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

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

Faruk Göloğlu, Robert Granger, Gary McGuire and Jens Zumbrägel

Claude Shannon Institute, 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 $L_{q^n}(1/3, 2/3)$ for computing arbitrary logarithms, but also in an heuristic *polynomial time* algorithm for finding the discrete logarithms of degree one elements. To illustrate the efficiency of the method, we have successfully solved the DLP in the finite field with 2^{1971} elements.

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 weakens (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 [16]; Optimal Extension Fields [2] and Weil descent for elliptic curves [7]; high-compression algebraic tori [18] and specialised index calculus [8]; quasi-cyclic or dyadic McEliece variants [17] and Grobner basis attacks [5], and more recently elliptic curves over binary fields [6], to a name just a few. In practice therefore, one should be wary of any additional structure, which may potentially weaken a system.

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 medium-sized base field. In 2006 Joux and Lercier designed a particularly efficient variation of the Function Field Sieve (FFS) algorithm for computing discrete logarithms [14], which at the time possessed the fastest asymptotic complexity of all known discrete logarithm algorithms for appropriately balanced q and n, namely $L_{q^n}(1/3, 3^{1/3})$, where

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

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

In 2012, Joux proposed a more efficient method of obtaining relations, dubbed 'pinpointing', in which relations can be generated efficiently without the need for sieving [11], which is advantageous when sieving is the dominant phase, rather than the linear algebra (or individual logarithm phase). With this technique the overall complexity of solving the DLP can be as low as $L_{q^n}(1/3, 2/3^{2/3}) = L_{q^n}(1/3, 0.961)$, subject to suitable modular conditions between q and n. To demonstrate the approach, Joux solved the DLP in two cases: an 1175-bit field and a 1425-bit field, setting records for medium-sized base fields, in this case prime fields.

* 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 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 the medium-sized base field FFS over binary fields this leads to the dramatic conclusion that the logarithms of degree one elements in some binary fields can be solved in *polynomial time*. As far as we are aware, no other algorithm for the collecting relations and 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 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 solve individual logarithms, which for a range of binary fields gives the fastest $L_{q^n}(1/3)$ algorithms to date.

We emphasise that our work is completely independent of [11]. Firstly, it does not use pinpointing, and secondly, it is applicable to far many more binary fields. In particular we impose no conditions on the extension degree n of the base field (except in one case).

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 [14]. 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 of degrees d_1 and d_2 respectively, and define two bivariate polynomials

$$f_1(X,Y) = X - g_1(Y), \quad f_2(X,Y) = -g_2(X) + Y.$$

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:

$$\mathbb{F}_q[X]/F(X) \text{ and } \mathbb{F}_q[Y]/G(Y),$$

where $G(Y)$ is the corresponding degree n irreducible factor of $-g_2(g_1(Y)) + Y$ over $\mathbb{F}_q$. 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 the respective representations of the finite field $\mathbb{F}_{q^n}$:

$$Xg_2(X) + ag_2(X) + bX + c = Yg_1(Y) + aY + bg_1(Y) + c. \tag{1}$$

The factor base consists simply of the degree one elements of $\mathbb{F}_q[X]/F(X)$ and $\mathbb{F}_q[Y]/G(Y)$; then for every triple (a, b, c) for which both sides of (1) splits over $\mathbb{F}_q$ — ie., 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 faster by 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 detailed 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 [14], 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 [14] as presented in the previous section, and detail some resulting consequences. From now on let $\mathbb{F}_q$ denote the finite field with 2^l elements.

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

• Whenever $k \mid l$ and $3k \leq l$, 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 2 special-$\mathfrak{q}$ to degree 3, or even degree 4 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 explicate these advantages in more detail. In addition to the above two observations, using non-prime base fields induces extra automorphisms of the factor base, 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

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 [10] for further details.

We have the following theorem due to Bluher [3] (and refined in the binary case by Helleseth and Kholosha [10]), which counts the number of $B \in \mathbb{F}_q$ for which f_B splits over $\mathbb{F}_q$.

