials, i.e., no sieving need take place. Secondly, as members of this family of polynomials have arbitrarily high degree, the other 'random' side can be made to have very small degree, which thus splits

with very high probability. These two differences result in our polynomial time relation generation.

The paper is organised as follows. In §2 we recall the Joux-Lercier variant of the FFS. In §3 we present our specialisation and our analysis of splitting probabilities, while in §4 we present our new descent methods and analyse the complexity of the resulting algorithms. In §5 we present our implementation results and conclude in §6.

2 The Medium-sized Base Field Function Field Sieve

In this section we briefly recall the 2006 FFS variant of Joux and Lercier [16]. Let $\mathbb{F}_{q^n}$ be the finite field in which discrete logarithms are to be solved, where q is a prime power. In order to represent $\mathbb{F}_{q^n}$, choose two univariate polynomials $g_1, g_2 \in \mathbb{F}_q[X]$ of degrees d_1 and d_2 respectively. Then whenever $X - g_1(g_2(X))$ possesses a degree n irreducible factor $F(X)$ over $\mathbb{F}_q$, one can represent $\mathbb{F}_{q^n}$ in two related ways. In particular, let $x \in \mathbb{F}_{q^n}$ be a root of $F(X) = 0$, and let $y := g_2(x)$, so that by construction $x = g_1(y)$ as well. These relations give an explicit isomorphism between $\mathbb{F}_q(x)$ and $\mathbb{F}_q(y)$, both of which represent $\mathbb{F}_{q^n}$.

In the most basic version of the algorithm (which also leads to the best complexity) one chooses $d_1 \approx d_2 \approx \sqrt{n}$, and considers elements of $\mathbb{F}_{q^n}$ represented by:

$$xy + ay + bx + c\,, \quad \text{with} \;\; a, b, c \in \mathbb{F}_q\,.$$

Substituting x by $g_1(y)$, and y by $g_2(x)$, we obtain the following equality in $\mathbb{F}_{q^n}$:

$$xg_2(x) + ag_2(x) + bx + c = yg_1(y) + ay + bg_1(y) + c\,. \tag{1}$$

The factor base consists simply of the degree one elements of $\mathbb{F}_q(x)$ and $\mathbb{F}_q(y)$; then for every triple (a, b, c) for which both sides of (1) split over $\mathbb{F}_q$ — i.e., when all of its roots are in $\mathbb{F}_q$ — in the respective factor bases, one obtains a relation. Determining such triples can naturally be made faster using sieving techniques. Once more than $2q$ such relations have been collected, one performs a linear algebra elimination to recover the individual logarithms. To compute arbitrary discrete logarithms, one uses a 'descent' method, as we detail in §4.

In order to assess the complexity of this algorithm, throughout the paper let $Q = q^n$, let $q = L_Q(1/3, \alpha)$, and let $L_Q(1/3, c_1)$ and $L_Q(1/3, c_2)$ denote the complexity of the sieving and linear algebra phases respectively. As shown in [16], heuristically one has

$$c_1 = \alpha + \frac{2}{3\sqrt{\alpha}} \quad \text{and} \quad c_2 = 2\alpha\,.$$

In order to generate sufficiently many relations, α must satisfy the condition:

$$2\alpha \geq \frac{2}{3\sqrt{\alpha}}\,.$$

For such α's, the complexity of the entire algorithm, including the descent phase, is minimised for $\alpha = 3^{-2/3}$, with resulting complexity $L_Q(1/3, 3^{1/3})$.

3 Specialisation to Binary Fields

We now present a specialisation of the construction of [16] as presented in the previous section, and detail some interesting consequences. From now on let $\mathbb{F}_q$ denote the finite field with 2^l elements.

All of our improvements and observations arise from a rather innocent-looking choice for g_2, namely $y = x^{2^k}$. Our primary motivation for this was to automatically eliminate half of the factor base, since any linear polynomial $(y+a)$ is equal to $(x+a^{2^{-k}})^{2^k}$, and so $\log(y+a) = 2^k \log(x + a^{2^{-k}})$. However, this selection has further serendipitous consequences, the central two being:

- Whenever $k \mid l$ and $l \geq 3k$, the probability of the l.h.s. of (1) splitting over $\mathbb{F}_q$ is approximately 2^{-3k}, instead of the expected $1/(2^k+1)!$. We show that for some asymptotic families of binary fields, this leads to a *polynomial time* algorithm to find the logarithms of all degree one elements of $\mathbb{F}_{q^n}$.
- As surprising as the above result is, for such families, the individual logarithm phase then has complexity $L_{q^n}(1/2)$. Hence one must ensure the complexity of the stages is balanced. Depending on the form of n, we show that the bottleneck of the descent changes from degree two to degree three special-$\mathfrak{q}$, since the x-side has the same form of the l.h.s. of (1), and thus enjoys the same higher splitting probability. This ensures that our claimed new $L_{q^n}(1/3)$ complexities are achieved across all the phases of the algorithm.

In the remainder of the paper we explain these advantages in more detail. In addition to the above two observations, for certain extensions which possess Galois-invariant factor bases, the use of non-prime base fields can induce extra automorphisms, which reduce its size further, see §5. Other practical speed ups arise from our choice $y = x^{2^k}$. The matrix-vector multiplications in Lanczos' algorithm consists of only cyclic rotations, i.e., shifts mod $q^n - 1$, and so no multiplications need to be performed. Furthermore, in the descent phase, one ordinarily needs to perform special-$\mathfrak{q}$ eliminations in both function fields. However, due to the simple relation between x and y, one is free to map from one side to the other in order to increase the probability of smoothness. One can also balance the degrees of both sides by utilising other auxiliary function fields arising from passing a power of 2 from the x-side to other side; this not only provides a practical speed up but is core to our new complexity results, see §4.

3.1 Higher Splitting Probabilities

Throughout this section, rather than use the field elements x, y as variables, we use X, Y to emphasise that the stated results are valid in the univariate polynomial ring over $\mathbb{F}_q$, which is implicitly either $\mathbb{F}_q[X]$ or $\mathbb{F}_q[Y]$, depending on which side of (1) is involved.

Assume $1 < k < l$. When $Y = X^{2^k}$ the l.h.s. of (1) becomes

$$X^{2^k+1} + aX^{2^k} + bX + c\,. \tag{2}$$

Assuming $c \neq ab$ and $b \neq a^{2^k}$, this polynomial may be transformed (up to a scalar factor) into the polynomial

$$f_B(\overline{X}) = \overline{X}^{2^k+1} + B\overline{X} + B\,, \quad \text{with} \quad B = \frac{(b + a^{2^k})^{2^k+1}}{(ab + c)^{2^k}}\,, \tag{3}$$

via

$$X = \frac{ab + c}{b + a^{2^k}}\,\overline{X} + a\,.$$

The polynomial f_B is related to $P_A(\overline{X}) = \overline{X}^{2^k+1} + \overline{X} + A$, which is well-studied in the literature, having arisen in several contexts including finite geometry, difference sets, as well as determining cross correlation between m-sequences; see references in [12] for further details.

We have the following theorem due to Bluher [3] (and refined in the binary case by Helleseth and Kholosha [12]), which counts the number of $B \in \mathbb{F}_q$ for which f_B splits over $\mathbb{F}_q$.

Theorem 1. *[12, Thm. 1] Let $d = \gcd(l, k)$. Then the number of $B \in \mathbb{F}_{2^l}^\times$ such that $f_B(\overline{X})$ has exactly $2^d + 1$ roots over $\mathbb{F}_{2^l}$ is*

$$\begin{cases} \dfrac{2^{l-d} - 1}{2^{2d} - 1} & \text{if } l/d \text{ odd,} \\[2ex] \dfrac{2^{l-d} - 2^d}{2^{2d} - 1} & \text{if } l/d \text{ even.} \end{cases}$$

Theorem 1 of [12] also states that f_B can have no more than $2^d + 1$ roots in $\mathbb{F}_q$, and so if $\gcd(l, k) < k$ then f_B can not split. Hence we must have $k \mid l$ for our application. Indeed we must also have $l \geq 3k$ in order for there to be at least one such B. Observe that under these two conditions, for B chosen uniformly at random from $\mathbb{F}_q$, the probability that f_B splits completely over $\mathbb{F}_q$ is approximately $1/2^{3k}$ – far higher than the splitting probability $1/(2^k+1)!$ for a degree $2^k + 1$ polynomial chosen uniformly at random.

Furthermore, the set S_B of all such B can be computed explicitly, without needing to perform any factorisations or smoothness tests. Indeed, the proof of Prop. 5 in [12] gives an explicit parameterisation of all such B: for $u \in G = \mathbb{F}_{2^l} \setminus \mathbb{F}_{2^{2k}}$, we have

$$S_B = \mathrm{Im}\left(u \longrightarrow \frac{(u + u^{2^{2k}})^{2^k+1}}{(u + u^{2^k})^{2^{2k}+1}}\right).$$

By analysing the form of this map, one can avoid obtaining repeated images. However, even a naive enumeration of elements of G requires at most $\widetilde{O}(q)$ $\mathbb{F}_q$-operations, which is comparable to the complexity of relation generation, as we now show.

3.2 Relation Generation

By exploiting the above transformation of (2) to (3) and the list S_B of precomputed B's for which (3) splits, one can construct polynomials of the form (2) which always split completely over $\mathbb{F}_q$. Indeed, for any (a, b) for which $b \neq a^{2^k}$, and for each $B \in S_B$, we simply compute via (3) the corresponding unique $c \in \mathbb{F}_q$. This ensures that (2) splits and therefore requires no sieving whatsoever.

In order to obtain a relation, we also require that

$$Y g_1(Y) + b g_1(Y) + aY + c \tag{4}$$

splits over $\mathbb{F}_q$, which we assume occurs with probability $1/(d_1+1)!$ for randomly chosen g_1. Since $|L_B| \approx q/2^{3k}$, for each (a, b) we expect to obtain

$$\frac{q}{2^{3k}\,(d_1+1)!}$$

relations. Since we need q relations, we expect to require about $2^{3k}\,(d_1 + 1)!$ pairs (a, b) to obtain sufficiently many. For each pair (a, b) this costs $O(q/2^{3k})$ 1-smoothness tests, or $\tilde{O}(q/2^{3k})$ $\mathbb{F}_q$-operations. Hence the total cost is $\tilde{O}(q\,(d_1+1)!)$. Finally, in order for there to be sufficiently many relations, we must have

$$\frac{q^3}{2^{3k}\,(d_1+1)!} > q\,, \quad \text{or} \quad q^2 > 2^{3k}\,(d_1+1)!\,.$$

Since we insist that $l \geq 3k$, having $q > (d_1+1)!$ is sufficient. In §4 we consider the impact of this approach on the full DLP complexity in two cases when $q = L_{q^n}(1/3, \alpha)$ and $n \approx 2^k \cdot d_1$: firstly for $2^k \approx d_1$ and secondly for $2^k \gg d_1$. However, we now consider the relation generation complexity when the base field cardinality is polynomially related to the extension degree.

3.3 Polynomial Time Relation Generation

With a view to reducing the complexity of degree one relation generation to a minimum for some example fields, we choose k as large as possible such that $k \mid l$ and $l \geq 3k$, and set d_1 to be as small as possible, assuming a g_1 can be found with $X - g_1(X^{2^k})$ possessing a degree n irreducible factor. Experimentally it seems that $d_1 = 3$ (or possibly $d_1 = 4$) is sufficient to produce an irreducible of any degree $n \leq 2^k$, for q sufficiently large. Of course, n may be as large as $2^k \cdot d_1$ in this case.

Writing $l = k \cdot k'$ with $k' \geq 3$ a constant, and $n \approx 2^k \cdot d_1$ with d_1 constant, as $l \to \infty$, the logarithms of the degree one factor base elements of $\mathbb{F}_{q^n}$ can be computed in heuristic *polynomial time*. In particular, as $n \approx 2^k \cdot d_1 = 2^{l/k'} \cdot d_1$, we have

$$Q = q^n \approx 2^{l \cdot 2^{l/k'} \cdot d_1}\,.$$

As $l \to \infty$, we therefore have

$$\frac{\log Q}{\log \log Q} = O(2^{l/k'}) \, .$$

The cost of relation generation is $\widetilde{O}(q\,(d_1+1)!) = \widetilde{O}(q) = \widetilde{O}(2^l) = \widetilde{O}(\log^{k'} Q)$, whereas the cost of sparse linear algebra, using Lanczos' algorithm [18] for instance, is the product of the row weight and the square of number of variables, namely

$$(2^{l/k'} + d_1)\,\widetilde{O}(q^2) = \widetilde{O}(\log^{2k'+1} Q) \, .$$

For the optimal choice $k' = 3$ the complexity is therefore $\widetilde{O}(\log^7 Q)$. We summarise this in the following:

Heuristic Result 1. *Let $q = 2^l$ with $l = k \cdot k'$ and $k' \geq 3$ a constant, let $d_1 \geq 3$ be constant, and assume $n \approx 2^k \cdot d_1$. Assuming that $Yg_1(Y) + aY + bg_1(Y) + c$ splits over $\mathbb{F}_q$ with probability $1/(d_1+1)!$ over all triples $(a,b,c) \in (\mathbb{F}_q)^3$, the logarithms of all degree one elements of $\mathbb{F}_{q^n}$ can be computed in time $\widetilde{O}(\log^{2k'+1} Q)$.*

Note that the set of degree one elements is always defined relative to a particular representation of $\mathbb{F}_{q^n}$. As it is easy to switch between any two representations of a finite field [20], one can always map to our $\mathbb{F}_q(x)$ first. Note also that the statement of Heuristic Result 1 implicitly assumes that the factor base contains a generator of $\mathbb{F}_{q^n}^\times$. A result of Chung proves that for all prime powers s and all $r \geq 1$ such that $s > (r-1)^2$, if $\mathbb{F}_{s^r} = \mathbb{F}_s(x)$ then $\{x + a \mid a \in \mathbb{F}_s\}$ generates $\mathbb{F}_{s^r}^\times$ [4, Thm. 8]. In our context we therefore need $q^{k'} > (n-1)^2 \approx q^2 \cdot d_1^2$ in order for our DLP algorithm to work, which is satisfied for our q and small d_1. However, the issue of whether there exists a generator in the stated factor base remains an open problem in general, see for instance [26].

3.4 An Extreme Case: $n = 2^k \pm 1$

If $n = 2^k \pm 1$ then the degree one relation generation becomes extremely fast. In particular, if $g_1(X) = \gamma X^{\mp 1}$ then as $g_2(X) = X^{2^k}$, we obtain the polynomials $X^{2^k \pm 1} + \gamma$. Furthermore, if $k \mid l$ then $X^{2^k \pm 1} + \gamma$ is irreducible whenever γ has no root of prime order $p \mid (2^k \pm 1)$. In both cases, (4) has degree two and splits with probability 1/2.

Table 1 contains timing data for relation generation for a family of fields with $q = 2^{3k}$ and $n = 2^k - 1$, which incorporates the factor base reduction technique arising from quotienting out by the action of the k-th power of Frobenius, which has order $3n$, see §5. We used an AMD Opteron 6128 processor clocked at 2.0 GHz. Note that the time is quasi-cubic in the bitlengh, in accordance with the discussion preceeding Heuristic Result 1.

Table 1. Relation generation times for $q = 2^{3k}$ and $n = 2^k - 1$

k	$\log_2(q^n)$	#vars	time
7	2667	5506	$2.3s$
8	6120	21932	$15.0s$
9	13797	87554	$122s$
10	30690	349858	$900s$

4 Individual Logarithms and Complexity Analysis

As unexpected as Heuristic Result 1 is, it does not by itself solve the DLP. Using a descent method à la [16, 5], computing individual logarithms unfortunately then has complexity $L_{q^n}(1/2)$. Hence one can not allow the extension degree n to grow as fast as Theorem 1 permits; it must be tempered relative to the base field size. With this in mind, we now consider the complexity of the descent, for q and n appropriately balanced so that the total complexity is $L_{q^n}(1/3)$.

For a generator $g \in \mathbb{F}_{q^n}^\times$ and a target element $h \in \langle g \rangle$, the descent proceeds by first finding an $i \in \mathbb{N}$ such that $z = h\,g^i$ is m-smooth for a suitable m, i.e., so that all of the irreducible factors of z have degrees $\leq m$. The goal of the descent is to eliminate every irreducible factor of z, by expressing each as a product of smaller degree irreducibles recursively, until only degree one elements remain, whose logarithms are known. We do so using the special-$\mathfrak{q}$ lattice approach from [16], as follows.

Let $p(x)$ be a degree d irreducible (considered as an element of $\mathbb{F}_q[X]$) which we wish to eliminate. Since $y = x^{2^k}$, we have

$$p(x)^{2^k} = \overline{p}(x^{2^k}) = \overline{p}(y)\,,$$

where the coefficients of $\overline{p}$ are those of p, powered by 2^k. Note that we also have

$$\overline{p}(y)^{2^{-k}} = p(x)\,,$$

and hence we can freely choose to eliminate p using either the x-side or the y-side of (1). For convenience we focus on the y-side. The corresponding lattice $L_{\overline{p}}$ is defined by:

$$L_{\overline{p}(Y)} = \{(w_0(Y), w_1(Y)) \in \mathbb{F}_q[Y]^2 : w_0(Y)\,g_1(Y) + w_1(Y) \equiv 0 \pmod{\overline{p}(Y)}\}\,.$$

A basis for this lattice is $(0, \overline{p}(Y)), (1, g_1(Y) \pmod{\overline{p}(Y)})$, which is clearly unbalanced. Using the extended Euclidean algorithm, we may construct a balanced basis $(u_0(Y), u_1(Y))$, $(v_0(Y), v_1(Y))$ for which the degrees are $\approx d/2$. Then for any $r(Y), s(Y) \in \mathbb{F}_q[Y]$ with $r(Y)$ monic we have

$$(w_0(Y), w_1(Y)) = \big(r(Y)u_0(Y) + s(Y)v_0(Y)\,,\, r(Y)u_1(Y) + s(Y)v_1(Y)\big) \in L_{\overline{p}(Y)}$$

and thus $\mathrm{RHS}(Y) \equiv 0 \pmod{\overline{p}(Y)}$, where

$$\mathrm{RHS}(Y) = w_0(Y)\, g_1(Y) + w_1(Y)\,.$$

When $\mathrm{RHS}(Y)/\overline{p}(Y)$ is $(d-1)$-smooth, we also check whether $\mathrm{LHS}(X)$ is also $(d-1)$-smooth, where

$$\mathrm{LHS}(X) = w_0(X^{2^k})\, X + w_1(X^{2^k})\,.$$

When both sides are $(d-1)$-smooth, we may replace $\overline{p}(Y)$ with a product of irreducibles of degree at most $d-1$, and then recurse.

Let $Q = q^n$. As in [16], we assume there is a parameter α such that:

$$n = \frac{1}{\alpha}\left(\frac{\log Q}{\log\log Q}\right)^{2/3}, \qquad q = \exp\left(\alpha\,\sqrt[3]{\log Q \cdot \log^2 \log Q}\right). \tag{5}$$

The three stages to consider are relation generation, linear algebra, and the descent, whose complexities we denote by $L_Q(1/3, c_1)$, $L_Q(1/3, c_2)$ and $L_Q(1/3, c_3)$, respectively. The total complexity is therefore $L_Q(1/3, c)$, where $c = \max\{c_1, c_2, c_3\}$. We next consider degree 2 elimination and then two special cases of field representation.

4.1 Degree 2 Elimination

We begin with degree 2 elimination as firstly it is the bottleneck in the descent, and secondly because one can exploit the higher splitting probability of the polynomials (2) as well. Let $\overline{p}(Y)$ be a degree 2 irreducible to be eliminated. A reduced basis $(u_0(Y), u_1(Y)), (v_0(Y), v_1(Y))$ for the lattice $L_{\overline{p}(Y)}$ can be found with degrees $(1,0), (0,1)$. Hence with r normalised to be 1 and $s \in \mathbb{F}_q$, we have

$$(w_0(Y), w_1(Y)) = \big(u_0(Y) + s\, v_0(Y), u_1(Y) + s\, v_1(Y)\big) \in L_{\overline{p}(Y)}$$

with degrees $(1,1)$. We have thus

$$w_0(Y)\, g_1(Y) + w_1(Y) \equiv 0 \pmod{\overline{p}(Y)}\,,$$

and so the remaining factor has degree $d_1 - 1$. The corresponding polynomial $\mathrm{LHS}(X)$ is

$$w_0(X^{2^k})\, X + w_1(X^{2^k})\,, \tag{6}$$

which is of the form $X^{2^k+1} + aX^{2^k} + bX + c$, and as a consequence of Theorem 1, it splits over $\mathbb{F}_q$ with probability approximately 2^{-3k}. However, as with relation generation, we can also ensure that $\mathrm{LHS}(X)$ always splits, with the following technique. Writing the basis elements explicitly as $(Y + u_{00}, u_{10}), (v_{00}, Y + v_{10})$, and with $r = 1$ and $s \in \mathbb{F}_q$ the lattice elements are $(w_0(Y), w_1(Y)) = (Y + u_{00} + sv_{00}, sY + u_{10} + sv_{10})$. Thus combining (6) and (3), for each $B \in S_B$ we find the set of roots $s \in \mathbb{F}_q$ that satisfy the $\mathbb{F}_q[S]$ polynomial

$$B \cdot (v_{00}S^2 + (u_{00} + v_{10})S + u_{10})^{2^k} + (S^{2^k} + v_{00}S + u_{00})^{2^k+1} = 0\,,$$

by computing its GCD with $S^q + S$. This technique extracts all such s algebraically for any B, which ensures that $\mathrm{LHS}(X)$ automatically splits.

On average one expects there to be one such $s \in \mathbb{F}_q$ for each B. Then for each such s we check whether $\mathrm{RHS}(Y)/\overline{p}(Y)$ splits, which we assume occurs with probability $1/(d_1-1)!$. In general we therefore need sufficiently many B's in S_B for this to occur with good probability, i.e., that $q/2^{3k} > (d_1-1)!$.

4.2 Case 1: $n \approx 2^k \cdot d_1$ and $2^k \approx d_1$

In this section we will show the following:

Heuristic Result 2 (i). *Let $q = 2^l$, let $k \mid l$ and let n be such that (5) holds. Then for $n \approx 2^k \cdot d_1$ where $2^k \approx d_1$, the DLP can be solved with complexity $L_Q(1/3, (8/9)^{1/3}) \approx L_Q(1/3, 0.961)$.*

This is the simplest case we present; however for the sake of completeness and ease of exposition, we explicitly tailor the derivation presented in §3.2. By our relation generation method, the l.h.s. polynomial (2) always splits, whereas the probability of (4) being smooth is approximately $1/\sqrt{n}!$. Using the standard approximation $\log n! \approx n \log n$, the logarithm of the probability P of both sides being smooth is therefore:

$$\log P \approx -\sqrt{n} \log \sqrt{n} = -\frac{1}{2}\sqrt{n} \log n \,.$$

The size of the sieving space is $q^3/2^{3k}$, and since we require q relations we must have:

$$\frac{q^3\, P}{2^{3k}} \geq q \,, \quad \text{or} \quad 2 \log q \geq \left(\frac{3}{2} + \frac{\sqrt{n}}{2}\right) \log n \approx \frac{\sqrt{n}}{2} \log n \,.$$

Ignoring low order terms, by (5) this is equivalent to

$$2\alpha \geq \frac{1}{3\sqrt{\alpha}} \,, \quad \text{or} \quad \alpha \geq 6^{-2/3} \,. \tag{7}$$

Given that we require q relations, the expected time to collect these relations is

$$\frac{q}{P} = L_Q\Big(1/3 \,,\, \alpha + \frac{1}{3\sqrt{\alpha}}\Big) \,,$$

and hence $c_1 = \alpha + \frac{1}{3\sqrt{\alpha}}$. Since the linear algebra is quadratic in the size of the factor base, we also have $c_2 = 2\alpha$.

For the descent, as in [16], let the smoothness bound be $m = \mu\sqrt{n}$. Then the probability of finding such an expression is

$$1 \,/\, L_Q\Big(1/3 \,,\, \frac{1}{3\mu\sqrt{\alpha}}\Big) \,.$$

If the descent is to be no more costly than either the relation generation or the linear algebra, then we must have

$$\frac{1}{3\mu\sqrt{\alpha}} \;\leq\; \max\left\{\alpha + \frac{1}{3\sqrt{\alpha}}\,,\, 2\alpha\right\}. \tag{8}$$

We also need to ensure three further conditions are satisfied. Firstly, that the cost of all the special-$\mathfrak{q}$ eliminations is no more than $L_Q(1/3, \max\{c_1, c_2\})$. Secondly, that there are enough (r, s) pairs to ensure a relation is found. And thirdly, that during the descent the degrees of the polynomials being tested for smoothness is really descending.

By the discussion in §4.1, in order to eliminate degree 2 elements we need $q \geq 2^{3k}\,(d_1 - 1)!$, or equivalently,

$$\alpha \geq \frac{1}{3\sqrt{\alpha}}\,, \quad \text{or} \quad \alpha \geq 3^{-2/3}\,.$$

Since for degree 3 special-$\mathfrak{q}$ LHS(X) will not have the form (2), we need to check that the smoothness probability does not impose an extra condition on α. For $\overline{p}(Y)$ a degree 3 irreducible to be eliminated, a reduced basis $(u_0(Y), u_1(Y))$, $(v_0(Y), v_1(Y))$ for the lattice $L_{\overline{p}(Y)}$ can be found with degrees $(1,1), (0,2)$. Hence with r now allowed to be monic of degree one and $s \in \mathbb{F}_q$, we have

$$(w_0(Y), w_1(Y)) = \big((Y + r_0)u_0(Y) + s\,v_0(Y)\,,\, (Y + r_0)u_1(Y) + s\,v_1(Y)\big) \in L_{\overline{p}(Y)},$$

with degrees $(2,2)$. As before, we have

$$w_0(Y)\,g_1(Y) + w_1(Y) \equiv 0 \pmod{\overline{p}(Y)}\,,$$

and the corresponding polynomial LHS(X) is

$$w_0(X^{2^k})\,X + w_1(X^{2^k})\,.$$

Once divided by $\overline{p}(Y)$, the degree of the Y-side is $d_1 - 1 \approx \sqrt{n}$ while the degree of the X-side is $2^{k+1} + 1 \approx 2\sqrt{n}$. The logarithm of the probability that a degree n polynomial over $\mathbb{F}_q$ is m-smooth, for q and n tending to infinity but m fixed, can be estimated by $-(n/m)\log(n/m)$, as shown in [16]. Therefore the log of the probability P of both sides being 2-smooth is:

$$\log P \approx -\frac{\sqrt{n}}{2}\log\frac{\sqrt{n}}{2} - \frac{2\sqrt{n}}{2}\log\frac{2\sqrt{n}}{2} \approx -\frac{3}{2}\sqrt{n}\log\frac{\sqrt{n}}{2} \approx -\frac{3}{4}\sqrt{n}\log n\,,$$

and therefore $P = 1/L_Q(1/3, \frac{1}{2\sqrt{\alpha}})$. Since the (r, s) search space has size q^2 (which is also the complexity of the linear algebra), we require that

$$2\alpha \geq \frac{1}{2\sqrt{\alpha}} \quad \text{or} \quad \alpha \geq 16^{-1/3}\,.$$

Since $16^{-1/3} < 3^{-2/3}$, this imposes no additional constraint on α. Hence we can set $\alpha = 3^{-2/3}$, and one can check that in this case, $c_1 = c_2 = c_3 = 2\alpha$, giving complexity

$$L_Q(1/3, (8/9)^{1/3}) \approx L_Q(1/3, 0.961)\,,$$

which is precisely the complexity Joux obtained using either optimal one-sided, or advanced pinpointing [13]. Furthermore for this α, (8) implies that $\mu \geq 1/2$. For an upper bound, note that for special-$\mathfrak{q}$ of degree $\mu\sqrt{n}$, the degree of RHS(Y) is about $\sqrt{n}(1-\mu/2)$, while the degree of LHS(X) is about $\mu n/2$, so that $\mu < 2$ ensures that the descent is effective.

4.3 Case 2: $n \approx 2^k \cdot d_1$ and $2^k \gg d_1$

In this section we will show the following:

Heuristic Result 2 (ii). *Let $q = 2^l$, let $k \mid l$ and let n be such that (5) holds. Then for $n \approx 2^k \cdot d_1$ where $2^k \gg d_1$, the DLP can be solved with complexity between $L_Q(1/3, (4/9)^{1/3}) \approx L_Q(1/3, 0.763)$ and $L_Q(1/3, (1/2)^{1/3}) \approx L_Q(1/3, 0.794)$.*

Observe that interestingly, these two complexities are precisely the square-roots of the complexities of Coppersmith' algorithm [5], for which $c = (32/9)^{1/3}$ and $4^{1/3}$, the lower of the two being the complexity of the ordinary FFS [1, 14].

For n and q of the form (5), we claim that $c_1 = \alpha$, $c_2 = 2\alpha$, and that there are sufficiently many relations available. In particular, if we write $d_1 = n^\beta$ with $\beta < 1/2$ and $2^k = n^{1-\beta}$, then again by our relation generation method, the l.h.s. polynomial (2) always splits, and the log of the probability P of both sides being 1-smooth is:

$$\log P \approx -\beta n^\beta \log n.$$

By (5) we have

$$\begin{aligned} -\beta n^\beta \log n &\approx -\frac{2\beta}{3\alpha^\beta}\left(\frac{\log Q}{\log\log Q}\right)^{2\beta/3}(\log\log Q) \\ &= -\frac{2\beta}{3\alpha^\beta}(\log Q)^{2\beta/3}(\log\log Q)^{1-2\beta/3}\,. \end{aligned}$$

Hence the expected time of the relation generation is

$$\frac{q}{P} = L_Q(1/3, \alpha) \cdot L_Q\Big(2\beta/3, \frac{2\beta}{3\alpha^\beta}\Big)\,.$$

For $\beta < 1/2$ the second term on the right is absorbed by the $o(1)$ term in the first term, and hence $c_1 = \alpha$ and $c_2 = 2\alpha$. The size of the sieving space is $q^3/2^{3k}$, and since we require q relations we must have:

$$\frac{q^3\,P}{2^{3k}} \geq q\,, \quad \text{or} \quad L_Q(1/3, 2\alpha) \geq L_Q\Big(2\beta/3, \frac{2\beta}{3\alpha^\beta}\Big)\,,$$

which holds for any $\alpha > 0$ when $\beta < 1/2$.

For the descent (as for Case 1) the cost of finding the first $\mu\sqrt{n}$-smooth relation is $L_Q(1/3, \frac{1}{3\mu\sqrt{\alpha}})$. And as before, for degree 2 special-$\mathfrak{q}$, the X-side has the same form and the condition on q arising from the search space being sufficiently large is always satisfied, since

$$q \geq 2^{3k}\,(d_1 - 1)! = n^{3(1-\beta)}\; L_Q\Big(2\beta/3, \frac{2\beta}{3\alpha^{\beta}}\Big)\,,$$

which holds for any $\alpha > 0$ when $\beta < 1/2$.

Hence degree 3 special-$\mathfrak{q}$ are the bottleneck. As in the first case, with r allowed to be monic of degree one and $s \in \mathbb{F}_q$, the degree of RHS(Y) is $d_1 - 1$ while the degree of LHS(X) is $2^{k+1} + 1$. These degrees are clearly unbalanced. However, we can employ the following tactic to balance them.

