WEAKNESS OF $\mathbb{F}_{3^{6\cdot 509}}$ FOR DISCRETE LOGARITHM CRYPTOGRAPHY

GORA ADJ, ALFRED MENEZES, THOMAZ OLIVEIRA,
AND FRANCISCO RODRÍGUEZ-HENRÍQUEZ

Abstract. In 2013, Joux, and then Barbulescu, Gaudry, Joux and Thomé, presented new algorithms for computing discrete logarithms in finite fields of small and medium characteristic. We show that these new algorithms render the finite field $\mathbb{F}_{3^{6\cdot 509}} = \mathbb{F}_{3^{3054}}$ weak for discrete logarithm cryptography in the sense that discrete logarithms in this field can be computed significantly faster than with the previous fastest algorithms. Our concrete analysis shows that the supersingular elliptic curve over $\mathbb{F}_{3^{509}}$ with embedding degree 6 that had been considered for implementing pairing-based cryptosystems at the 128-bit security level in fact provides only a significantly lower level of security. Our work provides a convenient framework and tools for performing a concrete analysis of the new discrete logarithm algorithms and their variants.

1. Introduction

Let $\mathbb{F}_q$ denote a finite field of order q, and let $Q = q^n$. The discrete logarithm problem (DLP) in $\mathbb{F}_Q$ is that of determining, given a generator g of $\mathbb{F}_Q^*$ and an element $h \in \mathbb{F}_Q^*$, the integer $x \in [0, Q-2]$ satisfying $h = g^x$. The integer x is called the discrete logarithm of h to the base g and is denoted by $\log_g h$. In the remainder of the paper, we shall assume that the characteristic of $\mathbb{F}_q$ is 2 or 3.

The fastest general-purpose algorithm known for solving the DLP is Coppersmith's 1984 index-calculus algorithm [18] with a running time[1] of $L_Q[\frac{1}{3}, (32/9)^{1/3}] \approx L_Q[\frac{1}{3}, 1.526]$, where as usual $L_Q[\alpha, c]$ with $0 < \alpha < 1$ and $c > 0$ denotes the expression

$$\exp\big((c + o(1))(\log Q)^{\alpha}(\log\log Q)^{1-\alpha}\big)$$

that is subexponential in $\log Q$. In 2006, Joux and Lercier [35] presented an algorithm with a running time of $L_Q[\frac{1}{3}, 3^{1/3}] \approx L_Q[\frac{1}{3}, 1.442]$ when q and n are balanced in the sense that

$$q = \exp\left(3^{-2/3} \cdot (\log Q)^{1/3}(\log\log Q)^{2/3}\right) \text{ and } n = 3^{2/3} \cdot \left(\frac{\log Q}{\log\log Q}\right)^{2/3}.$$

In 2012, Joux [31] introduced a 'pinpointing' technique that improves the running time of the Joux-Lercier algorithm to $L_Q[\frac{1}{3}, 2/3^{2/3}] \approx L_Q[\frac{1}{3}, 0.961]$.

In February 2013, Joux [32] presented a new DLP algorithm with a running time of $L_Q[\frac{1}{4} + o(1), c]$ (for some undetermined c) when q and n are balanced in the sense that

Date: July 15, 2013; updated on October 25, 2013.

This work was done while the first, third and fourth authors were visiting the University of Waterloo.

[1]All running times in this paper have been determined using *heuristic* arguments, and have not been rigorously proven.

$q \approx m$ where $n = 2m$. Also in February 2013, Göloğlu, Granger, McGuire and Zumbrägel [26] proposed a variant of the Joux-Lercier algorithm that imposes a further divisibility condition on ℓ where $q = 2^\ell$. The running time of the Gögloğlu et al. algorithm is (i) $L_Q[\frac{1}{3}, 2/3^{2/3}] \approx L_Q[\frac{1}{3}, 0.961]$ when $n \approx 2^m d_1$, $d_1 \approx 2^m$, and $m \mid \ell$; and (ii) between $L_Q[\frac{1}{3}, (2/3)^{2/3}] \approx L_Q[\frac{1}{3}, 0.763]$ and $L_Q[\frac{1}{3}, 1/2^{1/3}] \approx L_Q[\frac{1}{3}, 0.794]$ when $n \approx 2^m d_1$, $2^m \gg d_1$, and $m \mid \ell$. The new algorithms were used to compute discrete logarithms in $\mathbb{F}_{2^{8\cdot 3\cdot 255}} = \mathbb{F}_{2^{6120}}$ in only 750 CPU hours [27], and in $\mathbb{F}_{2^{8\cdot 3\cdot 257}} = \mathbb{F}_{2^{6168}}$ in only 550 CPU hours [33]. The astoundingly small computational effort expended in these experiments depends crucially on the special nature of the fields $\mathbb{F}_{2^{6120}}$ and $\mathbb{F}_{2^{6168}}$ — namely that $\mathbb{F}_{2^{6120}}$ is a degree-255 extension of $\mathbb{F}_{2^{8\cdot 3}}$ with $255 = 2^8 - 1$, and $\mathbb{F}_{2^{6168}}$ is a degree-257 extension of $\mathbb{F}_{2^{8\cdot 3}}$ with $257 = 2^8 + 1$. Despite these remarkable achievements, the effectiveness of the new algorithms for computing discrete logarithms in general finite fields of small characteristic remains unclear.

In June 2013, Barbulescu, Gaudry, Joux and Thomé [7] presented a new DLP algorithm that, for many choices of field sizes, is asymptotically faster than all previous algorithms. Most impressively, in the case where $q \approx n$ and $n \leq q + 2$, the discrete logarithm problem in $\mathbb{F}_{q^{2n}} = \mathbb{F}_Q$ can be solved in *quasi-polynomial time*

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

Note that (1) is asymptotically smaller than $L_Q[\alpha, c]$ for any $\alpha > 0$ and $c > 0$. However, the practical relevance of the new algorithm has not yet been determined.