Theorem 1. *[10, 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*

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

Theorem 1 of [10] 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 for $d = k$ and 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.

3.2 Relation generation

If one naively takes random triples (a, b, c) and tests whether both sides of (1) split over $\mathbb{F}_q$, the expected cost of generating q relations is $\widetilde{O}(q \cdot 2^{3k} \cdot (d_1 + 1)!)$ $\mathbb{F}_q$-operations. In order for there to be sufficiently many relations, we must have

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

Since we insist that $l \geq 3k$, having $q > (d_1+1)!$ suffices. As we are free to set d_1 to be as small an integer as possible, such that $g_1(Y)^{2^k} + Y$ contains a degree n irreducible factor, this condition can be assumed to be always satisfied (in practice it seems that $d_1 = 3$ may suffice to produce an irreducible of degree n, for q sufficiently large).

As naive as this approach is, it is already sufficient to provide an heuristic *polynomial time* algorithm for solving the discrete logarithms of all degree one elements of $\mathbb{F}_{q^n}$, for constant d_1 as $l \to \infty$ and $l = k' \cdot k$ with $k' \geq 3$ a constant. In particular, for $n \approx 2^k \cdot d_1 = 2^{l/k'} \cdot d_1$, 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 sieving is $\widetilde{O}(q \cdot 2^{3k}) = \widetilde{O}(2^{l(1+3/k')}) = \widetilde{O}(\log^{k'+3} Q)$, whereas the cost of sparse linear algebra, using Lanczos' algorithm [15] for instance, is the product of the row weight and the square of number of variables, namely

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

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 $Y \cdot g_1(Y) + aY + bg_1(Y) + c$ 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 $\widetilde{O}(\log^{2k'+1} Q)$.*

By exploiting the above transformation of (2) to (3), a simple improvement upon the naive approach stated at the beginning of this subsection is as follows. We begin by computing the list L_B of all B for which (2) splits. Indeed, the proof of Prop. 5 in [10] gives an explicit parameterisation of all such B: for $u \in G = \mathbb{F}_{2^l} \setminus \mathbb{F}_{2^{2k}}$, we have

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

Computing this list costs $\widetilde{O}(q)$ $\mathbb{F}_q$-operations, avoiding smoothness tests if preferable. Then for any fixed (a, b), for each $B \in L_B$ we compute via (3), the corresponding (unique) $c \in \mathbb{F}_q$ and check whether

$$Y g_1(Y) + bg_1(Y) + aY + c \tag{4}$$

splits over $\mathbb{F}_q$. Assuming that this occurs with probability $1/(d_1+1)!$, we expect to obtain about

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

relations. Since we need q relations, we expect to require about $(d_1+1)! \cdot 2^{3k}$ 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 only $\tilde{O}(q \cdot (d_1+1)!)$. Indeed, for fixed $d_1, k' \geq 3$ and $l = k' \cdot k$, as $l \to \infty$, the relation generation time is $\widetilde{O}(q)$.

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 á la [14, 4], 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 $z = h \cdot g^i$ 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 [14], as follows.

Let $p(X)$ be a degree d irreducible 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:

$$L_{\overline{p}(Y)} = \{(u_0(Y), u_1(Y)) \in \mathbb{F}_q[Y]^2 : u_0(Y)g_1(Y) + u_1(Y) \equiv 0 \pmod{\overline{p}(Y)}\}.$$

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

$$\mathrm{RHS}(Y) = (u_0(Y)r(Y) + v_0(Y)s(Y))g_1(Y) + (u_1(Y)r(Y) + v_1(Y)s(Y)).$$

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

$$\mathrm{LHS}(X) = (u_0(X^{2^k})r(X^{2^k}) + v_0(X^{2^k})s(X^{2^k}))X + (u_1(X^{2^k})r(X^{2^k}) + v_1(X^{2^k})s(X^{2^k})).$$

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 [14], we assume there is a parameter α such that:

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

The three stages to consider are sieving, 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 now consider three cases.