Since $g_1(Y)^{2^k} + Y = 0$, we let $X' = g_1(Y)^{2^a}$ and thus $Y = X'^{2^{k-a}}$. We are free to choose any $1 < a < k$, as an elimination of a special-$\mathfrak{q}$ using Y and X' can be written in terms of Y and X by powering by a power of 2. With r allowed to be monic of degree one and $s \in \mathbb{F}_q$ we have $(w_0(Y), w_1(Y)) \in L_{\overline{p}(Y)}$ with degrees $(2, 2)$, and our new expressions become

$$w_0(Y)\, g_1(Y)^{2^a} + w_1(Y) \equiv 0 \pmod{\overline{p}(Y)}\,.$$

The corresponding polynomial LHS(X') is

$$w_0(X'^{2^{k-a}})\, X' + w_1(X'^{2^{k-a}})\,.$$

Assuming the degrees are (approximately) the same, taking logs we have

$$k - a + 1 = \log_2(d_1) + a\,, \quad \text{or} \quad a = (k + 1 - \log_2(d_1))/2\,.$$

Since a must be an integer, rather than a real variable, we must choose the nearest integer to this value. In the best case, we can take a to be this exact value, and consequently both degrees are $\sqrt{2d_1}\,2^{k/2} = \sqrt{2}\sqrt{n}$. Therefore the log of the probability P of both sides being 2-smooth is:

$$\log P \approx -\frac{\sqrt{2}}{2}\sqrt{n}\log\Big(\frac{\sqrt{2}}{2}\sqrt{n}\Big) - \frac{\sqrt{2}}{2}\sqrt{n}\log\Big(\frac{\sqrt{2}}{2}\sqrt{n}\Big) \approx -\frac{\sqrt{2}}{2}\sqrt{n}\log n\,,$$

and hence $P = L_Q(1/3, -\frac{\sqrt{2}}{3\sqrt{\alpha}})$. In order to have a sufficiently large search space we must therefore have

$$2\alpha \geq \frac{\sqrt{2}}{3\sqrt{\alpha}}\,, \quad \text{or} \quad \alpha \geq 18^{-1/3}\,.$$

For $\alpha = 18^{-1/3}$ the descent initiation stipulates that $\mu \geq \alpha^{-3/2}/6 = 1/\sqrt{2}$, and any $\mu \in [1/\sqrt{2}, \sqrt{n})$ suffices. We therefore have a total complexity of

$$L_Q(1/3, 2\alpha) = L_Q(1/3, (4/9)^{1/3}) \approx L_Q(1/3, 0.763)\,.$$

On the other hand when we need to round a to the nearest integer, the degrees can become unbalanced so that the degree of one side is up to double the degree of the other. In this case a simple calculation shows that the optimal α is $16^{-1/3}$, giving a complexity of

$$L_Q(1/3, 2\alpha) = L_Q(1/3, (1/2)^{1/3}) \approx L_Q(1/3, 0.794)\,.$$

Naturally, for a ratio of degrees in $(1/2, 2)$, we get c-values in between. This situation is redolent of Coppersmith's algorithm [5], in which precisely the same issue arises when forcing a real variable to take integer arguments only.

Note that this degree balancing technique also works for special-$\mathfrak{q}$ of any degree, making the descent far more rapid than for Case 1.

Remark 1. Observe that the best-case complexity with $c = (4/9)^{1/3}$ is precisely the complexity of the oracle-assisted Static Diffie-Hellman Problem in finite fields of small characteristic [17, §3]. Our result may therefore seem unsurprising, since the complexity of computing the logarithms of the factor base elements is never more than the complexity of the descent, and is thus effectively free. However, this reasoning overlooks the fact that we are working with a medium-sized base field, as opposed to the traditional FFS setting with a very small base field. In contrast to the result in [17, §3], our complexities depend crucially on our degree two elimination method, in addition to the fast computation of degree one logarithms.

5 Application to the DLP in $\mathbb{F}_{2^{1971}}$ and $\mathbb{F}_{2^{3164}}$

In this section we provide details of our implementation for discrete logarithm computations in the finite fields with 2^{1971} (as announced in [9]) and 2^{3164} elements, respectively.

5.1 Discrete Logarithms in $\mathbb{F}_{2^{1971}}$

In order to represent the finite field with 2^{1971} elements we first defined $\mathbb{F}_q = \mathbb{F}_{2^{27}}$ by $\mathbb{F}_2[T]/(T^{27} + T^5 + T^2 + T + 1)$. Denoting by t a root of this irreducible in $\mathbb{F}_{2^{27}}$ we defined $\mathbb{F}_{2^{1971}} = \mathbb{F}_{q^{73}}$ by $\mathbb{F}_q[X]/(X^{73} + t)$. For x a root of $X^{73} + t$ in $\mathbb{F}_{q^{73}}$, we defined y by $y := x^8$, and we therefore also have $x = t/y^9$.

Since we use a Kummer extension, the elements of the factor base are related via the generator of the Galois group of $\mathbb{F}_{q^{73}}/\mathbb{F}_q$ [16, 13], and one can therefore quotient out by the action of this automorphism to reduce the number of variables from 2^{27} to $\approx 2^{27}/73$. As stated in §3, we can take this idea even further. In fact, $x^{2^9} = c\,x$ for $c = t^7 \in \mathbb{F}_q$, so the map $\sigma : a \to a^{2^9}$ is an additional automorphism which preserves the set of degree one factor base elements. The map σ^3 equals the Frobenius $a \to a^q$ (of order 73) and hence σ generates a group G of order 219. Considering the orbits of G acting on the factor base elements, we find 612 864 orbits of full size 219, seven of size 73,

and one of size 1, resulting in $N = 612\,872$ orbits, which gives the number of factor base variables.

Since the degrees of the polynomials relating x and y are nearly balanced, the complexity of our relation generation falls into Case 1 in §4.2, which matches Joux's optimal one-sided, or advanced pinpointing for Kummer extensions. However, for Kummer extensions for which the degrees are balanced — as opposed to being very skewed as in §3.4 where $2^k \gg d_1$ — the advanced pinpointing is faster in practice, and so we used it for relation generation. We computed approximately $10N$ relations in about 14 core-hours computation time. For simplicity, we keep only those relations with distinct factors; this ensures that each entry of the relation matrix is a power of two, and hence all element multiplications in the matrix-vector products consist of cyclic rotations modulo $2^{1971} - 1$.

After relation generation, we performed structured Gaussian elimination (SGE) (in a version based on [15]) to reduce the number of variables and thus to decrease the cost for the subsequent linear algebra step. During our experiments we made the observation that additional equations are indeed useful for reducing the number of variables. However, the benefit of SGE is unclear as the row weight is being increased. We therefore stopped the SGE at this point, which resulted in a $528\,812 \times 527\,766$ matrix of constant row weight 19. The running time here was about 10 minutes on a single core.

We obtained the following partial factorisation of $2^{1971} - 1$:

$$
\begin{aligned}
&7 \cdot 73^2 \cdot 439 \cdot 3943 \cdot 262657 \cdot 2298041 \cdot 10178663167 \cdot 27265714183 \cdot 9361973132609 \\
&\cdot 1406791071629857 \cdot 5271393791658529 \cdot 6711658986174134 17 \cdot 2762194134676763431 \\
&\cdot 4815314615204347717321 \cdot 421859275529837631474313737 19 \\
&\cdot 2206836284671480716039792791233921 6441 \\
&\cdot 7813353937053182028691100246843597594051790 97 \cdot C_{338} \,,
\end{aligned}
$$

where C_{338} is a 338-digit composite. We took as our modulus for the linear algebra step the product of C_{338} and the six largest prime factors of the cofactor, which has 507 digits. We applied a parallel version of Lanczos' algorithm (see [18]) using OpenMP on an SGI Altix ICE 8200EX cluster using Intel (Westmere) Xeon E5650 hex-core processors and GNU Multi-Precision library [11], taking 2220 core-hours in total.

For the DLP we took as (a presumed) generator $g = x + 1 \in \mathbb{F}_{2^{1971}}^{\times}$ and the target element was set as usual to be

$$x_\pi = \sum_{i=0}^{72} \tau(\lfloor \pi \cdot q^{i+1} \rfloor \bmod q)\, x^i \,,$$

where τ takes the binary representation of an integer and maps to $\mathbb{F}_q$ via $2^i \mapsto t^i$. We first solved the target logarithm in the subgroups of order the first 11 terms in the factorisation using either linear search or Pollard's rho [22].

The descent proceeded by first finding an $i \in \mathbb{N}$ such that

$$x_\pi \, g^i = z_1/z_2 \, ,$$

where both z_1 and z_2 were 7-smooth. We implemented the descent in such a way that at the early phase of the algorithm the expected subsequent costs are as small as possible. This means that we try to find factorisations which consist of as many small degree factors as possible. We used about 40 core-hours to find an exponent i with favourable factorisation patterns and found $i = 47\,147\,576$ to be a good choice. We then spent about 3 hours to perform the descent down to degree 3. As stated in §3 and §4, at each stage during the descent, we can eliminate a given special-$\mathfrak{q}$ on either the x-side or on the y-side, one of which may be much faster. Computing the elimination probabilities we found that eliminating on the y-side is always faster. Indeed, for degree 2 special-$\mathfrak{q}$ we *must* perform this on the y-side, as it is not possible to do so on the x-side, due to the factorisation patterns of (2).

At this point we were left with 103 special-$\mathfrak{q}$ of degree 3, as opposed to the ≈ 500 expected with a random 7-smooth split of $x_\pi \, g^i$. The expected cost of eliminating each of these is $2^{25.1}$ 2-smoothness tests. These special-$\mathfrak{q}$ elements were resolved on the same SGI Altix ICE 8200EX cluster in about 850 core-hours, using Shoup's Number Theory Library [24], resulting in 1140 special-$\mathfrak{q}$ elements of degree 2. Using the technique of §4.1, we reduced the cost of the elimination of each of these by a factor of $2^9 = 2^{3k}$, and all their logarithms were computed in 5 core-hours, completing the descent.

Thus the running time for solving an instance of the discrete logarithm problem completely in the finite field $\mathbb{F}_{2^{1971}}$ sums to $14 + 2220 + 898 = 3132$ core-hours in total. Finally, we found that $\log_g(x_\pi)$ equals

$$\begin{aligned}
&1199298421535410686609114637198885584518685275544716335236895900760902198 79\\
&5745784008181148775933944656038305197825417423602365358899373622007711173 61\\
&6782694231011634031353555222808041139032152735559059010822822482400219287 87\\
&8207304028565280573096588688279004416835100344085961912427000601289864337 52\\
&1100022143802898875460611252245879711978727508058465196231404376457393629 38\\
&2354173616116810825627780459657892709561158924173579400674739684346062992 68\\
&2942919573782264511826207837453495025029601399274531964897400652447954895 83\\
&2792082788276833244090734244664394109767021620395395133776731154834 39 \, .
\end{aligned}$$

5.2 Discrete Logarithms in $\mathbb{F}_{2^{3164}}$

For this case we defined $\mathbb{F}_q = \mathbb{F}_{2^{28}} = \mathbb{F}_2[T]/(T^{28} + T + 1)$. We denote by t a root of this irreducible in $\mathbb{F}_{2^{28}}$. Furthermore, let $\mathbb{F}_{q^{113}} = \mathbb{F}_q[X]/(X^{113} + t)$ and denote by x a root of $X^{113} + t$ in $\mathbb{F}_{2^{3164}}$. We defined y by $y = x^{16}$, and we therefore also have $x = t/y^7$.

As in the previous section we use the Kummer extension idea of [16, 13] to reduce the size of the factor base. Again we can use a larger group than just the

Galois group of $\mathbb{F}_{q^{113}}/\mathbb{F}_q$, since $x^{2^{14}} = c\,x$ for $c = t^9 + t^8 + t^5 + t^4 \in \mathbb{F}_q$ and thus the map $\sigma : a \to a^{2^{14}}$ is an additional factor base preserving automorphism. The map σ^2 equals the Frobenius $a \to a^q$ and hence σ generates a group G of order 226. Considering the orbits of G acting on the factor base elements, we find $N = 1\,187\,841$ orbits in total, which gives the number of factor base variables.

For relation generation, since $16 > 7$ the degrees are unbalanced and hence more favourable toward the use of our relation generation method as given in §3.2. It produces one relation in just under a second, so that more than N relations can be found in about 350 core-hours. However, thanks to our choice of g_2, Joux's pinpointing methods *also* benefit from the higher splitting probability as explained by Theorem 1, and so for this Kummer extension, it is still preferable to use Joux's advanced pinpointing method, which generates about $10N$ relations in approximately 2 hours on a single-core.

With the structured Gaussian elimination step in mind we computed approximately $10N$ relations and performed SGE on this matrix to reduce the number of variables, where we stopped again at the point when the row weight is being increased. The result was a $1\,066\,010 \times 1\,064\,991$ matrix of constant row weight 25, which constitutes a reduction of 10.3% in the number of variables.

The full factorisation of $2^{3164} - 1$ (obtained from the Cunningham tables [25]) is:

$$
\begin{aligned}
&3 \cdot 5 \cdot 29 \cdot 43 \cdot 113^2 \cdot 127 \cdot 227 \cdot 1583 \cdot 3391 \cdot 6329 \cdot 23279 \cdot 48817 \cdot 58309 \cdot 65993 \cdot 85429 \\
&\cdot 1868569 \cdot 2362153 \cdot 116163097 \cdot 636190001 \cdot 7920714887 \cdot 54112378027 \\
&\cdot 1066818132868207 \cdot 94395483835364237 \cdot 362648335437701461 \cdot 491003369344660409 \\
&\cdot 15079116213901326178369 \cdot 10384593717069655112945804582584321 \\
&\cdot 16210807687504089730597044158159945072569569899134297641 53 \\
&\cdot 25492807273453795564805967522921896342698297652509936704025490424226 49 \\
&\cdot 47852903674919527709794449504727427687484814404052312692462789051543 17 \\
&\cdot 94732691570793956856759198414911779734119524415635396799864941098330965 56 \\
&026935578510143423 7 \\
&\cdot 30893732435679706159469738259014519623666572271820219584074344744581789 67 \\
&789139446879970022670238264606111325817550047 99 \\
&\cdot 33248138195822034659908271092377125566098001373614163921550203376275101 35 \\
&820887988159907760592109751241079357983631847413209086969671 21 \cdot P_{190}\,,
\end{aligned}
$$

where P_{190} is a 190-digit prime.

We then ran a parallel version of the Lanczos' algorithm on several nodes of the SGI Altix ICE 8200EX cluster, using MPI and OpenMP parallelisation techniques on 144 cores and again the GNU Multi-Precision library [11], taking 85488 core-hours in total. Note that since the nodes we used for the computation were not very "well-connected," the total running time would have been reduced to around 30000 core-hours if we had run our algorithm on 12 cores.

For the DLP we took as our (proven) generator $g = x + t + 1 \in \mathbb{F}_{2^{3164}}^{\times}$ and a target element set as usual to be $x_\pi = \sum_{i=0}^{113} \tau(\lfloor \pi \cdot q^{i+1} \rfloor \bmod q)\, x^i$.

As before the descent proceeded by first finding an $i \in \mathbb{N}$ such that $x_\pi\, g^i = z_1/z_2$, where both z_1 and z_2 were here 16-smooth. At each stage, we choose to sieve for the special-$\mathfrak{q}$ on the y-side.

In this case we put even more effort in analysing and optimising the descent in the earlier stages so that the expected subsequent costs will be minimised. In fact we associated a cost k_d to each factor of degree d arising in the factorisation of the l.h.s. and r.h.s. polynomials, which we estimated by considering the distribution of factorisation pattern.

We used about 70 core-hours to find the 16-smooth initial fraction z_1/z_2, then spent 210 core-hours for the descent down to degree 4, and used 340 core-hours for processing the degree 4 polynomials. At this point we had 71 polynomials of degree 3, which needed an expected number of $2^{34.1}$ 2-smoothness tests to be resolved. These special-$\mathfrak{q}$ elements have been processed by the same SGI Altix ICE 8200EX cluster in about 20972 core-hours, using Shoup's Number Theory Library [24], and resulted in 1239 special-$\mathfrak{q}$ elements of degree 2. Finally, using the technique in §4.1, these elements were eliminated in about 10 core-hours, completing the descent.

The running time for solving an instance of the discrete logarithm problem completely in the finite field $\mathbb{F}_{2^{3164}}$ sums to 350+85488+20972+210+340+10 = 107092 core-hours (as already indicated, this figure would be reduced to around 52000 core-hours if Lanczos' algorithm was run on 12 cores). Finally, we found that $\log_g(x_\pi)$ equals

$$
\begin{aligned}
&2410958672084703779901202077261642209070514313288787533385808717024878456 57\\
&1268831206349103676532335755385717747797766545731784956477016880944817731 73\\
&1405243895025293868522646360493835468855617633181786341747893370309598402 58\\
&2718996263618673697554067799885512742832012390129483899153002417393400439 16\\
&1058228340028972042930361976940653379032557934518587736643501300307220916 66\\
&2531725410704479482997812210193428607010640365444303319677531146468063350 63\\
&3002030742348610674716684119982045443191768323538019822219249958042954261 67\\
&1123069707959607989886446311000373932915585804124069420045551161487903876 54\\
&9604900084297695444007900819088072394071341577241660482464194055035573980 35\\
&8979998525931969540314396297687768509998877208705617419130555318640416547 07\\
&8404337954037532005208916171502547565867282159415513550648407797656823989 93\\
&1563900000242491107399569193500692930336704230702995815576366649937212045 36\\
&8630387367148801640963557811787088923027864916437813 3\,.
\end{aligned}
$$

Observe that this computation also breaks the elliptic curve DLP for supersingular curves defined over $\mathbb{F}_{2^{791}}$, with embedding degree 4. However, since 791 is not prime, even before this break, such curves would not have been recommended, due to the potential applicability of Weil descent attacks [8].

6 Conclusion

We have presented and analysed new variants of the medium-sized base field FFS, for binary fields, which have complexities as low as $L_{q^n}(1/3, (4/9)^{1/3})$ for computing arbitrary logarithms. Furthermore, for fields possessing a subfield of an appropriate size, we have provided the first ever heuristic *polynomial time* algorithm for finding the discrete logarithms of degree one and two elements, which have both been verified experimentally. To illustrate the efficiency of the methods, we have successfully solved the DLP in the finite fields $\mathbb{F}_{2^{1971}}$ and $\mathbb{F}_{2^{3164}}$, setting a record for binary fields.

It would be interesting to know whether there are more general theorems on splitting behaviours for other polynomials arising during the descent, and also to what extent the known theorems apply to other characteristics.

Acknowledgements

The authors would like to extend their thanks to the Irish Centre for High-End Computing (ICHEC) — and Gilles Civario in particular — for their support throughout the course of our computations.

References

1. Leonard M. Adleman and Ming-Deh A. Huang. Function field sieve method for discrete logarithms over finite fields. *Inform. and Comput.*, 151(1-2):5–16, 1999.
2. Daniel V. Bailey, Christof Paar, Gabor Sarkozy, and Micha Hofri. Computation in optimal extension fields. In *Conference on The Mathematics of Public Key Cryptography, The Fields Institute for Research in the Mathematical Sciences*, pages 12–17, 2000.
3. Antonia W. Bluher. On $x^{q+1} + ax + b$. *Finite Fields and Their Applications*, 10(3):285–305, 2004.
4. F. R. K. Chung. Diameters and eigenvalues. *J. Amer. Math. Soc.*, 2(2):187–196, 1989.
5. Don Coppersmith. Fast evaluation of logarithms in fields of characteristic two. *IEEE Trans. Inform. Theory*, 30(4):587–593, 1984.
6. Jean-Charles Faugère, Ayoub Otmani, Ludovic Perret, and Jean-Pierre Tillich. Algebraic cryptanalysis of McEliece variants with compact keys. In Henri Gilbert, editor, *EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 279–298. Springer, Heidelberg, 2010.
7. Jean-Charles Faugère, Ludovic Perret, Christophe Petit, and Guénaël Renault. Improving the complexity of index calculus algorithms in elliptic curves over binary fields. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 27–44. Springer, Heidelberg, 2012.
8. Pierrick Gaudry, Florian Hess, and Nigel P. Smart. Constructive and destructive facets of Weil descent on elliptic curves. *J. Cryptology*, 15(1):19–46, 2002.
9. Faruk Göloğlu, Robert Granger, Gary McGuire, and Jens Zumbrägel. Discrete Logarithms in $GF(2^{1971})$. NMBRTHRY list, 19 Feb 2013.
10. Robert Granger and Frederik Vercauteren. On the discrete logarithm problem on algebraic tori. In Victor Shoup, editor, *CRYPTO 2005*, volume 3621 of *LNCS*, pages 66–85. Springer, Heidelberg, 2005.
11. Torbjörn Granlund and the GMP development team. *GNU MP: The GNU Multiple Precision Arithmetic Library*, 5.0.5 edition, 2012. `http://gmplib.org/`.
12. Tor Helleseth and Alexander Kholosha. $x^{2^l+1} + x + a$ and related affine polynomials over $\mathrm{GF}(2^k)$. *Cryptogr. Commun.*, 2(1):85–109, 2010.

13. Antoine Joux. Faster index calculus for the medium prime case application to 1175-bit and 1425-bit finite fields. In Thomas Johansson and Phong Q. Nguyen, editors, *EUROCRYPT 2013*, volume 7881 of *LNCS*, pages 177–193. Springer, Heidelberg, 2013.
14. Antoine Joux and Reynald Lercier. The function field sieve is quite special. In Claus Fieker and David R. Kohel, editors, *Algorithmic number theory (Sydney, 2002)*, volume 2369 of *LNCS*, pages 431–445. Springer, Heidelberg, 2002.
15. Antoine Joux and Reynald Lercier. Improvements to the general number field sieve for discrete logarithms in prime fields: a comparison with the gaussian integer method. *Math. Comput.*, 72(242):953–967, 2003.
16. Antoine Joux and Reynald Lercier. The function field sieve in the medium prime case. In Serge Vaudenay, editor, *EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 254–270. Springer, Heidelberg, 2006.
17. Antoine Joux, Reynald Lercier, David Naccache, and Emmanuel Thomé. Oracle-assisted static diffie-hellman is easier than discrete logarithms. In Matthew G. Parker, editor, *Cryptography and Coding*, volume 5921 of *LNCS*, pages 351–367. Springer, Heidelberg, 2009.
18. Brian A. LaMacchia and Andrew M. Odlyzko. Solving large sparse linear systems over finite fields. In Alfred J. Menezes and Scott A. Vanstone, editors, *CRYPTO '90*, volume 537 of *LNCS*, pages 109–133. Springer, Heidelberg, 1991.
19. Arjen K. Lenstra and Hendrik W. Lenstra, Jr., editors. *The development of the number field sieve*, volume 1554 of *Lecture Notes in Mathematics*. Springer, Heidelberg, 1993.
20. Hendrik W. Lenstra, Jr. Finding isomorphisms between finite fields. *Math. Comp.*, 56(193):329–347, 1991.
21. Rafael Misoczki and Paulo S. Barreto. Compact McEliece keys from Goppa codes. In Michael J. Jacobson, Jr., Vincent Rijmen, and Reihaneh Safavi-Naini, editors, *Selected Areas in Cryptography*, volume 5867 of *LNCS*, pages 376–392. Springer, Heidelberg, 2009.
22. J. M. Pollard. Monte carlo methods for index computation ($\bmod p$). *Math. Comp.*, 32(143):918–924, 1978.
23. Karl Rubin and Alice Silverberg. Torus-based cryptography. In Dan Boneh, editor, *CRYPTO 2003*, volume 2729 of *LNCS*, pages 349–365. Springer, Heidelberg, 2003.
24. Victor Shoup. *NTL: A library for doing number theory*, 5.5.2 edition, 2009. `http://www.shoup.net/ntl/`.
25. Sam Wagstaff et al. The Cunningham Project. `http://homes.cerias.purdue.edu/~ssw/cun/index.html`.
26. Daqing Wan. Generators and irreducible polynomials over finite fields. *Math. Comp.*, 66(219):1195–1212, 1997.

Solving a 6120-bit DLP on a Desktop Computer*

Faruk Göloğlu, Robert Granger, Gary McGuire, and Jens Zumbrägel

Complex & Adaptive Systems Laboratory and
School of Mathematical Sciences
University College Dublin, Ireland
{farukgologlu,robbiegranger}@gmail.com, {gary.mcguire,jens.zumbragel}@ucd.ie

Abstract. In this paper we show how some recent ideas regarding the discrete logarithm problem (DLP) in finite fields of small characteristic may be applied to compute logarithms in some very large fields extremely efficiently. By combining the polynomial time relation generation from the authors' CRYPTO 2013 paper, an improved degree two elimination technique, and an analogue of Joux's recent small-degree elimination method, we solved a DLP in the record-sized finite field of 2^{6120} elements, using just a single core-month. Relative to the previous record set by Joux in the field of 2^{4080} elements, this represents a 50% increase in the bitlength, using just 5% of the core-hours. We also show that for the fields considered, the parameters for Joux's $L_Q(1/4 + o(1))$ algorithm may be optimised to produce an $L_Q(1/4)$ algorithm.

Keywords: Discrete logarithm problem, binary finite fields

1 Introduction

The understanding of the hardness of the DLP in the multiplicative group of finite extension fields could be said to be undergoing a mini-revolution. It began with Joux's 2012 paper in which he introduced a method of relation generation dubbed 'pinpointing', which reduces the time required to obtain the logarithms of the elements of the factor base [11]. For medium-sized base fields, this technique has heuristic complexity as low as $L_Q(1/3, 2/3^{2/3}) \approx L_Q(1/3, 0.961)$[1], where

$$L_Q(a, c) = \exp\left((c + o(1))\,(\log Q)^a (\log\log Q)^{1-a}\right),$$

and Q is the cardinality of the finite field. This improves upon the previous best by Joux and Lercier [17] $L_Q(1/3, 3^{1/3}) \approx L_Q(1/3, 1.442)$. To demonstrate the practicality of this approach, Joux solved two example DLPs in fields of bitlength 1175 and 1425 respectively, both with prime base fields.

Soon afterwards the present authors showed that in the context of binary fields (and more generally small characteristic fields), finding relations for the

* Research supported by the Claude Shannon Institute, Science Foundation Ireland Grant 06/MI/006. The fourth author was in addition supported by SFI Grant 08/IN.1/I1950.

[1] On foot of recent communications [13], the complexity may in fact be $L_Q(1/3, 2^{1/3})$.

factor base can be *polynomial time* in the size of the field [6]. By extending the basic idea to eliminate degree two elements during the descent phase, for medium-sized base fields an heuristic complexity as low as $L_Q(1/3, (4/9)^{1/3}) \approx L_Q(1/3, 0.763)$ was achieved; this approach was demonstrated via the solution of a DLP in the field $\mathbb{F}_{2^{1971}}$ [7], and in the field $\mathbb{F}_{2^{3164}}$.

After the initial publication of [6], Joux released a preprint [12] detailing an algorithm for solving the discrete logarithm problem for fields of the form $\mathbb{F}_{q^{2n}}$, with $q = p^\ell$ and $n \approx q$, which was used in the solving of a DLP in $\mathbb{F}_{2^{1778}}$ [14], and later in $\mathbb{F}_{2^{4080}}$ [15]. This algorithm has heuristic complexity $L_Q(1/4 + o(1))$, and also has an heuristic polynomial time relation generation method, similar in principle to that in [6]. While the degree two element elimination in [6] is arguably superior, for other small degrees, Joux's elimination method is faster, resulting in the stated complexity. Joux's discrete logarithm computation in $\mathbb{F}_{2^{4080}}$ [15] required about 14,100 core-hours: 9,300 core-hours for the computation of the logarithms of all degree one and two elements; and 4,800 core-hours for the descent step, i.e., for computing the logarithm of an arbitrary element. For this computation, the field $\mathbb{F}_{2^{4080}}$ was represented as a degree 255 Kummer extension of $\mathbb{F}_{2^{16}}$, i.e., $\mathbb{F}_{(q^2)^{q-1}}$ with $q = 2^8$, as per [12]. The use of Kummer extensions (with extension degree either $q - 1$ or $q + 1$) gives a reduction in the size of the degree one and two factor base [17, 11, 12]; they are therefore preferable when it comes to setting record DLP computations.

The relation generation method in [6, §3.3] applies to larger base fields of the form $\mathbb{F}_{q^k}$ with $k \geq 3$ (rather than $k = 2$) and extension degrees up to $n \approx q\delta_1$ with $\delta_1 \geq 1$ a small integer. Hence the methods in this paper naturally apply to any extension degree. Note that this representation offers greater flexibility than Joux's (which can represent extension degrees up to $q + \delta_1'$) for essentially the same algorithmic cost, and may therefore provide a more practical DLP break when small base fields need to be embedded into larger ones in order to apply the attacks. However, here we choose to focus on Kummer extensions of degree $q \pm 1$, as these optimise the relation generation efficiency [6, §3.4], and linear algebra step. While the two DLP breaks in the fields $\mathbb{F}_{2^{1971}}$ and $\mathbb{F}_{2^{3164}}$ contained therein did not fully exploit the above 'extreme' fields in which the extension degree is polynomially related to the size of the base field, thanks to Joux's fast small-degree elimination method, one can now do this more efficiently. Hence, with a view to solving the DLP in larger fields than before and in as short a time as possible, in this work we identify a family of fields for which the DLP is very easily solved, relative to other fields of a similar size. While this does not mean other fields of a similar size are infeasible to break, it requires more time in practice to find the logarithms of the factor base elements, with the complexities remaining the same.

One benefit of using base fields with $k \geq 3$ is that there is an efficient probabilistic elimination technique for degree two elements [6, §4.1]. For any

fixed $k \geq 4$ the elimination probability very quickly tends to 1 for increasing q. In this paper we present an improved technique which allows one to find the logarithm of degree two elements extremely fast, once the logarithms of all degree one elements are known. However, for $k = 3$ the elimination probability is $1/(2(\delta_1 - 1)!)$, or exactly $1/2$ for $\mathbb{F}_{2^{6120}} = \mathbb{F}_{(q^3)^{q-1}}$ with $q = 2^8$. Therefore the natural next choice is to set $k = 4$ and solve a DLP in $\mathbb{F}_{2^{8160}} = \mathbb{F}_{(q^4)^{q-1}}$. This would require solving a sparse linear system in $\approx 4.2 \cdot 10^6$ variables, and a slightly more costly descent step. Instead of carrying out this computation, we devised a technique for the 6120 bit case for which the elimination of each degree two element took only 0.03 seconds, and which required solving a much smaller linear system in 21,932 variables. This culminated in the resolution of a DLP in $\mathbb{F}_{2^{6120}}$ in under 750 core-hours [8], which represents a 50% increase in bitlength over the previous record, whilst requiring just 5% of the computation time.

We note that the solving of DLPs in $\mathbb{F}_{2^{6120}} = \mathbb{F}_{2^{24 \cdot 255}}$ renders insecure all pairing-based protocols based on supersingular curves of genus one and two over $\mathbb{F}_{2^{255}}$, since the correponding embedding degrees are 4 and 12 (in the best cases), respectively [1]. However, since 255 is not prime, such curves would not be recommended due to possible Weil descent attacks [5]. In any case, the Jacobians of the curves do not have prime or nearly prime order and so are not cryptographically interesting. As stated above, we could just as easily have solved the corresponding DLP with extension degree $q + 1$ rather than $q - 1$, i.e., with extension degree 257 rather than 255. However, since the full factorisation of $2^{6120} - 1$ is known, we were able to use a proven generator and so for completeness we chose to solve this case.[2]

Since our break of the DLP in $\mathbb{F}_{2^{6120}}$ may be considered as a proof-of-concept implementation for our approach, at the time we were not overly concerned with the issue of complexity. Indeed, as the elimination times are reasonable and as just noted, comparable to Joux's elimination timings, further experimentation is needed to ascertain if the performance is comparable for larger systems. However, one basic difference between the two approaches is that the quadratic systems which arise when using our analogue of Joux's small-degree elimination method are not bilinear, and hence are not guaranteed to enjoy the same resolution complexity, as given in Spaenlehauer's thesis [25, Cor. 6.30]. Therefore, we can not currently argue that the heuristic complexity is the same. Nevertheless, we show that with a better choice of parameter and a tighter analysis, the final part of the descent in Joux's

[2] Forty days after the announcement of our full DLP break in $\mathbb{F}_{2^{6120}} = \mathbb{F}_{2^{24 \cdot 255}}$ [8] – and after the submission of this paper – Joux announced a break of the DLP in a 1843-bit subgroup of $\mathbb{F}_{2^{6168}}^{\times} = \mathbb{F}_{2^{24 \cdot 257}}^{\times}$, using a nearly identical degree two elimination technique and the same descent parameters, in under 550 core-hours [16]. Noting that the logarithms were not computed in the full multiplicative group and that this computation was performed on faster processors, it is clear that the number of our core-hours and Joux's are comparable. In this case too the corresponding Jacobians do not have prime or nearly prime order.

$L_Q(1/4 + o(1))$ algorithm may be improved to an $L_Q(1/4)$ algorithm, for the fields we consider, i.e., those for which the extension degree is polynomially related to the size of the basefield. Since the other phases of the algorithm have complexity $L_Q(1/4)$, or lower, the overall complexity for solving the DLP is $L_Q(1/4)$ as well.

The remainder of the paper is organised as follows. §2 explains our field setup and algorithm in detail. §3 covers the other essential algorithms and issues regarding the computation. §4 gives the details of a discrete logarithm computation in $\mathbb{F}_{2^{6120}}$, while finally in §5 we briefly address the issue of complexity.

2 The Algorithm

The following describes the field setup and index calculus method that we use for our discrete logarithm computation.

2.1 Setup

We consider here Kummer extensions, which are our focus for efficiency reasons; the general case can be found in [6, §3.3] and is recalled in §5.

Let ℓ, k be positive integers, $q := 2^\ell$, and $n := q - 1$. We construct the finite field $\mathbb{F}_{(q^k)^n}$ of bit length $\ell kn = \ell k(q-1)$ in which we solve the DLP, as follows[3]. As stated in the introduction, the case $n := q + 1$ follows *mutatis mutandis*.