The aforementioned advances in DLP algorithms are potentially relevant to the security of pairing-based cryptosystems that use bilinear pairings derived from supersingular elliptic curves E or genus-2 hyperelliptic curves C defined over finite fields $\mathbb{F}_q$ of characteristic 2 or 3. Such a symmetric pairing, classified as a Type 1 pairing in [24], is a non-degenerate bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ where $\mathbb{G}$ and $\mathbb{G}_T$ are groups of prime order N. Here, $\mathbb{G}$ is either a subgroup of $E(\mathbb{F}_q)$, the group of $\mathbb{F}_q$-rational points on E, or a subgroup of $\mathrm{Jac}_C(\mathbb{F}_q)$, the jacobian of C over $\mathbb{F}_q$, and $\mathbb{G}_T$ is the order-N subgroup of $\mathbb{F}_{q^k}^*$ where k is the embedding degree (the smallest positive integer such that $\#\mathbb{G} \mid (q^k - 1)$). A necessary condition for the security of pairing-based cryptosystems that employ the pairing e is the intractability of the discrete logarithm problem in $\mathbb{G}_T$. Hence, any advance in algorithms for solving the DLP in $\mathbb{F}_{q^k}$ can potentially impact the security of pairing-based cryptosystems.

Three symmetric pairings that have received a great deal of attention in the literature are: (i) the $k = 6$ pairings derived from supersingular elliptic curves $Y^2 = X^3 - X + 1$ and $Y^2 = X^3 - X - 1$ over $\mathbb{F}_{3^\ell}$; (ii) the $k = 4$ pairings derived from supersingular elliptic curves $Y^2 + Y = X^3 + X$ and $Y^2 + Y = X^3 + X + 1$ over $\mathbb{F}_{2^\ell}$; and (iii) the $k = 12$ pairing derived from supersingular genus-2 curves $Y^2 + Y = X^5 + X^3$ and $Y^2 + Y = X^5 + X^3 + 1$ over $\mathbb{F}_{2^\ell}$; in all cases, ℓ is chosen to be prime. These symmetric pairings were considered in some early papers [15, 22, 9, 23] on pairing-based cryptography. Since then, many papers have reported on software and hardware implementation of these pairings; some examples are [8, 28, 41, 3, 29, 13, 16, 19, 12, 4, 1].

In all the papers cited in the previous paragraph, the pairing parameters were chosen under the assumption that Coppersmith's algorithm is the fastest method for finding discrete logarithms in $\mathbb{F}_{q^k}$. For example, to achieve the 128-bit security level, [3] chose

$\ell = 1223$ for the $k = 4$ pairing and $\ell = 509$ for the $k = 6$ pairing, [16] chose $\ell = 439$ for the $k = 12$ pairing, and [4] chose $\ell = 367$ for the $k = 12$ pairing. These choices were made because Coppersmith's algorithm, as analyzed by Lenstra [37], has running time approximately 2^{128} for computing logarithms in $\mathbb{F}_{2^{4\cdot 1223}}$, $\mathbb{F}_{3^{6\cdot 509}}$, $\mathbb{F}_{2^{12\cdot 439}}$, and $\mathbb{F}_{2^{12\cdot 367}}$, respectively.

In 2012, Hayashi et al. [30] reported on their implementation of the Joux-Lercier algorithm for computing logarithms in $\mathbb{F}_{3^{6\cdot 97}}$. Their work demonstrated that in practice the Joux-Lercier algorithm is considerably faster than Coppersmith's algorithm for DLP computations in $\mathbb{F}_{3^{6\cdot 97}}$; note that the $k = 6$ pairing with $\ell = 97$ was considered in [9, 23]. In contrast, the largest discrete logarithm computation reported using Coppersmith's algorithm (and its generalizations [2, 34]) is the April 2013 computation by Barbulescu et al. [5] of logarithms in $\mathbb{F}_{2^{809}}$; note that 809 is prime and $3^{6\cdot 97} \approx 2^{922}$. Shinohara et al. [42] estimated that $\mathbb{F}_{3^{6\cdot 509}}$ offers only 111-bits of security against Joux-Lercier attacks, considerably less than the assumed 128-bits of security against Coppersmith attacks.

The purpose of this paper is to demonstrate that the new algorithms by Joux [32] and Barbulescu et al. [7] can be combined to solve the discrete logarithm problem in $\mathbb{F}_{3^{6\cdot 509}}$ significantly faster than the Joux-Lercier algorithm. More precisely, we estimate that logarithms in this field can be computed in $2^{73.7}$ time with the new algorithms, which is less than the estimate of $2^{102.69}$ for the Joux-Lercier algorithm with pinpointing; the unit of time here is the cost of an integer multiplication modulo an 804-bit prime. While the $2^{73.7}$ computation is certainly a formidable challenge, we argue that it is already within the realm of feasibility for a very well-funded adversary. Thus, we conclude that $\mathbb{F}_{3^{6\cdot 509}}$ does not offer adequate security for discrete logarithm cryptosystems and, in particular, the supersingular elliptic curve over $\mathbb{F}_{3^{509}}$ with embedding degree 6 is not suitable for implementing pairing-based cryptosystems.

We also analyze the efficacy of the new algorithms for computing discrete logarithms in $\mathbb{F}_{2^{12\cdot 367}}$ and conclude that the supersingular genus-2 curve over $\mathbb{F}_{2^{367}}$ with embedding degree 12 should be considered weak and not employed in pairing-based cryptography.

The remainder of the paper is organized as follows. §2 collects some results on the number of smooth polynomials over a finite field. The new discrete logarithm algorithms are outlined in §3. Our estimates for discrete logarithm computations in $\mathbb{F}_{3^{6\cdot 509}}$ and $\mathbb{F}_{2^{12\cdot 367}}$ are presented in §4 and Appendix A, respectively. We draw our conclusions in §5.

2. Smooth polynomials

2.1. **Number of smooth polynomials.** The number of monic polynomials of degree n over $\mathbb{F}_q$ is q^n. The number of monic irreducible polynomials of degree n over $\mathbb{F}_q$ is

$$I_q(n) = \frac{1}{n} \sum_{d|n} \mu(n/d) q^d, \tag{2}$$

where μ is the Möbius function. A polynomial in $\mathbb{F}_q[X]$ is said to be m-smooth if all its irreducible factors in $\mathbb{F}_q[X]$ have degree at most m. Define

$$F(u, z) = \prod_{\ell=1}^{m} \left(1 + \frac{uz^\ell}{1 - z^\ell}\right)^{I_q(\ell)}.$$

$F(u, z)$ is the generating function for m-smooth monic polynomials in $\mathbb{F}_q[X]$, where u marks the number of distinct irreducible factors, and z marks the degree of the polynomial. Thus, the number of monic m-smooth degree-n polynomials in $\mathbb{F}_q[X]$ that have exactly k distinct monic irreducible factors is

$$N_q(m, n, k) = [u^k z^n]F(u, z) \tag{3}$$

where [ ] denotes the coefficient operator, whereas the total number of monic m-smooth degree-n polynomials in $\mathbb{F}_q[X]$ is

$$N_q(m, n) = [z^n]F(1, z). \tag{4}$$

Furthermore, the average number of distinct monic irreducible factors among all monic m-smooth degree-n polynomials in $\mathbb{F}_q[X]$ is

$$A_q(m, n) = \frac{[z^n]\left(\frac{\partial F}{\partial u}\big|_{u=1}\right)}{N_q(m, n)}. \tag{5}$$