4.1 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 $L_Q(1/3, 2/3^{2/3}) \approx L_Q(1/3, 0.961)$.*

This is the simplest case we present; however for the sake of completeness and ease of exposition of the latter cases, we explicitly tailor the derivation presented in §3.2. By Theorem 1 the probability of (2) being smooth is $1/2^{3k}$, whereas the probability of (4) being smooth is approximately $1/\sqrt{n}!$. Using the standard approximation $\log n! \approx n \log n$, the log of the probability P of both sides being smooth is:

$$\log P = -\log 2^{3k} - \sqrt{n}\log\sqrt{n} = -\frac{3}{2}\log n - \frac{1}{2}\sqrt{n}\log n \approx -\frac{1}{2}\sqrt{n}\log n.$$

The size of the sieving space is q^3, and since we require q relations we must have:

$$q^3 \cdot P \geq q, \quad \text{or} \quad 2 \cdot \log q \geq \frac{1}{2}\sqrt{n}\log n.$$

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

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

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

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

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

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

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

$$\frac{1}{3\mu\sqrt{\alpha}} \leq \max\{\alpha + \frac{1}{3\sqrt{\alpha}}, 2\alpha\}. \tag{7}$$

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.

The natural bottleneck in the descent is for degree 2 special-$\mathfrak{q}$, therefore let $\overline{p}(Y)$ be a degree 2 irreducible to be eliminated. A reduced basis for the lattice $L_{\overline{p}(Y)}$ can always be found with degrees $(1,1),(0,1)$; in fact, it can even be of the form

$$(u_{01}Y, u_{11}Y + u_{10}), (1, v_{11}Y + v_{10}).$$

Hence with r normalised to be 1 and $s \in \mathbb{F}_q$, we have

$$(u_{01}Y + s)g_1(Y) + (u_{11}Y + u_{10}) + (v_{11}Y + v_{10})s \in L_{\overline{p}(Y)},$$

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

$$(u_{01}X^{2^k} + s)X + (u_{11}X^{2^k} + u_{10}) + (v_{11}X^{2^k} + v_{10})s,$$

which is thus of the form $X^{2^k+1} + aX^{2^k} + bX + c$, and so by Theorem 1, splits over $\mathbb{F}_q$ with probability 2^{-3k}. We therefore need to ensure that there are sufficiently many $s \in \mathbb{F}_q$ for this to occur, i.e., $q > 2^{3k} \cdot (d_1 - 1)!$, or equivalently,

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

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 for the lattice $L_{\overline{p}(Y)}$ can always be found with degrees $(1,1),(1,2)$; in fact, it can even be of the form

$$(u_{01}Y + u_{00}, u_{11}Y + u_{10}), (v_{10}Y, v_{21}Y^2 + v_{11}Y + v_{10}).$$

Hence with r now allowed to be monic of degree one and $s \in \mathbb{F}_q$, we have

$$((u_{01}Y + u_{00})r(Y) + (v_{10} \cdot sY))g_1(Y) + (u_{11}Y + u_{10})r(Y) + (v_{21}Y^2 + v_{11}Y + v_{10})s \in L_{\overline{p}(Y)}.$$

The corresponding polynomial LHS(X) is

$$((u_{01}X^{2^k} + u_{00})r(X^{2^k}) + (v_{10} \cdot sX^{2^k}))X + (u_{11}X^{2^k} + u_{10})r(X^{2^k}) + (v_{21}X^{2^{k+1}} + v_{11}X^{2^k} + v_{10})s \quad (8)$$

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 [14]. Therefore the log of the probability P of both sides being 2-smooth is:

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

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

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

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

$$L_Q(1/3, 2/3^{2/3}) \approx L_Q(1/3, 0.961),$$

which is precisely the complexity that Joux obtains using advanced pinpointing for Kummer extensions [11]. However, the crucial difference between what we have presented so far and Joux's work is that we make no assumptions on the form of n, and we do not need a specific pinpointing strategy, even if our more general approach may be viewed as a form of pinpointing.