We express our base field $\mathbb{F}_{q^k}$ as a degree k extension of $\mathbb{F}_q$. Then we choose $\gamma \in \mathbb{F}_{q^k}$ such that the polynomial $X^n + \gamma$ is irreducible in $\mathbb{F}_{q^k}[X]$ and define $\mathbb{F}_{(q^k)^n}$ as the Kummer extension

$$\mathbb{F}_{q^k}(x) \cong \mathbb{F}_{q^k}[X]/\big((X^n + \gamma)\,\mathbb{F}_{q^k}[X]\big)\,,$$

where x is a root of the polynomial $X^n + \gamma$ in $\mathbb{F}_{(q^k)^n}$. Note that a Kummer extension of degree n over $\mathbb{F}_{q^k}$ exists if and only if $n \mid q^k - 1$. Throughout the paper, the upper case letters $X, W, \ldots$ are used for indeterminates and the lower case letters $x, w, \ldots$ are reserved for finite fields elements that are roots of polynomials.

The following table displays the bit length ℓkn of the finite field $\mathbb{F}_{(q^k)^n}$ for various choices of the numbers ℓ and k.

$k \setminus \ell$	6	7	8	9
3	1134	2667	6120	13797
4	1512	3556	8160	18396
5	1890	4445	10200	22995
6	2268	5334	12240	27594

[3] Our choice of representation of the finite field $\mathbb{F}_{(q^k)^n}$ will be advantageous for our method to solve the DLP. Note that it is a computationally easy problem to switch between two different representations of a finite field [22].

In §4, we will give the details of the discrete logarithm computation when $\ell kn = 6120$. The algorithm we explain in this section may be successfully applied to any of the above parameters with $k \geq 4$, whereas for $k = 3$ one would normally be required to precompute the logarithms of all degree two elements using a method analogous to Joux's [12]. However, for $k = 3$ and $\ell = 8$, precomputation can be avoided entirely; see §4.4.

2.2 Factor Base and Automorphisms

The factor base we use consists of the elements in $\mathbb{F}_{(q^k)^n}$ which have degree one in the polynomial representation over $\mathbb{F}_{q^k}$, i.e., we consider the set $\{x+a \mid a \in \mathbb{F}_{q^k}\}$. As noted in [17, 11, 6], factor base preserving automorphisms of $\mathbb{F}_{(q^k)^n}$, which are provided by Kummer extensions, can be used to significantly reduce the number of variables involved in the linear algebra step. Indeed, the map $\sigma := \mathrm{Frob}^\ell : \alpha \to \alpha^q$ satisfies $\sigma(x) = \gamma x$ with $\gamma \in \mathbb{F}_{q^k}$, and thus preserves the factor base. Furthermore, for $\varphi := \sigma^k = \mathrm{Frob}^{\ell k} : \alpha \to \alpha^{q^k}$ we have $\varphi(x) = \mu x$ with $\mu \in \mathbb{F}_q$ a primitive n-th root of unity, and thus we find

$$(x+a)^{q^{kj+i}} = \sigma^{kj+i}(x+a) = \sigma^i(\varphi^j(x+a)) = \sigma^i(\mu^j x + a) = \mu^j \gamma^{e_i} x + a^{q^i} ,$$

where $e_0 = 0$ and $e_i = qe_{i-1} + 1$ for $1 \leq i < k$; thus it follows that

$$\log\left(x + \frac{a^{q^i}}{\mu^j \gamma^{e_i}}\right) = q^{kj+i} \log(x+a)$$

for all $0 \leq j < n$ and $0 \leq i < k$.

The automorphism σ generates a group of order kn, which acts on the set of q^k factor base elements, thus dividing the factor base into about N orbits, where $N \approx \frac{q^k}{kn} \approx \frac{1}{k} q^{k-1}$ is the number of variables to consider.

2.3 Relation Generation

In order to generate relations between the factor base elements we use the method from [6, §3.1-4]. We exploit properties of polynomials of the form

$$F_B(X) := X^{q+1} + BX + B ,$$

which have been studied by Bluher [2] and Helleseth/Kholosha [10]. We recall in particular the following result of Bluher [2] (see also [10, 6]):

Theorem 1. *The number of elements $B \in \mathbb{F}_{q^k}^\times$ such that the polynomial $F_B(X)$ splits completely over $\mathbb{F}_{q^k}$ equals*

$$\frac{q^{k-1}-1}{q^2-1} \quad \textit{if } k \textit{ odd}, \qquad \frac{q^{k-1}-q}{q^2-1} \quad \textit{if } k \textit{ even}.$$

Let $B \in \mathbb{F}_{q^k}^{\times}$ be an element such that $F_B(X)$ splits and denote its roots by μ_i, for $i = 1, \ldots, q+1$. For arbitrary $a, b \in \mathbb{F}_{q^k}$ (with $a^q \neq b$) there exists $c \in \mathbb{F}_{q^k}$ with $(a^q + b)^{q+1} = B\,(ab + c)^q$ and we then find that

$$f(X) := F_B\Big(\frac{ab+c}{a^q+b}\,X + a\Big) = X^{q+1} + aX^q + bX + c$$

and that $f(X)$ also splits over $\mathbb{F}_{q^k}$, with roots $\nu_i := \frac{ab+c}{a^q+b}\,\mu_i + a$.

Now by the definition of $\mathbb{F}_{(q^k)^n}$ we have $x^n = \gamma$ and thus $x^q = \gamma x$, with $\gamma \in \mathbb{F}_{q^k}$. Hence in $\mathbb{F}_{(q^k)^n}$ we have

$$f(x) = \gamma x^2 + a\gamma x + bx + c = \gamma(x^2 + (a + \tfrac{b}{\gamma})x + \tfrac{c}{\gamma}) = \gamma g(x)\ ,$$

where $g(X) := X^2 + (a + \frac{b}{\gamma})X + \frac{c}{\gamma}$. Hence, if the polynomial $g(X)$ splits, i.e., if $g(X) = (X + \xi_1)(X + \xi_2)$, which heuristically occurs with probability 1/2, then we find a relation of factor base elements, namely

$$\prod_{i=1}^{q+1}(x + \nu_i) = \gamma(x + \xi_1)(x + \xi_2)\ .$$

Such a relation corresponds to a linear relation between the logarithms of the factor base elements. Once we have found more than N relations we can solve the discrete logarithms of the factor base elements by means of linear algebra; see §3.3.

2.4 Individual Logarithms

After the logarithms of the factor base elements have been found, a general individual discrete logarithm can be computed, as is common, by a descent strategy. The basic idea of this method is trying to write an element, given by its polynomial representation over $\mathbb{F}_{q^k}$, as a product in $\mathbb{F}_{(q^k)^n}$ of factors represented by lower degree polynomials. By applying this principle recursively a descent tree is constructed, and one can eventually express a given target element by a product of factor base elements, thus solving the DLP.

While for large degree polynomials it is relatively easy to find an expression involving lower degree polynomials by a standard approach, this method becomes increasingly less efficient as the degree becomes smaller. In addition, the number of small degree polynomials in the descent tree grows significantly with lower degree. We therefore propose new methods for degree 2 elimination and small degree descent, which are inspired by the recent works [6] and [12] respectively.

Degree 2 Elimination Given a polynomial $Q(X) := X^2 + q_1 X + q_0 \in \mathbb{F}_{q^k}[X]$ we aim at expressing the corresponding finite field element $Q(x) \in \mathbb{F}_{(q^k)^n}$ as a product of factor base elements. In essence, what we do is just the reverse of the degree one relation generation, with the polynomial $g(X)$ set to be $Q(X)$.

In particular, we compute – when possible – $a, b, c \in \mathbb{F}_{q^k}$ such that, up to a multiplicative constant in $\mathbb{F}_{q^k}^\times$, $Q(x) = x^2 + q_1 x + q_0$ equals $x^{q+1} + ax^q + bx + c$ where the polynomial $X^{q+1} + aX^q + bX + c$ splits into linear factors (cf. [6, §4.1]).

As $x^n = \gamma$ holds, we have $x^{q+1} + ax^q + bx + c = \gamma(x^2 + (a + \frac{b}{\gamma})x + \frac{c}{\gamma})$ and comparing coefficients we find $\gamma q_0 = c$ and $\gamma q_1 = \gamma a + b$. Now letting $B \in \mathbb{F}_{q^k}^\times$ be an element satisfying the splitting property of Theorem 1 and combining the previous equations with $(a^q + b)^{q+1} = B\,(ab + c)^q$ we arrive at the condition

$$(a^q + \gamma a + \gamma q_1)^{q+1} + B(\gamma a^2 + \gamma q_1 a + \gamma q_0)^q = 0\,.$$

Considering $\mathbb{F}_{q^k}$ as a degree k extension over $\mathbb{F}_q$ this equation gives a quadratic system in the k $\mathbb{F}_q$-components of a, which can be solved very fast by a Gröbner basis method.

Heuristically, for each of the above B's the probability of success of this method, i.e., when an $a \in \mathbb{F}_{q^k}$ as above exists, is $1/2$. Note that if $k = 3$ there is just one single B in the context of Theorem 1, and so this direct method fails in half of the cases. However, as noted earlier, this issue can be resolved under certain circumstances, e.g., for $\ell = 8$; see §4.4.

Small Degree Descent The following describes the Gröbner basis descent of Joux [12] applied in the context of the polynomials $F_B(X) = X^{q+1} + BX + B$ of Theorem 1. Let $f(X)$ and $g(X)$ be polynomials over $\mathbb{F}_{q^k}$ of degree δ_f and δ_g respectively. We substitute X by the rational function $\frac{f(X)}{g(X)}$ and thus find that the polynomial

$$P(X) := f(X)^{q+1} + Bf(X)\,g(X)^q + Bg(X)^{q+1}$$

factors into polynomials of degree at most $\delta = \max\{\delta_f, \delta_g\}$. Since $x^q = \gamma x$ holds in $\mathbb{F}_{(q^k)^n}$ the element $P(x)$ can also be represented by a polynomial of degree 2δ.

Now given a monic polynomial $Q(X) \in \mathbb{F}_{q^k}[X]$ of degree 2δ (resp. $2\delta - 1$) to be eliminated we consider the equation $P(x) = Q(x)$ (resp. $P(x) = (x+a)Q(x)$ with some random fixed $a \in \mathbb{F}_{q^k}$). It results as above in a quadratic system of $\mathbb{F}_q$-variables representing the coefficients of $f(X)$ and $g(X)$ in $\mathbb{F}_{q^k}$, and can be solved by a Gröbner basis algorithm. In order to minimise the number of variables involved we set $f(X)$ to be monic of degree $\delta_f = \delta$ and $g(X)$ of degree $\delta_g = \delta - 1$, resulting in $k\delta + k\delta = 2k\delta$ variables in $\mathbb{F}_q$. Since the number of equations to be satisfied equals $2k\delta$ as well, we find a solution of this system with good probability.

Large Degree Descent This part of the descent is somewhat classical (see [17] for example), but includes the degree balancing technique described in [6, §4], which makes the descent far more rapid when the base field $\mathbb{F}_{q^k}$ is a degree k extension of a non-prime field. In the finite field $\mathbb{F}_{(q^k)^n}$ we let $y := x^q$ and $\bar{x} := x^{2^{\ell-a}}$ for some suitably chosen integer $1 < a < k$. Then $y = \bar{x}^{2^a}$ and $\bar{x} = (\frac{y}{\gamma})^{2^{\ell-a}}$ holds. Now for given $Q(X) \in \mathbb{F}_{q^k}[X]$ of degree d representing $Q(y)$ we consider the lattice

$$L := \left\{(w_0, w_1) : Q(X) \mid (\tfrac{X}{\gamma})^{2^{\ell-a}} w_0(X) + w_1(X)\right\} \subseteq \mathbb{F}_{q^k}[X]^2 \,.$$

By Gaussian lattice reduction we find a basis (u_0, u_1), (v_0, v_1) of L of degree $\approx d/2$ and can thus generate lattice elements $(w_0, w_1) = r(u_0, u_1) + s(v_0, v_1)$ of low degree. In $\mathbb{F}_{(q^k)^n}$ we then consider the equation

$$\bar{x} w_0(\bar{x}^{2^a}) + w_1(\bar{x}^{2^a}) = \bar{x} w_0(y) + w_1(y) = (\tfrac{y}{\gamma})^{2^{\ell-a}} w_0(y) + w_1(y) \,,$$

where the right-hand side is divisible by $Q(y)$ by construction, and a is chosen so as to make the degrees of both sides as close as possible. The descent is successful whenever a lattice element (w_0, w_1) is found such that the involved polynomials $X w_0(X^{2^a}) + w_1(X^{2^a})$ and $\frac{1}{Q(x)}(X^{2^{\ell-a}} w_0(X) + \gamma^{2^{\ell-a}} w_1(X))$ are $(d-1)$-smooth, i.e., have only factors of degree less than d.

3 Other Essentials

In this section we give an explicit account of further basics required for a discrete logarithm computation.

3.1 Factorisation of the Group Order

The factorisation of the group order $|\mathbb{F}^\times_{(q^k)^n}| = 2^{\ell kn} - 1$ is of interest for several reasons. Firstly it indicates the difficulty of solving the associated DLP using the Pohlig-Hellman algorithm. Secondly it enables one to provably find a generator. Finally, it determines the small factors for which we apply Pollard's rho method, and the large factors for the linear algebra computation. Since the complexity of the Special Number Field Sieve [20] is much higher than the present DLP algorithms, it is unlikely that one can completely factorise $2^{\ell kn} - 1$ in cases of interest in a reasonable time. In these cases it is vital to at least know all the small prime factors of the group order, which can be accomplished using the Elliptic Curve Method [21] and the identity

$$2^{\ell kn} - 1 = \prod_{d \mid \ell kn} \Phi_d(2) \,,$$

where $\Phi_d \in \mathbb{Z}[x]$ denotes the d-th cyclotomic polynomial.

3.2 Pohlig-Hellman and Pollard's Rho Method

In order to compute a discrete logarithm in a group G of order m we can use any factorisation of $m = m_1 \cdot \ldots \cdot m_r$ into pairwise coprime factors m_i and compute the discrete log modulo each factor. Indeed, if we are to compute $z = \log_\alpha \beta$ it suffices to compute $\log_{\alpha^{c_i}} \beta^{c_i}$ with $c_i = m/m_i$, which determines $z \bmod m_i$. With the information of $z \bmod m_i$ for all i one easily determines $z \pmod m$ by the Chinese Remainder Theorem.

For the small prime (power) factors of m we use Pollard's rho method to compute the discrete logarithm modulo each factor. Regarding the large factors of m we find it most efficient to combine them into a single product m_*, so that in the linear algebra step of the index calculus method we work over the ring $\mathbb{Z}_{m_*}$. Note that each iteration of the Lanczos method that we use for the linear algebra problem requires the inversion of a random element in $\mathbb{Z}_{m_*}$; this is the reason why we separate the small factors of the group order from the large ones.

3.3 Linear Algebra

The relation generation phase of the index calculus method produces linear relations among the logarithms of the factor base elements. As the factor base logs are also related by the automorphism group as explained in §2.2 the number N of variables is reduced and the linear relations will have coefficients being powers of 2. Once $M > N$ relations have been generated we have to find a nonzero solution vector for the linear system. To ensure that the matrix is of maximal rank $N - 1$ we generate $M \approx N + 100$ relations. As noted earlier the number of variables N is expected to be about $\frac{q^k}{kn} \approx \frac{1}{k} q^{k-1}$.

We let B be the $M \times N$ matrix of the relations' coefficients, which is a matrix of constant row-weight $q + 3$. We have to find a nonzero vector v of length N such that $Bv = 0$ modulo m_*, the product of the large prime factors of the group order m. A common approach in index calculus algorithms is to reduce the matrix size at this stage by using a structured Gaussian elimination (SGE) method. In our case, however, the matrix is not extremely sparse while its size is quite moderate, hence the expected benefit from SGE would be minimal and we refrained from this step.

We use the iterative Lanczos method [19, 18] to solve the linear algebra problem, which we briefly describe here. Let $A = B^t B$, which is a symmetric $N \times N$ matrix. We let $v \in \mathbb{Z}_{m^*}^N$ be random, $w = Av$, and find a vector $x \in \mathbb{Z}_{m^*}^N$ such that $Ax = w$ holds (since $A(x - v) = 0$ we have thus found a kernel element). We compute the following iteration

$$w_0 = w\,, \qquad v_0 = Aw_0\,, \qquad w_1 = v_0 - \frac{(v_0, v_0)}{(v_0, w_0)} w_0$$

$$v_i = Aw_i\,, \qquad w_{i+1} = v_i - \frac{(v_i, v_i)}{(v_i, w_i)} w_i - \frac{(v_i, v_{i-1})}{(v_{i-1}, w_{i-1})} w_{i-1}$$

and stop once $(v_j, w_j) = 0$; if $w_j \neq 0$ the algorithm fails, otherwise we find the solution vector

$$x = \sum_{i=0}^{j-1} \frac{(w, w_i)}{(v_i, w_i)} w_i \, .$$

Performing the above iteration consists essentially of several matrix-vector products, scalar-vector multiplications, and vector-vector inner products. As the matrix is sparse and consists of entries being powers of 2 the matrix-vector products can be carried out quite efficiently. Therefore, the scalar multiplications and inner products consume a significant part of the computation time. We have used a way to reduce the number of inner products per iteration, as was suggested recently [23].

Indeed, using the A-orthogonality $(v_i, w_j) = w_i^t A w_j = 0$ for $i \neq j$ we find that

$$(v_i, v_{i-1}) = (v_i, w_i) \quad \text{and} \quad (w, w_{i+1}) = -\frac{(v_i, v_i)}{(v_i, w_i)}(w, w_i) - \frac{(v_i, v_{i-1})}{(v_{i-1}, w_{i-1})}(w, w_{i-1}).$$

Now at each iteration, given w_i we compute the matrix-vector product Bw_i and the inner product $a_i := (v_i, w_i) = (Bw_i, Bw_i)$, as well as $v_i = Aw_i = B^t(Bw_i)$ and $b_i := (v_i, v_i) = (Aw_i, Aw_i)$. We then have the simplified iteration

$$w_0 = w \, , \quad w_1 = v_0 - \frac{b_0}{a_0} w_0 \, , \quad w_{i+1} = v_i - \frac{b_i}{a_i} w_i - \frac{a_i}{a_{i-1}} w_{i-1}$$

and the solution vector $x = \sum_{i=0}^{j-1} \frac{c_i}{a_i} w_i$, where $c_i := (w, w_i)$ can be computed by the iteration

$$c_0 = (w, w) \, , \quad c_1 = a_0 - \frac{b_0}{a_0} c_0 \, , \quad c_{i+1} = -\frac{b_i}{a_i} c_i - \frac{a_i}{a_{i-1}} c_{i-1} \, .$$

We see that each iteration requires merely two matrix-vector products, three scalar multiplications, and two inner products.

3.4 Target Element

In order to set ourselves a DLP challenge we construct the 'random' target element $\beta \in \mathbb{F}_{(q^k)^n}$ using the binary digits expansion of the mathematical constant π. More precisely, considering the q^k-ary expansion

$$\pi = 3 + \sum_{i=1}^{\infty} c_i \, q^{-ki} \quad \text{with} \quad c_i \in S_{q^k} := \{0, 1, \ldots, q^k - 1\}$$

we use a bijection between the sets S_{q^k} and $\mathbb{F}_{q^k}$, which is defined by the mappings $\varphi_q : \mathbb{F}_q \to \{0, \ldots, q-1\}$: $\sum_{i=0}^{\ell-1} a_i t^i \mapsto \sum_{i=0}^{\ell-1} a_i 2^i$ and $\varphi : \mathbb{F}_{q^k} \to S_{q^k}$: $\sum_{j=0}^{k-1} b_j w^j \mapsto \sum_{j=0}^{k-1} \varphi_q(b_j) q^j$, and construct in this way the target element

$$\beta_\pi := \sum_{i=0}^{n-1} \varphi^{-1}(c_{i+1}) \, x^i \in \mathbb{F}_{(q^k)^n} \, .$$

4 Discrete Logarithms in $\mathbb{F}_{2^{6120}}$

In this section we document the breaking of DLP in the case $\ell = 8$ and $k = 3$, i.e., in $\mathbb{F}_{2^{6120}}$. The salient features of the computation are:

- The relation generation for degree one elements took 15 seconds[4].
- The corresponding linear algebra took 60.5 core-hours.
- In contrast to [15, 12], we computed the logarithm of degree 2 irreducibles on the fly; each took on average 0.03 seconds.
- The descent was designed so as to significantly reduce the number of bottleneck (degree 6) eliminations. As a result, the individual logarithm phase took just under 689 core-hours.

4.1 Setup

We first defined $\mathbb{F}_{2^8}$ using the irreducible polynomial $T^8 + T^4 + T^3 + T + 1$. Letting t be a root of this polynomial, we defined $\mathbb{F}_{2^{24}}/\mathbb{F}_{2^8}$ using the irreducible polynomial $W^3 + t$. Letting w be a root of this polynomial, we finally defined $\mathbb{F}_{2^{6120}}/\mathbb{F}_{2^{24}}$ using the irreducible polynomial $X^{255} + w + 1$, where we denote a root of this polynomial by x.

We chose as a generator $g = x + w$, which has order $2^{6120} - 1$; this was proven via the prime factorisation of $2^{6120} - 1$, which is provided in [8]. As usual, the target element was set to be β_π as explained in §3.4.

4.2 Relation Generation

Our factor base is simply the set of degree one elements of $\mathbb{F}_{2^{6120}}/\mathbb{F}_{2^{24}}$. As detailed in §2.2, quotienting out by the action of the 8-th power of Frobenius produces 21,932 distinct orbits. To obtain relations, as explained in §2.3, we make essential use of the single polynomial $X^{257} + X + 1$, which splits completely over $\mathbb{F}_{2^{24}}$. In particular, letting $y := x^{256}$ so that $x = \frac{y}{w+1}$, the $\mathbb{F}_{2^{6120}}$ element $xy + ay + bx + c$ corresponds to $X^{257} + aX^{256} + bX + c$ on the one hand, and $\frac{X^2}{w+1} + aX + \frac{bX}{w+1} + c$ on the other. The first of these transforms to $X^{257} + X + 1$ if and only if $(a^{256} + b)^{257} = (ab + c)^{256}$. So for randomly chosen (a, b) we compute c and check whether the corresponding quadratic splits. If it does – which occurs with probability $1/2$ – we obtain a relation. Thanks to the simplicity of this approach, we collected 22,932 relations and wrote these to a matrix in 15 seconds using C++/NTL [24].

[4] In our inital announcement [8] we stated a running time of 60 seconds for the relation generation. The reason for this higher running time was an unnecessary step of ordering the matrix entries, which we have discounted here.

4.3 Linear Algebra

We took as our modulus the product of the largest 35 factors of $2^{6120}-1$ listed in [8], which has bitlength 5121. We ran a parallelised C/GMP [9] implementation of Lanczos' algorithm on four of the Intel (Westmere) Xeon E5650 hex-core processors of ICHEC's SGI Altix ICE 8200EX Stokes cluster. This took 60.5 core-hours (just over 2.5 hours wall time).

4.4 Individual Logarithm

Degree 2 Elimination For computing the discrete logarithm of a degree two element $Q(x) = x^2 + q_1 x + q_0$ we try to equate $Q(x)$ with $x^{257} + ax^{256} + bx + c$, where $(a^{256} + b)^{257} = (ab + c)^{256}$. If this fails we apply the following strategy, making use of the fact that $\mathbb{F}_{2^{24}}$ can also be viewed as a field extension of $\mathbb{F}_{2^6}$. We consider $y = x^{256}$ and $\bar{x} = x^4$, so that $y = \bar{x}^{64}$ and $\bar{x} = (\frac{y}{\gamma})^4$ holds, and apply the large degree descent method to $\bar{Q}(X) := Q(\frac{X}{\gamma})$ (note that $\bar{Q}(y) = Q(x)$). Considering the lattice L (see §2.4) we construct a basis of the form $(X + u_0, u_1)$, $(v_0, X + v_1)$, where $u_0, u_1, v_0, v_1 \in \mathbb{F}_{2^{24}}$. Then for $s \in \mathbb{F}_{2^{24}}$ we have lattice elements $(X + u_0 + sv_0, sX + u_1 + sv_1) \in L$. Now for each $B \in \mathbb{F}_{2^{24}}$ such that $X^{65} + BX + B$ splits, we solve for $s \in \mathbb{F}_{2^{24}}$ satisfying

$$(v_0 s^2 + (u_0 + v_1)s + u_1)^{64} = B\,(s^{64} + v_0 s + u_0)^{65}\,,$$

which can be expressed as a quadratic system in the $\mathbb{F}_{2^6}$-components of s, and thus solved by a Gröbner basis computation over $\mathbb{F}_{2^6}$. We then have an equation

$$\bar{x}^{65} + a\bar{x}^{64} + b\bar{x} + c = \tfrac{1}{\gamma^4}(y^5 + by^4 + a\gamma^4 y + c\gamma^4)$$

with $a = s$, $b = \gamma s + q_1$, and $c = \frac{q_0}{\gamma}$, where the left-hand side polynomial splits, while the right-hand side polynomial contains $\bar{Q}(X)$.

The polynomial $X^5 + bX^4 + a\gamma^4 X + c\gamma^4 = \bar{Q}(X)R(X)$ has the property that $R(X)$ always factors into a linear and an irreducible quadratic polynomial over $\mathbb{F}_{q^k}$. Indeed, by a result of Bluher [2, Thm. 4.3], for any $B \in \mathbb{F}_{2^{24}}$ and any $d \geq 1$, the number of roots in $\mathbb{F}_{2^{24d}}$ of the polynomial $F_B(X) = X^5 + BX + B$ equals either 0, 1, 2, or 5. Since $X^5 + bX^4 + a\gamma^4 X + c\gamma^4$ can be rewritten as $X^5 + BX + B$ via a linear transformation (except when $a\gamma^4 = b^4$), the same holds also regarding the $\mathbb{F}_{2^{24d}}$-roots of this polynomial. Now applying Bluher's result for $d = 1$ we see that $R(X)$ can not split into linear factors, and by Bluher's result for $d = 3$ we conclude that $R(X)$ can not be irreducible. Hence, $R(X)$ is the product of linear and a quadratic polynomial, which we call $Q'(X)$.

Now if $Q'(X)$ is resolvable by the direct method, we have successfully eliminated the original polynomial $Q(X)$. The number of B such that $X^{65} + BX + B$ splits over $\mathbb{F}_q$ equals 64, according to Theorem 1, and by experiment, for each one the success probability to find a resolvable polynomial $Q'(X)$ is about 0.4.

Performing the Descent Using C++/NTL we first used continued fractions to express the target element β_π as a ratio of two 27-smooth polynomials, which took 10 core-hours, and then we applied the three different descent strategies as explained in §2.4.

We used the large degree descent strategy to express all of the featured polynomials using polynomials of degree 6 or less. This took a further 495 core-hours. While we could have performed this part of the descent more efficiently, as noted above we opted to find expressions which resulted in a relatively small number of degree 6 polynomials – which are the bottleneck eliminations for the subsequent descent – namely 326.

For degrees 6 down to 3 we used the analogue of Joux's small degree elimination method, based on the same polynomial that we used for relation generation, i.e., $X^{257} + X + 1$, rather than the polynomial $X^{256} + X$ that was used in [15], since the resulting performance was slightly better. Finally, we performed the degree 2 elimination as outlined above.

For convenience we coded the eliminations of polynomials of degrees 6 down to 2 in Magma [3] V2.16-12, using Faugere's F4 algorithm [4]. The total time for this part was just over 183.5 core-hours on a 2 GHz AMD Opteron computer.

For the logarithm modulo the cofactor of our modulus we used either linear search or Pollard's rho method, which took 20 minutes in total in C++/NTL. Thus the total time for the descent was just under 689 hours.

Finally, we found[5] that $\beta_\pi = g^{\log}$, with $\log =$

$$
\begin{aligned}
&1385875983639786926254757112831231710092363615038969923664959317045177002 8\\
&0127178022234894098617581360131441835074256363730624426814293233474272521 5\\
&9816612695792811682544311096540425383793880859540411103523802710777217882 2\\
&9392818734034519997318151400734817665137153584492793145567973524462468603 1\\
&7946750124475689474406274942356035936501674050933448909201029834522226732 2\\
&4777189708322321728205157364501360361304236778271636187781793837439382431 3\\
&0190736247863876184140375416811202840446593831929074368525263920877243047 7\\
&5451631271825250968111451400502733404381769675255289127346639350098221570 8\\
&4440038078851633249658388252243638191800820016703218635024510775134697959 6\\
&3146961536667161689514819480910600667301847667581377739443038754298308672 0\\
&5463918144256843911730747265146154193438041627833661739775057161236346096 2\\
&3656687525127784306232997304447548656106220435690856847147127938378103853 8\\
&8188844637969899060760798432481272520208397058864360712136505751867074569 4\\
&8584072378916942925369140868417196479573481032711481021729162865973588174 0
\end{aligned}
$$

[5] Magma verification code for this solution is available from [8].

9638991330560767785803399636173490553715036202472051577266078120885550543 4
3310557665700142118756029406335757638504575030790870743765853044705204113 2
0246292255375711457573555286060236699317039454479326718281128961423275142 7
8756942569053283328334404963552130259600089719251203669529880729403296453 0
9596913770872045463489601327600955441059801982552454932024128315938919847 8
8152417957691939817112366182063687529915365150361180214451234387656883256 1
4935599440505114958596916307530702664795603568367158954644853995513272611 2
0349386559612918562034222476803870290784735209511603344725254750716806726 2
3661587292720329606182512044312194357156139201340952037872975243254476081 5
5493700212295341594940726213723209985229839483842290764319139767329023834 4
1830460409758599159285365304456971453176680449737096483324156185041 .

4.5 Total Running Time

The total running time is $689 + 60.5 = 749.5$ core-hours. Note that most of the computation (all except the linear algebra part) was performed on a personal computer. On a modern quad-core PC, the total running time would be around a week.

5 Complexity Considerations

In this section we prove a tighter complexity result than that given in [12] for the new small-degree stage of the descent. As stated in §1, the systems arising from the small-degree elimination in §2.4 are quadratic, but not bilinear. As such, they do not necessarily enjoy the same resolution complexity as bilinear quadratic systems, as given by a theorem due to Spaenlehauer [25, Cor. 6.30]. However, if one instead reverts to using the polynomial $X^q - X$, then one can argue as follows.

Let the fields under consideration be $\mathbb{F}_{(q^k)^n}$, with $k \geq 3$ fixed, $n \approx q\delta_1$ and $\delta_1 \geq 1$ a small integer, as per the field representation described in [6, §3.3], and $q \to \infty$. This is achieved by finding a polynomial p_1 of degree δ_1 such that $p_1(X^q) - X \equiv 0 \pmod{I(X)}$, with $I(X)$ irreducible of degree n. By letting $x \in \mathbb{F}_{(q^k)^n}$ be a root of $I(X)$ and $y := x^q$, one also has $x = p_1(y)$, and therefore two related representations of $\mathbb{F}_{(q^k)^n}$.

For simplicity we assume $\delta_1 = 1$; the case $\delta_1 > 1$ can be treated similarly. The cardinality of $\mathbb{F}_{(q^k)^n}$ is $\approx q^{kq}$ and we have

$$\begin{aligned} L_{q^{kq}}(1/4, c) &= \exp\left((c + o(1))(kq \log q)^{1/4} (\log(kq \log q))^{3/4}\right) \\ &= \exp\left((ck^{1/4} + o(1))\, q^{1/4} \log q\right). \end{aligned} \tag{1}$$

We now recall Joux's elimination method. The final part of the descent starts with an element $Q(x)$ of degree $D \approx \alpha_1 q^{1/2}$ which is to be eliminated;

here, α_1 is a constant that depends on the efficiency of the classical large-degree descent. For a parameter $1 < d < D/2$ yet to be optimised, we substitute $X = f(X)/g(X)$ into $X^q - X$ with $\deg(f) = d$ and $\deg(g) = D - d$, both with yet-to-be determined $\mathbb{F}_{q^k}$ coefficients. In this case one has the $\mathbb{F}_{(q^k)^n}$-relation

$$f(x)^q g(x) - f(x)g(x)^q = \left(f(x)^q g(x) - f(x)g(x)^q\right) \bmod I(x). \tag{2}$$

By the factorisation of $X^q - X$ over $\mathbb{F}_q$, the LHS of Eq. (2) has irreducible factors of degree at most $D - d$. On the RHS one stipulates that it be zero mod $Q(x)$. This condition can be expressed as a bilinear quadratic system in the dk $\mathbb{F}_q$-components of the coefficients of f and the $(D-d)k$ $\mathbb{F}_q$-components of the coefficients of g. Since $Q(x)$ has D coefficients in $\mathbb{F}_{q^k}$ one expects there to be $O(1)$ solutions to this system when both f and g are monic. Hence by varying the leading coefficient of one of them, one expects many solutions.