For any given q, m and n, $N_q(m, n)$ can be obtained by using a symbolic algebra package such as Maple [40] to compute the first $n + 1$ terms of the Taylor series expansion of $F(1, z)$ and then extracting the coefficient of z^n. Similarly, one can compute $N_q(m, n, k)$ and $A_q(m, n)$. For example, we used Maple 17 on a 3.2 GHz Intel Xeon CPU X5672 machine to compute $N_{3^{12}}(30, 254)$ in 3.2 seconds, $A_{3^{12}}(30, 254) = 14.963$ in 102.9 seconds, and $N_{3^{12}}(30, 254, 9)$ in 4305 seconds.

Example 1. ($q = 3^{12}$, $n = 33$, $m = 7$) Table 1 lists, for each $t \in [1, 33]$, the proportion of monic 7-smooth degree-33 polynomials in $\mathbb{F}_{3^{12}}[X]$ that have exactly t distinct monic irreducible factors, and the proportion that have at most t distinct monic irreducible factors (cf. §4.6). We see that most 7-smooth degree-33 polynomials in $\mathbb{F}_{3^{12}}[X]$ will have 6, 7, 8, 9 or 10 distinct monic irreducible factors. In fact, the average number of distinct monic irreducible factors is $A_{3^{12}}(7, 33) = 8.072$.

2.2. **Smoothness testing.** A degree-d polynomial $f \in \mathbb{F}_q[X]$ can be tested for m-smoothness by computing

$$w(X) = f'(X) \cdot \prod_{i=\lceil m/2 \rceil}^{m} (X^{q^i} - X) \mod f(X) \tag{6}$$

and checking whether $w(X) = 0$ [18]. Here, f' denotes the formal derivative of f. If f indeed is m-smooth, then $w(X) = 0$. On the other hand, if f is not m-smooth then a necessary condition to have $w(X) = 0$ is that f be divisible by the square of an irreducible polynomial of degree $> m$. Since randomly selected polynomials f are unlikely to satisfy this condition, the vast majority of polynomials that pass the smoothness test are indeed m-smooth. The polynomials that are declared to be m-smooth are then factored using a general-purpose polynomial factorization algorithm, at which time the polynomials falsely declared to be m-smooth are identified.

Without loss of generality, we can assume that f is monic. Then the product of two polynomials of degree $< d$ can be multiplied modulo f in time $2d^2$, where the unit of time is an $\mathbb{F}_q$-multiplication. To compute $w(X)$, one first precomputes $X^q \bmod f$. This can be accomplished by repeated square-and-multiplication at a cost of at most $2||q||_2$ modular multiplications, where $||q||_2$ denotes the bitlength of q. Then, $X^{qi} \bmod f$ for $2 \leq i \leq d-1$

t	Exactly t distinct irred. factors	At most t distinct irred. factors
1	0.00000000000000000000000000000000	0.00000000000000000000000000000000
2	0.00000000000000000000000000000000	0.00000000000000000000000000000000
3	0.00000000000000000000000000000000	0.00000000000000000000000000000000
4	0.00000000000000000000000000000000	0.00000000000000000000000000000000
5	0.00694396258725365727783584 2266	0.00694396258725365727783584 2266
6	0.09450823523737471257706302 1493	0.10145219782462836985489886 3759
7	0.25239461704744106489033780 3216	0.35384681487206943474523666 6975
8	0.30099673202313362746357465 5796	0.65484354689520306220881132 2771
9	0.20850598472151827911125834 8913	0.86334953161672134132006967 1684
10	0.09566691512298691622545503 3585	0.95901644673970825754552470 5269
11	0.03143115350072971206911975 9222	0.99044760024043796961464446 4492
12	0.00778172966693396340229811 1588	0.99822932990737193301694257 6080
13	0.00150455397702231797030251 0303	0.99973388388439425098724508 6382
14	0.00023317550477421944799744 9230	0.99996705938916847043524253 5613
15	0.00002953983934862953106205 0590	0.99999659922851709996630458 6203
16	0.00000310502611457732272818 3235	0.99999970425463167728903276 9438
17	0.00000027391385389124067914 1964	0.99999997816848556852971191 1402
18	0.00000002045574507926935020 3882	0.99999999862423064779906211 5284
19	0.00000000130146987379573081 8942	0.99999999992570052159479293 4226
20	0.00000000007085333588538066 1976	0.99999999999655385748017359 6201
21	0.00000000000330881436907745 8356	0.99999999999986267184925105 4557
22	0.00000000000013263322345658 5201	0.99999999999999530507270763 9758
23	0.00000000000004557715382559 271	0.99999999999999986278809019 9029
24	0.00000000000000133804578363 524	0.99999999999999999659266856 2553
25	0.00000000000000003336091284 541	0.99999999999999999992875984 7094
26	0.00000000000000000070002399 226	0.99999999999999999999876224 6320
27	0.00000000000000000001220209 967	0.99999999999999999999982456 287
28	0.00000000000000000000017346 134	0.99999999999999999999999802 422
29	0.00000000000000000000000195 878	0.99999999999999999999999998 300
30	0.00000000000000000000000001 690	0.99999999999999999999999999 990
31	0.00000000000000000000000000 010	1.00000000000000000000000000 000
32	0.00000000000000000000000000 000	1.00000000000000000000000000 000
33	0.00000000000000000000000000 000	1.00000000000000000000000000 000

TABLE 1. Proportion of monic 7-smooth degree-33 polynomials in $\mathbb{F}_{3^{12}}[X]$ that have exactly t (resp. at most t) distinct monic irreducible factors.

can be computed by repeated multiplication of $X^q \bmod f$ with itself at a cost of approximately d modular multiplications, and $X^{q^i} \bmod f$ for $2 \le i \le m$ can be computed by repeated exponentiation by q with each exponentiating having cost d^2 $\mathbb{F}_q$-multiplications. Finally, the product in (6) can be computed using $m/2$ modular multiplications at a cost of md^2 $\mathbb{F}_q$-multiplications. The total cost for testing m-smoothness of f is thus

$$S_q(m,d) = 2d^2(d+m+2||q||_2) \quad \mathbb{F}_q\text{-multiplications.} \tag{7}$$