Furthermore, for this α (7) implies that $\mu > 1/2$. 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 the decent degrees actually descend to one.

Incidentally, in this case, if for degree 2 special-$\mathfrak{q}$ we allow s to be degree 1, we obtain the condition $2\alpha > \frac{2}{3\sqrt{\alpha}}$, giving exactly the same optimal α. In the following case this no longer holds, and we obtain a dramatic improvement.

4.2 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 $L_Q(1/3, (2/3)^{2/3}) \approx L_Q(1/3, 0.761)$.*

Observe that this is the square-root of the complexity of the original FFS [1, 12], for which $c = (32/9)^{1/3}$. 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 the log of the probability P of both sides being smooth is:

$$\log P = -\log n^{3(1-\beta)} - n^\beta \log n^\beta \approx -\beta n^\beta \log n.$$

By (5) we have

$$-\beta n^\beta \log n = -\frac{2\beta}{3\alpha^\beta} \cdot \left(\frac{\log Q}{\log\log Q}\right)^{2\beta/3} \cdot (\log\log Q) = -\frac{2\beta}{3\alpha^\beta} \cdot (\log Q)^{2\beta/3}(\log\log Q)^{1-2\beta/3}.$$

Hence the expected time of the relation generation is

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

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, and since we require q relations we must have:

$$q^3 \cdot P \geq q, \quad \text{or} \quad L_Q(1/3, 2\alpha) > L_Q\left(2\beta/3, \frac{2\beta}{3\alpha^\beta}\right),$$

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

$$q > 2^{3k} \cdot (d_1 - 1)! = n^{3(1-\beta)} \cdot L_Q\left(2\beta/3, \frac{2\beta}{3\alpha^\beta}\right),$$

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 now 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 now allowed to be monic of degree one and $s \in \mathbb{F}_q$, our new expressions become

$$((u_{01}Y + u_{00})r(Y) + (v_{10} \cdot sY)) \cdot g_1(Y)^{2^a} + (u_{11}Y + u_{10})r(Y) + (v_{21}Y^2 + v_{11}Y + v_{10})s \in L_{\overline{p}(Y)}.$$

The corresponding polynomial LHS(X') is

$$\begin{aligned}&((u_{01}X'^{2^{k-a}} + u_{00})r(X'^{2^{k-a}}) + (v_{10} \cdot sX'^{2^{k-a}}))X' + (u_{11}X'^{2^{k-a}} + u_{10})r(X'^{2^{k-a}}) \\ &+ (v_{21}X'^{2^{k-a+1}} + v_{11}X'^{2^{k-a}} + v_{10})s.\end{aligned}$$

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

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

Although we must take the nearest integer to this a, asymptotically we may assume it is exact. Such a choice ensures that both degrees are $\sqrt{2d_1} \cdot 2^{k/2} = \sqrt{2} \cdot \sqrt{n}$. Therefore the log of the probability P of both sides being 2-smooth is:

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

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

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

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

$$L_Q(1/3, 2\alpha) = L_Q(1/3, (2/3)^{2/3}) \approx L_Q(1/3, 0.761).$$

4.3 Case 3: $n = 2^k - 1$

In this section we will show the following:

Heuristic Result 2(iii): *Let $q = 2^l$, let $k(k-1) \mid l$ and let n be such that (5) holds. Then for $n = 2^k - 1$ the DLP can be solved with complexity $L_Q(1/3, 2/3)$.*

One case where the assumption in Heuristic Result 1 that the splitting probability is $1/(d_1+1)!$ is clearly false, is when $g_1(Y) = Y^{2^{k_1}}$, since this is the basis of Theorem 1. However, the natural idea to exploit Theorem 1 in this way only works when $d_2 = 2^{k-1}$ and $d_1 = 2$, as otherwise the intersection of the factors on each side is almost empty. When $d_1 = 2$, (4) covers all degree 3 polynomials over $\mathbb{F}_q$ and so the set of splitting roots are randomly distributed in $\mathbb{F}_q$. This ensures that we can perform the linear algebra elimination, with the only condition being $(k-1) \mid l$.