The degree of the RHS of Eq. (2) depends on the representation of the field $\mathbb{F}_{(q^k)^n}$. Recall that in Joux's field representation, one has $h_0(X)$, $h_1(X)$ of very low degree δ_{h_0}, δ_{h_1} such that $h_1(X)X^q - h_0(X) \equiv 0 \pmod{I(X)}$, with $I(X)$ irreducible of degree n and $n \approx q$. Now on the RHS of Eq. (2) one replaces each occurrence of x^q by $h_0(x)/h_1(x)$, and thus the cofactor of $Q(x)$ on the RHS has degree $(D-d)(\max\{\delta_{h_0}, \delta_{h_1}\} - 1)$. For each solution to the bilinear quadratic system, it is tested for $(D-d)$-smoothness, and when it is, one has successfully represented $Q(x)$ as a product of at most q field elements of degree at most $D - d$ (ignoring the negligible number of factors from the cofactor).

Using our field representation, recall that $y = x^q$ and hence

$$f(x)^q = \sum_{i=0}^{d} f_i^q y^i \quad \text{and} \quad g(x)^q = \sum_{j=0}^{D-d} g_j^q y^j \ .$$

Then also using $x = p_1(y)$, the RHS of Eq. (2) becomes:

$$\left(\sum_{i=0}^{d} f_i^q y^i\right)\left(\sum_{j=0}^{D-d} g_j p_1(y)^j\right) - \left(\sum_{i=0}^{d} f_i p_1(y)^i\right)\left(\sum_{j=0}^{D-d} g_j^q y^j\right),$$

so that the cofactor of $Q(y)$ has degree $(D-d)(\delta_1 - 1)$ in y.

By repeating the above elimination technique recursively for each element occurring in the product until only degree one or degree two elements remain, the logarithm of $Q(x)$ is computed. So what is the optimal d? Joux's analysis [12] indicates that $d = O(q^{1/4}(\log q)^{1/2})$ should be used, giving an overall complexity of $\exp\left((c' + o(1))\, q^{1/4}(\log q)^{3/2}\right)$ for some c', which is $L_{q^{kq}}(1/4 + o(1), c')$, due to the presence of the extra $(\log q)^{1/2}$ factor, relative to Eq. (1).

However, one can instead set $d \approx \alpha_2 q^{1/4}$, as we now show (the constant α_2 is to be optimised later). Let $C(D, d)$ be the cost of expressing a degree D element as a product of elements of degree at most d, when the numerator f

has degree d at each step. If $C_0(D,d)$ is the cost of resolving the corresponding bilinear quadratic system, we have

$$\begin{aligned} C(D,d) &= C_0(D,d) + q\,C(D-d,d) \\ &= C_0(D,d) + q\,\big(C_0(D-d,d) + q\,C(D-2d,d)\big) \\ &= \cdots = \sum_{i=0}^{\lfloor D/d \rfloor - 1} q^i C_0(D-id,d)\,. \end{aligned}$$

Since $C_0(D-id,d) \le C_0(D,d)$ for all i and since $\sum_{i=0}^{\lfloor D/d \rfloor - 1} q^i \le q^{D/d}$ we get the upper bound

$$C(D,d) \le q^{D/d} C_0(D,d)\,.$$

As in [12], we need the following essential lemma.

Lemma 1 ([25, Cor. 6.30]). *The arithmetic complexity (measured in $\mathbb{F}_q$-operations) of computing a Gröbner basis of a generic bilinear system $f_1,\ldots,f_{n_x+n_y} \in \mathbb{F}_q[x_0,\ldots,x_{n_x-1},y_0,\ldots,y_{n_y-1}]$ with Faugere's F4 algorithm [4] is bounded by*

$$O\Big(\min(n_x,n_y)\,(n_x+n_y)\binom{n_x+n_y+\min(n_x,n_y)+2}{\min(n_x,n_y)+2}^{\omega}\Big)\,,$$

where ω is the exponent of matrix multiplication.

Hence, using the estimate $\binom{a+2}{b+2} \le (\frac{a}{b})^2\binom{a}{b} \le (\frac{a}{b})^2(e\,\frac{a}{b})^b = e^b(\frac{a}{b})^{b+2}$, we have

$$C_0(D,d) = O\Big(k^2 Dd\binom{k(D+d)+2}{kd+2}^{\omega}\Big) = O\Big(k^2 Dde^{k\omega d}\left(\frac{D+d}{d}\right)^{k\omega d+2\omega}\Big)\,,$$

and, neglecting the lower order terms, we get

$$\log C_0(D,d) = \big(k\omega d\log(D/d)\big)(1+o(1))\,.$$

Therefore, we have

$$\begin{aligned} \log C(D,d) &= \big((D/d)\log q + k\omega d\log(D/d)\big)(1+o(1)) \\ &= \Big(\left(\frac{\alpha_1}{\alpha_2} + \frac{k\omega\alpha_2}{4}\right) q^{1/4}\log q\Big)(1+o(1))\,, \end{aligned}$$

and in particular, for the optimal choice $\alpha_2 = (4\alpha_1/k\omega)^{1/2}$, we get

$$\log C(D,d) = \big((k\omega\alpha_1)^{1/2} q^{1/4}\log q\big)(1+o(1))\,.$$

Thus, taking into account Eq. (1), we arrive at the complexity

$$C(D,d) = L_{q^{kq}}(1/4\,,\,k^{1/4}(\omega\alpha_1)^{1/2})\,. \tag{3}$$

Observe that the number of degree $d \approx \alpha_2 q^{1/4}$ elements in such an expression for the initial degree $D \approx \alpha_1 q^{1/2}$ element is $O(q^{(\alpha_1/\alpha_2)q^{1/4}})$. Note that this choice of d represents the optimal balance between the number of nodes in the descent tree at level d and the cost of resolving the bilinear systems.

Moreover, exactly the same argument shows that $C(\alpha_j q^{1/2^j}, \alpha_{j+1} q^{1/2^{j+1}}) = L_{q^{kq}}(1/2^{j+1})$, and so the cost of expressing each of the $L_{q^{kq}}(1/4)$ degree $\alpha_2 q^{1/4}$ elements in terms of elements of degree $\alpha_3 q^{1/8}$ is $L_{q^{kq}}(1/8)$, and therefore for any $j > 1$ the total cost down to degree $\alpha_j q^{1/2^j}$ never exceeds $L_{q^{kq}}(1/4)$. After $j = \lceil \log_2 \log_2 q \rceil$ of the above sequence of steps we have $\lfloor q^{1/2^j} \rfloor = 1$, and the total cost is precisely that given in Eq. (3).

As the complexity of the initial splitting of a target element into a product of elements of degree at most $\alpha_0 q^{3/4}$ is $L_{q^{kq}}(1/4)$, as is the complexity of classical descent from degree $\alpha_0 q^{3/4}$ to degree $\alpha_1 q^{1/2}$, the above tighter analysis demonstrates that for the fields considered, Joux's algorithm has complexity $L_{q^{kq}}(1/4)$ as well, for both his and our field representations. We have omitted the determination of the optimal parameters α_0 and α_1, since this is beyond our focus on proving that the full algorithm is $L(1/4)$.

References

1. Paulo S. L. M. Barreto, Steven D. Galbraith, Colm Ó' hÉigeartaigh, and Michael Scott. Efficient pairing computation on supersingular abelian varieties. *Des. Codes Cryptogr.*, 42(3):239–271, 2007.
2. Antonia W. Bluher. On $x^{q+1} + ax + b$. *Finite Fields and Their Applications*, 10(3):285–305, 2004.
3. Wieb Bosma, John Cannon, and Catherine Playoust. The Magma algebra system. I. The user language. *J. Symbolic Comput.*, 24(3-4):235–265, 1997.
4. Jean-Charles Faugére. A new efficient algorithm for computing Gröbner bases (F_4). *J. Pure Appl. Algebra*, 139(1-3):61–88, 1999.
5. Pierrick Gaudry, Florian Hess, and Nigel P. Smart. Constructive and destructive facets of Weil descent on elliptic curves. *J. Cryptology*, 15(1):19–46, 2002.
6. Faruk Göloğlu, Robert Granger, Gary McGuire, and Jens Zumbrägel. On the function field sieve and the impact of higher splitting probabilities: Application to discrete logarithms in $\mathbb{F}_{2^{1971}}$ and $\mathbb{F}_{2^{3164}}$. In Ran Canetti and Juan Garay, editors, *CRYPTO 2013*, volume 8043 of *LNCS*, pages 109–128. Springer, Heidelberg, 2013.
7. Faruk Göloğlu, Robert Granger, Gary McGuire, and Jens Zumbrägel. Discrete Logarithms in $GF(2^{1971})$. NMBRTHRY list, 19 Feb 2013.
8. Faruk Göloğlu, Robert Granger, Gary McGuire, and Jens Zumbrägel. Discrete Logarithms in $GF(2^{6120})$. NMBRTHRY list, 11 Apr 2013.
9. Torbjörn Granlund and the GMP development team. *GNU MP: The GNU Multiple Precision Arithmetic Library*, 5.0.5 edition, 2012. http://gmplib.org/.
10. Tor Helleseth and Alexander Kholosha. $x^{2^l+1} + x + a$ and related affine polynomials over GF(2^k). *Cryptogr. Commun.*, 2(1):85–109, 2010.
11. Antoine Joux. Faster index calculus for the medium prime case. Application to 1175-bit and 1425-bit finite fields. In Thomas Johansson and Phong Q. Nguyen, editors, *EUROCRYPT 2013*, volume 7881 of *LNCS*, pages 177–193. Springer, Heidelberg, 2013.
12. Antoine Joux. A new index calculus algorithm with complexity $L(1/4 + o(1))$ in very small characteristic. Cryptology ePrint Archive, Report 2013/095, 2013. http://eprint.iacr.org/.

13. Antoine Joux. Personal communication, 2013.
14. Antoine Joux. Discrete Logarithms in $GF(2^{1778})$. NMBRTHRY list, 11 Feb 2013.
15. Antoine Joux. Discrete Logarithms in $GF(2^{4080})$. NMBRTHRY list, 22 Mar 2013.
16. Antoine Joux. Discrete Logarithms in $GF(2^{6168})$. NMBRTHRY list, 21 May 2013.
17. Antoine Joux and Reynald Lercier. The function field sieve in the medium prime case. In Serge Vaudenay, editor, *EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 254–270. Springer, Heidelberg, 2006.
18. Brian A. LaMacchia and Andrew M. Odlyzko. Solving large sparse linear systems over finite fields. In Alfred J. Menezes and Scott A. Vanstone, editors, *CRYPTO '90*, volume 537 of *LNCS*, pages 109–133. Springer, Heidelberg, 1991.
19. Cornelius Lanczos. An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. *J. Research Nat. Bur. Standards*, 45:255–282, 1950.
20. Arjen K. Lenstra and Hendrik W. Lenstra, Jr., editors. *The development of the number field sieve*, volume 1554 of *Lecture Notes in Mathematics*. Springer, Heidelberg, 1993.
21. Hendrik W. Lenstra, Jr. Factoring integers with elliptic curves. *Ann. of Math. (2)*, 126(3):649–673, 1987.
22. Hendrik W. Lenstra, Jr. Finding isomorphisms between finite fields. *Math. Comp.*, 56(193):329–347, 1991.
23. Ilya Popovyan. Efficient parallelization of lanczos type algorithms. Cryptology ePrint Archive, Report 2011/416, 2011. http://eprint.iacr.org/.
24. Victor Shoup. *NTL: A library for doing number theory*, 5.5.2 edition, 2009. http://www.shoup.net/ntl/.
25. Pierre-Jean Spaenlehauer. Solving multihomogeneous and determinantal systems algorithms - complexity - applications. Ph.D. thesis, Université Pierre et Marie Curie (UPMC), 2012.

Breaking '128-bit Secure' Supersingular Binary Curves*

(or how to solve discrete logarithms in $\mathbb{F}_{2^{4\cdot 1223}}$ and $\mathbb{F}_{2^{12\cdot 367}}$)

Robert Granger[1], Thorsten Kleinjung[1], and Jens Zumbrägel[2]

[1] Laboratory for Cryptologic Algorithms, EPFL, Switzerland
[2] Institute of Algebra, TU Dresden, Germany
robbiegranger@gmail.com, thorsten.kleinjung@epfl.ch, jens.zumbragel@ucd.ie

Abstract. In late 2012 and early 2013 the discrete logarithm problem (DLP) in finite fields of small characteristic underwent a dramatic series of breakthroughs, culminating in a heuristic quasi-polynomial time algorithm, due to Barbulescu, Gaudry, Joux and Thomé. Using these developments, Adj, Menezes, Oliveira and Rodríguez-Henríquez analysed the concrete security of the DLP, as it arises from pairings on (the Jacobians of) various genus one and two supersingular curves in the literature, which were originally thought to be 128-bit secure. In particular, they suggested that the new algorithms have no impact on the security of a genus one curve over $\mathbb{F}_{2^{1223}}$, and reduce the security of a genus two curve over $\mathbb{F}_{2^{367}}$ to 94.6 bits. In this paper we propose a new field representation and efficient general descent principles which together make the new techniques far more practical. Indeed, at the '128-bit security level' our analysis shows that the aforementioned genus one curve has approximately 59 bits of security, and we report a total break of the genus two curve.

Keywords: Discrete logarithm problem, supersingular binary curves, pairings, finite fields

1 Introduction

The role of small characteristic supersingular curves in cryptography has been a varied and an interesting one. Having been eschewed by the cryptographic community for succumbing spectacularly to the subexponential MOV attack in 1993 [40], which maps the DLP from an elliptic curve (or more generally, the Jacobian of a higher genus curve) to the DLP in a small degree extension of the base field of the curve, they made a remarkable comeback with the advent of pairing-based cryptography in 2001 [42, 31, 9]. In particular, for the latter it was reasoned that the existence of a subexponential attack on the DLP does not *ipso facto* warrant their complete exclusion; rather, provided that the finite field DLP into which the elliptic curve DLP embeds is sufficiently hard, this state of affairs would be acceptable.

* The second author acknowledges the support of the Swiss National Science Foundation, via grant numbers 206021-128727 and 200020-132160, while the third author acknowledges the support of the Irish Research Council, grant number ELEVATEPD/2013/82.

Neglecting the possible existence of native attacks arising from the supersingularity of these curves, much research effort has been expended in making instantiations of the required cryptographic operations on such curves as efficient as possible [6, 17, 14, 28, 27, 5, 30, 7, 11, 18, 3, 1], to name but a few, with the associated security levels having been estimated using Coppersmith's algorithm from 1984 [12, 39]. Alas, a series of dramatic breakthrough results for the DLP in finite fields of small characteristic have potentially rendered all of these efforts in vain.

The first of these results was due to Joux, in December 2012, and consisted of a more efficient method – dubbed 'pinpointing' – to obtain relations between factor base elements [32]. For medium-sized base fields, this technique has heuristic complexity as low as $L(1/3, 2^{1/3}) \approx L(1/3, 1.260)^{\dagger}$, where as usual $L(\alpha, c) = L_Q(\alpha, c) = \exp((c + o(1))(\log Q)^{\alpha}(\log\log Q)^{1-\alpha})$, with Q the cardinality of the field. This improved upon the previous best complexity of $L(1/3, 3^{1/3}) \approx L(1/3, 1.442)$ due to Joux and Lercier [37]. Using this technique Joux solved example DLPs in fields of bitlength 1175 and 1425, both with prime base fields.

Then in February 2013, Göloğlu, Granger, McGuire and Zumbrägel used a specialisation of the Joux-Lercier doubly-rational function field sieve (FFS) variant [37], in order to exploit a well-known family of 'splitting polynomials', i.e., polynomials which split completely over the base field [19]. For fields of the form $\mathbb{F}_{q^{kn}}$ with $k \geq 3$ fixed ($k = 2$ is even simpler) and $n \approx dq$ for a fixed integer $d \geq 1$, they showed that for binary (and more generally small characteristic) fields, relation generation for degree one elements runs in heuristic *polynomial time*, as does finding the logarithms of degree two elements (if q^k can be written as $q'^{k'}$ for $k' \geq 4$), once degree one logarithms are known. For medium-sized base fields of small characteristic a heuristic complexity as low as $L(1/3, (4/9)^{1/3}) \approx L(1/3, 0.763)$ was attained; this approach was demonstrated via the solution of example DLPs in the fields $\mathbb{F}_{2^{1971}}$ [21] and $\mathbb{F}_{2^{3164}}$.

After the initial publication of [19], Joux released a preprint [33] detailing an algorithm for solving the discrete logarithm problem for fields of the form $\mathbb{F}_{q^{2n}}$, with $n \leq q + d$ for some very small d, which was used to solve a DLP in $\mathbb{F}_{2^{1778}}$ [34] and later in $\mathbb{F}_{2^{4080}}$ [35]. For $n \approx q$ this algorithm has heuristic complexity $L(1/4 + o(1), c)$ for some undetermined c, and also has a heuristic polynomial time relation generation method, similar in principle to that in [19]. While the degree two element elimination method in [19] is arguably superior – since elements can be eliminated on the fly – for other small degrees Joux's elimination method is faster, resulting in the stated complexity.

In April 2013 Göloğlu *et al.* combined their approach with Joux's to solve an example DLP in the field $\mathbb{F}_{2^{6120}}$ [22] and later demonstrated that Joux's algorithm can be tweaked to have heuristic complexity $L(1/4, c)$ [20], where c can be as low as $(\omega/8)^{1/4}$ [24], with ω the linear algebra constant, i.e., the

† The original paper states a complexity of $L(1/3, (8/9)^{1/3}) \approx L(1/3, 0.961)$; however, on foot of recent communications the constant should be as stated.

exponent of matrix multiplication. Then in May 2013, Joux announced the solution of a DLP in the field $\mathbb{F}_{2^{6168}}$ [36].

Most recently, in June 2013, Barbulescu, Gaudry, Joux and Thomé announced a *quasi-polynomial time* for solving the DLP [4], for fields $\mathbb{F}_{q^{kn}}$ with $k \geq 2$ fixed and $n \leq q + d$ with d very small, which for $n \approx q$ has heuristic complexity

$$(\log q^{kn})^{O(\log \log q^{kn})}. \tag{1}$$

Since (1) is smaller than $L(\alpha, c)$ for any $\alpha > 0$, it is asymptotically the most efficient algorithm known for solving the DLP in finite fields of small characteristic, which can always be embedded into a field of the required form. Interestingly, the algorithmic ingredients and analysis of this algorithm are much simpler than for Joux's $L(1/4 + o(1), c)$ algorithm.

Taken all together, one would expect the above developments to have a substantial impact on the security of small characteristic parameters appearing in the pairing-based cryptography literature. However, all of the record DLP computations mentioned above used Kummer or twisted Kummer extensions (those with n dividing $q^k \mp 1$), which allow for a reduction in the size of the factor base by a factor of kn and make the descent phase for individual logarithms relatively easy. While such parameters are preferable for setting records (most recently in $\mathbb{F}_{2^{9234}}$ [26]), none of the parameters featured in the literature are of this form, and so it is not *a priori* clear whether the new techniques weaken existing pairing-based protocol parameters.

A recent paper by Adj, Menezes, Oliveira and Rodríguez-Henríquez has begun to address this very issue [2]. Using the time required to compute a single multiplication modulo the cardinality of the relevant prime order subgroup as their basic unit of time, which we denote by M_r, they showed that the DLP in the field $\mathbb{F}_{3^{6 \cdot 509}}$ costs at most $2^{73.7}\ M_r$. One can arguably interpret this result to mean that this field has 73.7 bits of security[†]. This significantly reduces the intended security level of 128 bits (or 111 bits as estimated by Shinohara *et al.* [43], or 102.7 bits for the Joux-Lercier FFS variant with pinpointing, as estimated in [2]). An interesting feature of their analysis is that during the descent phase, some elimination steps are performed using the method from the quasi-polynomial time algorithm of Barbulescu *et al.*, when one might have expected these steps to only come into play at much higher bitlengths, due to the high arity of the arising descent nodes.

In the context of binary fields, Adj *et al.* considered in detail the DLP in the field $\mathbb{F}_{2^{12 \cdot 367}}$, which arises via a pairing from the DLP on the Jacobian of a supersingular genus two curve over $\mathbb{F}_{2^{367}}$, first proposed in [3], with embedding degree 12. Using all of the available techniques they provided an upper bound

[†] The notion of bit security is quite fuzzy; for the elliptic curve DLP it is usually intended to mean the logarithm to the base 2 of the expected number of group operations, however for the finite field DLP different authors have used different units, perhaps because the cost of various constituent algorithms must be amortised into a single cost measure. In this work we time everything in seconds, while to achieve a comparison with [2] we convert to M_r.

of $2^{94.6}$ M_r for the cost of breaking the DLP in the embedding field, which is some way below the intended 128-bit security level. In their conclusion Adj *et al.* also suggest that a commonly implemented genus one supersingular curve over $\mathbb{F}_{2^{1223}}$ with embedding degree 4 $[30, 7, 11, 18, 1]$, is not weakened by the new algorithmic advances, i.e., its security remains very close to 128 bits.

In this work we show that the above security estimates were incredibly optimistic. Our techniques and results are summarised as follows.

- **Field representation:** We introduce a new field representation that can have a profound effect on the resulting complexity of the new algorithms. In particular it permits the use of a smaller q than before, which not only speeds up the computation of factor base logarithms, but also the descent (both classical and new).
- **Exploit subfield membership:** During the descent phase we apply a *principle of parsimony*, by which one should always try to eliminate an element in the target field, and only when this is not possible should one embed it into an extension field. So although the very small degree logarithms may be computed over a larger field, the descent cost is *greatly reduced* relative to solving a DLP in the larger field.
- **Further descent tricks:** The above principle also means that elements can automatically be rewritten in terms of elements of smaller degree, via factorisation over a larger field, and that elements can be eliminated via Joux's Gröbner basis computation method [33] with $k = 1$, rather than $k > 1$, which increases its degree of applicability.
- **'128-bit secure' genus one DLP:** We show that the DLP in $\mathbb{F}_{2^{4\cdot 1223}}$ can be solved in approximately 2^{40} s, or 2^{59} M_r, with r a 1221-bit prime.
- **'128-bit secure' genus two DLP:** We report a total break of the DLP in $\mathbb{F}_{2^{12\cdot 367}}$ (announced in [25]), which took about 52240 core-hours.
- **$\mathbf{L(1/4, c)}$ technique only:** Interestingly, using our approach the elimination steps à la Barbulesu *et al.* [4] were not necessary for the above estimate and break.

The rest of the paper is organised as follows. In §2 we describe our field representation and our target fields. In §3 we present the corresponding polynomial time relation generation method for degree one elements and degree two elements (although we do not need the latter for the fields targeted in the present paper), as well as how to apply Joux's small degree elimination method [33] with the new representation. We then apply these and other techniques to $\mathbb{F}_{2^{4\cdot 1223}}$ in §4 and to $\mathbb{F}_{2^{12\cdot 367}}$ in §5 . Finally, we conclude in §6.

2 Field Representation and Target Fields

In this section we introduce our new field representation and the fields whose DLP security we will address. This representation, as well as some preliminary security estimates, were initially presented in [23].

2.1 Field Representation

Although we focus on binary fields in this paper, for the purposes of generality, in this section we allow for extension fields of arbitrary characteristic. Hence let $q = p^l$ for some prime p, and let $\mathbb{K} = \mathbb{F}_{q^{kn}}$ be the field under consideration, with $k \geq 1$.

We choose a positive integer d_h such that $n \leq qd_h + 1$, and then choose $h_0, h_1 \in \mathbb{F}_{q^k}[X]$ with $\max\{\deg(h_0), \deg(h_1)\} = d_h$ such that

$$h_1(X^q)X - h_0(X^q) \equiv 0 \pmod{I(X)}, \tag{2}$$

where $I(X)$ is an irreducible degree n polynomial in $\mathbb{F}_{q^k}[X]$. Then $\mathbb{K} = \mathbb{F}_{q^k}[X]/(I(X))$. Denoting by x a root of $I(X)$, we introduce the auxiliary variable $y = x^q$, so that one has two isomorphic representations of $\mathbb{K}$, namely $\mathbb{F}_{q^k}(x)$ and $\mathbb{F}_{q^k}(y)$, with $\sigma : \mathbb{F}_{q^k}(y) \to \mathbb{F}_{q^k}(x) : y \mapsto x^q$. To establish the inverse isomorphism, note that by (2) in $\mathbb{K}$ we have $h_1(y)x - h_0(y) = 0$, and hence $\sigma^{-1} : \mathbb{F}_{q^k}(x) \to \mathbb{F}_{q^k}(y) : x \mapsto h_0(y)/h_1(y)$.

The knowledgeable reader will have observed that our representation is a synthesis of two other useful representations: the one used by Joux [33], in which one searches for a degree n factor $I(X)$ of $h_1(X)X^q - h_0(X)$; and the one used by Göloğlu *et al.* [19, 20], in which one searches for a degree n factor $I(X)$ of $X - h_0(X^q)$. The problem with the former is that it constrains n to be approximately q. The problem with the latter is that the polynomial $X - h_0(X^q)$ is insufficiently general to represent all degrees n up to qd_h. By changing the coefficient of X in the latter from 1 to $h_1(X^q)$, we greatly increase the probability of overcoming the second problem, thus combining the higher degree coverage of Joux's representation with the higher degree possibilities of [19, 20].

The *raison d'être* of using this representation rather than Joux's representation is that for a given n, by choosing $d_h > 1$, one may use a smaller q. So why is this useful? Well, since the complexity of the new descent methods is typically a function of q, then subject to the satisfaction of certain constraints, one may use a smaller q, thus reducing the complexity of solving the DLP. This observation was our motivation for choosing field representations of the above form.

Another advantage of having an h_1 coefficient (which also applies to Joux's representation) is that it increases the chance of there being a suitable (h_1, h_0) pair with coefficients defined over a proper subfield of $\mathbb{F}_{q^k}$, which then permits one to apply the factor base reduction technique of [37], see §4 and §5.

2.2 Target Fields

For $i \in \{0,1\}$ let $E_i/\mathbb{F}_{2^p} : Y^2 + Y = X^3 + X + i$. These elliptic curves are supersingular and can have prime or nearly prime order only for p prime, and have embedding degree 4 [16, 6, 17]. We focus on the curve

$$E_0/\mathbb{F}_{2^{1223}} : Y^2 + Y = X^3 + X, \tag{3}$$

which has a prime order subgroup of cardinality $r_1 = (2^{1223} + 2^{612} + 1)/5$, of bitlength 1221. This curve was initially proposed for 128-bit secure protocols [30] and has enjoyed several optimised implementations [7, 11, 18, 1]. Many smaller p have also been proposed in the literature (see [5, 16], for instance), and are clearly weaker.

For $i \in \{0,1\}$ let $H_i/\mathbb{F}_{2^p} : Y^2 + Y = X^5 + X^3 + i$. These genus two hyperelliptic curves are supersingular and can have a nearly prime order Jacobian only for p prime (note that 13 is always a factor of $\#\mathrm{Jac}_{H_0}(\mathbb{F}_{2^p})$, since $\#\mathrm{Jac}_{H_0}(\mathbb{F}_2) = 13$), and have embedding degree 12 [5, 16]. We focus on the curve

$$H_0/\mathbb{F}_{2^{367}} : Y^2 + Y = X^5 + X^3, \tag{4}$$

with $\#\mathrm{Jac}_H(\mathbb{F}_{2^{367}}) = 13 \cdot 7170258097 \cdot r_2$, and $r_2 = (2^{734} + 2^{551} + 2^{367} + 2^{184} + 1)/(13 \cdot 7170258097)$ is a 698-bit prime, since this was proposed for 128-bit secure protocols [3], and whose security was analysed in depth by Adj *et al.* in [2].

3 Computing the Logarithms of Small Degree Elements

In this section we adapt the polynomial time relation generation method from [19] and Joux's small degree elimination method [33] to the new field representation as detailed in §2.1. Note that henceforth, we shall refer to elements of $\mathbb{F}_{q^{kn}} = \mathbb{F}_{q^k}[X]/(I(X))$ as field elements or as polynomials, as appropriate, and thus use x and X (and y and Y) interchangeably. We therefore freely apply polynomial ring concepts, such as degree, factorisation and smoothness, to field elements.

In order to compute discrete logarithms in our target fields we apply the usual index calculus method. It consists of a precomputation phase in which by means of (sparse) linear algebra techniques one obtains the logarithms of the factor base elements, which will consist of the low degree irreducible polynomials. Afterwards, in the individual logarithm phase, one applies procedures to recursively rewrite each element as a product of elements of smaller degree, in this way building up a *descent* tree, which has the target element as its root and factor base elements as its leaves. This proceeds in several stages, starting with a continued fraction descent of the target element, followed by a special-Q lattice descent (referred to as degree-balanced classical descent, see [19]), and finally using Joux's Gröbner basis descent [33] for the lower degree elements. Details of the continued fraction and classical descent steps are given

in §4, while in this section we provide details of how to find the logarithms of elements of small degree.

We now describe how the logarithms of degree one and two elements (when needed) are to be computed. We use the relation generation method from [19], rather than Joux's method [33], since it automatically avoids duplicate relations. For $k \geq 2$ we first precompute the set $\mathcal{S}_k$, where

$$\mathcal{S}_k = \{(a,b,c) \in (\mathbb{F}_{q^k})^3 \mid X^{q+1} + aX^q + bX + c \text{ splits completely over } \mathbb{F}_{q^k}\}.$$

For $k = 2$, this set of triples is parameterised by $(a, a^q, \mathbb{F}_q \ni c \neq a^{q+1})$, of which there are precisely $q^3 - q^2$ elements. For $k \geq 3$, $\mathcal{S}_k$ can also be computed very efficiently, as follows. Assuming $c \neq ab$ and $b \neq a^q$, the polynomial $X^{q+1} + aX^q + bX + c$ may be transformed (up to a scalar factor) into the polynomial $f_B(\overline{X}) = \overline{X}^{q+1} + B\overline{X} + B$, where $B = \frac{(b-a^q)^{q+1}}{(c-ab)^q}$, and $X = \frac{c-ab}{b-a^q}\overline{X} - a$. The set $\mathcal{L}$ of $B \in \mathbb{F}_{q^k}$ for which f_B splits completely over $\mathbb{F}_{q^k}$ can be computed by simply testing for each such B whether this occurs, and there are precisely $(q^{k-1}-1)/(q^2-1)$ such B if k is odd, and $(q^{k-1}-q)/(q^2-1)$ such B if k is even [8]. Then for any (a,b) such that $b \neq a^q$ and for each $B \in \mathcal{L}$, we compute via $B = \frac{(b-a^q)^{q+1}}{(c-ab)^q}$ the corresponding (unique) $c \in \mathbb{F}_{q^k}$, which thus ensures that $(a,b,c) \in \mathcal{S}_k$. Note that in all cases we have $|\mathcal{S}_k| \approx q^{3k-3}$.

3.1 Degree 1 Logarithms

We define the factor base $\mathcal{B}_1$ to be the set of linear elements in x, i.e., $\mathcal{B}_1 = \{x - a \mid a \in \mathbb{F}_{q^k}\}$. Observe that the elements linear in y are each expressible in $\mathcal{B}_1$, since $(y-a) = (x - a^{1/q})^q$.

As in [37, 19, 20], the basic idea is to consider elements of the form $xy + ay + bx + c$ with $(a,b,c) \in \mathcal{S}_k$. The above two field isomorphisms induce the following equality in $\mathbb{K}$:

$$x^{q+1} + ax^q + bx + c = \frac{1}{h_1(y)}\big(yh_0(y) + ayh_1(y) + bh_0(y) + ch_1(y)\big). \tag{5}$$

When the r.h.s. of (5) also splits completely over $\mathbb{F}_{q^k}$, one obtains a relation between elements of $\mathcal{B}_1$ and the logarithm of $h_1(y)$. One can either adjoin $h_1(y)$ to the factor base, or simply use an $h_1(y)$ which splits completely over $\mathbb{F}_{q^k}$.

We assume that for each $(a,b,c) \in \mathcal{S}_k$ that the r.h.s. of (5) – which has degree $d_h + 1$ – splits completely over $\mathbb{F}_{q^k}$ with probability $1/(d_h+1)!$. Hence in order for there to be sufficiently many relations we require that

$$\frac{q^{3k-3}}{(d_h+1)!} > q^k, \quad \text{or equivalently} \quad q^{2k-3} > (d_h+1)!. \tag{6}$$

When this holds, the expected cost of relation generation is $(d_h+1)! \cdot q^k \cdot S_{q^k}(d_h+1, 1)$, where $S_{q^k}(n,m)$ denotes the cost of testing whether a degree n

polynomial is m-smooth, i.e., has all of its irreducible factors of degree $\leq m$. The cost of solving the resulting linear system using sparse linear algebra techniques is $O(q^{2k+1})$ arithmetic operations modulo the order r subgroup in which one is working.