We will mostly be interested in the case $q = 3^{12}$. Then, $X^q \bmod f$ can be determined by first precomputing $X^3, X^6, \ldots, X^{3(d-1)} \bmod f$ by repeated multiplication by X. Thereafter, cubing a polynomial modulo f can be accomplished by cubing the coefficients

of the polynomial, and then multiplying the precomputed polynomials by these cubes (and adding the results). In this way, we get a loose upper bound of $3d^2 + 11d^2 = 14d^2$ $\mathbb{F}_{3^{12}}$-multiplications of the cost to compute $X^{3^{12}} \bmod f$, and the total cost for testing m-smoothness of f becomes

$$S_{3^{12}}(m, d) = 2d^2(d + m + 7) \quad \mathbb{F}_{3^{12}}\text{-multiplications.} \tag{8}$$

3. New DLP algorithm of Joux and Barbulescu et al.

The DLP algorithm we describe is due to Joux [32], with a descent step from the quasi-polynomial time algorithm (QPA) of Barbulescu et al. [7]. For lack of a better name, we will call this algorithm the "new DLP algorithm".

Let $\mathbb{F}_{q^{2n}}$ be a finite field where $n \leq q + 2$. The elements of $\mathbb{F}_{q^{2n}}$ are represented as polynomials of degree at most $n - 1$ over $\mathbb{F}_{q^2}$. Let $N = q^{2n} - 1$. Let g be an element of order N in $\mathbb{F}^*_{q^{2n}}$, and let $h \in \mathbb{F}^*_{q^{2n}}$. We wish to compute $\log_g h$. The algorithm proceeds by first finding the logarithms of all degree-one (§3.2) and degree-two (§3.3) elements in $\mathbb{F}_{q^{2n}}$. Then, in the *descent stage*, $\log_g h$ is expressed as a linear combination of logarithms of degree-one and degree-two $\mathbb{F}_{q^{2n}}$ elements. The descent stage proceeds in several steps, each expressing the logarithm of a degree-D element as a linear combination of the logarithms of elements of degree $\leq m$ for some $m < D$. Four descent methods are used; these are described in §3.4–§3.7. The cost of each step is given in Table 2.

Finding logarithms of linear polynomials (§3.2)	
Relation generation	$6q^2 \cdot S_{q^2}(1, 3)$
Linear algebra	$q^5 \cdot M_N$
Finding logarithms of irreducible quadratic polynomials (§3.3)	
Relation generation	$q^{16}/N_{q^2}(1, 6) \cdot S_{q^2}(1, 6)$
Linear algebra	$q^7 \cdot M_N$
Descent (Degree D to degree m)	
Continued-fraction (§3.4)	$\{D = n - 1\}$ $(q^{n-1}/N_{q^2}(m, (n-1)/2))^2 \cdot S_{q^2}(m, (n-1)/2)$
Classical (§3.5)	$q^{2(t_1-D+t_2)}/(N_{q^2}(m, t_1 - D)N_{q^2}(m, t_2)) \cdot \min(S_{q^2}(m, t_1 - D), S_{q^2}(m, t_2))$
QPA (§3.6)	$q^{6D+2}/N_{q^2}(m, 3D) \cdot S_{q^2}(m, 3D) + q^5 \cdot M_N$
Gröbner bases (§3.7)	$G_{q^2}(m, D) + q^{6m-2D}/N_{q^2}(m, 3m - D) \cdot S_{q^2}(m, 3m - D)$

Table 2. Estimated costs of the main steps of the new DLP algorithm for computing discrete logarithms in $\mathbb{F}_{q^{2n}}$. M_N and M_{q^2} denote the costs of a multiplication modulo N and a multiplication in $\mathbb{F}_{q^2}$. The smoothness testing cost $S_{q^2}(m, D)$ is given in (7). See §3.5 for the definitions of t_1 and t_2. The Gröbner basis cost $G_{q^2}(m, D)$ is defined in §3.7.

Notation. For $\gamma \in \mathbb{F}_{q^2}$, $\overline{\gamma}$ denotes the element γ^q. For $P \in \mathbb{F}_{q^2}[X]$, $\overline{P}$ denotes the polynomial obtained by raising each coefficient of P to the power q. The cost of an integer multiplication modulo N is denoted by M_N, and the cost of a multiplication in $\mathbb{F}_{q^2}$ is denoted by M_{q^2}. The projective general linear group of order 2 over $\mathbb{F}_q$ is denoted $\mathrm{PGL}_2(\mathbb{F}_q)$. $\mathcal{P}_q$ is a set of distinct representatives of the left cosets of $\mathrm{PGL}_2(\mathbb{F}_q)$ in $\mathrm{PGL}_2(\mathbb{F}_{q^2})$; note that $\#\mathcal{P}_q = q^3 + q$.

3.1. **Setup.** Select polynomials $h_0, h_1 \in \mathbb{F}_{q^2}[X]$ of degree at most 2 so that $h_1X^q - h_0$ has an irreducible factor I_X of degree n in $\mathbb{F}_{q^2}[X]$; we will henceforth assume that $\max(\deg h_0, \deg h_1) = 2$. In order to avoid the "traps" discussed in [17], we further assume that each irreducible factor $J \in \mathbb{F}_{q^2}[X]$ of $(h_1X^q - h_0)/I_X$ satisfies the following two conditions: (i) $\gcd(\deg J, n) = 1$; and (ii) $\deg J > m$ where m is the integer specified in the continued-fraction descent stage (§3.4). We have $X^q \equiv h_0/h_1 \pmod{I_X}$, so $\mathbb{F}_{q^{2n}}$ is represented as $\mathbb{F}_{q^{2n}} = \mathbb{F}_{q^2}[X]/(I_X)$ and the elements of $\mathbb{F}_{q^{2n}}$ can be represented as polynomials in $\mathbb{F}_{q^2}[X]$ of degree at most $n-1$. Let g be a generator of $\mathbb{F}_{q^{2n}}^*$.