We are therefore looking for polynomials $Y^{2^k} + \gamma Y$ with irreducible factors of about the same degree. If $k \mid l$ as well, then $f_\gamma = Y^{2^k-1} + \gamma$ is irreducible whenever γ has no root of prime order $p \mid (2^k - 1)$. When reducible, the polynomial f_γ may have other fairly high-degree factors (up to $(2^k-1)/3$), but then our complexity improvement is less pronounced when using the embedding into the ring $\mathbb{F}_q[Y]/(Y^{2^k} + \gamma Y)$.

Hence let $l = k(k-1)$ and let $q = 2^l$. The case where l is a greater multiple of $k(k-1)$ merely increases the cost of the sieving and linear algebra, and can be dealt with similarly. We have $Y = X^{2^{k-1}}$ and $X = Y^2/\gamma$.

By a similar argument to Case 2, we have $c_1 = \alpha$ and $c_2 = 2\alpha$ for any $\alpha > 0$, and there are sufficiently many relations.

For the descent, the complexity of finding an initial $\mu\sqrt{n}$-smooth element is $L_Q(1/3, \frac{1}{3\mu\sqrt{\alpha}})$. For the same reason as in Case 2, degree 2 special-$\mathfrak{q}$ are easy to eliminate. Let $\overline{p}(Y)$ be a degree 3 special-$\mathfrak{q}$. Then as before, a reduced basis for $L_{\overline{p}(Y)}$ can always be found with degrees $(1,1), (1,2)$; in fact, it can even be of the form

$$(Y + u_{00}, u_{11}Y + u_{10}), (Y, v_{12}Y^2 + v_{11}Y + v_{10}).$$

Hence with $r = 1$ and $s \in \mathbb{F}_q$ we have

$$((u_{00} + s)Y)Y^2 + (u_{11}Y + u_{10}) + (v_{12}Y^2 + v_{11}Y + v_{10})s \in L_{\overline{p}(Y)},$$

which is equal to $\mathfrak{q}$ in $\mathbb{F}_q[Y]/\mathbb{F}_q^\times$. The corresponding polynomial on the LHS is

$$((u_{00} + s)X^{2^{k-1}})X + (u_{11}X^{2^{k-1}} + u_{10}) + (v_{12}\beta X + v_{11}X^{2^{k-1}} + v_{10})s,$$

where the Y^2 term is replaced by $X^{2^k} = \beta X$. Hence this is of the form (2), and again, these are easy to eliminate. Hence degree 4 special-$\mathfrak{q}$ are the bottleneck, as unfortunately LHS(X) does not have the form required to apply Theorem 1.

As in Case 2, we therefore take a slightly different approach. Since $Y = X^{2^{k-1}}$ and $X = Y^2/\gamma$, for $1 < a \leq k-1$ we can set $X' = Y^{2^{k-a}}/\beta$ and $Y = X'^{2^a}$. The degrees of the reduced lattice basis $(u_0, u_1), (v_0, v_1)$ after Gaussian reduction are $(2,2), (1,2)$. Hence RHS(Y) has the form

$$((u_{02}Y^2 + u_{01}Y + u_{00}) + (v_{01}Y + v_{00})s(Y))Y^{2^{k-a}}/\gamma + (u_{12}Y^2 u_{11}Y + u_{10}) + (v_{12}Y^2 + v_{11}Y + v_{10})s(Y),$$

while the LHS will have the form

$$\begin{aligned}&(u_{02}X'^{2^{a+1}} + u_{01}X'^{2^a} + u_{00}) + (v_{01}X'^{2^a} + v_{00})s(X'^{2^a}))X \\ &+ (u_{12}X'^{2^{a+1}} u_{11}X'^{2^a} + u_{10}) + (v_{12}X'^{2^{a+1}} + v_{11}X'^{2^a} + v_{10})s(X'^{2^a}).\end{aligned}$$