3.2 Degree 2 Logarithms

For degree two logarithms, there are several options. The simplest is to apply the degree one method over a quadratic extension of $\mathbb{F}_{q^k}$, but in general (without any factor base automorphisms) this will cost $O(q^{4k+1})$ modular arithmetic operations. If $k \geq 4$ then subject to a condition on q, k and d_h, it is possible to find the logarithms of irreducible degree two elements on the fly, using the techniques of [19, 20]. In fact, for the DLP in $\mathbb{F}_{2^{12\cdot 367}}$ we use both of these approaches, but for different base fields, see §5.

Although not used in the present paper, for completeness we include here the analogue in our field representation of Joux's approach [33]. Since this approach forms the basis of the higher degree elimination steps in the quasi-polynomial time algorithm of Barbulescu *et al.*, its analogue in our field representation should be clear.

We define $\mathcal{B}_{2,u}$ to be the set of irreducible elements of $\mathbb{F}_{q^k}[X]$ of the form $X^2 + uX + v$. For each $u \in \mathbb{F}_{q^k}$ one expects there to be about $q^k/2$ such elements[†]. As in [33], for each $u \in \mathbb{F}_{q^k}$ we find the logarithms of all the elements of $\mathcal{B}_{2,u}$ simultaneously. To do so, consider (5) but with x on the l.h.s. replaced with $Q = x^2+ux$. Using the field isomorphisms we have that $Q^{q+1}+aQ^q+bQ+c$ is equal to

$$\begin{aligned}&(y^2+u^q y)\big((\tfrac{h_0(y)}{h_1(y)})^2+u(\tfrac{h_0(y)}{h_1(y)})\big) + a(y^2+u^q y) + b\big((\tfrac{h_0(y)}{h_1(y)})^2+u(\tfrac{h_0(y)}{h_1(y)})\big) + c\\ &= \tfrac{1}{h_1(y)^2}\big((y^2+u^q y)(h_0(y)^2+uh_0(y)h_1(y)+ah_1(y)^2)+b(h_0(y)^2+uh_0(y)h_1(y))+ch_1(y)^2\big).\end{aligned}$$

The degree of the r.h.s. is $2(d_h + 1)$, and when it splits completely over $\mathbb{F}_{q^k}$ we have a relation between elements of $\mathcal{B}_{2,u}$ and degree one elements, whose logarithms are presumed known, which we assume occurs with probability $1/(2(d_h + 1))!$. Hence in order for there to be sufficiently many relations we require that

$$\frac{q^{3k-3}}{(2(d_h+1))!} > \frac{q^k}{2}, \quad \text{or equivalently} \quad q^{2k-3} > (2(d_h+1))!/2. \tag{7}$$

Observe that (7) implies (6). When this holds, the expected cost of relation generation is $(2(d_h + 1))! \cdot q^k \cdot S_{q^k}(2(d_h + 1), 1)/2$. The cost of solving the resulting linear system using sparse linear algebra techniques is again $O(q^{2k+1})$ modular arithmetic operations, where now both the number of variables and the average weight is halved relative to the degree one case. Since there are q^k

[†] For binary fields there are precisely $q^k/2$ irreducibles, since $X^2 + uX + v$ is irreducible if and only if $\mathrm{Tr}_{\mathbb{F}_{q^k}/\mathbb{F}_2}(v/u^2) = 1$.

such u, the total expected cost of this stage is $O(q^{3k+1})$ modular arithmetic operations, which may of course be parallelised.

3.3 Joux's Small Degree Elimination with the New Representation

As in [33], let Q be a degree d_Q element to be eliminated, let $F(X) = \sum_{i=0}^{d_F} f_i X^i$, $G(X) = \sum_{j=0}^{d_G} g_j X^j \in \mathbb{F}_{q^k}[X]$ with $d_F + d_G + 2 \geq d_Q$, and assume without loss of generality $d_F \geq d_G$. Consider the following expression:

$$G(X) \prod_{\alpha \in \mathbb{F}_q} (F(X) - \alpha\, G(X)) = F(X)^q G(X) - F(X) G(X)^q \tag{8}$$

The l.h.s. is $\max(d_F, d_G)$-smooth. The r.h.s. can be expressed modulo $h_1(X^q)X - h_0(X^q)$ in terms of $Y = X^q$ as a quotient of polynomials of relatively low degree by using

$$F(X)^q = \sum_{i=0}^{d_F} f_i^q Y^i, \quad G(X)^q = \sum_{j=0}^{d_G} g_j^q Y^j \quad \text{and} \quad X \equiv \frac{h_0(Y)}{h_1(Y)}.$$

Then the numerator of the r.h.s. becomes

$$\Big(\sum_{i=0}^{d_F} f_i^q Y^i\Big)\Big(\sum_{j=0}^{d_G} g_j^q h_0(Y)^j h_1(Y)^{d_F - j}\Big) - \Big(\sum_{i=0}^{d_F} f_i^q h_0(Y)^i h_1(Y)^{d_F - i}\Big)\Big(\sum_{j=0}^{d_G} g_j^q Y^j\Big). \tag{9}$$

Setting (9) to be 0 modulo $Q(Y)$ gives a system of d_Q equations over $\mathbb{F}_{q^k}$ in the $d_F + d_G + 2$ variables $f_0, \ldots, f_{d_F}, g_0, \ldots, g_{d_G}$. By choosing a basis for $\mathbb{F}_{q^k}$ over $\mathbb{F}_q$ and expressing each of the $d_F + d_G + 2$ variables $f_0, \ldots, f_{d_F}, g_0, \ldots, g_{d_G}$ in this basis, this system becomes a bilinear quadratic system† of kd_Q equations in $(d_F + d_G + 2)k$ variables. To find solutions to this system, one can specialise $(d_F + d_G + 2 - d_Q)k$ of the variables in order to make the resulting system generically zero-dimensional while keeping its bilinearity, and then compute the corresponding Gröbner basis, which may have no solution, or a small number of solutions. For each solution, one checks whether (9) divided by $Q(Y)$ is $(d_Q - 1)$-smooth: if so then Q has successfully been rewritten as a product of elements of smaller degree; if no solutions give a $(d_Q - 1)$-smooth cofactor, then one begins again with another specialisation.

The degree of the cofactor of $Q(Y)$ is upper bounded by $d_F(1 + d_h) - d_Q$, so assuming that it behaves as a uniformly chosen polynomial of such a degree one can calculate the probability ρ that it is $(d_Q - 1)$-smooth using standard combinatorial techniques.

Generally, in order for Q to be eliminable by this method with good probability, the number of solutions to the initial bilinear system must be greater than $1/\rho$. To estimate the number of solutions, consider the action of $\mathrm{Gl}_2(\mathbb{F}_{q^k})$

† The bilinearity makes finding solutions to this system easier [45], and is essential for the complexity analysis in [33] and its variant in [20].

on the set of pairs (F, G). The subgroups $\mathrm{Gl}_2(\mathbb{F}_q)$ and $\mathbb{F}_{q^k}^\times$ (via diagonal embedding) both act trivially on the set of relations, modulo multiplication by elements in $\mathbb{F}_{q^k}^\times$. Assuming that the set of (F, G) quotiented out by the action of the compositum of these subgroups (which has cardinality $\approx q^{k+3}$), generates distinct relations, one must satisfy the condition

$$q^{(d_F+d_G+1-d_Q)k-3} > 1/\rho \ . \tag{10}$$

Note that while (10) is preferable for an easy descent, one may yet violate it and still successfully eliminate elements by using various tactics, as demonstrated in §5.

4 Concrete Security Analysis of $\mathbb{F}_{2^{4\cdot 1223}}$

In this section we focus on the DLP in the 1221-bit prime order r_1 subgroup of $\mathbb{F}_{2^{4\cdot 1223}}^\times$, which arises from the MOV attack applied to the genus one supersingular curve (3). By embedding $\mathbb{F}_{2^{4\cdot 1223}}$ into its degree two extension $\mathbb{F}_{2^{8\cdot 1223}} = \mathbb{F}_{2^{9784}}$ we show that, after a precomputation taking approximately 2^{40} s, individual discrete logarithms can be computed in less than 2^{34} s.

4.1 Setup

We consider the field $\mathbb{F}_{2^{8\cdot 1223}} = \mathbb{F}_{q^n}$ with $q = 2^8$ and $n = 1223$ given by the irreducible factor of degree n of $h_1(X^q)X - h_0(X^q)$, with

$$h_0 = X^5 + tX^4 + tX^3 + X^2 + tX + t \ , \quad h_1 = X^5 + X^4 + X^3 + X^2 + X + t \ ,$$

where t is an element of $\mathbb{F}_{2^2} \setminus \mathbb{F}_2$. Note that the field of definition of this representation is $\mathbb{F}_{2^2}$.

Since the target element is contained in the subfield $\mathbb{F}_{2^{4\cdot 1223}}$, we begin the classical descent over $\mathbb{F}_{2^4}$, we switch to $\mathbb{F}_q = \mathbb{F}_{2^8}$, i.e., $k = 1$, for the Gröbner basis descent, and, as explained below, we work over $\mathbb{F}_{q^k}$ with either $k = 1$ or a few $k > 1$ to obtain the logarithms of all factor base elements.

4.2 Linear Algebra Cost Estimate

In this precomputation we obtain the logarithms of all elements of degree at most four over $\mathbb{F}_q$. Since the degree 1223 extension is defined over $\mathbb{F}_{2^2}$ in our field representation, by the action of the Galois group $\mathrm{Gal}(\mathbb{F}_q/\mathbb{F}_{2^2})$ on the factor base, the number of irreducible elements of degree j whose logarithms are to be computed can be reduced to about $2^{8j}/(4j)$ for $j \in \{1, 2, 3, 4\}$.

One way to obtain the logarithms of these elements is to carry out the degree 1 relation generation method from §3.1, together with the elementary observation that an irreducible polynomial of degree k over $\mathbb{F}_q$ splits completely over $\mathbb{F}_{q^k}$. First, computing degree one logarithms over $\mathbb{F}_{q^3}$ gives the logarithms of irreducible elements of degrees one and three over $\mathbb{F}_q$. Similarly, computing

degree one logarithms over $\mathbb{F}_{q^4}$ gives the logarithms of irreducible elements of degrees one, two, and four over $\mathbb{F}_q$. The main computational cost consists in solving the latter system arising from $\mathbb{F}_{q^4}$, which has size 2^{28} and an average row weight of 256.

However, we propose to reduce the cost of finding these logarithms by using $k = 1$ only, in the following easy way. Consider §3.3, and observe that for each polynomial pair (F, G) of degree at most d, one obtains a relation between elements of degree at most d when the numerator of the r.h.s. is d-smooth (ignoring factors of h_1). Note that we are not setting the r.h.s. numerator to be zero modulo Q or computing any Gröbner bases. Up to the action of $\mathrm{Gl}_2(\mathbb{F}_q)$ (which gives equivalent relations) there are about q^{2d-2} such polynomial pairs. Hence, for $d \geq 3$ there are more relations than elements if the smoothness probability of the r.h.s. is sufficiently high. Notice that $k = 1$ implies that the r.h.s. is divisible by $h_1(Y)Y - h_0(Y)$, thus increasing its smoothness probability and resulting in enough relations for $d = 3$ and for $d = 4$. After having solved the much smaller system for $d = 3$ we know the logarithms of all elements up to degree three, so that the average row weight for the system for $d = 4$ can be reduced to about $\frac{1}{4} \cdot 256 = 64$ (irreducible degree four polynomials on the l.h.s.). As above the size of this system is 2^{28}.

The cost for generating the linear systems is negligible compared to the linear algebra cost. For estimating the latter cost we consider Lanczos' algorithm to solve a sparse $N \times N$, $N = 2^{28}$, linear system with average row weight $W = 64$. As noted in [41, 20] this algorithm can be implemented such that

$$N^2 \left(2\,W\,\mathrm{ADD} + 2\,\mathrm{SQR} + 3\,\mathrm{MULMOD}\right) \tag{11}$$

operations are used. On our benchmark system, an AMD Opteron 6168 processor at 1.9 GHz, using [29] our implementation of these operations took 62 ns, 467 ns and 1853 ns for an ADD, a SQR and a MULMOD, respectively, resulting in a linear algebra cost of 2^{40} s.

As in [2], the above estimate ignores communication costs and other possible slowdowns which may arise in practice. An alternative estimate can be obtained by considering a problem of a similar size over $\mathbb{F}_2$ and extrapolating from [38]. This gives an estimated time of 2^{42} s, or for newer hardware slightly less. Note that this computation was carried out using the block Wiedemann algorithm [13], which we recommend in practice because it allows one to distribute the main part of the computation. For the sake of a fair comparison with [2] we use the former estimate of 2^{40} s.

4.3 Descent Cost Estimate

We assume that the logarithms of elements up to degree four are known, and that computing these logarithms with a lookup table is free.

Small Degree Descent. We have implemented the small degree descent of §3.3 in Magma [10] V2.20-1, using Faugere's F4 algorithm [15]. For each

degree from 5 to 15, on the same AMD Opteron 6168 processor we timed the Gröbner basis computation between 10^6 and 100 times, depending on the degree. Then using a bottom-up recursive strategy we estimated the following average running times in seconds for a full logarithm computation, which we present to two significant figures:

$$C[5,\ldots,15] = [\,0.038\,,\,2.1\,,\,2.1\,,\,93\,,\,95\,,\,180\,,\,190\,,\,3200\,,\,3500\,,\,6300\,,\,11000\,]\,.$$

Degree-Balanced Classical Descent. From now on, we make the conservative assumption that a degree n polynomial which is m-smooth, is a product of n/m degree m polynomials. In practice the descent cost will be lower than this, however, the linear algebra cost is dominating, so this issue is inconsequential for our security estimate. The algorithms we used for smoothness testing are detailed in the full version of the paper.

For a classical descent step with degree balancing we consider polynomials $P(X^{2^a}, Y) \in \mathbb{F}_q[X, Y]$ for a suitably chosen integer $0 \le a \le 8$. It is advantageous to choose P such that its degree in one variable is one; let d be the degree in the other variable. In the case $\deg_{X^{2^a}}(P) = 1$, i.e., $P = v_1(Y)X^{2^a} + v_0(Y)$, $\deg v_i \le d$, this gives rise to the relation

$$L_v^{2^a} = \Big(\frac{R_v}{h_1(X)^{2^a}}\Big)^{2^8} \quad \text{where} \quad \begin{array}{l} L_v = \tilde{v}_1(X^{2^{8-a}})X + \tilde{v}_0(X^{2^{8-a}})\,, \\ R_v = v_1(X)h_0(X)^{2^a} + v_0(X)h_1(X)^{2^a} \end{array}$$

in $\mathbb{F}_q[X]/(h_1(X^q)X - h_0(X^q))$ with $\deg L_v \le 2^{8-a}d + 1$, $\deg R_v \le d + 5 \cdot 2^a$, and $\tilde{v}_i$ being v_i with its coefficients powered by 2^{8-a}, for $i = 0, 1$. Similarly, in the case $\deg_Y(P) = 1$, i.e., $P = w_1(X^{2^a})Y + w_0(X^{2^a})$, $\deg w_i \le d$, we have the relation

$$L_w^{2^a} = \Big(\frac{R_w}{h_1(X)^{2^a d}}\Big)^{2^8} \quad \text{where} \quad \begin{array}{l} L_w = \tilde{w}_1(X)X^{2^{8-a}} + \tilde{w}_0(X)\,, \\ R_w = h_1(X)^{2^a d}\big(w_1\big((\frac{h_0(X)}{h_1(X)})^{2^a}\big)X + w_0\big((\frac{h_0(X)}{h_1(X)})^{2^a}\big)\big) \end{array}$$

with $\deg L_w \le d + 2^{8-a}$, $\deg R_w \le 5 \cdot 2^a d + 1$ and again $\tilde{w}_i$ being w_i with its coefficients powered by 2^{8-a}, for $i = 0, 1$.

The polynomials v_i (respectively w_i) are chosen in such a way that either the l.h.s. or the r.h.s. is divisible by a polynomial $Q(X)$ of degree d_Q. Gaussian reduction provides a lattice basis $(u_0, u_1), (u_0', u_1')$ such that the polynomial pairs satisfying the divisibility condition above are given by $ru_i + su_i'$ for $i = 0, 1$, where $r, s \in \mathbb{F}_q[X]$. For nearly all polynomials Q it is possible to choose a lattice basis of polynomials with degree $\approx d_Q/2$ which we will assume for all Q appearing in the analysis; extreme cases can be avoided by look-ahead or backtracking techniques. Notice that a polynomial Q over $\mathbb{F}_{2^4} \subset \mathbb{F}_q$ can be rewritten as a product of polynomials which are also over $\mathbb{F}_{2^4}$, by choosing the basis as well as r and s to be over $\mathbb{F}_{2^4}$. This will be done in all steps of the classical descent. The polynomials r and s are chosen to be of degree four, resulting in 2^{36} possible pairs (multiplying both by a common non-zero constant gives the same relation).

In the final step of the classical eliminations (from degree 26 to 15) we relax the criterion that the l.h.s. and r.h.s. are 15-smooth, allowing also irreducibles of even degree up to degree 30, since these can each be split over $\mathbb{F}_q$ into two polynomials of half the degree, thereby increasing the smoothness probabilities. Admittedly, if we follow our worst-case analysis stipulation that all polynomials at this step have degree 26, then one could immediately split each of them into two degree 13 polynomials. However, in practice one will encounter polynomials of all degrees ≤ 26 and we therefore carry out the analysis without using the splitting shortcut, which will still provide an overestimate of the cost of this step.

In the following we will state the logarithmic cost (in seconds) of a classical descent step as $c_l + c_r + c_s$, where 2^{c_l} and 2^{c_r} denote the number of trials to get the left hand side and the right hand side m-smooth, and 2^{c_s} s is the time required for the corresponding smoothness test. Our smoothness tests were benchmarked on the AMD Opteron 6168 processor.

- $\mathbf{d_Q = 26}$ **to** $\mathbf{m = 15}$**:** We choose $\deg_{X^{2^a}} P = 1$, $a = 5$, Q on the right, giving $d = 17$ and $(\deg(L_v), \deg(R_v)) = (137, 151)$. On average the smoothness test $S_{2^8}(137, 30)$ takes 1.9 ms, giving a logarithmic cost of $13.4 + 15.6 - 9.0$, hence $2^{20.0}$ s. The expected number of factors is 19.2, so the subsequent cost will be less than $2^{17.7}$ s. Note that, as explained above, we use the splitting shortcut for irreducibles of even degree up to 30, resulting in the higher than expected smoothness probabilities.
- $\mathbf{d_Q = 36}$ **to** $\mathbf{m = 26}$**:** We choose $\deg_{X^{2^a}} P = 1$, $a = 5$, Q on the right, giving $d = 22$ and $(\deg(L_v), \deg(R_v)) = (177, 146)$. On average the smoothness test $S_{2^8}(146, 26)$ takes 1.9 ms, giving a logarithmic cost $18.7 + 13.6 - 9.0$, hence $2^{23.3}$ s. The expected number of factors is 12.4, so the subsequent cost will be less than $2^{23.9}$ s.
- $\mathbf{d_Q = 94}$ **to** $\mathbf{m = 36}$**:** We choose $\deg_Y P = 1$, $a = 0$, Q on the left, giving $d = 51$ and $(\deg(L_w), \deg(R_w)) = (213, 256)$. On average the smoothness test $S_{2^8}(213, 36)$ takes 5.1 ms, giving a logarithmic cost $15.0 + 20.3 - 7.5$, hence $2^{27.8}$ s. The expected number of factors is 13.0, so the subsequent cost will be less than $2^{28.4}$ s.

Continued Fraction Descent. For the continued fraction descent we multiply the target element by random powers of the generator and express the product as a ratio of two polynomials of degree at most 611. For each such expression we test if both the numerator and the denominator are 94-smooth. On average the smoothness test $S_{2^8}(611, 94)$ takes 94 ms, giving a logarithmic cost of $17.7+17.7-3.4$, hence $2^{32.0}$ s. The expected number of degree 94 factors on both sides will be 13, so the subsequent cost will be less than $2^{32.8}$ s.

Total Descent Cost. The cost for computing an individual logarithm is therefore upper-bounded by $2^{32.0}$ s $+ 2^{32.8}$ s $< 2^{34}$ s.

4.4 Summary

The main cost in our analysis is the linear algebra computation which takes about 2^{40} s, with the individual logarithm stage being considerably faster. In order to compare with the estimate in [2], we write the main cost in terms of M_r which gives 2^{59} M_r, and thus an improvement by a factor of 2^{69}. Nevertheless, solving a system of cardinality 2^{28} is still a formidable challenge, but perhaps not so much for a well-funded adversary. For completeness we note that if one wants to avoid a linear algebra step of this size, then one can work over different fields, e.g., with $q = 2^{10}$ and $k = 2$, or $q = 2^{12}$ and $k = 1$. However, while this allows a partitioning of the linear algebra into smaller steps as described in §3.2 but at a slightly higher cost, the resulting descent cost is expected to be significantly higher.

5 Solving the DLP in $\mathbb{F}_{2^{12\cdot 367}}$

In this section we present the details of our solution of a DLP in the 698-bit prime order r_2 subgroup of $\mathbb{F}^\times_{2^{12\cdot 367}} = \mathbb{F}^\times_{2^{4404}}$, which arises from the MOV attack applied to the Jacobian of the genus two supersingular curve (4). Note that the prime order elliptic curve $E_1/\mathbb{F}_{2^{367}} : Y^2+Y = X^3+X+1$ with embedding degree 4 also embeds into $\mathbb{F}_{2^{4404}}$, so that logarithms on this curve could have easily been computed as well.

5.1 Setup

To compute the target logarithm, as stated in §1 we applied a principle of parsimony, namely, we tried to solve all intermediate logarithms in $\mathbb{F}_{2^{12\cdot 367}}$, considered as a degree 367 extension of $\mathbb{F}_{2^{12}}$, and only when this was not possible did we embed elements into the extension field $\mathbb{F}_{2^{24\cdot 367}}$ (by extending the base field to $\mathbb{F}_{2^{24}}$) and solve them there.

All of the classical descent down to degree 8 was carried out over $\mathbb{F}_{2^{12\cdot 367}}$, which we formed as the compositum of the following two extension fields. We defined $\mathbb{F}_{2^{12}}$ using the irreducible polynomial $U^{12}+U^3+1$ over $\mathbb{F}_2$, and defined $\mathbb{F}_{2^{367}}$ over $\mathbb{F}_2$ using the degree 367 irreducible factor of $h_1(X^{64})X - h_0(X^{64})$, where $h_1 = X^5 + X^3 + X + 1$, and $h_0 = X^6 + X^4 + X^2 + X + 1$. Let u and x be roots of the extension defining polynomials in U and X respectively, and let $c = (2^{4404} - 1)/r_2$. Then $g = x + u^7$ is a generator of $\mathbb{F}^\times_{2^{4404}}$ and $\bar{g} = g^c$ is a generator of the subgroup of order r_2. As usual, our target element was chosen to be $\bar{x}_\pi = x_\pi^c$ where

$$x_\pi = \sum_{i=0}^{4403} (\lfloor \pi \cdot 2^{i+1} \rfloor \bmod 2) \cdot u^{11-(i \bmod 12)} \cdot x^{\lfloor i/12 \rfloor}.$$

The remaining logarithms were computed using a combination of tactics, over $\mathbb{F}_{2^{12}}$ when possible, and over $\mathbb{F}_{2^{24}}$ when not. These fields were constructed

as degree 2 and 4 extensions of $\mathbb{F}_{2^6}$, respectively. To define $\mathbb{F}_{2^6}$ we used the irreducible polynomial $T^6 + T + 1$. We then defined $\mathbb{F}_{2^{12}}$ using the irreducible polynomial $V^2 + tV + 1$ over $\mathbb{F}_{2^6}$, and $\mathbb{F}_{2^{24}}$ using the irreducible polynomial $W^4 + W^3 + W^2 + t^3$ over $\mathbb{F}_{2^6}$.

5.2 Degree 1 Logarithms

It was not possible to find enough relations for degree 1 elements over $\mathbb{F}_{2^{12}}$, so in accordance with our stated principle, we extended the base field to $\mathbb{F}_{2^{24}}$ to compute the logarithms of all 2^{24} degree 1 elements. We used the polynomial time relation generation from §3.1, which took 47 hours. This relative sluggishness was due to the r.h.s. having degree $d_h + 1 = 7$, which must split over $\mathbb{F}_{2^{24}}$. However, this was faster by a factor of 24 than it would have been otherwise, thanks to h_0 and h_1 being defined over $\mathbb{F}_2$. This allowed us to use the technique from [37] to reduce the size of the factor base via the automorphism $(x+a) \mapsto (x+a)^{2^{367}}$, which fixes x but has order 24 on all non-subfield elements of $\mathbb{F}_{2^{24}}$, since $367 \equiv 7 \bmod 24$ and $\gcd(7, 24) = 1$. This reduced the factor base size to 699252 elements, which was solved in 4896 core hours on a 24 core cluster using Lanczos' algorithm, approximately 24^2 times faster than if we had not used the automorphisms.

5.3 Individual Logarithm

We performed the standard continued fraction initial split followed by degree-balanced classical descent as in §4.3, using Magma [10] and NTL [44], to reduce the target element to an 8-smooth product in 641 and 38224 core hours respectively. The most interesting part of the descent was the elimination of the elements of degree up to 8 over $\mathbb{F}_{2^{12}}$ into elements of degree one over $\mathbb{F}_{2^{24}}$, which we detail below. This phase was completed using Magma and took a further 8432 core hours. However, we think that the combined time of the classical and non-classical parts could be reduced significantly via a backwards-induction analysis of the elimination times of each degree.

Small Degree Elimination. As stated above we used several tactics to achieve these eliminations. The first was the splitting of an element of even degree over $\mathbb{F}_{2^{12}}$ into two elements of half the degree (which had the same logarithm modulo r_2) over the larger field. This automatically provided the logarithms of all degree 2 elements over $\mathbb{F}_{2^{12}}$. Similarly elements of degree 4 and 8 over $\mathbb{F}_{2^{12}}$ were rewritten as elements of degree 2 and 4 over $\mathbb{F}_{2^{24}}$, while we found that degree 6 elements were eliminable more efficiently by initially continuing the descent over $\mathbb{F}_{2^{12}}$, as with degree 5 and 7 elements.

The second tactic was the application of Joux's Gröbner basis elimination method from §3.3 to elements over $\mathbb{F}_{2^{12}}$, as well as elements over $\mathbb{F}_{2^{24}}$. However, in many cases condition (10) was violated, in which case we had to

employ various recursive strategies in order to eliminate elements. In particular, elements of the same degree were allowed on the r.h.s. of relations, and we then attempted to eliminate these using the same (recursive) strategy. For degree 3 elements over $\mathbb{F}_{2^{12}}$, we even allowed degree 4 elements to feature on the r.h.s. of relations, since these were eliminable via the factorisation into degree 2 elements over $\mathbb{F}_{2^{24}}$.

In Figure 1 we provide a flow chart for the elimination of elements of degree up to 8 over $\mathbb{F}_{2^{12}}$, and for the supporting elimination of elements of degree up to 4 over $\mathbb{F}_{2^{24}}$. Nearly all of the arrows in Figure 1 were necessary for these field parameters (the exceptions being that for degrees 4 and 8 over $\mathbb{F}_{2^{12}}$ we could have initially continued the descent along the bottom row, but this would have been slower). The reason this 'non-linear' descent arises is due to q being so small, and d_H being relatively large, which increases the degree of the r.h.s. cofactors, thus decreasing the smoothness probability. Indeed these tactics were only borderline applicable for these parameters; if h_0 or h_1 had degree any larger than 6 then not only would most of the descent have been much harder, but it seems that one would be forced to compute the logarithms of degree 2 elements over $\mathbb{F}_{2^{24}}$ using Joux's linear system method from §3.2, greatly increasing the required number of core hours. As it was, we were able to eliminate degree 2 elements over $\mathbb{F}_{2^{24}}$ on the fly, as we describe explicitly below.

Finally, we note that our descent strategy is considerably faster than the alternative of embedding the DLP into $\mathbb{F}_{2^{24\cdot 367}}$ and performing a full descent in this field, even with the elimination on the fly of degree 2 elements over $\mathbb{F}_{2^{24}}$, since much of the resulting computation would constitute superfluous effort for the task in hand.

Degree 2 Elimination over $\mathbb{F}_{2^{24}}$. Let $Q(Y)$ be a degree two element which is to be eliminated, i.e., written as a product of degree one elements. As in [19, 20] we first precompute the set of 64 elements $B \in \mathbb{F}_{2^{24}}$ such that the polynomial $f_B(X) = X^{65} + BX + B$ splits completely over $\mathbb{F}_{2^{24}}$ (in fact these B's happen to be in $\mathbb{F}_{2^{12}}$, but this is not relevant to the method). We then find a Gaussian-reduced basis of the lattice $L_{Q(Y)}$ defined by

$$L_{Q(Y)} = \{(w_0(Y), w_1(Y)) \in \mathbb{F}_{2^{24}}[Y]^2 \colon w_0(Y)h_0(Y) + w_1(Y)h_1(Y) \equiv 0 \pmod{Q(Y)}\}.$$

Such a basis has the form $(u_0, Y + u_1), (Y + v_0, v_1)$, with $u_i, v_i \in \mathbb{F}_{2^{24}}$, except in rare cases, see Remark 1. For $s \in \mathbb{F}_{2^{24}}$ we obtain lattice elements $(w_0(Y), w_1(Y)) = (Y + v_0 + su_0, sY + v_1 + su_1)$.

Using the transformation detailed in §3, for each $B \in \mathbb{F}_{2^{24}}$ such that f_B splits completely over $\mathbb{F}_{2^{24}}$ we perform a Gröbner basis computation to find the set of $s \in \mathbb{F}_{2^{24}}$ that satisfy

$$B = \frac{(s^{64} + u_0 s + v_0)^{65}}{(u_0 s^2 + (u_1 + v_0)s + v_1)^{64}} ,$$

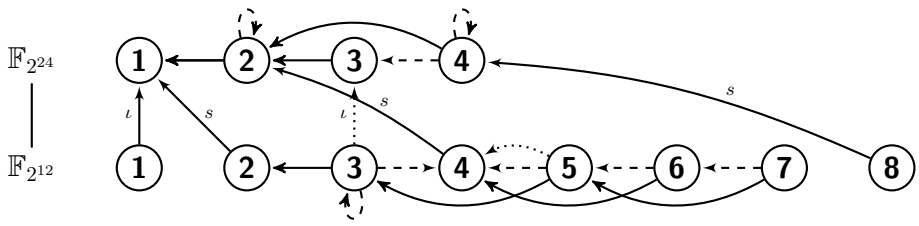

Fig. 1. This diagram depicts the set of strategies employed to eliminate elements over $\mathbb{F}_{2^{12}}$ of degree up to 8. The encircled numbers represent the degrees of elements over $\mathbb{F}_{2^{12}}$ on the bottom row, and over $\mathbb{F}_{2^{24}}$ on the top row. The arrows indicate how an element of a given degree is rewritten as a product of elements of other degrees, possibly over the larger field. Unadorned solid arrows indicate the maximum degree of elements obtained on the l.h.s. of the Gröbner basis elimination method; likewise dashed arrows indicate the degrees of elements obtained on the r.h.s. of the Gröbner basis elimination method, when these are greater than those obtained on the l.h.s. Dotted arrows indicate a fall-back strategy when the initial strategy fails. An s indicates that the element is to be split over the larger field into two elements of half the degree. An ι indicates that an element is promoted to the larger field. Finally, a loop indicates that one must use a recursive strategy in which further instances of the elimination in question must be solved in order to eliminate the element in question.

by first expressing s in a $\mathbb{F}_{2^{24}}/\mathbb{F}_{2^6}$ basis, which results in a quadratic system in 4 variables. This ensures that the l.h.s. splits completely over $\mathbb{F}_{2^{24}}$. For each such s we check whether the r.h.s. cofactor of $Q(Y)$, which has degree 5, is 1-smooth. If this occurs, we have successfully eliminated $Q(Y)$.

However, one expects on average just one s per B, and so the probability of $Q(Y)$ being eliminated in this way is $1-(1-1/5!)^{64} \approx 0.415$, which was borne out in practice to two decimal places. Hence, we adopted a recursive strategy in which we stored all of the r.h.s. cofactors whose factorisation degrees had the form $(1,1,1,2)$ (denoted type 1), or $(1,2,2)$ (denoted type 2). Then for each type 1 cofactor we checked to see if the degree 2 factor was eliminable by the above method. If none were eliminable we stored every type 1 cofactor of each degree 2 irreducible occurring in the list of type 1 cofactors of $Q(Y)$. If none of these were eliminable (which occurred with probability just 0.003), then we reverted to the type 2 cofactors, and adopted the same strategy just specified for each of the degree 2 irreducible factors. Overall, we expected our strategy to fail about once in every $6 \cdot 10^6$ such $Q(Y)$. This happened just once during our descent, and so we multiplied this $Q(Y)$ by a random linear polynomial over $\mathbb{F}_{2^{24}}$ and performed a degree 3 elimination, which necessitates an estimated 32 degree 2 polynomials being simultaneously eliminable by the above method, which thanks to the high probability of elimination, will very likely be successful for any linear multiplier.