3.2. **Finding logarithms of linear polynomials.** Let $\mathcal{B}_1 = \{X + a \mid a \in \mathbb{F}_{q^2}\}$, and note that $\#\mathcal{B}_1 = q^2$. To compute the logarithms of $\mathcal{B}_1$-elements, we first generate linear relations of these logarithms. Let $a, b, c, d \in \mathbb{F}_{q^2}$ with $ad - bc \neq 0$. Substituting $Y \mapsto (aX + b)/(cX + d)$ into the systematic equation

$$Y^q - Y = \prod_{\alpha \in \mathbb{F}_q} (Y - \alpha), \tag{9}$$

and then multiplying by $(cX + d)^{q+1}$ yields

$$\begin{aligned}(\overline{a}h_0 + \overline{b}h_1)(cX + d) - (aX + b)(\overline{c}h_0 + \overline{d}h_1) \\ \equiv h_1 \cdot (cX + d) \cdot \prod_{\alpha \in \mathbb{F}_q} [(a - \alpha c)X + (b - \alpha d)] \pmod{I_X}.\end{aligned} \tag{10}$$

Note that the left side of (10) is a polynomial of degree (at most) 3. If this polynomial is 1-smooth, then taking logarithms of both sides of (10) yields a linear relation of the logarithms of $\mathcal{B}_1$-elements[2] and the logarithm of h_1. As explained in [7], in order to avoid redundant relations one selects quadruples (a, b, c, d) from $\mathcal{P}_q$; here we are identifying a quadruple (a, b, c, d) with the matrix $\left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right)$.

Now, the probability that the left side of (10) is 1-smooth is

$$\frac{N_{q^2}(1, 3)}{q^6} = \binom{q^2 + 2}{3} / q^6 \approx \frac{1}{6}.$$

Thus, after approximately $6q^2$ trials one expects to obtain (slightly more than) q^2 relations. The cost of the relation generation stage is $6q^2 \cdot S_{q^2}(1, 3)$. The logarithms can then be obtained by using standard techniques for solving sparse systems of linear equations [36]. The expected cost of the linear algebra is $q^5 \cdot M_N$ since each equation has approximately q nonzero terms.

3.3. **Finding logarithms of irreducible quadratic polynomials.** Let $u \in \mathbb{F}_{q^2}$, and let $Q(X) = X^2 + uX + v \in \mathbb{F}_{q^2}[X]$ be an irreducible quadratic. Define $\mathcal{B}_{2,u}$ to be the set of all irreducible quadratics of the form $X^2 + uX + w$ in $\mathbb{F}_{q^2}[X]$; one expects that $\#\mathcal{B}_{2,u} \approx (q^2 - 1)/2$. The logarithms of all elements in $\mathcal{B}_{2,u}$ are found simultaneously using one application of QPA descent (see §3.6). More precisely, one first collects relations of

[2]It is understood that all polynomials of the right side of (10) and factors of the left side of (10) should be made monic. The same holds for (17) and (19).

the form (17), where the left side of (17) factors as a product of linear polynomials (whose logarithms are known). The expected number of relations one can obtain is

$$\frac{N_{q^2}(1,6)}{q^{12}} \cdot (q^3+q).$$

Provided that this number is significantly greater than $\#\mathcal{B}_{2,u}$, the matrix $\mathcal{H}(Q)$ is expected to have full (column) rank. One can then solve the resulting system of linear equations to obtain the logarithms of all irreducible translates $Q+w$ of Q. This step is repeated for each $u \in \mathbb{F}_{q^2}$. Hence, there are q^2 independent linear systems of equations to be solved.

For each $u \in \mathbb{F}_{q^2}$, the cost of relation generation is $q^{14}/N_{q^2}(1,6) \cdot S_{q^2}(1,6)$, while the linear algebra cost is $q^5 \cdot M_N$.

3.4. **Continued-fraction descent.** Recall that we wish to compute $\log_g h$, where $h \in \mathbb{F}_{q^{2n}} = \mathbb{F}_{q^2}[X]/(I_X)$. Note that $\deg h \leq n-1$; we will henceforth assume that $\deg h = n-1$. The descent stage begins by multiplying h by a random power of g. The extended Euclidean algorithm is used to express the resulting field element h' in the form $h' = w_1/w_2$ where $\deg w_1, \deg w_2 \approx n/2$ [14]; for simplicity, we shall assume that n is odd and $\deg w_1 = \deg w_2 = (n-1)/2$. This process is repeated until both w_1 and w_2 are m-smooth for some chosen $m < (n-1)/2$. This gives $\log_g h'$ as a linear combination of logarithms of polynomials of degree at most m. The expected cost of this continued-fraction descent step is approximately

$$\left(\frac{q^{n-1}}{N_{q^2}(m,(n-1)/2)}\right)^2 \cdot S_{q^2}(m,(n-1)/2). \tag{11}$$

The expected number of distinct irreducible factors of w_1 and w_2 is $2A_{q^2}(m,(n-1)/2)$. In the analysis, we shall assume that each of these irreducible factors has degree exactly m. The logarithm of each of these degree-m polynomials is then expressed as a linear combination of logarithms of smaller degree polynomials using one of the descent methods described in §3.5, §3.6 and §3.7.

3.5. **Classical descent.** Let p be the characteristic of $\mathbb{F}_q$, and let $q = p^\ell$. Let $s \in [1,\ell]$, and let $R \in \mathbb{F}_{q^2}[X,Y]$. Then

$$R(X, X^{p^s})^{p^{\ell-s}} = R'(X^{p^{\ell-s}}, X^q) \equiv R'(X^{p^{\ell-s}}, \frac{h_0}{h_1}) \pmod{I_X},$$

where R' is obtained from R by raising all its coefficients to the power $p^{\ell-s}$. For the sake of simplicity, we will assume in this section that $h_1 = 1$ and so

$$R(X, X^{p^s})^{p^{\ell-s}} \equiv R'(X^{p^{\ell-s}}, h_0) \pmod{I_X}. \tag{12}$$

Let $Q \in \mathbb{F}_{q^2}[X]$ with $\deg Q = D$, and let $m < D$. In the Joux-Lercier descent method [35], as modified by Joux [32], one selects suitable parameters d_1, d_2 and searches for a polynomial $R \in \mathbb{F}_{q^2}[X,Y]$ such that (i) $\deg_X R \leq d_1$ and $\deg_Y R \leq d_2$; (ii) $Q \mid R_1$ where $R_1 = R(X, X^{p^s})$; and (iii) R_1/Q and R_2 are m-smooth where $R_2 = R'(X^{p^{\ell-s}}, h_0)$. Taking logarithms of both sides of (12) then gives an expression for $\log_g Q$ in terms of the logarithms of polynomials of degree at most m.