When s is a linear polynomial, the respective degrees are $2^{k-a}-2$ and $3\cdot 2^a$. Balancing these degrees as before, asymptotically with k they become about $\sqrt{3}\cdot 2^{k/2}$. Hence the logarithm of the probability of each side being 3-smooth is

$$-\frac{\sqrt{n}}{\sqrt{3}}\cdot\log\frac{\sqrt{n}}{\sqrt{3}}\approx-\frac{\sqrt{n}}{2\sqrt{3}}\cdot\log n.$$

The probability of both sides being 3-smooth is

$$\exp\Big(-\frac{\sqrt{n}}{\sqrt{3}}\cdot\log n\Big)=L_Q\Big(1/3,-\frac{2}{\sqrt{27}\sqrt{\alpha}}\Big).$$

Since there are only q^2 such s ensuring these degrees, we must have

$$2\alpha>\frac{2}{\sqrt{27}\sqrt{\alpha}},\quad\text{or}\quad\alpha>1/3.$$

For this α, we must have $\mu>\frac{1}{6\alpha^{3/2}}=\sqrt{3}/2\approx 0.866$. For an upper bound, we have a lot more freedom than before, since for any degree d special-$\mathfrak{q}$, the degree of both sides can balanced to approximately $2^{k/2}\cdot\sqrt{d/2+1}$. Hence setting $m=\mu\sqrt{n}$ the degrees of both sides is $2^{k/2}\cdot\sqrt{\frac{\mu\cdot 2^{k/2}}{2}+1}\approx 2^{3k/4}\sqrt{\frac{\mu}{2}}$. Since this should be less than 2^k, any $\mu<2^{k/2}$ will do. In practice of course, a smaller μ reduces the number of special q's to be eliminated in total. Hence the total complexity of the algorithm is

$$L_Q(1/3,2/3).$$

We summarise all three cases in the following:

Heuristic Result 2. *Let $q=2^l$, $k\mid l$ and n be such that (5) holds. Then we have:*

(i) For $n\approx 2^k\cdot d_1$ and $2^k\approx d_1$ the DLP can be solved with complexity $L_Q(1/3,2/3^{2/3})\approx L_Q(1/3,0.961)$;
(ii) For $n\approx 2^k\cdot d_1$ and $2^k\gg d_1$ the DLP can be solved with complexity $L_Q(1/3,(2/3)^{2/3})\approx L_Q(1/3,0.761)$.
(iii) For $n=2^k-1$ and $(k-1)\mid l$, the DLP can be solved with complexity $L_Q(1/3,2/3)$.

5 Application to $\mathbb{F}_{2^{1971}}$

In this section we give details of our implementation and report our results. Let $\mathbb{F}_q=\mathbb{F}_{2^{27}}=\mathbb{F}_2[t]/(t^{27}+t^5+t^2+t+1)$ and let $\mathbb{F}_{q^{73}}=\mathbb{F}_q[X]/(X^{73}+t)$ be the finite field of order 2^{1971}. In this field it holds that $Y=X^8$ and $X=t/Y^9$.

We use the Kummer extension idea of [14, 11] to reduce the size of the factor base from 2^{27} to $\approx 2^{27}/73$. As stated in §3 we can use a larger group than just the Galois group of $\mathbb{F}_{q^{73}}/\mathbb{F}_q$ to reduce the number of variables. 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 612864 orbits of full size 219, seven of size 73, and one of size 1, resulting in $N=612872$ orbits, which gives the number of factor base variables.

For relation generation, we began by using Joux's pinpointing method from [11]; however we then developed a simple new pinpointing method which arises from the technique in §3.2, which we will include in an updated version. We computed approximately $10\,N$ relations in about 14 core hours computation time. For simplicity, we keep only those relations with distinct

factors, as this ensures that each entry of the relation matrix is a power of two, as this ensures that all element multiplications in the matrix-vector products consist of cyclic rotations modulo $2^{1971} - 1$, due to our choice of $g_2(X)$ and the factor base reduction method.

After relation generation, we performed structured Gaussian elimination (SGE) (in a version based on [13]) 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 528812×527766 matrix of constant row weight 19. The running time here was about 10 minutes on a single core.