5.4 Summary

Finally, after a total of approximately 52240 core hours (or 2^{48} M_{r_2}), we found that $\bar{x}_\pi = \bar{g}^{\log}$, with (see [25] for a Magma verification script) log =

$$
\begin{aligned}
&4093208920214235164093447733900702563725614097945142354192285387447360 4\\
&3901535168472140823368768956390251106223098014527287101738254282676469 5\\
&5984311476789554547579576647584875422721159476118231281401707689324 2\,.
\end{aligned}
$$

Remark 1. During the descent, we encountered several polynomials $Q(Y)$ that were apparently not eliminable via the Gröbner basis method. We discovered that they were all factors of $h_1(Y) \cdot c + h_0(Y)$ for $c \in \mathbb{F}_{2^{12}}$ or $\mathbb{F}_{2^{24}}$, and hence $h_0(Y)/h_1(Y) \equiv c \pmod{Q(Y)}$. This implies that (9) is equal to $F(c)G^{(q)}(Y) + F^{(q)}(Y)G(c)$ modulo $Q(Y)$, where $G^{(q)}$ denotes the Frobenius twisted G and similarly for $F^{(q)}$. This cannot become 0 modulo $Q(Y)$ if the degrees of F and G are smaller than the degree of Q, unless F and G are both constants. However, thanks to the field representation, finding the logarithm of these $Q(Y)$ turns out to be easy. In particular, if $h_1(Y) \cdot c + h_0(Y) = Q(Y) \cdot R(Y)$ then $Q(Y) = h_1(Y)((h_0/h_1)(Y) + c)/R(Y) = h_1(Y)(X + c)/R(Y)$, and thus modulo r_2 we have $\log(Q(y)) \equiv \log(x + c) - \log(R(y))$, since $\log(h_1(y)) \equiv 0$. Since $(x + c)$ is in the factor base, if we are able to compute the logarithm of $R(y)$, then we are done. In all the cases we encountered, the cofactor $R(y)$ was solvable by the above methods.

6 Conclusion

We have introduced a new field representation and efficient descent principles which together make the recent DLP advances far more practical. As example demonstrations, we have applied these techniques to two binary fields of central interest to pairing-based cryptography, namely $\mathbb{F}_{2^{4\cdot 1223}}$ and $\mathbb{F}_{2^{12\cdot 367}}$, which arise as the embedding fields of (the Jacobians of) a genus one and a genus two supersingular curve, respectively. When initially proposed, these fields were believed to be 128-bit secure, and even in light of the recent DLP advances, were believed to be 128-bit and 94.6-bit secure. On the contrary, our analysis indicates that the former field has approximately 59 bits of security and we have implemented a total break of the latter.

References

1. Jithra Adikari, M. Anwar Hasan, and Christophe Nègre. Towards faster and greener cryptoprocessor for eta pairing on supersingular elliptic curve over $\mathbb{F}_{2^{1223}}$. In *Selected Areas in Cryptography—SAC 2012*, volume 7707 of *LNCS*, pages 166–183. Springer, 2012.
2. Gora Adj, Alfred Menezes, Thomaz Oliveira, and Francisco Rodríguez-Henríquez. Weakness of $\mathbb{F}_{3^{6\cdot 509}}$ for discrete logarithm cryptography. In *Pairing-based Cryptography—Pairing 2013*, volume 8365 of *LNCS*, pages 20–44. Springer, 2013.

3. Diego F. Aranha, Jean-Luc Beuchat, Jérémie Detrey, and Nicolas Estibals. Optimal eta pairing on supersingular genus-2 binary hyperelliptic curves. In *Topics in Cryptology—CT-RSA 2012*, volume 7178 of *LNCS*, pages 98–115. Springer, 2012.
4. Razvan Barbulescu, Pierrick Gaudry, Antoine Joux, and Emmanuel Thomé. A heuristic quasi-polynomial algorithm for discrete logarithm in finite fields of small characteristic. In *Advances in Cryptology—EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 1–16. Springer, 2014.
5. Paulo S. L. M. Barreto, Steven D. Galbraith, Colm Ó' Héigeartaigh, and Michael Scott. Efficient pairing computation on supersingular abelian varieties. *Des. Codes Cryptography*, 42(3):239–271, March 2007.
6. Paulo S. L. M. Barreto, Hae Yong Kim, Ben Lynn, and Michael Scott. Efficient algorithms for pairing-based cryptosystems. In *Advances in Cryptology—CRYPTO 2002*, volume 2442 of *LNCS*, pages 354–368. Springer, 2002.
7. Jean-Luc Beuchat, Emmanuel López-Trejo, Luis Martínez-Ramos, Shigeo Mitsunari, and Francisco Rodríguez-Henríquez. Multi-core implementation of the Tate pairing over supersingular elliptic curves. In *Cryptology and Network Security—CANS 2009*, volume 5888 of *LNCS*, pages 413–432. Springer, 2009.
8. Antonia W. Bluher. On $x^{q+1} + ax + b$. *Finite Fields and Their Applications*, 10(3):285–305, 2004.
9. Dan Boneh and Matt Franklin. Identity-based encryption from the Weil pairing. In *Advances in Cryptology—CRYPTO 2001*, volume 2139 of *LNCS*, pages 213–229. Springer, 2001.
10. Wieb Bosma, John Cannon, and Catherine Playoust. The Magma algebra system. I. The user language. *J. Symbolic Comput.*, 24(3-4):235–265, 1997.
11. Sanjit Chatterjee, Darrel Hankerson, and Alfred Menezes. On the efficiency and security of pairing-based protocols in the type 1 and type 4 settings. In *Arithmetic of Finite Fields*, volume 6087 of *LNCS*, pages 114–134. Springer, 2010.
12. Don Coppersmith. Fast evaluation of logarithms in fields of characteristic two. *IEEE Transactions on Information Theory*, 30(4):587–593, 1984.
13. Don Coppersmith. Solving homogeneous linear equations over GF(2) via block Wiedemann algorithm. *Mathematics of Computation*, 62(205):333–350, 1994.
14. Iwan Duursma and Hyang-Sook Lee. Tate pairing implementation for hyperelliptic curves $y^2 = x^p - x + d$. In *Advances in Cryptology—ASIACRYPT 2003*, volume 2894 of *LNCS*, pages 111–123. Springer, 2003.
15. Jean-Charles Faugère. A new efficient algorithm for computing Gröbner bases (F_4). *J. Pure Appl. Algebra*, 139(1-3):61–88, 1999.
16. Steven D. Galbraith. Supersingular curves in cryptography. In *Advances in Cryptology—ASIACRYPT 2001*, volume 2248 of *LNCS*, pages 495–513. Springer, 2001.
17. Steven D. Galbraith, Keith Harrison, and David Soldera. Implementing the Tate pairing. In *Algorithmic Number Theory—ANTS-V*, volume 2369 of *LNCS*, pages 324–337. Springer, 2002.
18. Santosh Ghosh, Dipanwita Roychowdhury, and Abhijit Das. High speed cryptoprocessor for η_t pairing on 128-bit secure supersingular elliptic curves over characteristic two fields. In *Cryptographic Hardware and Embedded Systems—CHES 2011*, volume 6917 of *LNCS*, pages 442–458. Springer, 2011.
19. Faruk Göloğlu, Robert Granger, Gary McGuire, and Jens Zumbrägel. On the function field sieve and the impact of higher splitting probabilities: Application to discrete logarithms in $\mathbb{F}_{2^{1971}}$ and $\mathbb{F}_{2^{3164}}$. In *Advances in Cryptology—CRYPTO 2013*, volume 8043 of *LNCS*, pages 109–128. Springer, 2013.
20. Faruk Göloğlu, Robert Granger, Gary McGuire, and Jens Zumbrägel. Solving a 6120-bit DLP on a desktop computer. In *Selected Areas in Cryptography—SAC 2013*, volume 8282 of *LNCS*, pages 136–152. Springer, 2014.
21. Faruk Göloğlu, Robert Granger, Gary McGuire, and Jens Zumbrägel. Discrete Logarithms in $GF(2^{1971})$. NMBRTHRY list, 19/2/2013.
22. Faruk Göloğlu, Robert Granger, Gary McGuire, and Jens Zumbrägel. Discrete Logarithms in $GF(2^{6120})$. NMBRTHRY list, 11/4/2013.

23. Robert Granger. On the function field sieve and the impact of higher splitting probabilities, 2013. Presentation at the 17th Workshop on Elliptic Curve Cryptography, 16 September 2013.
24. Robert Granger. Solving a 6120-bit DLP on a desktop computer, 2013. Presentation at Selected Areas in Cryptography 2013, 15 August 2013.
25. Robert Granger, Thorsten Kleinjung, and Jens Zumbrägel. Discrete logarithms in the Jacobian of a genus 2 supersingular curve over $GF(2^{367})$. NMBRTHRY list, 30/1/2014.
26. Robert Granger, Thorsten Kleinjung, and Jens Zumbrägel. Discrete Logarithms in $GF(2^{9234})$. NMBRTHRY list, 31/1/2014.
27. Robert Granger, Dan Page, and Martijn Stam. Hardware and software normal basis arithmetic for pairing-based cryptography in characteristic three. *IEEE Trans. Computers*, 54(7):852–860, 2005.
28. Robert Granger, Dan Page, and Martijn Stam. On small characteristic algebraic tori in pairing-based cryptography. *LMS J. Comput. Math.*, 9:64–85, 2006.
29. Torbjörn Granlund and the GMP development team. *GNU MP: The GNU Multiple Precision Arithmetic Library*, 5.0.5 edition, 2012. `http://gmplib.org/`.
30. Darrel Hankerson, Alfred Menezes, and Michael Scott. Software implementation of pairings. In *Identity-Based Cryptography, vol. 2*, Cryptology and Information Security, pages 188–206. IOS Press, 2008.
31. Antoine Joux. A one round protocol for tripartite Diffie-Hellman. In *Algorithmic Number Theory—ANTS-IV*, volume 1838 of *Lecture Notes in Comput. Sci.*, pages 385–393. Springer, Berlin, 2000.
32. Antoine Joux. Faster index calculus for the medium prime case. Application to 1175-bit and 1425-bit finite fields. In *Advances in Cryptology—EUROCRYPT 2013*, volume 7881 of *LNCS*, pages 177–193. Springer, 2013.
33. Antoine Joux. A new index calculus algorithm with complexity $L(1/4 + o(1))$ in very small characteristic. In *Selected Areas in Cryptography—SAC 2013*, volume 8282 of *LNCS*, pages 355–379. Springer, 2014.
34. Antoine Joux. Discrete Logarithms in $GF(2^{1778})$. NMBRTHRY list, 11/2/2013.
35. Antoine Joux. Discrete Logarithms in $GF(2^{4080})$. NMBRTHRY list, 22/3/2013.
36. Antoine Joux. Discrete Logarithms in $GF(2^{6168})$. NMBRTHRY list, 21/5/2013.
37. Antoine Joux and Reynald Lercier. The function field sieve in the medium prime case. In *Advances in Cryptology—EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 254–270. Springer, 2006.
38. Thorsten Kleinjung, Kazumaro Aoki, Jens Franke, Arjen K. Lenstra, Emmanuel Thomé, Joppe W. Bos, Pierrick Gaudry, Alexander Kruppa, Peter L. Montgomery, Dag Arne Osvik, Herman J. J. te Riele, Andrey Timofeev, and Paul Zimmermann. Factorization of a 768-bit RSA modulus. In *Advances in Cryptology—CRYPTO 2010*, volume 6223 of *LNCS*, pages 333–350. Springer, 2010.
39. Arjen K. Lenstra. Unbelievable security: Matching AES security using public key systems. In *Advances in Cryptology—ASIACRYPT 2001*, volume 2248 of *LNCS*, pages 67–86. Springer, 2001.
40. Alfred J. Menezes, Tatsuaki Okamoto, and Scott A. Vanstone. Reducing elliptic curve logarithms to logarithms in a finite field. *IEEE Trans. Inform. Theory*, 39(5):1639–1646, 1993.
41. Ilya Popovyan. Efficient parallelization of lanczos type algorithms. Cryptology ePrint Archive, Report 2011/416, 2011. `http://eprint.iacr.org/`.
42. Ryuichi Sakai, Shigeo Mitsunari, and Masao Kasahara. Cryptographic schemes based on pairing over elliptic curve. *IEIC Technical Report*, 101(214):75–80, 2001.
43. Naoyuki Shinohara, Takeshi Shimoyama, Takuya Hayashi, and Tsuyoshi Takagi. Key length estimation of pairing-based cryptosystems using η_t pairing. In *Information Security Practice and Experience*, volume 7232 of *LNCS*, pages 228–244. Springer, 2012.
44. Victor Shoup. *NTL: A library for doing number theory*, 5.5.2 edition, 2009. `http://www.shoup.net/ntl/`.
45. Pierre-Jean Spaenlehauer. Solving multihomogeneous and determinantal systems algorithms - complexity - applications. Ph.D. thesis, Université Pierre et Marie Curie (UPMC), 2012.

On the discrete logarithm problem in finite fields of fixed characteristic

Robert Granger*, Thorsten Kleinjung**, and Jens Zumbrägel***

Laboratory for Cryptologic Algorithms
School of Computer and Communication Sciences
École polytechnique fédérale de Lausanne
Switzerland
{robert.granger,thorsten.kleinjung,jens.zumbragel}@epfl.ch

Abstract. For q a prime power, the discrete logarithm problem (DLP) in $\mathbb{F}_q^\times$ consists in finding, for any $g \in \mathbb{F}_q^\times$ and $h \in \langle g \rangle$, an integer x such that $g^x = h$. For each prime p we exhibit infinitely many extension fields $\mathbb{F}_{p^n}$ for which the DLP in $\mathbb{F}_{p^n}^\times$ can be solved in expected quasi-polynomial time.

1 Introduction

In this paper we prove the following result.

Theorem 1. *For every prime p there exist infinitely many explicit extension fields $\mathbb{F}_{p^n}$ with for which the DLP in $\mathbb{F}_{p^n}^\times$ can be solved in expected quasi-polynomial time*

$$\exp\big((1/\log 2 + o(1))\log^2 n\big). \tag{1}$$

Theorem 1 is an easy corollary of the following much stronger result, which we prove by presenting a randomised algorithm for solving any such DLP.

Theorem 2. *Given a prime power q that is not a power of 4, an integer $k \geq 18$, polynomials $h_0, h_1 \in \mathbb{F}_{q^k}[X]$ of degree at most two and an irreducible degree l factor I of $h_1 X^q - h_0$, the DLP in $\mathbb{F}_{q^{kl}}^\times$ where $\mathbb{F}_{q^{kl}} \cong \mathbb{F}_{q^k}[X]/(I)$ can be solved in expected time*

$$q^{\log_2 l + O(k)}. \tag{2}$$

To deduce Theorem 1 from Theorem 2, note that thanks to Kummer theory, when $l = q-1$ such h_0, h_1 are known to exist; indeed, for all k there exists an $a \in \mathbb{F}_{q^k}$ such that $I := X^{q-1} - a \in \mathbb{F}_{q^k}[X]$ is irreducible and therefore $I \mid X^q - aX$. By setting $q = p^i$ for any $i \geq 1$ (odd for $p = 2$), $k \geq 18$ with $k = o(\log q)$, $l = q - 1 = p^i - 1$ and finally $n = ik(p^i - 1)$, applying (2) proves

* Supported by the Swiss National Science Foundation via grant number 200021-156420.
** Supported by the Swiss National Science Foundation via grant number 200020-132160.
*** The work was partially done while the author was with the Institute of Algebra, TU Dresden, Germany, supported by the Irish Research Council via grant number ELEVATEPD/2013/82.

that the DLP in this representation of $\mathbb{F}_{p^n}^\times$ can be solved in expected time (1). As one can compute an isomorphism between any two representations of $\mathbb{F}_{p^n}^\times$ in polynomial time [41], this completes the proof. Observe that by using the same argument one may also replace the prime p in Theorem 1 by any prime power that is not a power of 4.

In order to apply Theorem 2 to the DLP in $\mathbb{F}_{p^n}^\times$ with p fixed and arbitrary n, one should first embed the DLP into one in an appropriately chosen $\mathbb{F}_{q^{kn}}^\times$. By this we mean that $q = p^i$ should be at least $n-2$ (so that h_0, h_1 may exist) but not too large, and that $18 \leq k = o(\log q)$, so that the resulting complexity (2) is given by (1) as $n \to \infty$. Proving that appropriate $h_0, h_1 \in \mathbb{F}_{q^k}[X]$ exist for such q and k would complete our approach and prove the far stronger result that the DLP in $\mathbb{F}_{p^n}^\times$ can be solved in expected time (1) for all sufficiently large n. However, this seems to be a very hard problem, even if heuristically it would appear to be almost certain. What is striking about Theorem 2 is that in contrast to all finite field DLP algorithms from the past thirty years, it is rigorous, and our algorithm is therefore guaranteed to work once an appropriate field representation is found.

Note that if one could prove the existence of an infinite sequence of primes p (or more generally prime powers) for which $p-1$ is quasi-polynomially smooth in $\log p$, then the Pohlig-Hellman algorithm [45] (discovered independently by Silver) would also give a rigorous – and deterministic – quasi-polynomial time algorithm for solving the DLP in such fields, akin to Theorem 1. However, such a sequence is not known to exist and even if it were, Theorem 1 is arguably more interesting since our algorithm exploits properties of the fields in question rather than just the factorisation of the order of their multiplicative groups. Furthermore, the fields to which our algorithm applies are explicit, whereas it may be very hard to find members of such a sequence of primes (or prime powers), should one exist.

Gauss was probably the first to define discrete logarithms – or indices, as he called them, with respect to a primitive root – noting their usefulness for computing n-th roots modulo primes [50, art. 57–60]. Since he suggested the use of look-up tables for this purpose, the algorithm he used for computing logarithms in the tiny examples to which he applied the technique was almost certainly just tabulation via exponentiation. However, Gauss noted in art. 58 that the table need only consist of indices for primes, implicitly assuming that integers less than the modulus can be factorised efficiently. In the early 1920s Kraitchik developed this observation into what is now called the Index Calculus Method (ICM) [38, 39]; evidently a very natural idea, it was also discovered independently by Cunningham at around the same time, see [53], and rediscovered by Adleman [1], Merkle [43] and Pollard [46] in the late 1970s. In this context the ICM proceeds by first defining a *factor base* consisting of primes up to some *smoothness bound* B. One then searches for multiplicative relations between elements of the factor base; one can do this for instance

by computing random powers of the primitive root g modulo p and storing those which are B-smooth. These relations between factor base elements (and g) each induce a linear equation between their logarithms with respect to g, and once there are sufficiently many relations the logarithms of the factor base elements can be computed via a linear algebra elimination. The second phase of the ICM consists of computing the logarithm of a target element h which is not B-smooth. In this setting one can multiply h by random powers of g until the product is B-smooth, at which point its logarithm is easily determined. Exploiting the distribution of $L_p(1/2)$-smooth integers amongst integers less than p [14, 12, 13] gives a heuristic $L_p(1/2)$ algorithm for the DLP in $\mathbb{F}_p^\times$ [1]; here, as is usual for such algorithms, we use the following measure of subexponentiality:

$$L_p(\alpha, c) = \exp\big((c + o(1))(\log p)^\alpha (\log\log p)^{1-\alpha}\big),$$

where for simplicity we sometimes suppress the subscript, the constant c, or both. The algorithm just described can be made rigorous for both prime fields and fixed characteristic extension fields [47, 18].

In 1984 Coppersmith proposed the first heuristic $L(1/3, c)$ algorithm for fields of the form $\mathbb{F}_{2^n}$ [10, 11] with the constant c being a periodic function of n satisfying $(32/9)^{1/3} < c < 4^{1/3}$. Coppersmith's algorithm exhibits similar periodic behaviour for extensions fields of any fixed characteristic. In 1994 Adleman proposed the Function Field Sieve (FFS) [2] – an analogue of the famous Number Field Sieve [40] – which can also be seen as a generalisation of Coppersmiths algorithm. This was refined by Adleman and Huang in 1999, achieving a heuristic complexity of $L(1/3, (32/9)^{1/3})$ for extension fields of any fixed characteristic [3].

For fixed characteristic extension fields, the main difference between the $L(1/2)$ and $L(1/3)$ algorithms is that during relation generation the former generates elements of degree $\approx n$ and searches for sufficiently many which are $\tilde{O}(n^{1/2})$-smooth (where the $\tilde{O}$ indicates suppressed log factors), whereas algorithms of the latter type generate elements of degree $\tilde{O}(n^{2/3})$ and search for sufficiently many which are $\tilde{O}(n^{1/3})$-smooth. In the former case the elements can be generated uniformly and so one can apply smoothness results to obtain a rigorous algorithm. Crucially, for the $L(1/3)$ algorithms the elements generated are not uniformly distributed amongst elements of that degree and hence the complexity analysis is only heuristic. A second difference is that during the individual logarithm phase of the $L(1/3)$ algorithms one needs to recursively express a target element as a product of irreducible elements of lower degrees – with one iteration of this process being known as an *elimination* of that element – which produces a tree with the target element at its root and the elements produced by this process at its nodes. After sufficiently many iterations the elements at the leaves of this tree will be contained entirely in the factor base and so the logarithm of the target element can easily be computed

via backtracking. Since this process descends through elements of lower and lower degree, the individual logarithm phase is also known as the *descent*.

In order to obtain algorithms of better complexity – at least for the first phase of the ICM – there are two natural directions that one could explore: firstly, one could attempt to generate relations between elements of lower degree, which heuristically would have a higher probability of being smooth; or secondly, one could attempt to generate relations which have better than expected smoothness properties (or possibly a combination of both). The second idea is perhaps far less obvious and more nuanced than the first; indeed until recently it does not seem to have been appreciated that it was even a possibility, most likely because from an algorithm analysis perspective it is desirable that the expected smoothness properties hold. For nearly three decades there was no progress in either direction; the only development in fixed characteristic being a practical improvement [34], while for so-called medium characteristic fields – those for which the base field cardinality satisfies $q = L_{q^n}(1/3)$ – a slight reduction in the constant was achieved, to $c = 3^{1/3} \approx 1.44$ [35] and to $c = 2^{1/3} \approx 1.26$ [28], the latter using a clever method to amplify one relation into many others. Note that we mention the medium characteristic developments because they can be applied to fixed characteristic extensions for appropriate extension degrees. Given the immense importance of the DLP to public key cryptography ever since its inception in 1976 [17], this plateau in progress could have been taken as strong evidence of the problem's hardness. However, in 2013 a series of algorithmic breakthroughs occurred which demonstrated that for fixed characteristic fields the DLP is, at least heuristically, far easier than originally believed.

In particular, in February 2013, Göloğlu, Granger, McGuire and Zumbrägel showed that for binary (and more generally fixed characteristic) fields of a certain form, relation generation for degree one elements runs in heuristic *polynomial time*, as does computing the logarithms of degree two elements using a technique which eliminates them on the fly, i.e., individually and quickly [19, 20], which was previously the bottleneck in the descent when using the standard techniques. This was the first example of the second idea alluded to above as it demonstrated how to generate relations which are 1-smooth for arbitrarily large degree, completely contradicting the usual smoothness heuristics. However, the efficient elimination of higher degree elements remained an unresolved problem. For fields of essentially the same form Joux independently gave: a degree one relation generation method which is isomorphic to that of Göloğlu *et al.*; a very different degree two elimination method; and a new small degree element elimination method which resulted in an algorithm with heuristic complexity $L(1/4 + o(1))$ [30, 29]. Combinations and variations of these techniques led to several large scale DLP computations and records [31, 22, 32, 23, 33, 25, 26, 21, 24], the largest of which at the time of writing was in the field $\mathbb{F}_{2^{9234}}$.

Then in June 2013, for fields of the same form and of bitlength λ, Barbulescu, Gaudry, Joux and Thomé announced a heuristic *quasi-polynomial time* algorithm (referred to hereafter as the original QPA) for solving the DLP [5], which has complexity

$$\lambda^{O(\log \lambda)}. \tag{3}$$

Since (3) is smaller than $L(\alpha)$ for any $\alpha > 0$, it is asymptotically the most efficient algorithm known for solving the DLP in finite fields of fixed characteristic. It also results in an immediate $L(\alpha+o(1))$ algorithm when $q = L_{q^n}(\alpha)$ for $0 \leq \alpha < 1/3$. The principal idea behind the elimination steps of the original QPA may be viewed as a generalisation of Joux's degree two elimination method [29], which finds the logarithms of all translates of a degree two element simultaneously via the collection of suitable relations and a subsequent linear algebra elimination.

The principal idea behind our new QPA may be viewed as a generalisation of the degree two elimination method of [20]. In particular, let h be an element of degree $2d$ that we wish to eliminate, which we assume is irreducible when considered as an element of the polynomial ring. By taking a degree d extension of the base field, h factors into a product of d irreducible quadratics. Applying the degree two elimination method of [20] to any one of these quadratics enables one to rewrite the quadratic as a product of linear elements over the degree d extension of the base field. To return to the original base field one simply applies the relevant norm, which takes the linear elements to powers of irreducible elements of order dividing d and the quadratic element back to the original element h which was to be eliminated, thus completing its elimination. If the target element has degree a power of two then this elimination can be applied recursively, halving the degree (or more) of the elements in the descent tree upon each iteration. Central to our proof of Theorem 2 is our demonstration that this recursive step can always be carried out successfully. For the purpose of building a full DLP algorithm which may be applied to any target element, one can use a Dirichlet-type theorem due to Wan [52, Thm. 5.1] to ensure that any field element is equivalent to an irreducible of degree a power of two only slightly larger than the extension degree of the field in question.

A remarkable property of the above descent method is that it does not require any smoothness assumptions about non-uniformly distributed polynomials, in contrast to all previous index calculus algorithms, including the original QPA. So while the polynomial time relation generation techniques of [20, 29] in a sense *resisted* smoothness heuristics, our new descent method *completely eliminates* them. We emphasise that our new QPA is radically different from the original QPA of Barbulescu *et al.*, while it is its very algebraic nature that makes our rigorous analysis possible. Given the essential use of smoothness heuristics in the original QPA, as well as one other heuristic, it

seems unlikely that it can be made rigorous, even if the existence of appropriate field representations are assumed or proven. Furthermore, while not of central interest to the results of the present paper, we remark that our elimination steps are extremely practical, even for relatively small fields [44, 37], whereas the bitlengths for which the original QPA becomes effective have yet to be determined.

The sequel is organised as follows. In Section 2 we describe our algorithm and explain why the steps are sufficient for our purpose. We then give a brief review of the FFS in Section 3 and fix some notation. In Section 4 we provide details of the building block behind our new descent and explain why its successful application implies Theorem 2, and hence Theorem 1. In Section 5 we complete the proof of these theorems by demonstrating that the descent step is indeed always successful. We conclude in Section 6.

2 The algorithm

As per Theorem 2, let q be a prime power that is not a power of 4 and let $k \geq 18$ be an integer; the reasons for these bounds are explained in Sections 4 and 5. We also assume there exist $h_0, h_1, I \in \mathbb{F}_{q^k}[X]$ with $\deg(h_0), \deg(h_1) \leq 2$ and I a degree l irreducible factor of $h_1 X^q - h_0$. Finally, let $g \in \mathbb{F}_{q^{kl}}^{\times}$ and let $h \in \langle g \rangle$ be the target element for the DLP to base g.

We now present our algorithm, which differs slightly from the traditional ICM as described in Section 1 in that it does not first compute the logarithms of the factor base elements and then apply a descent strategy. Instead, one computes many descents for elements of the form $g^\alpha h^\beta$ (just one more than the number of factor base elements suffices) and then applies a linear algebra elimination. This approach and its analysis was first used by Enge and Gaudry [18], however the algorithm and argument we present follows very closely those used by Diem in the context of the elliptic curve DLP [16]. A small but important difference between our algorithm and Diem's is that we cannot assume that we know the factorisation of the order of the relevant group, since the fastest proven factorisation algorithms have complexity $L(1/2)$ [47, 51, 42] and are therefore insufficient for our purpose.

Input: A prime power q; an integer $k \geq 18$; a positive integer l; polynomials $h_0, h_1, I \in \mathbb{F}_{q^k}[X]$ with $\deg(h_0), \deg(h_1) \leq 2$ and I a degree l irreducible factor of $h_1 X^q - h_0$; $g \in \mathbb{F}_{q^{kl}}^{\times}$ and $h \in \langle g \rangle$.

Output: An integer x such that $g^x = h$.

1. Let $N = q^{kl} - 1$, let $\mathcal{F} = \{F \in \mathbb{F}_{q^k}[X] \mid \deg F \leq 1,\ F \neq 0\} \cup \{h_1\}$ and denote its elements by $F_1, \ldots, F_m$, where $m = |\mathcal{F}| = q^{2k}$.
2. Construct a matrix $R = (r_{i,j}) \in (\mathbb{Z}/N\mathbb{Z})^{(m+1)\times m}$ and column vectors $\alpha, \beta \in (\mathbb{Z}/N\mathbb{Z})^{m+1}$ as follows. For each i with $1 \leq i \leq m+1$ choose $\alpha_i, \beta_i \in \mathbb{Z}/N\mathbb{Z}$

uniformly and independently at random and apply the (randomised) descent algorithm of Section 4 to $g^{\alpha_i}h^{\beta_i}$ to express this as

$$g^{\alpha_i}h^{\beta_i} = \prod_{j=1}^{m} F_j^{r_{i,j}}.$$

3. Compute a lower row echelon form R' of R by using invertible row transformations; apply these row transformations also to α and β, and denote the results by α' and β'.
4. If $\gcd(\beta'_1, N) > 1$, go to Step 2.
5. Return an integer x such that $\alpha'_1 + x\beta'_1 \equiv 0 \pmod{N}$.

We now explain why the algorithm is correct and discuss the running time, treating the descent in Step 2 as a black box algorithm for now. Henceforth, we assume that any random choices used in the descent executions are independent from each other and of the randomness of α and β. For the correctness, note that $g^{\alpha'_1}h^{\beta'_1} = 1$ holds after Step 3, since the first row of R' vanishes. Thus for any integer x such that $\alpha'_1 + x\beta'_1 \equiv 0 \pmod{N}$ we have $g^x = h$, provided that β'_1 is invertible in $\mathbb{Z}/N\mathbb{Z}$.

Lemma 1. *After Step 3 of the algorithm the element $\beta'_1 \in \mathbb{Z}/N\mathbb{Z}$ is uniformly distributed. Therefore, the algorithm succeeds with probability $\varphi(N)/N$, where φ denotes Euler's phi function.*

Proof. We follow the argument from [18, Sec. 5] and [16, Sec. 2.3]. As $h \in \langle g \rangle$, for any fixed value $\beta_i = b \in \mathbb{Z}/N\mathbb{Z}$ the element $g^{\alpha_i}h^b$ is uniformly distributed over the group $\langle g \rangle$, therefore the element $g^{\alpha_i}h^{\beta_i}$ is independent of β_i. As the executions of the descent algorithm are assumed to be independent, we have that the row $(r_{i,1}, \ldots, r_{i,m})$ is also independent of β_i. It follows that the matrix R is independent of the vector β. Then the (invertible) transformation matrix $U \in (\mathbb{Z}/N\mathbb{Z})^{(m+1)\times(m+1)}$ is also independent of β, so that $\beta' = U\beta$ is uniformly distributed over $(\mathbb{Z}/N\mathbb{Z})^{m+1}$, since β is. From this the lemma follows. □

Regarding the running time, for Step 3 we note that a lower row echelon form of R can be obtained using invertible row transformations as for the Smith normal form, which along with the corresponding transformation matrices can be computed in polynomial time [36], so that Step 3 takes time polynomial in m and $\log N$. Furthermore, from [48] we obtain $N/\varphi(N) \in O(\log\log N)$. Altogether this implies that the DLP algorithm has quasipolynomial expected running time (in $\log N$), provided the descent is quasipolynomial. We defer a detailed complexity analysis of the descent to Section 4.

Observe that the algorithm does not require g to be a generator of $\mathbb{F}_{q^{kl}}^{\times}$, which is in practice hard to test without factorising N. In fact, the algorithm

gives rise to a Monte Carlo method for deciding group membership $h \in \langle g \rangle$. Indeed, if a discrete logarithm $\log_g h$ has been computed, then obviously $h \in \langle g \rangle$; thus if $h \notin \langle g \rangle$, we always must have $\gcd(\beta_1', N) > 1$ in Step 4.