A family of polynomials R satisfying (i) and (ii) can be constructed by finding the null space of the $D \times (D+\delta)$ matrix whose columns are indexed by monomials $X^i Y^j$ for $D+\delta$

pairs $(i,j) \in [0,d_1] \times [0,d_2]$, and whose X^iY^j-th column entries are the coefficients of the polynomial $X^i(X^{p^s})^j \bmod Q$. The components of the vectors in the null space of this matrix can be interpreted as the coefficients of polynomials $R \in \mathbb{F}_{q^2}[X,Y]$ satisfying (i) and (ii). The dimension of this null space is expected to be δ, and so the null space is expected to contain $(q^2)^{\delta-1}$ monic polynomials. Let $\deg R_1 = t_1$ and $\deg R_2 = t_2$. We have $t_1 \le d_1 + p^s d_2$ and $t_2 \le p^{\ell-s}d_1 + 2d_2$; the precise values of t_1 and t_2 depend on the (i,j) pairs chosen (see §4.5 for an example). In order to ensure that the null space includes a monic polynomial R such that both R_1/Q and R_2 are m-smooth, the parameters must be selected so that

$$q^{2\delta-2} \gg \frac{q^{2(t_1-D)}}{N_{q^2}(m,t_1-D)} \cdot \frac{q^{2t_2}}{N_{q^2}(m,t_2)}. \tag{13}$$

Ignoring the time to compute the null space, the expected cost of finding a polynomial R satisfying (i)–(iii) is

$$\frac{q^{2(t_1-D)}}{N_{q^2}(m,t_1-D)} \cdot \frac{q^{2t_2}}{N_{q^2}(m,t_2)} \cdot \min(S_{q^2}(m,t_1-D), S_{q^2}(m,t_2)). \tag{14}$$

The expected number of distinct irreducible factors of R_1/Q and R_2 is $A_{q^2}(m,t_1-D) + A_{q^2}(m,t_2)$. In the analysis, we shall assume that each of these irreducible factors has degree exactly m.

3.6. **QPA descent.** The QPA descent method is so named because it was a crucial step in the Barbulescu et al. quasi-polynomial time algorithm for the DLP in finite fields of small characteristic [7].

Let $Q \in \mathbb{F}_{q^2}[X]$ with $\deg Q = D$, and let $m \in [\lceil D/2 \rceil, D-1]$. Let $(a,b,c,d) \in \mathcal{P}_q$, and recall that $\#\mathcal{P}_q = q^3 + q$. Substituting $Y \mapsto (aQ+b)/(cQ+d)$ into the systematic equation (9) and multiplying by $(cQ+d)^{q+1}$ yields

$$(aQ+b)^q(cQ+d) - (aQ+b)(cQ+d)^q = (cQ+d) \prod_{\alpha \in \mathbb{F}_q} [(a-\alpha c)Q + (b - \alpha d)]. \tag{15}$$

The left side of (15) can be written as

$$\begin{aligned}(\overline{a}\overline{Q}(X^q) + \overline{b})(cQ+d) - (aQ+b)(\overline{c}\overline{Q}(X^q) + \overline{d}) \\ \equiv (\overline{a}\overline{Q}(\frac{h_0}{h_1}) + \overline{b})(cQ+d) - (aQ+b)(\overline{c}\overline{Q}(\frac{h_0}{h_1}) + \overline{d}) \pmod{I_X}.\end{aligned}$$

Hence

$$\begin{aligned}(\overline{a}\overline{Q}(\frac{h_0}{h_1}) + \overline{b})(cQ+d) - (aQ+b)(\overline{c}\overline{Q}(\frac{h_0}{h_1}) + \overline{d}) \\ \equiv (cQ+d) \prod_{\alpha \in \mathbb{F}_q} [(a-\alpha c)Q + (b - \alpha d)] \pmod{I_X}.\end{aligned} \tag{16}$$

Multiplying (16) by h_1^D yields

$$\begin{aligned}(\overline{a}\widetilde{Q} + \overline{b})(cQ+d) - (aQ+b)(\overline{c}\widetilde{Q} + \overline{d}) \\ \equiv h_1^D \cdot (cQ+d) \cdot \prod_{\alpha \in \mathbb{F}_q} [(a-\alpha c)Q + (b - \alpha d)] \pmod{I_X},\end{aligned} \tag{17}$$

where $\widetilde{Q}(X) = h_1^D \cdot \overline{Q}(h_0/h_1)$. Note that the polynomial on the left side of (17) has degree $\leq 3D$. If this polynomial is m-smooth, then (17) yields a linear relation of the logarithms of some degree-m polynomials and logarithms of translates of Q. After collecting slightly more than q^2 such relations, one searches for a linear combination of these relations that eliminates all translates of Q except for Q itself. To achieve this, consider row vectors in $(\mathbb{Z}_N)^{q^2}$ with coordinates indexed by elements $\lambda \in \mathbb{F}_{q^2}$. For each relation, we define a vector v whose entry v_λ is 1 if $Q - \lambda$ appears in the right side of (17), and 0 otherwise. If the resulting matrix $\mathcal{H}(Q)$ of row vectors has full column rank, then one obtains an expression for $\log_g Q$ in terms of the logarithms of polynomials of degree $\leq m$. The number of distinct polynomials of degree $\leq m$ in this expression is expected to be $A_{q^2}(m, 3D) \cdot q^2$; in the analysis we shall assume that each of these polynomials has degree exactly m.

Since the probability that a degree-$3D$ polynomial is m-smooth is $N_{q^2}(m, 3D)/(q^2)^{3D}$, one must have

$$\frac{N_{q^2}(m, 3D)}{q^{6D}} \cdot (q^3 + q) \gg q^2 \tag{18}$$

in order to ensure that $\mathcal{H}(Q)$ has $\gg q^2$ rows, whereby $\mathcal{H}(Q)$ can be expected to have full rank.

Example 2. If $q = 3^6$ (so $q^2 = 531,441$), $D = 11$, and $m = 6$, then the left side of (18) is approximately 86,885 so QPA descent from 11 to 6 fails. On the other hand, if $q = 3^6$, $D = 11$, and $m = 7$ then the left side of (18) is approximately 570,172 so QPA descent from 11 to 7 should succeed. For $q = 3^6$ and each $D \in [2, 28]$, Table 3 shows the smallest $m \in [\lceil D/2 \rceil, D - 1]$ for which inequality (18) is satisfied.