We then applied a parallel version of the Lanczos algorithm (see [15]) using OpenMP on an SGI Altix ICE 8200EX cluster using Intel (Westmere) Xeon E5650 hex-core processors and GNU Multi-Precision library [9], taking 2220 core hours in total. We took as generator $g = X + 1 \in \mathbb{F}_{2^{1971}}^{\times}$ and a target element 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 obtained the following partial factorisation of $2^{1971} - 1$:

$$C338 \cdot 7 \cdot 73^2 \cdot 439 \cdot 3943 \cdot 262657 \cdot 2298041 \cdot 10178663167 \cdot 27265714183 \cdot 9361973132609 \cdot$$
$$1406791071629857 \cdot 5271393791658529 \cdot 671165898617413417 \cdot 2762194134676763431 \cdot$$
$$4815314615204347717321 \cdot 421859275529837631474313737 19 \cdot$$
$$22068362846714807160397927912339216441 \cdot 781335393705318202869110024684359759405179097,$$

where $C338$ is a 338-digit composite. We solved the log in the subgroups of order the first eleven terms using linear search or Pollard rho, with the remaining 507-digit factor being the modulus for Lanczos' algorithm.

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, ie., all of their irreducible factors of degrees are at most 7. At each stage, we have two choices for how to sieve for that special-$\mathfrak{q}$; on the LHS or on the RHS, one of which may be much faster. Note that 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. And for theses special-$\mathfrak{q}$, we combined (8) and (3) so that for each $B \in \mathbb{F}_q$ for f_B splits, we compute the set of $s \in \mathbb{F}_q$ that satisfy

$$B = \frac{\left(\frac{s}{u_{01}} + \left(\frac{u_{11} + v_{11} s}{u_{01}} \right)^{2^k} \right)^{2^k + 1}}{\left(\frac{u_{11} + v_{11} s}{u_{01}} \cdot \frac{s}{u_{01}} + \frac{u_{10} + v_{10} s}{u_{01}} \right)^{2^k}}.$$

For each such s we check whether the RHS is 1 smooth as normal. This chops a factor of 2^9 from the elimination of each degree 2 special-$\mathfrak{q}$.

As one can see from Table 1, sieving on the Y-side is always the fastest. Choosing degree 7 special-$\mathfrak{q}$ means that the maximum cost of an elimination is $2^{25.1}$ occuring at the degree 3 special-$\mathfrak{q}$ eliminations.

Table 1. Individual logarithm data for $n = 1971$, with $Y = X^8$ and $X = t/Y^9$.

Setup			Special-Q on X-side			Special-Q on Y-side		
$\delta(Q)$	#trials	basis δ's	$\delta(X$-side)	$\delta(Y$-side)	#trials	$\delta(X$-side)	$\delta(Y$-side)	#trials
2	$2^{126.3}$	$(1,0),(0,1)$		N.A.		9	8	$2^{15.3}$
3	$2^{75.9}$	$(1,1),(0,2)$	6	19	$2^{27.8}$	17	7	$2^{25.1}$
4	$2^{51.3}$	$(2,1),(1,2)$	6	19	$2^{15.4}$	17	7	$2^{13.7}$
5	$2^{37.1}$	$(2,2),(1,3)$	5	28	$2^{17.6}$	25	6	$2^{15.0}$
6	$2^{28.1}$	$(3,2),(2,3)$	5	28	$2^{12.3}$	25	6	$2^{10.3}$
7	$2^{21.9}$	$(3,3),(2,4)$	4	37	$2^{14.7}$	33	5	$2^{12.1}$

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. Then we spent about 3 hours to perform the descent up to degree 3. At this point we were left with 103 special-$\mathfrak{q}$ of degree 3 (in comparison to about 500 special-$\mathfrak{q}$ of degree 3 in a naive implementation). These special-$\mathfrak{q}$ elements have been resolved on the same SGI Altix ICE 8200EX cluster in about 850 core hours, using Victor Shoup's Number Theory Library [19], resulting in 1140 special-Q elements of degree 2. These elements were subsequently resolved in 5 core hours, by which time the descent is finished.