Practitioners may have noticed inefficiencies in the algorithm. In particular, in the usual index calculus method one precomputes the logarithms of all factor base elements and then applies a single descent to the target element to obtain its logarithm. Moreover, one usually first computes the logarithm in $\mathbb{F}_{q^{kl}}^\times / \mathbb{F}_{q^k}^\times$, i.e., one ignores multiplicative constants and therefore includes only monic polynomials in the factor base, obtaining the remaining information by solving an additional DLP in $\mathbb{F}_{q^k}^\times$. However, the setup as presented simplifies and facilitates our rigorous analysis.

3 Overview of the Function Field Sieve

In this section we briefly review the classical FFS and describe some of the recent techniques. The knowledgeable reader may omit this section, having familiarised themself with the notation via a brief look at Fig. 1.

Given the embedding of $\mathbb{F}_{p^n}$ into $\mathbb{F}_{q^{kl}}$ as described in the introduction, we focus purely on the latter. A relation in $\mathbb{F}_{q^{kl}}$ is an equality of products of elements in $\mathbb{F}_{q^{kl}}^\times$, or, equivalently, a linear combination of logarithms of elements in $\mathbb{F}_{q^{kl}}^\times$ whose sum is zero. All variants of the FFS rely on the following basic method for obtaining relations. Let $R = \mathbb{F}_{q^k}[X, Y]$ and let $f_1, f_2 \in R$ be two irreducible polynomials such that $R_{12} = R/(f_1, f_2)$ is a finite ring surjecting onto the target field $\mathbb{F}_{q^{kl}}$. Furthermore, for $i = 1, 2$, let $R_i = \mathbb{F}_{q^k}[X, Y]/(f_i)$ and $Z_i \in R$ such that the quotient field $\mathrm{Quot}(R_i)$ is a finite extension of the rational function field $\mathrm{Quot}(Q_i)$ where $Q_i = \mathbb{F}_{q^k}[Z_i]$. This is summarised in Fig. 1.

Via the maps π, φ_1 and φ_2, logarithms in $\mathbb{F}_{q^{kl}}^\times$ can be extended to a notion of logarithms in $R_i \setminus (\pi \circ \varphi_i)^{-1}(0)$, $i = 1, 2$. Therefore, relations can also be viewed as linear combinations of logarithms of elements in R_1 and in R_2 whose sum is zero. It is always implicitly assumed that all logarithms are defined, i.e., that the sets $(\pi \circ \varphi_i)^{-1}(0)$, $i = 1, 2$, are avoided.

A polynomial $P \in R$ gives rise to a relation by decomposing $P \bmod f_i$ in R_i for $i = 1$ and $i = 2$ (and mapping down to R_{12} or $\mathbb{F}_{q^{kl}}$ if desired). Sufficiently many non-trivial relations amongst elements of a set of bounded size allow one to compute logarithms in this set. If the multiplicative closure of such a set is $\mathbb{F}_{q^{kl}}^\times$, arbitrary logarithms can be computed by expressing an element as a product of elements of this set. As was described in Section 1, this is done by following a descent strategy in which elements, also called special-Q, are recursively rewritten as 'easier' elements using relations as above.

In the classical FFS the polynomials f_1, f_2 are chosen such that their degrees are as low as possible, typically of the form $f_1 = Y - a(X)$, $f_2 = \sum_{j=0}^{d} b_j(X) Y^j$ with $\deg_X(a) = e$, $\deg_X(b_j) < e$ and $de > l$, and $Z_1 = Z_2 = X$

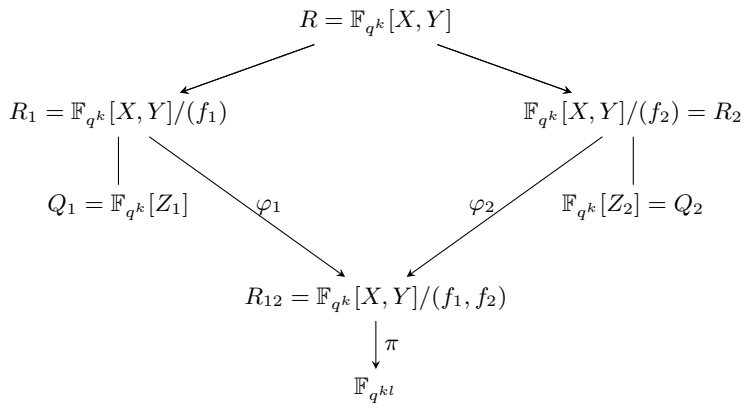

Fig. 1: Setup for the FFS

so that the extensions $\mathrm{Quot}(R_i)/\mathrm{Quot}(Q_i)$, $i = 1, 2$, are of degree 1 and degree d, respectively. By choosing P as a low-degree polynomial, the degrees of the norms $N_{\mathrm{Quot}(R_i)/\mathrm{Quot}(Q_i)}(P \bmod f_i)$, $i = 1, 2$, are not too big and therefore the chance of both norms splitting into low-degree polynomials is sufficiently high. With judiciously selected parameters this gives a heuristic running time of $L(1/3)$.

The main difference between the classical FFS and the recent variations [20, 29, 5] is where the relation generation begins. In the recent variations a product of low-degree polynomials $\tilde{P} = \prod \tilde{P}_j$ in R_1 is constructed in such a way that it can be lifted to a low-degree polynomial $P \in R$ and such that its reduction $P \bmod f_2$ is of sufficiently low degree, where by low degree we mean that the norm has low degree. This can be achieved by choosing q to be of the order of l, $f_1 = Y - X^{q\dagger}$ and f_2 of low degree. Then $R_1 = \mathbb{F}_{q^k}[X]$ and low-degree polynomials $F, G \in R_1$ give rise to relations via

$$\tilde{P} = F^q G - F G^q = G \prod_{\alpha \in \mathbb{F}_q} (F - \alpha G) = \prod \tilde{P}_j, \tag{4}$$

since F^q (resp. G^q) can be expressed as a degree $\deg F$ (resp. $\deg G$) polynomial in Y, and thus $\tilde{P}$ can be lifted to a low-degree polynomial P. This yields a

† An interesting historical aside is that this specialisation was also proposed by Shinohara *et al* in January 2012 in order to half the size of the factor base when q is a power of the characteristic [49, Sec. 4.1], but its impact on relation generation was not considered. Furthermore, in December 2012 Joux used $f_1 = Y - X^d$ for medium characteristic fields with prime base fields [28], which does not help in finding a relation, but does allow one to generate many relations once one has been found, via transformations of the roots. Viewed in this context the selection of $f_1 = Y - X^q$ in [19] and [30] is a very natural (and indeed fertile) one, even if the ensuing approaches diverge in terms of field representation, relation generation and small degree elements elimination.

heuristic polynomial time algorithm for finding relations between elements of $\mathbb{F}_{q^{kl}}$ that are, via φ_1 and π, images of polynomials of bounded degree.

In the descent phase it is advantageous to choose f_2 such that its degree in X or in Y is one (cf. [24] and [29] respectively), which implies that $\mathrm{Quot}(R_2) = \mathrm{Quot}(Q_2)$ with $Z_2 = Y$ or $Z_2 = X$, respectively. More precisely, writing $f_2 = h_1X - h_0$ or $f_2 = h_1Y - h_0$ respectively, with $h_i \in Q_2$, $i = 0, 1$, implies $R_2 = \mathbb{F}_{q^k}[Z_2][\frac{1}{h_1}]$. Up to the logarithm of h_1, the logarithm of a polynomial of R_1 can be related to the logarithm of a corresponding polynomial in R_2 (the same polynomial for $Z_2 = X$ and a Frobenius twist for $Z_2 = Y$) which allows one to view a special-Q (the element to be eliminated) as coming from R_1 or from R_2. In the latter case, the condition that a polynomial $Q \in R_2$, a lift of the special-Q element, divides P mod f_2 for a P arising via (4), can be expressed as a bilinear quadratic system which gives, for appropriate parameter choices, an algorithm with heuristic running time $L(1/4 + o(1))$.

In the other case, namely the special-Q element being lifted to $Q \in R_1$, a certain set of polynomials in R_1 containing Q is chosen in such a way that pairs F, G from this set generate via (4) sufficiently many relations with P mod f_2 splitting into polynomials of sufficiently low degree. Solving a linear system of equations then expresses the logarithm of the special-Q element as a linear combination of logarithms of polynomials in R_2 of sufficiently low degree (and h_1), resulting in the original QPA.

Actually, the relations in the original QPA (and in [29]) are generated in a slightly different manner by applying linear fractional transformations to the polynomial $A = X^q - X = \prod_{\alpha \in \mathbb{F}_q}(X - \alpha)$. The subgroup $\mathrm{PGl}_2(\mathbb{F}_q) \subset \mathrm{PGl}_2(\mathbb{F}_{q^k})$ is the largest subgroup fixing this polynomial, so that the action of $\mathrm{PGl}_2(\mathbb{F}_{q^k})/\mathrm{PGl}_2(\mathbb{F}_q)$ on A produces $\frac{q^{3k}-q^k}{q^3-q}$ polynomials, each splitting into linear polynomials and whose only non-zero terms correspond to the monomials X^{q+1}, X^q, X and 1.

4 The Descent

In this section we detail the building block behind our new descent and explain why its successful application implies Theorem 2, and hence Theorem 1. In the terminology of the previous section, the setup for $\mathbb{F}_{q^{kl}}$ has $f_1 = Y - X^q$ and $f_2 = h_1Y - h_0$ with $h_i \in \mathbb{F}_{q^k}[X]$ of degree at most two for $i = 0, 1$, with $h_1X^q - h_0$ having an irreducible factor I of degree l, i.e., $R_{12} = \mathbb{F}_{q^k}[X, Y]/(f_1, f_2)$ surjects onto $\mathbb{F}_{q^{kl}}$. This implies $R_1 = \mathbb{F}_{q^k}[X]$ and $R_2 = \mathbb{F}_{q^k}[X][\frac{1}{h_1}]$. We assume (without loss of generality) that h_0 and h_1 are coprime. Since the logarithm of h_1 will appear in almost every relation and h_1 is of degree at most two, we adjoin it to the factor basis $\mathcal{F}$, and for the sake of simplicity it will be suppressed in the following description.

4.1 On-the-fly degree two elimination

In this subsection we review the on-the-fly degree two elimination method from [20], adjusted for the present framework. In [6] the affine portion of the set of polynomials obtained as linear fractional transformations of $X^q - X$ is parameterised as follows. Let $\mathcal{B}$ be the set of $B \in \mathbb{F}_{q^k}$ such that the polynomial $X^{q+1} - BX + B$ splits completely over $\mathbb{F}_{q^k}$, the cardinality of which is approximately q^{k-3} [6, Lemma 4.4]. Scaling and translating these polynomials means that all the polynomials $X^{q+1} + aX^q + bX + c$ with $c \neq ab$, $b \neq a^q$ and $B = \frac{(b-a^q)^{q+1}}{(c-ab)^q}$ split completely over $\mathbb{F}_{q^k}$ whenever $B \in \mathcal{B}$.

Let Q (viewed as polynomial in R_2) be an irreducible quadratic polynomial to be eliminated and let $L_Q \subset \mathbb{F}_{q^k}[X]^2$ be the lattice defined by

$$L_Q = \{(w_0, w_1) \in \mathbb{F}_{q^k}[X]^2 \mid w_0 h_0 + w_1 h_1 \equiv 0 \pmod{Q}\}. \tag{5}$$

In the case that Q divides $w_0 h_0 + w_1 h_1 \neq 0$ for some $w_0, w_1 \in \mathbb{F}_{q^k}$, then $Q = w(w_0 h_0 + w_1 h_1)$ for some $w \in \mathbb{F}_{q^k}^\times$, since the degree on the right hand side is at most two. Therefore, the relation generated from $P = w_0 Y + w_1 \in R$ relates the logarithm of Q with the logarithm of $w_0 X^q + w_1 = (w_0^{1/q} X + w_1^{1/q})^q \in R_1$ (and the logarithm of h_1). In the other case, L_Q has a basis of the form $(1, u_0 X + u_1), (X, v_0 X + v_1)$ with $u_i, v_i \in \mathbb{F}_{q^k}$. Since the polynomial $P = XY + aY + bX + c$ maps to $\frac{1}{h_1}((X+a)h_0 + (bX+c)h_1)$ in R_2, Q divides P mod f_2 if and only if $(X + a, bX + c) \in L_Q$. Note that the numerator of P mod f_2 is of degree at most three, thus it can at worst contain a linear factor besides Q. If the triple (a, b, c) also satisfies $c \neq ab$, $b \neq a^q$ and $\frac{(b-a^q)^{q+1}}{(c-ab)^q} \in \mathcal{B}$, then P mod f_1 splits into linear factors and the logarithm of Q has been rewritten in terms of logarithms of linear polynomials.

Algorithmically, a triple (a, b, c) satisfying all conditions can be found in several ways. Choosing a $B \in \mathcal{B}$, considering $(X + a, bX + c) = a(1, u_0 X + u_1) + (X, v_0 X + v_1)$ and rewriting $b = u_0 a + v_0$ and $c = u_1 a + v_1$ gives the condition

$$B = \frac{(-a^q + u_0 a + v_0)^{q+1}}{(-u_0 a^2 + (-v_0 + u_1)a + v_1)^q}. \tag{6}$$

By expressing a in an $\mathbb{F}_{q^k}/\mathbb{F}_q$ basis, (6) results in a quadratic system in k variables [21]. Using a Gröbner basis algorithm the running time is exponential in k. Alternatively, and this is one of the key observations for the present work, equation (6) can be considered as a polynomial of degree q^2+q in a whose roots can be found in polynomial time in q and in k by taking a GCD with $a^{q^k} - a$ in $\mathbb{F}_{q^k}[a]$ [20]. One can also check for random (a, b, c) such that the lattice condition holds, whether $X^{q+1} + aX^q + bX + c$ splits into linear polynomials, which happens with probability q^{-3}. This is also polynomial time in q and in k.

These degree 2 elimination methods will fail when Q divides $h_1 X^q - h_0$, because this would imply that the polynomial P mod $f_1 = X^{q+1} + aX^q + bX + c$

is divisible by Q whenever $P \bmod f_2$ is, a problem first discussed in [8]. Such polynomials Q or their roots will be called traps of level 0. Similarly, these degree 2 elimination methods might also fail when Q divides $h_1 X^{q^{k+1}} - h_0$, in which case such polynomials Q or their roots will be called traps of level k. By considering the degrees it follows easily that there are at most $q+2$ traps of level 0 and at most $q^{k+1}+2$ traps of level k for $k > 0$.

4.2 Elimination requirements

As we will see shortly, the on-the-fly degree two elimination method can be transformed into an elimination method for irreducible even degree polynomials. We now present a theorem which states that under some assumptions this degree two elimination is guaranteed to succeed, and subsequently demonstrate that it implies Theorem 2, and hence Theorem 1.

Let $\mathcal{T}$ be the set of *trap roots*, i.e., the set of roots of $h_1 X^{q^{kd+1}} - h_0$ for all $d \geq 0$ minus the set of roots of the field-defining polynomial I. A polynomial in R_1 or R_2 is said to be *good* if it has no roots in $\mathcal{T}$. The same definitions are used when the base field of R_1 and R_2 is extended.

Theorem 3. *Let $q > 61$ be a prime power that is not a power of 4, let $k \geq 18$ be an integer and let $h_0, h_1 \in \mathbb{F}_{q^k}[X]$ be of degree at most two with $h_1 X^q - h_0$ having an irreducible degree l factor. Moreover, let $d \geq 1$ be an integer, let $Q \in \mathbb{F}_{q^{kd}}[X]$ be an irreducible quadratic good polynomial such that Q does not divide $w_0 h_0 + w_1 h_1 \neq 0$ for $w_0, w_1 \in \mathbb{F}_{q^{kd}}$, and let $(1, u_0 X + u_1), (X, v_0 X + v_1)$ be a basis of the lattice L_Q in (5). Then the number of solutions $(a, B) \in \mathbb{F}_{q^{kd}} \times \mathcal{B}$ of (6) resulting in good descendents is lower bounded by q^{kd-5}.*

This theorem is of central importance for our rigorous analysis and will be proved in Section 5.

4.3 Degree $2d$ elimination and descent complexity

Now we demonstrate how the on-the-fly degree two elimination gives rise to a method for eliminating irreducible even degree polynomials, which will be the crucial building block for our descent algorithm. As per Theorem 3, let $q > 61$ be a prime power that is not a power of 4, let $k \geq 18$, and let h_0, h_1, I as before.

Proposition 1. *Let $Q \in R_2$ be an irreducible good polynomial of degree $2d$ with $d \geq 1$. Then the logarithm of Q can be rewritten as a linear combination of at most $q+2$ logarithms of irreducible good polynomials of degrees dividing d, in a running time polynomial in q and in d.*

Proof. Over the extension $\mathbb{F}_{q^{kd}}$ the polynomial Q splits into d irreducible good quadratic polynomials; let Q' be one of them. As explained earlier, we may

assume that Q does not divide $w_0h_0 + w_1h_1 \neq 0$ for some $w_0, w_1 \in \mathbb{F}_{q^{kd}}$. By Theorem 3 we may apply the on-the-fly degree two elimination method for $Q' \in \mathbb{F}_{q^{kd}}[X]$, which gives a polynomial $P' \in \mathbb{F}_{q^{kd}}[X, Y]$ such that $P' \bmod f_1$ splits into a product of at most $q + 1$ good polynomials of degree one over $\mathbb{F}_{q^{kd}}$ and such that $(P' \bmod f_2)h_1$ is a product of Q' and a good polynomial of degree at most one. Let P be the product of all conjugates of P' under $\mathrm{Gal}(\mathbb{F}_{q^{kd}}/\mathbb{F}_{q^k})$. Since the product of all conjugates of a linear polynomial under $\mathrm{Gal}(\mathbb{F}_{q^{kd}}/\mathbb{F}_{q^k})$ is the d_1-th power of an irreducible degree d_2 polynomial for d_1 and d_2 satisfying $d_1d_2 = d$, the rewriting assertion of the claim follows.

The three steps of this method – computing Q', the on-the-fly degree two elimination (when the second or third approach listed above for solving (6) is used), and the computation of the polynomial norms – all have running time polynomial in q and in d, which proves the running time assertion of the claim. □

By recursively applying Proposition 1 we can express the logarithm of a good irreducible polynomial of degree 2^e, $e \geq 1$, in terms of at most $(q + 2)^e$ logarithms of linear polynomials. The final step of this recursion, namely eliminating up to $(q + 2)^{e-1}$ quadratic polynomials, dominates the running time, which is thus upper bounded by $(q + 2)^e$ times a polynomial in q.

Lemma 2. *Any element in R_{12} can be lifted to an irreducible good polynomial of degree 2^e whenever $2^e > 4n$, where $n = \deg(h_1X^q - h_0)$.*

Proof. By the effective Dirichlet-type theorem on irreducibles in arithmetic progressions [52, Thm. 5.1], for $2^e > 4n$ the probability of irreducibility for a random lift is lower bounded by 2^{-e-1}. One may actually find an irreducible polynomial of degree 2^e which is good, since the number of traps ($< 2q^{2^{e-1}+1}$) is much smaller than the number ($> q^{2^e-n}2^{-e-1}$) of irreducibles produced by this Dirichlet-type theorem. □

Putting everything altogether, this proves the expected quasi-polynomial running time of the descent and therefore the running time of our algorithm in Section 2, establishing Theorem 2.

Finally note that during an elimination step, one need not use the basic building block as stated, which takes the norms of the linear polynomials produced back down to $\mathbb{F}_{q^k}$. Instead, one need only take their norms to a subfield of index 2, thus becoming quadratic polynomials, and then recurse, as depicted in Fig. 2.

5 Proof of Theorem 3

In this section we prove Theorem 3, which by the arguments of the previous section demonstrates the correctness of our algorithm and our main theorems.

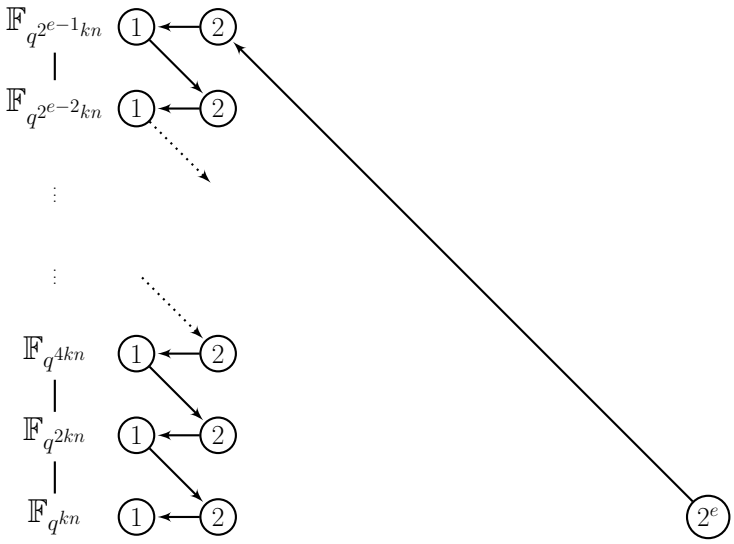

Fig. 2: Elimination of irreducible polynomials of degree a power of 2 when considered as elements of $\mathbb{F}_{q^k}[X]$. The arrow directions $\nwarrow, \leftarrow$ and $\searrow$ indicate factorisation, degree 2 elimination and taking a norm with respect to the indicated subfield, respectively.

5.1 Notation and statement of supporting results

Let $K = \mathbb{F}_{q^{kd}}$ with $kd \geq 18$, let $L = \mathbb{F}_{q^{2kd}}$ be its quadratic extension, and let Q be an irreducible quadratic polynomial in $K[X]$ such that $(1, u_0X + u_1), (X, v_0X + v_1)$ is a basis of its associated lattice L_Q in (5). Then Q is a scalar multiple of $-u_0X^2 + (-u_1 + v_0)X + v_1$.

Let $\mathcal{B}$ be the set of $B \in K$ such that the polynomial $X^{q+1} - BX + B$ splits completely over K. Using an elementary extension of [27, Theorem 5] the set $\mathcal{B}$ can be characterised as the image of $K \setminus \mathbb{F}_{q^2}$ under the map

$$u \mapsto \frac{(u - u^{q^2})^{q+1}}{(u - u^q)^{q^2+1}}. \tag{7}$$

By this and (6), in order to eliminate Q we need to find $(a, u) \in K \times (K \setminus \mathbb{F}_{q^2})$ satisfying

$$(u - u^{q^2})^{q+1}(-u_0a^2 + (-v_0 + u_1)a + v_1)^q - (u - u^q)^{q^2+1}(-a^q + u_0a + v_0)^{q+1} = 0.$$

The two terms have a common factor $(u - u^q)^{q+1}$ which motivates the following definitions. Let $\alpha = -u_0$, $\beta = u_1 - v_0$, $\gamma = v_1$ and $\delta = -v_0$ with $\alpha, \beta, \gamma, \delta \in K$,

as well as

$$D = \frac{U^{q^2} - U}{U^q - U} = \prod_{\epsilon \in \mathbb{F}_{q^2} \setminus \mathbb{F}_q} (U - \epsilon),$$
$$E = U^q - U = \prod_{\epsilon \in \mathbb{F}_q} (U - \epsilon),$$
$$F = \alpha A^2 + \beta A + \gamma = \alpha(A - \rho_1)(A - \rho_2) \quad \text{with} \quad \rho_1, \rho_2 \in L,$$
$$G = A^q + \alpha A + \delta \quad \text{and}$$
$$P = D^{q+1} F^q - E^{q^2 - q} G^{q+1} \in K[A, U].$$

Note that F equals $Q(-A)$ (up to a scalar), so that $\deg(F) = 2$, F is irreducible and $\rho_1, \rho_2 \notin K$. We consider the curve C defined by $P = 0$ and are interested in the number of (affine) points $(a, u) \in C(K)$ with $u \notin \mathbb{F}_{q^2}$. More precisely, we want to prove the following.

Theorem 4. *Let $q > 61$ be a prime power that is not a power of 4. If the conditions*

$$(*) \qquad \rho_1^q + \alpha\rho_2 + \delta \neq 0$$
$$(**) \qquad \rho_1^q + \alpha\rho_1 + \delta \neq 0$$

hold then there are at least q^{kd-1} pairs $(a, u) \in K \times (K \setminus \mathbb{F}_{q^2})$ satisfying $P(a, u) = 0$.

The relation of the two conditions to the quadratic polynomial Q as well as properties of traps are described in the following propositions.

Proposition 2. *If condition $(*)$ is not satisfied, then Q divides $h_1 X^q - h_0$, i.e., Q is a trap of level 0. If condition $(**)$ is not satisfied, then Q divides $h_1 X^{q^{kd+1}} - h_0$, i.e., Q is a trap of level kd. In particular, if Q is a good polynomial then conditions $(*)$ and $(**)$ are satisfied.*

Proposition 3. *Let $(a, u), (a', u') \in K \times (K \setminus \mathbb{F}_{q^2})$ be two solutions of $P = 0$ with $a \neq a'$, corresponding to the polynomials $\mathcal{P}_a = XY + aY + bX + c$ and $\mathcal{P}_{a'} = XY + a'Y + b'X + c'$, respectively. Then $\mathcal{P}_a \bmod f_1$ and $\mathcal{P}_{a'} \bmod f_1$ have no common roots. Furthermore, the common roots of $\mathcal{P}_a \bmod f_2$ and $\mathcal{P}_{a'} \bmod f_2$ are precisely the roots of Q.*

Now we explain how (for $q > 61$ not a power of 4) Theorem 3 follows from the above theorem and the propositions. By Proposition 2 an irreducible quadratic good polynomial Q satisfies the two conditions of Theorem 4. Since the map (7) is $q^3 - q : 1$ on $K \setminus \mathbb{F}_{q^2}$, there are at least q^{kd-4} solutions $(a, B) \in K \times \mathcal{B}$ of (6), which contain at least q^{kd-4} different values $a \in K$. As there are at most $q^{d'+1} + 2$ traps of level d' for any $d' \geq 0$, the set of traps of level dividing

$\frac{kd}{2}$ and of traps of level 0 has cardinality at most $q^{\frac{kd}{2}+3}$. By Proposition 3 a trap root can appear in $\mathcal{P}_a$ mod f_j for at most two values a, at most once for $j = 1$ and at most once for $j = 2$. Hence there are at most $q^{\frac{kd}{2}+4} < q^{kd-5}$ values a for which a trap root appears in $\mathcal{P}_a$ mod f_j, $j = 1, 2$. Thus there are at least q^{kd-5} different values a for which a solution (a, B) leads to an elimination into logarithms of good polynomials. This finishes the proof of Theorem 3, hence we focus on proving the theorem and the two propositions above.

5.2 Outline of the proof method

The main step of the proof of the theorem consists in showing that, subject to conditions $(*)$ and $(**)$, there exists an absolutely irreducible factor P_1 of P that lies already in $K[A, U]$. Since the (total) degree of P_1 is at most $q^3 + q$, restricting to the component of the curve defined by P_1 and using the Weil bound for possibly singular plane curves gives a lower bound on the cardinality of $C(K)$ which is large enough to prove the theorem after accounting for projective points and points with second coordinate in $\mathbb{F}_{q^2}$. This argument is given in the next subsection before dealing with the more involved main step.

For proving the main step the action of $\mathrm{PGl}_2(\mathbb{F}_q)$ on the variable U is considered. An absolutely irreducible factor P_1 of P is stabilised by a subgroup $S_1 \subset \mathrm{PGl}_2(\mathbb{F}_q)$ satisfying some conditions. The first step is to show that, after possibly switching to another absolutely irreducible factor, there are only a few cases for the subgroup. Then for each case it is shown that the factor is defined over $K[A, U]$ or that one of the conditions on the parameters is not satisfied.

Proving the propositions will be done in the final subsection.

5.3 Weil bound

Corollary 2.5 of [4] shows that for an absolutely irreducible plane curve C of degree d' we have

$$|\#C(K) - q^{kd} - 1| \leq (d' - 1)(d' - 2)q^{\frac{kd}{2}}.$$

Since $\deg_A(P) = q^2 + q$ there are at most $q^4 + q^3$ affine points with $u \in \mathbb{F}_{q^2}$. The number of points at infinity is at most $d' = q^3 + q^2 < q^4$. Denoting by $C(K)\tilde{}$ the set of affine points in $C(K)$ with second coordinate $u \notin \mathbb{F}_{q^2}$ one obtains

$$|\#C(K)\tilde{}| > q^{kd} - (q^4 + q^3) - d' - (d'^2 - 2)q^{\frac{kd}{2}} > q^{kd} - q^{\frac{kd}{2}+8} \geq q^{kd-1},$$

since $kd \geq 18$, thus proving the theorem if there exists an absolutely irreducible factor defined over $K[A, U]$.

5.4 $\mathbf{PGl_2}$ action

Here the following convention for the action of $\mathrm{PGl}_2(\mathbb{F}_q)$ on $\mathbb{P}^1$ and on polynomials is used. A matrix $\begin{pmatrix} a & b \\ c & d \end{pmatrix} \in \mathrm{PGl}_2(\mathbb{F}_q)$ acts on $\mathbb{P}^1(M)$, where M is an arbitrary field containing $\mathbb{F}_q$, by $(x_0 : x_1) \mapsto \begin{pmatrix} a & b \\ c & d \end{pmatrix}(x_0 : x_1) = (ax_0 + bx_1 : cx_0+dx_1)$ or, via $\mathbb{P}^1(M) = M \cup \{\infty\}$, by $x \mapsto \frac{ax+b}{cx+d}$. This is an action on the left, i.e., for $\sigma, \tau \in \mathrm{PGl}_2(\mathbb{F}_q)$ and $x \in \mathbb{P}^1(M)$ the following holds: $\sigma(\tau(x)) = (\sigma\tau)(x)$. On a homogeneous polynomial H in the variables $(X_0 : X_1)$ the action of $\sigma = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$ is given by $H^\sigma(X_0 : X_1) = H(aX_0+bX_1 : cX_0+dX_1)$. This is an action on the right, satisfying $H^{(\sigma\tau)} = (H^\sigma)^\tau$. In the following we will usually use this action on the dehomogenised polynomials given by $H^\sigma(X) = H(\frac{aX+b}{cX+d})$, clearing denominators in the appropriate way.

The polynomial $P \in (K[A])[U]$ is invariant under $\mathrm{PGl}_2(\mathbb{F}_q)$ acting on the variable U; this can be checked by considering the actions of $\begin{pmatrix} 1 & b \\ 0 & 1 \end{pmatrix}$, $\begin{pmatrix} a & 0 \\ 0 & 1 \end{pmatrix}$ and $\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$, and noticing that $\mathrm{PGl}_2(\mathbb{F}_q)$ is generated by these matrices. Let

$$P = s \prod_{i=1}^{g} P_i, \qquad P_i \in (\overline{K}[A])[U], \ s \in \overline{K}[A],$$

be the decomposition of P in $(\overline{K}[A])[U]$ into irreducible factors P_i and possibly reducible s. Notice that s must divide F^q and G^{q+1}, hence it divides a power of $\gcd(F, G)$. As F is irreducible, $\gcd(F, G)$ is either constant or of degree two. In the latter case ρ_1 is a root of G contradicting condition $(**)$. Therefore one can assume that $s \in \overline{K}$ is a constant.

Let

$$P = F^q \prod_{i=1}^{q^3-q} (U - r_i), \qquad r_i \in \overline{K(A)},$$

be the decomposition of P in $\overline{K(A)}[U]$. Then $\mathrm{PGl}_2(\mathbb{F}_q)$ permutes the set $\{r_i\}$ and, since fixed points of $\mathrm{PGl}_2(\mathbb{F}_q)$ lie in $\mathbb{F}_{q^2}$ but $r_i \notin \mathbb{F}_{q^2}$, the action is free. Since $\#\,\mathrm{PGl}_2(\mathbb{F}_q) = q^3 - q$ the action is transitive.

Therefore the action on the decomposition over $\overline{K}[A, U]$ is also transitive (adjusting the P_i by scalars in $\overline{K}[A]$ if necessary). Denoting by $S_i \subset \mathrm{PGl}_2(\mathbb{F}_q)$ the stabiliser of P_i it follows that all S_i are conjugates of each other, thus they have the same cardinality and hence $q^3 - q = g \cdot \#S_i$. Moreover the degree of P_i in U is constant, namely $\#S_i$, and also the degree of P_i in A is constant, thus $g \mid q^2 + q = \deg_A(P)$. In particular, $q - 1 \mid \#S_i$.