D	m	$3D$	ratio	D	m	$3D$	ratio	D	m	$3D$	ratio
2	1	6	1.0125	11	7	33	1.0729	20	13	60	1.0435
3	2	9	5.2634	12	8	36	1.6867	21	14	63	1.3637
4	3	12	9.6178	13	9	39	2.3927	22	15	66	1.7168
5	3	15	1.1193	14	9	42	1.0597	23	15	69	1.0382
6	4	18	2.6162	15	10	45	1.5305	24	16	72	1.3140
7	5	21	4.3795	16	11	48	2.0631	25	17	75	1.6154
8	5	24	1.0924	17	11	51	1.0503	26	17	78	1.0340
9	6	27	1.9687	18	12	54	1.4317	27	18	81	1.2762
10	7	30	2.9948	19	13	57	1.8571	28	19	84	1.5388

TABLE 3. Let $q = 3^6$. For each $D \in [2, 28]$, the table gives the smallest $m \in [\lceil D/2 \rceil, D - 1]$ for which QPA descent from degree D to degree m could succeed, i.e., inequality (18) is satisfied. The fourth column shows the ratio $[N_{q^2}(m, 3D) \cdot (q^3 + q)/q^{6D}]/q^2$.

The expected cost of the relation generation portion of QPA descent is

$$\frac{q^{6D}}{N_{q^2}(m, 3D)}\, q^2 \cdot S_{q^2}(m, 3D),$$

while the cost of the linear algebra is $q^5 \cdot M_N$.

3.7. **Gröbner bases descent.** Let $Q \in \mathbb{F}_{q^2}[X]$ with $\deg Q = D$, and let $m = \lceil (D+1)/2 \rceil$. In Joux's new descent method [32, §5.3], one finds degree-m polynomials[3] $k_1, k_2 \in \mathbb{F}_{q^2}[X]$ such that $Q \mid G$, where

$$G = h_1^m (k_1^q k_2 - k_1 k_2^q) \bmod I_X.$$

We then have

$$h_1^m \cdot k_2 \cdot \prod_{\alpha \in \mathbb{F}_q} (k_1 - \alpha k_2) \equiv G(X) \pmod{I_X}$$

as can be seen by making the substitution $Y \mapsto k_1/k_2$ into the systematic equation (9) and clearing denominators. Define $\widetilde{k}(X) = h_1^m \cdot \overline{k}(h_0/h_1)$ and note that $\deg \widetilde{k} = 2m$. We thus have $G \equiv \widetilde{k}_1 k_2 - k_1 \widetilde{k}_2 \pmod{I_X}$, and consequently $G = \widetilde{k}_1 k_2 - k_1 \widetilde{k}_2$ provided that $3m < n$. It follows that $G(X) = Q(X)R(X)$ for some $R \in \mathbb{F}_{q^2}[X]$ with $\deg R = 3m - D$. If R is m-smooth, we obtain a linear relationship between $\log_g Q$ and logs of degree-m polynomials by taking logarithms of both sides of the following:

$$h_1^m \cdot k_2 \cdot \prod_{\alpha \in \mathbb{F}_q} (k_1 - \alpha k_2) \equiv Q(X)R(X) \pmod{I_X}. \tag{19}$$

To determine (k_1, k_2, R) that satisfy

$$\widetilde{k}_1 k_2 - k_1 \widetilde{k}_2 = Q(X)R(X), \tag{20}$$

one can transform (20) into a system of multivariate bilinear equations over $\mathbb{F}_q$. Specifically, each coefficient of k_1, k_2 and R is written using two variables over $\mathbb{F}_q$, the two variables representing the real and imaginary parts of that coefficient (which is in $\mathbb{F}_{q^2}$). The coefficients of $\widetilde{k}_1$ and $\widetilde{k}_2$ can then be written in terms of the coefficients of k_1 and k_2. Hence, equating coefficients of X^i of both sides of (20) yields $3m+1$ quadratic equations. The real and imaginary parts of each of these equations are equated, yielding $6m + 2$ bilinear equations in $10m - 2D + 6$ variables over $\mathbb{F}_q$. This system of equations can be solved by finding a Gröbner basis for the ideal it generates. Finally, solutions (k_1, k_2, R) are tested until one is found for which R is m-smooth. This yields an expression for $\log_g Q$ in terms of the logarithms of approximately $q + 1 + A_{q^2}(m, 3m - D)$ polynomials of degree (at most) m; in the analysis we shall assume that each of the polynomials has degree exactly m.

Now, the number of candidate pairs (k_1, k_2) is $((q^2)^{m+1})^2 = q^{4(m+1)}$. Since $(q^2)^{3m-D+1}$ of the $(q^2)^{3m+1}$ degree-$(3m)$ polynomials in $\mathbb{F}_{q^2}[X]$ are divisible by $Q(X)$, the number of solutions (k_1, k_2, R) is expected to be approximately

$$\frac{q^{2(3m-D+1)}}{q^{2(3m+1)}} \cdot q^{4(m+1)} = q^{4(m+1)-2D}.$$

However, the number of distinct R obtained will be significantly less than $q^{4(m+1)-2D}$. For example, any two pairs (k_1', k_2') and (k_1'', k_2'') with $k_1'/k_2' = k_1''/k_2''$ will generate the same R, so the expected number of distinct R is at most $q^{4(m+1)-2D}/(q^2 - 1)$. Let us denote by $R(m, D)$ the expected number of distinct R obtainable. Then the condition

$$R(m, D) \gg \frac{q^{2(3m-D)}}{N_{q^2}(m, 3m - D)}, \tag{21}$$

[3]More generally, the degrees of k_1 and k_2 can be different.

can ensure that there exists a solution (k_1, k_2, R) for which R is m-smooth.

The number $R(m, D)$ has not been determined in general. For the case $m = 1$ and $D = 2$, one must select $k_1 = aX + b$ and $k_2 = cX + d$ with $(a, b, c, d) \in \mathcal{P}_q$ to avoid collisions; hence $R(1, 2) \leq \frac{q^4}{q^8}(q^3 + q) \approx \frac{1}{q}$ and descending from 2 to 1 can be expected to succeed only for 1 out of every q quadratics; this is indeed what we observed in our experiments. In general, the success of the Gröbner bases descent step is best determined experimentally (cf. §4.7).

It is difficult to determine the exact cost $G_{q^2}(m, D)$ of the Gröbner basis finding step. After the Gröbner basis is found, the cost to find an m-smooth R is $(q^2)^{3m-D}/N_{q^2}(m, 3m-D) \cdot S_{q^2}(m, 3m - D)$.