Thus the running time for solving an instance of the discrete logarithm problem completely in the finite field $\mathbb{F}_{2^{1971}}$ is $14 + 2220 + 898 = 3132$ core hours in total. In particular, we have:

$$\begin{aligned}\log_g(X_\pi) = {} & 11992984215354106866091146371988855845186852755447163352368959007609021987957457840081811487759339446560383051978254174236023653588993736220077111736167826942310116340313\\ & 53555222808041139032152735559059010822822482400219287878207304028565280573096588688279004416835100344085961912427000601289864337521100022143802898875460611252245879711978\\ & 72750805846519623140437645739362938235417361611681082562778045965789270956115892417357940067473968434606299268294291957378226451182620783745349502502960139927453196489740\\ & 06524479548958327920827882768332440907342446643941097670216203953951337767311548343 9.\end{aligned}$$

6 Conclusions

We have presented and analysed a new variant of the medium-sized base field FFS, for binary fields. We have established new complexity results as well as the intriguing fact that the logarithm of degree one elements can be solved in polynomial time. We have also presented the results of an implementation of the method, setting a DLP record in the field $\mathbb{F}_{2^{1971}}$.

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 extend 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. Don Coppersmith. Fast evaluation of logarithms in fields of characteristic two. *IEEE Transactions on Information Theory*, 30(4):587–593, 1984.
5. Jean-Charles Faugère, Ayoub Otmani, Ludovic Perret, and Jean-Pierre Tillich. Algebraic cryptanalysis of McEliece variants with compact keys. In *Advances in cryptology—EUROCRYPT 2010*, volume 6110 of *Lecture Notes in Comput. Sci.*, pages 279–298. Springer, Berlin, 2010.
6. Jean-Charles Faugre, Ludovic Perret, Christophe Petit, and Gunal Renault. Improving the complexity of index calculus algorithms in elliptic curves over binary fields. In *Advances in Cryptology EUROCRYPT 2012*, volume 7237 of *Lecture Notes in Computer Science*, pages 27–44. Springer Berlin Heidelberg, 2012.
7. 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.
8. Robert Granger and Frederik Vercauteren. On the discrete logarithm problem on algebraic tori. In *Advances in cryptology—CRYPTO 2005*, volume 3621 of *Lecture Notes in Comput. Sci.*, pages 66–85. Springer, Berlin, 2005.
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). *Cryptography and Communications*, 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. *IACR Cryptology ePrint Archive*, 2012:720, 2012.
12. Antoine Joux and Reynald Lercier. The function field sieve is quite special. In *Algorithmic number theory (Sydney, 2002)*, volume 2369 of *Lecture Notes in Comput. Sci.*, pages 431–445. Springer, Berlin, 2002.
13. 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, April 2003.
14. Antoine Joux and Reynald Lercier. The function field sieve in the medium prime case. In Serge Vaudenay, editor, *EUROCRYPT 2006*, volume 4004 of *Lecture Notes in Computer Science*, pages 254–270. Springer, 2006.
15. Brian A. LaMacchia and Andrew M. Odlyzko. Solving large sparse linear systems over finite fields. In *CRYPTO 1990*, pages 109–133, 1990.
16. Arjen K. Lenstra and Hendrik W. Lenstra, Jr., editors. *The development of the number field sieve*, volume 1554 of *Lecture Notes in Mathematics*. Springer-Verlag, Berlin, 1993.
17. 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*, pages 376–392. Springer-Verlag, Berlin, Heidelberg, 2009.
18. Karl Rubin and Alice Silverberg. Torus-based cryptography. In *Advances in cryptology—CRYPTO 2003*, volume 2729 of *Lecture Notes in Comput. Sci.*, pages 349–365. Springer, Berlin, 2003.
19. Victor Shoup. *NTL: A library for doing number theory*, 5.5.2 edition, 2009. `http://www.shoup.net/ntl/`.