5.5 Subgroups of PGl_2

The classification of subgroups of $\mathrm{PSl}_2(\mathbb{F}_q)$ is well known [15] and allows to determine all subgroups of $\mathrm{PGl}_2(\mathbb{F}_q)$ [7]. Since $\#S_i$ is divisible by $q-1$ (in particular $\#S_i > 60$), only the following subgroups are of interest (per conjugation class only one subgroup is listed):

1. the cyclic group $\begin{pmatrix} * & 0 \\ 0 & 1 \end{pmatrix}$ of order $q-1$,
2. the dihedral group $\begin{pmatrix} * & 0 \\ 0 & 1 \end{pmatrix} \cup \begin{pmatrix} 0 & 1 \\ * & 0 \end{pmatrix}$ of order $2(q-1)$ as well as, in odd characteristic, its two dihedral subgroups

$$\left\{ \begin{pmatrix} a & 0 \\ 0 & 1 \end{pmatrix} \mid a \neq 0 \text{ a square} \right\} \cup \left\{ \begin{pmatrix} 0 & 1 \\ c & 0 \end{pmatrix} \mid c \neq 0 \text{ a square} \right\} \quad \text{and}$$
$$\left\{ \begin{pmatrix} a & 0 \\ 0 & 1 \end{pmatrix} \mid a \neq 0 \text{ a square} \right\} \cup \left\{ \begin{pmatrix} 0 & 1 \\ c & 0 \end{pmatrix} \mid c \text{ not a square} \right\},$$

   both of order $q-1$,
3. the Borel subgroup $\begin{pmatrix} * & * \\ 0 & 1 \end{pmatrix}$ of order q^2-q,
4. if q is odd, $\mathrm{PSl}_2(\mathbb{F}_q)$ of index 2,
5. if $q = q'^2$ is a square, $\mathrm{PGl}_2(\mathbb{F}_{q'})$ of order $q'^3 - q' = q'(q-1)$, and
6. $\mathrm{PGl}_2(\mathbb{F}_q)$.

In the last case P is absolutely irreducible, thus it remains to investigate the first five cases which will be done in the next subsection.

Remark: The condition $q > 61$ rules out some small subgroups as A_4, S_4, and A_5. In many of the finitely many cases $q \leq 61$ the proof of the theorem also works (e.g., q not a square and $q-1 \nmid 120$). The condition of q not being a power of even exponent of 2 eliminates the fifth case in characteristic 2; removing this condition would be of some interest.

5.6 The individual cases

Since the stabilisers S_i are conjugates of each other, one can assume without loss of generality that S_1 is one of the explicit subgroups given in the previous subsection. Then the polynomial P_1 is invariant under certain transformations of U, so that P_1 and P can be rewritten in terms of another variable as stated in the following.

If a polynomial (in the variable U) is invariant under $U \mapsto aU$, $a \in \mathbb{F}_q^\times$, it can be considered as a polynomial in the variable $V = U^{q-1}$. For the polynomials D and E^{q-1} one obtains

$$D = \frac{V^{q+1}-1}{V-1} \qquad \text{and} \qquad E^{q-1} = V(V-1)^{q-1}.$$

Similarly, in the case of odd q, if a polynomial is invariant under $U \mapsto aU$ for all squares $a \in \mathbb{F}_q^\times$, it can be rewritten in the variable $V' = U^{\frac{q-1}{2}}$. For D and E^{q-1} this gives

$$D = \frac{V'^{2q+2} - 1}{V'^2 - 1} \qquad \text{and} \qquad E^{q-1} = V'^2(V'^2 - 1)^{q-1}.$$

If a polynomial is invariant under $U \mapsto U + b$, $b \in \mathbb{F}_q$, it can be considered as a polynomial in $\tilde{V} = U^q - U$ which gives

$$D = \tilde{V}^{q-1} + 1 \qquad \text{and} \qquad E^{q-1} = \tilde{V}^{q-1}.$$

Combining the above yields that a polynomial which is invariant under both $U \mapsto aU$, $a \in \mathbb{F}_q^\times$, and $U \mapsto U + b$, $b \in \mathbb{F}_q$, can be considered as a polynomial in $W = \tilde{V}^{q-1} = (U^q - U)^{q-1}$. For D and E^{q-1} one obtains

$$D = W + 1 \qquad \text{and} \qquad E^{q-1} = W.$$

This is now applied to the various cases for S_1.

The cyclic case Rewriting P and P_1 in terms of $V = U^{q-1}$ one obtains

$$P = \Big(\frac{V^{q+1} - 1}{V - 1}\Big)^{q+1} F^q - V^q(V-1)^{q^2-q} G^{q+1}$$

and $\deg_V(P_1) = 1$, i.e., $P_1 = p_1 V - p_0$ with $p_i \in \overline{K}[A]$, $\gcd(p_0, p_1) = 1$, $\max(\deg(p_0), \deg(p_1)) = 1$ and it can be assumed that p_0 is monic.

The divisibility $P_1 \mid P$ transforms into the following polynomial identity in $\overline{K}[A]$:

$$\Big(\frac{p_0^{q+1} - p_1^{q+1}}{p_0 - p_1}\Big)^{q+1} F^q = p_1^q p_0^q (p_0 - p_1)^{q^2-q} G^{q+1}.$$

The degree of the first factor on the left hand side is either $q^2 + q$ or $q^2 - 1$ (if $p_0 - \zeta p_1$ is constant for some $\zeta \in \mu_{q+1}(\mathbb{F}_{q^2}) \setminus \{1\}$). Since the degrees of the other factors are all divisible by q, the latter case is impossible. Since $\deg(F) = 2$ one gets $\deg(F^q) = 2q$. Furthermore, $\deg((p_0 p_1)^q) \in \{q, 2q\}$, $\deg((p_0 - p_1)^{q^2-q}) \in \{0, q^2 - q\}$ and $\deg(G^{q+1}) = q^2 + q$ which implies $\deg(p_0 - p_1) = 0$, $\deg(p_0) = \deg(p_1) = 1$ since $q > 2$.

Let $p_0 - p_1 = c_1 \in \overline{K}$; in the following c_i will be some constants in $\overline{K}$. Since the first factor on the left hand side is coprime to $p_0 p_1$, it follows

$$\frac{p_0^{q+1} - p_1^{q+1}}{p_0 - p_1} = c_2 G, \quad F = c_3 p_0 p_1 \quad \text{and} \quad c_2^{q+1} c_3^q = c_1^{q^2-q}.$$

Exchanging ρ_1 and ρ_2, if needed, one obtains

$$p_0 = A - \rho_1, \quad p_1 = A - \rho_2, \quad c_3 = \alpha \quad \text{and} \quad c_1 = \rho_2 - \rho_1.$$

Considering the coefficient of A^q in the equation for G gives $c_2 = 1$ and evaluating this equation at $A = \rho_2$ gives

$$\rho_1^q + \alpha\rho_2 + \delta = 0.$$

This means that condition $(*)$ does not hold.

The dihedral cases The case of the dihedral group of order $2(q-1)$ is considered first. Then, as above, P and P_1 can be expressed in terms of V, and, since P and P_1 are also invariant under $V \mapsto \frac{1}{V}$, they can be expressed in terms of $W_+ = V + \frac{1}{V}$. This gives $\deg_{W_+}(P_1) = 1$ and with $\mathcal{Z} = \mu_{q+1}(\mathbb{F}_{q^2}) \setminus \{1\}$

$$D^{q+1}V^{-\frac{q^2+q}{2}} = \prod_{\zeta\in\mathcal{Z}} (W_+ - (\zeta + \zeta^q))^{\frac{q+1}{2}} \qquad \text{and}$$

$$PV^{-\frac{q^2+q}{2}} = \Big(\prod_{\zeta\in\mathcal{Z}} (W_+ - (\zeta + \zeta^q))^{\frac{q+1}{2}}\Big) F^q - (W_+ - 2)^{\frac{q^2-q}{2}} G^{q+1}.$$

In characteristic 2 each factor of the product over $\mathcal{Z}$ appears twice, thus justifying their exponent $\frac{q+1}{2}$.

By writing $P_1 = p_1 W_+ - p_0$, with $p_i \in \overline{K}[A]$, $\gcd(p_0, p_1) = 1$, $\max(\deg(p_0), \deg(p_1)) = 2$ and p_0 being monic, the divisibility $P_1 \mid P$ transforms into the following polynomial identity in $\overline{K}[A]$:

$$\Big(\prod_{\zeta\in\mathcal{Z}} (p_0 - (\zeta + \zeta^q)p_1)^{\frac{q+1}{2}}\Big) F^q = p_1^q (p_0 - 2p_1)^{\frac{q^2-q}{2}} G^{q+1}.$$

Again the degree of the first factor on the left hand side must be divisible by q (respectively, $\frac{q}{2}$ in characteristic 2), and since $p_0 - (\zeta + \zeta^q)p_1$ can be constant or linear for at most one sum $\zeta + \zeta^q$, the degree of the first factor must be $q^2 + q$ for $q > 4$. Also the degree of $p_0 - 2p_1$ must be zero since $q > 2$ and thus the degree of p_1 is 2, as well as the degree of F.

In even characteristic $p_0 - 2p_1 = p_0$ is a constant, thus $p_0 = 1$ (p_0 is monic). The involution $\zeta \mapsto \zeta^q = \zeta^{-1}$ on $\mathcal{Z}$ has no fixed points, and, denoting by $\mathcal{Z}_2$ a set of representatives of $\mathcal{Z}$ modulo the involution, one obtains

$$\prod_{\zeta\in\mathcal{Z}_2} (1 - (\zeta + \zeta^q)p_1) = c_1 G, \quad F = c_2 p_1 \quad \text{and} \quad c_1^{q+1} c_2^q = 1.$$

Modulo F one gets $F \mid c_1 G - 1$ which implies $c_1 \in K$. Thus $c_2 \in K$, $p_1 \in K[A]$ and therefore $P_1 \in K[A, U]$.

In odd characteristic the factor corresponding to $\zeta = -1$, namely $(p_0 + 2p_1)^{\frac{q+1}{2}}$, is coprime to the other factors in the product and coprime to $p_1(p_0 - 2p_1)$. Hence $p_0 + 2p_1$ must be a square and its square root must divide G. Moreover, one gets $F = c_1 p_1$. Since $p_0 - 2p_1 = c_2$ is a constant and p_0 is

monic, one gets $c_1 = 2\alpha$, implying $p_1 \in K[A]$. Since $p_0 + 2p_1 = 4p_1 + c_2$ is a square, its discriminant is zero, thus $c_2 \in K$ and hence $P_1 \in K[A, U]$.

If S_1 is one of the two dihedral subgroups of order $q-1$ (which implies that q is odd), the argumentation is similar. The polynomials P and P_1 are expressed in terms of $V' = U^{\frac{q-1}{2}}$ and then, since $U \mapsto \frac{1}{cU}$ becomes $V' \mapsto c^{-\frac{q-1}{2}} \frac{1}{V'}$ with $c^{-\frac{q-1}{2}} = \pm 1$, in terms of $W'_+ = V' + \frac{1}{V'}$ or $W'_- = V' - \frac{1}{V'}$, respectively. In the first case P is rewritten as

$$PV'^{-(q^2+q)} = \Big(\prod_{\zeta \in \mathcal{Z}'} (W'_+ - (\zeta + \zeta^{-1}))^{\frac{q+1}{2}} \Big) F^q - (W'_+ - 2)^{\frac{q^2-q}{2}} (W'_+ + 2)^{\frac{q^2-q}{2}} G^{q+1}$$

where $\mathcal{Z}' = \mu_{2(q+1)}(\mathbb{F}_{q^2}) \setminus \{\pm 1\}$. By setting $P_1 = p_1 W'_+ - p_0$ with $p_i \in \overline{K}[A]$, $\gcd(p_0, p_1) = 1$, $\max(\deg(p_0), \deg(p_1)) = 1$ and p_0 being monic, one obtains

$$\Big(\prod_{\zeta \in \mathcal{Z}'} (p_0 - (\zeta + \zeta^{-1}) p_1)^{\frac{q+1}{2}} \Big) F^q = p_1^{2q} (p_0 - 2p_1)^{\frac{q^2-q}{2}} (p_0 + 2p_1)^{\frac{q^2-q}{2}} G^{q+1}.$$

Since one of $p_0 \pm 2p_1$ is not constant, the degree of the right hand side exceeds the degree of the left hand side for $q > 5$ which is a contradiction.

In the second case P is rewritten as

$$PV'^{-(q^2+q)} = \Big(\prod_{\zeta \in \mathcal{Z}'} (W'_- - (\zeta - \zeta^{-1}))^{\frac{q+1}{2}} \Big) F^q - W'^{q^2-q}_- G^{q+1}$$

and by setting $P_1 = p_1 W'_- - p_0$ with $p_i \in \overline{K}[A]$, $\gcd(p_0, p_1) = 1$, $\max(\deg(p_0), \deg(p_1)) = 1$ and p_0 being monic, one obtains

$$\Big(\prod_{\zeta \in \mathcal{Z}'} (p_0 - (\zeta - \zeta^{-1}) p_1)^{\frac{q+1}{2}} \Big) F^q = p_1^{2q} p_0^{q^2-q} G^{q+1}.$$

Considering the degrees for $q > 5$ it follows that p_0 must be constant and hence p_1 is of degree one. Since p_1 is coprime to the first factor on the left hand side, it must divide F^q which implies $\rho_1 = \rho_2 \in K$, contradicting the irreducibility of F.

The Borel case In this case, rewriting P and P_1 in terms of $W = (U^q - U)^{q-1}$ gives

$$P = (W+1)^{q+1} F^q - W^q G^{q+1}$$

and $\deg_W(P_1) = 1$, $P_1 = p_1 W - p_0$, with $p_i \in \overline{K}[A]$, $\gcd(p_0, p_1) = 1$, $\max(\deg(p_0), \deg(p_1)) = q$ and p_1 being monic. Then the divisibility $P_1 \mid P$ transforms into the following polynomial identity in $\overline{K}[A]$:

$$(p_0 + p_1)^{q+1} F^q = p_1 p_0^q G^{q+1}.$$

From $\deg(G^{q+1}) = q^2 + q$, $\deg(p_1 p_0^q) \geq q$ and $\deg(F^q) \leq 2q$ it follows that the degree of $p_0 + p_1$ must be q. This implies $\deg(F^q) = \deg(p_1 p_0^q)$, thus $\deg(p_0) \leq 2$ and therefore $\deg(p_1) = q$, $\deg(p_0) \leq 1$ since $q > 2$.

Since $p_0 + p_1$ is coprime to $p_0 p_1$, it follows

$$p_0 + p_1 = c_1 G, \quad p_1 = \tilde{p}^q, \quad F = c_2 \tilde{p} p_0 \quad \text{and} \quad c_1^{q+1} c_2^q = 1$$

for a monic linear polynomial $\tilde{p} \in \overline{K}[A]$.

Exchanging ρ_1 and ρ_2, if needed, one obtains

$$\tilde{p} = A - \rho_1, \quad p_0 = c_3(A - \rho_2), \quad c_1 = 1, \quad c_2 = 1 \quad \text{and} \quad c_3 = \alpha.$$

Evaluating $p_0 + p_1 = G$ at $A = 0$ gives

$$\rho_1^q + \alpha \rho_2 + \delta = 0.$$

This means that condition $(*)$ does not hold.

The $\mathbf{PSl_2}$ case This case can only occur for odd q, and then P splits as $P = sP_1P_2$ with a scalar $s \in \overline{K}$. The map $U \mapsto aU$ for a non-square $a \in \mathbb{F}_q$ exchanges P_1 and P_2. Since $\mathrm{PSl}_2(\mathbb{F}_q)$ is a normal subgroup of $\mathrm{PGl}_2(\mathbb{F}_q)$, P_2 is invariant under $\mathrm{PSl}_2(\mathbb{F}_q)$ as well. By rewriting P in terms of $W' = (U^q - U)^{\frac{q-1}{2}}$ one obtains

$$P = (W'^2 + 1)^{q+1} F^q - W'^{2q} G^{q+1} = sP_1(W')P_1(-W').$$

Denoting by $p_0 \in \overline{K}[A]$ the constant coefficient of $P_1 \in (\overline{K}[A])[W']$ this becomes modulo W'

$$F^q = sp_0^2$$

which implies $\rho_1 = \rho_2 \in K$, contradicting the irreducibility of F.

The case $\mathbf{PGl_2(\mathbb{F}_{q'})}$ Since $\mathrm{PGl}_2(\mathbb{F}_{q'}) \subset \mathrm{PSl}_2(\mathbb{F}_q)$ in odd characteristic, one can reduce this case to the previous case as follows.

Let $I_1 \subset \{1, \ldots, g\}$ be the subset of i such that S_i is a conjugate of S_1 by an element in $\mathrm{PSl}_2(\mathbb{F}_q)$, and let $I_2 = \{1, \ldots, g\} \setminus I_1$. These two sets correspond to the two orbits of the action of $\mathrm{PSl}_2(\mathbb{F}_q)$ on the S_i (or P_i). Both orbits contain $\#I_1 = \#I_2 = \frac{g}{2}$ elements and an element in $\mathrm{PGl}_2(\mathbb{F}_q) \setminus \mathrm{PSl}_2(\mathbb{F}_q)$ transfers one orbit into the other.

Let $\tilde{P}_j = \prod_{i \in I_j} P_i$, $j = 1, 2$, then P splits as $P = s\tilde{P}_1\tilde{P}_2$, $s \in \overline{K}$, and both $\tilde{P}_j$, $j = 1, 2$, are invariant under $\mathrm{PSl}_2(\mathbb{F}_q)$. Notice that the absolute irreducibility of P_1 and P_2 was not used in the argument in the PSl_2 case.

5.7 Traps

In the following the propositions are proven.

Let Q be an irreducible quadratic polynomial in $K[X]$ such that $(1, u_0X + u_1), (X, v_0X + v_1)$ is a basis of the lattice L_Q, so that Q is a scalar multiple of $-u_0X^2+(-u_1+v_0)X+v_1 = F(-X)$ and has roots $-\rho_1$ and $-\rho_2$. By definition of L_Q the pair (h_0, h_1) must be in the dual lattice (scaled by Q), given by the basis $(u_0X + u_1, -1), (v_0X + v_1, -X)$.

For the assertions concerning conditions $(*)$ and $(**)$, assume that $\rho_1, \rho_2 \in L \setminus K$ and that

$$\rho_1^q + \alpha\rho_j + \delta = 0$$

holds for $j = 1$ or $j = 2$.

First consider the case $j = 2$, i.e., condition $(*)$. To show that $-\rho_i$, $i = 1, 2$, are roots of $h_1X^q - h_0$ it is sufficient to show this for the basis of the dual lattice of L_Q given above. For $(u_0X + u_1, -1)$ one computes

$$-(-\rho_1^q) - u_0(-\rho_1) - u_1 = \rho_1^q - \alpha\rho_1 - \beta + \delta = -\alpha\rho_2 - \alpha\rho_1 - \beta = 0,$$

and for $(v_0X + v_1, -X)$ one obtains

$$-(-\rho_1)(-\rho_1^q) - v_0(-\rho_1) - v_1 = (-\rho_1^q - \delta)\rho_1 - \gamma = \alpha\rho_1\rho_2 - \gamma = 0.$$

Therefore $h_1X^q - h_0$ is divisible by Q, which is then a trap of level 0.

In the case $j = 1$ an analogous calculation shows that $-\rho_i^q$, $i = 1, 2$, are roots of $h_1X^{q^{kd+1}} - h_0$, namely for $(u_0X + u_1, -1)$ one has

$$-(-\rho_2^{q^{kd+1}}) - u_0(-\rho_2) - u_1 = \rho_1^q - \alpha\rho_2 - \beta + \delta = -\alpha\rho_1 - \alpha\rho_2 - \beta = 0$$

and for $(v_0X + v_1, -X)$ one gets

$$-(-\rho_2)(-\rho_2^{q^{kd+1}}) - v_0(-\rho_2) - v_1 = (-\rho_1^q - \delta)\rho_2 - \gamma = \alpha\rho_1\rho_2 - \gamma = 0$$

Therefore $h_1X^{q^{kd+1}} - h_0$ is divisible by Q, which is then a trap of level kd. This finishes the proof of Proposition 2.

Regarding Proposition 3, note that a solution (a, B) gives rise to the polynomial $\mathcal{P}_a = a(u_0X + (Y + u_1)) + ((Y + v_0)X + v_1)$. If, for $j = 1$ or $j = 2$, ρ is a root of $\mathcal{P}_a \bmod f_j$ for two different values of a, then ρ is a root of $u_0X + (Y + u_1) \bmod f_j$ and of $(Y + v_0)X + v_1 \bmod f_j$. Since

$$-X(u_0X+(Y+u_1))+(Y+v_0)X+v_1 = -u_0X^2+(-u_1+v_0)X+v_1 = F(-X),$$

which equals Q up to a scalar, it follows that ρ is also a root of Q. Furthermore, in the case $j = 1$ the polynomial $\mathcal{P}_a \bmod f_1$ splits completely, so that $\rho \in K$, contradicting the irreducibility of Q. This completes the proof of Proposition 3.

6 Conclusion

We have proposed the first rigorous randomised quasi-polynomial time algorithm for solving the DLP in infinitely many finite fields of any fixed characteristic. Interestingly, our algorithm does not rely on the notion of smoothness. Furthermore, subject to a conjecture on the existence of irreducibles of a particular form, our algorithm applies to all extension fields of any fixed characteristic. Resolving this conjecture is therefore an important open problem.

Other questions worthy of future consideration include whether or not there exists a polynomial time algorithm (either rigorous or heuristic) for the DLP in fixed characteristic fields, or even harder, what is the true complexity of the DLP in the fixed characteristic case? Note that a result of F.R.K. Chung implies that for fields of our form there is no quasi-polynomial lower bound on the number of linear elements that must be multiplied in order to represent an arbitrary field element [9, Thm. 8]; indeed a very small (polynomial) number suffices. Hence there is no representational barrier to obtaining a polynomial time algorithm, when the factor base consists of linear elements. Another important question is can the recent ideas be generalised to prime field DLPs? If this were possible, then it is highly likely that such ideas could be transferred to the notorious integer factorisation problem.

Acknowledgements

The authors are indebted to Claus Diem for explaining how one can obviate the need to compute the logarithms of the factor base elements, and wish to thank him also for some enlightening discussions.

References

1. Leonard M. Adleman. A subexponential algorithm for the discrete logarithm problem with applications to cryptography. In *Proceedings of the 20th Annual Symposium on Foundations of Computer Science*, SFCS '79, pages 55–60, Washington, DC, USA, 1979. IEEE Computer Society.
2. Leonard M. Adleman. The function field sieve. In Leonard M. Adleman and Ming-Deh Huang, editors, *Algorithmic Number Theory*, volume 877 of *Lecture Notes in Computer Science*, pages 108–121. Springer Berlin Heidelberg, 1994.
3. Leonard M. Adleman and Ming-Deh A. Huang. Function field sieve method for discrete logarithms over finite fields. *Inform. and Comput.*, 151(1-2):5–16, 1999.
4. Yves Aubry and Marc Perret. A Weil theorem for singular curves. In *Arithmetic, geometry and coding theory (Luminy, 1993)*, pages 1–7. de Gruyter, Berlin, 1996.
5. Razvan Barbulescu, Pierrick Gaudry, Antoine Joux, and Emmanuel Thomé. A heuristic quasi-polynomial algorithm for discrete logarithm in finite fields of small characteristic. In *Advances in Cryptology—EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 1–16. Springer, 2014.
6. Antonia W. Bluher. On $x^{q+1} + ax + b$. *Finite Fields and Their Applications*, 10(3):285–305, 2004.

7. Peter J. Cameron, Gholam R. Omidi, and Behruz Tayfeh-Rezaie. 3-designs from PGL$(2, q)$. *Electron. J. Combin.*, 13(1):Research Paper 50, 11, 2006.
8. Qi Cheng, Daqing Wan, and Jincheng Zhuang. Traps to the bgjt-algorithm for discrete logarithms. *LMS Journal of Computation and Mathematics*, 17:218–229, 2014.
9. Fan-Rong K. Chung. Diameters and eigenvalues. *J. Amer. Math. Soc.*, 2(2):187–196, 1989.
10. Don Coppersmith. Evaluating logarithms in GF(2^n). In *Proceedings of the Sixteenth Annual ACM Symposium on Theory of Computing*, STOC '84, pages 201–207, New York, NY, USA, 1984. ACM.
11. Don Coppersmith. Fast evaluation of logarithms in fields of characteristic two. *IEEE Trans. Inf. Theor.*, 30(4):587–594, 1984.
12. Nicolaas G. De Bruijn. On the number of positive integers $\leq x$ and free of prime factors $> y$. *Indagationes Mathematicae*, 13:50–60, 1951.
13. Nicolaas G. De Bruijn. On the number of positive integers $\leq x$ and free of prime factors $> y$, II. *Indagationes Mathematicae*, 28:239–247, 1966.
14. Karl Dickman. On the frequency of numbers containing prime factors of a certain relative magnitude. *Arkiv för Matematik, Astonomi och Fysik*, 22A (10):1–14, 1930.
15. Leonard E. Dickson. *Linear groups: With an exposition of the Galois field theory.* Teubner, Leipzig, 1901.
16. Claus Diem. On the discrete logarithm problem in elliptic curves. *Compositio Mathematica*, 147:75–104, 1 2011.
17. Whitfield Diffie and Martin E. Hellman. New directions in cryptography. *IEEE Trans. Inf. Theor.*, 22(6):644–654, September 2006.
18. Andreas Enge and Pierrick Gaudry. A general framework for subexponential discrete logarithm algorithms. *Acta Arithmetica*, 102:83–103, 2002.
19. Faruk Göloğlu, Robert Granger, Gary McGuire, and Jens Zumbrägel. On the function field sieve and the impact of higher splitting probabilities. Available from `eprint.iacr.org/2013/074`, 15th Feb 2013.
20. Faruk Göloğlu, Robert Granger, Gary McGuire, and Jens Zumbrägel. On the function field sieve and the impact of higher splitting probabilities. In Ran Canetti and Juan A. Garay, editors, *Advances in Cryptology—CRYPTO 2013*, volume 8043 of *LNCS*, pages 109–128. Springer, 2013.
21. Faruk Göloğlu, Robert Granger, Gary McGuire, and Jens Zumbrägel. Solving a 6120-bit DLP on a desktop computer. In *Selected Areas in Cryptography—SAC 2013*, volume 8282 of *LNCS*, pages 136–152. Springer, 2014.
22. Faruk Göloğlu, Robert Granger, Gary McGuire, and Jens Zumbrägel. Discrete Logarithms in $GF(2^{1971})$. NMBRTHRY list, 19/2/2013.
23. Faruk Göloğlu, Robert Granger, Gary McGuire, and Jens Zumbrägel. Discrete Logarithms in $GF(2^{6120})$. NMBRTHRY list, 11/4/2013.
24. Robert Granger, Thorsten Kleinjung, and Jens Zumbrägel. Breaking '128-bit secure' supersingular binary curves - (or how to solve discrete logarithms in $\mathbb{F}_{2^{4\cdot 1223}}$ and $\mathbb{F}_{2^{12\cdot 367}}$). In *Advances in Cryptology—CRYPTO 2014*, volume 8617 of *LNCS*, pages 126–145. Springer, 2014.
25. Robert Granger, Thorsten Kleinjung, and Jens Zumbrägel. Discrete logarithms in the Jacobian of a genus 2 supersingular curve over $GF(2^{367})$. NMBRTHRY list, 30/1/2014.
26. Robert Granger, Thorsten Kleinjung, and Jens Zumbrägel. Discrete Logarithms in $GF(2^{9234})$. NMBRTHRY list, 31/1/2014.
27. Tor Helleseth and Alexander Kholosha. $x^{2^l+1} + x + a$ and related affine polynomials over GF(2^k). *Cryptogr. Commun.*, 2(1):85–109, 2010.
28. Antoine Joux. Faster index calculus for the medium prime case. application to 1175-bit and 1425-bit finite fields. In Thomas Johansson and Phong Q. Nguyen, editors, *Advances in Cryptology—EUROCRYPT 2013*, volume 7881 of *LNCS*, pages 177–193. Springer, 2013.

29. Antoine Joux. A new index calculus algorithm with complexity $L(1/4 + o(1))$ in small characteristic. In Tanja Lange, Kristin Lauter, and Petr Lisoněk, editors, *Selected Areas in Cryptography—SAC 2013*, volume 8282 of *LNCS*, pages 355–379. Springer, 2014.
30. Antoine Joux. A new index calculus algorithm with complexity $L(1/4 + o(1))$ in very small characteristic. Available from `eprint.iacr.org/2013/095`, 20th Feb 2013.
31. Antoine Joux. Discrete Logarithms in $GF(2^{1778})$. NMBRTHRY list, 11/2/2013.
32. Antoine Joux. Discrete Logarithms in $GF(2^{4080})$. NMBRTHRY list, 22/3/2013.
33. Antoine Joux. Discrete Logarithms in $GF(2^{6168})$. NMBRTHRY list, 21/5/2013.
34. Antoine Joux and Reynald Lercier. The function field sieve is quite special. In Claus Fieker and David R. Kohel, editors, *Algorithmic number theory (Sydney, 2002)*, volume 2369 of *LNCS*, pages 431–445. Springer, 2002.
35. Antoine Joux and Reynald Lercier. The function field sieve in the medium prime case. In Serge Vaudenay, editor, *Advances in Cryptology—EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 254–270. Springer, 2006.
36. Ravindran Kannan and Achim Bachem. Polynomial algorithms for computing the Smith and Hermite normal forms of an integer matrix. *SIAM J. Comput.*, 8(4):499–507, 1979.
37. Thorsten Kleinjung. Discrete logarithms in $\mathrm{GF}(2^{1279})$. NMBRTHRY list, 17/10/2014.
38. Maurice Kraitchik. *Théorie des nombres*, volume 1. Paris: Gauthier-Villars, 1922.
39. Maurice Kraitchik. *Recherches sur la théorie des nombres*, volume 1. Paris: Gauthier-Villars, 1924.
40. Arjen K. Lenstra and Hendrik W. Lenstra, Jr., editors. *The development of the number field sieve*, volume 1554 of *Lecture Notes in Mathematics*. Springer, Heidelberg, 1993.
41. Hendrik W. Lenstra, Jr. Finding isomorphisms between finite fields. *Math. Comp.*, 56(193):329–347, 1991.
42. Hendrik W. Lenstra, Jr. and Carl Pomerance. A rigorous time bound for factoring integers. *J. Amer. Math. Soc.*, 5(3):483–516, 1992.
43. Ralph C. Merkle. *Secrecy, Authentication, and Public Key Systems*. PhD thesis, Stanford University, Stanford, CA, USA, 1979.
44. Cecile Pierrot and Antoine Joux. Discrete logarithm record in characteristic 3, $\mathrm{GF}(3^{5\cdot 479})$ a 3796-bit field. NMBRTHRY list, 15/9/2014.
45. Stephen C. Pohlig and Martin E. Hellman. An improved algorithm for computing logarithms over gf(p) and its cryptographic significance (corresp.). *IEEE Trans. Inf. Theory*, 24(1):106–110, 1978.
46. John M. Pollard. Monte Carlo Methods for Index Computation (mod p). *Mathematics of Computation*, 32:918–924, 1978.
47. Carl Pomerance. Fast, rigorous factorization and discrete logarithm algorithms. In *Discrete algorithms and complexity (Kyoto, 1986)*, volume 15 of *Perspect. Comput.*, pages 119–143. Academic Press, Boston, MA, 1987.
48. J. Barkley Rosser and Lowell Schoenfeld. Approximate formulas for some functions of prime numbers. *Illinois J. Math.*, 6:64–94, 1962.
49. Naoyuki Shinohara, Takeshi Shimoyama, Takuya Hayashi, and Tsuyoshi Takagi. Key length estimation of pairing-based cryptosystems using η_t pairing. In Mark D. Ryan, Ben Smyth, and Guilin Wang, editors, *Information Security Practice and Experience*, volume 7232 of *Lecture Notes in Computer Science*, pages 228–244. Springer Berlin Heidelberg, 2012.
50. Carl F. Gauss (translated by Arthur A. Clarke). *Disquisitiones Arithmeticae*. Yale University Press, 1965.
51. Brigitte Vallée. Generation of elements with small modular squares and provably fast integer factoring algorithms. *Math. Comp.*, 56(194):823–849, 1991.
52. Daqing Wan. Generators and irreducible polynomials over finite fields. *Mathematics of Computation*, 66:1195–1212, 1997.
53. A. E. Western and J. C. P. Miller. *Tables of indices and primitive roots*. Royal Society Mathematical Tables, vol. 9, Cambridge University Press, 1968.

Hiermit versichere ich, dass ich die vorliegende Arbeit ohne unzulässige Hilfe Dritter und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe; die aus fremden Quellen direkt oder inhaltlich übernommenen Stellen sind als solche kenntlich gemacht.

Bei allen eingereichten gemeinschaftlichen Arbeiten erstreckt sich meine Mitarbeit auf sämtliche Aspekte, einschließlich der Ideenfindung, der Ausarbeitung und der Implementierung.

Es wurden zuvor keine Habilitationsvorhaben unternommen.

Ich erkenne die Habilitationsordnung der Fakultät für Mathematik und Naturwissenschaften der Technischen Universität Dresden vom 12. Dezember 2010, in der geänderten Fassung mit Gültigkeit vom 19. Februar 2014, an.

Dresden, den 2. Juni 2015