4. Computing discrete logarithms in $\mathbb{F}_{3^{6\cdot 509}}$

We present a concrete analysis of the DLP algorithm described in §3 for computing discrete logarithms in $\mathbb{F}_{3^{6\cdot 509}}$. In fact, this field is embedded in the quadratic extension field $\mathbb{F}_{3^{12\cdot 509}}$, and it is the latter field where the DLP algorithm of §3 is executed. Thus, we have $q = 3^6 = 729$, $n = 509$, and $N = 3^{12\cdot 509} - 1$. Note that $3^{12\cdot 509} \approx 2^{9681}$. We wish to find $\log_g h$, where g is a generator of $\mathbb{F}^*_{3^{12\cdot 509}}$ and $h \in \mathbb{F}^*_{3^{12\cdot 509}}$.

As mentioned in §1, our main motivation for finding discrete logarithms in $\mathbb{F}_{3^{6\cdot 509}}$ is to attack the elliptic curve discrete logarithm problem in $E(\mathbb{F}_{3^{509}})$, where E is the supersingular elliptic curve $Y^2 = X^3 - X + 1$ with $\#E(\mathbb{F}_{3^{509}}) = 7r$, and where $r = (3^{509} - 3^{255} + 1)/7$ is an 804-bit prime. Note that $r^2 \nmid N$. The elliptic curve discrete logarithm problem in the order-r subgroup of $E(\mathbb{F}_{3^{509}})$ can be efficiently reduced to the discrete logarithm problem in the order-r subgroup of $\mathbb{F}^*_{3^{12\cdot 509}}$. In the latter problem, we are given two elements α, β of order r in $\mathbb{F}^*_{3^{12\cdot 509}}$ and we wish to find $\log_\alpha \beta$. It can readily be seen that $\log_\alpha \beta = (\log_g \beta)/(\log_g \alpha) \bmod r$. Thus, we will henceforth assume that h has order r and that we only need to find $\log_g h \bmod r$. An immediate consequence of this restriction is that all the linear algebra in the new algorithm has to be performed modulo the 804-bit r instead of modulo the 9681-bit N.

The parameters for each step of the algorithm were carefully chosen in order to balance the running time of the steps. We also took into account the degree to which each step could be parallelized on conventional computers. A summary of the parameter choices for the descent is given in Figure 1. The costs of each step are given in Table 4.

4.1. **Setup.**

We chose the representations

$$\mathbb{F}_{3^6} = \mathbb{F}_3[U]/(U^6 + 2U^4 + U^2 + 2U + 2)$$

and

$$\mathbb{F}_{3^{12}} = \mathbb{F}_{3^6}[V]/(V^2 + U^{365}).$$

We selected

$$h_0 = (U^{553}V + U^{343})X^2 + (U^{535}V + U^{417})X + (U^{172}V + U^{89}) \in \mathbb{F}_{3^{12}}[X]$$

and $h_1 = 1$, and $I_X \in \mathbb{F}_{3^{12}}[X]$ to be the monic degree-509 irreducible factor of $X^{3^6} - h_0$. The other irreducible factors have degrees 43, 55 and 122.

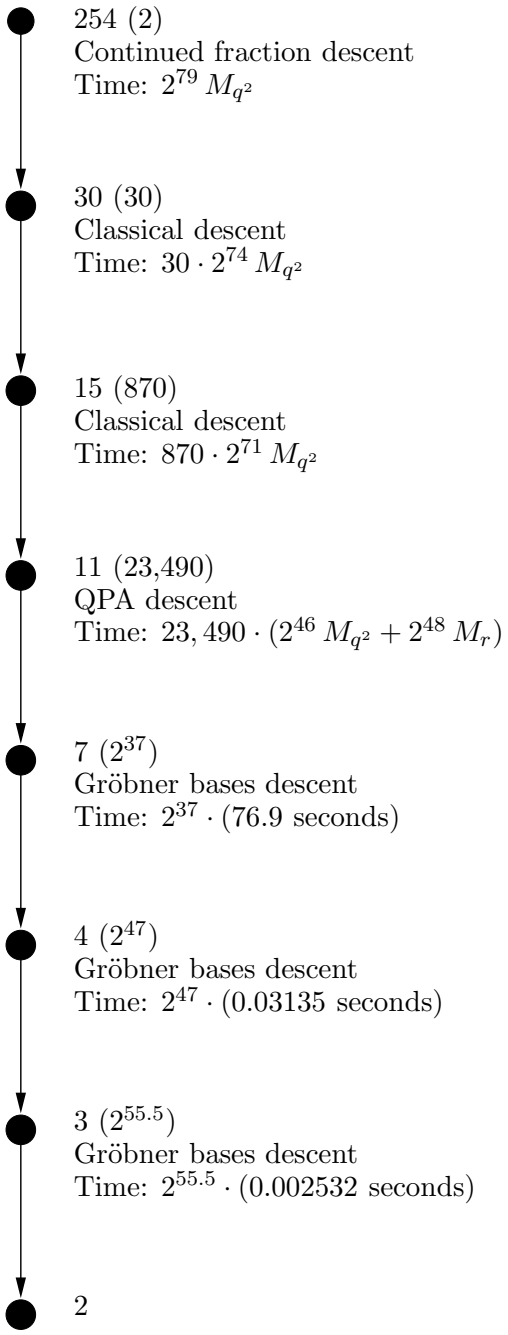

FIGURE 1. A typical path of the descent tree for computing an individual logarithm in $\mathbb{F}_{3^{12\cdot 509}}$ ($q = 3^6$). The numbers in parentheses next to each node are the expected number of nodes at that level. 'Time' is the expected time to generate all nodes at a level.

Finding logarithms of linear polynomials		
Relation generation	$2^{30}M_{q^2}$	$2^{22}M_r$
Linear algebra	$2^{48}M_r$	$2^{48}M_r$
Finding logarithms of irreducible quadratic polynomials		
Relation generation	$3^{12} \cdot 2^{39}M_{q^2}$	$2^{50}M_r$
Linear algebra	$3^{12} \cdot 2^{48}M_r$	$2^{67}M_r$
Descent		
Continued-fraction (254 to 30)	$2^{79}M_{q^2}$	$2^{71}M_r$
Classical (30 to 15)	$30 \cdot 2^{74}M_{q^2}$	$2^{71}M_r$
Classical (15 to 11)	$870 \cdot 2^{71}M_{q^2}$	$2^{73}M_r$
QPA (11 to 7)	$23,490 \cdot (2^{46}M_{q^2} + 2^{48}M_r)$	$2^{63}M_r$
Gröbner bases (7 to 4)	$2^{37} \cdot$ (76.9 seconds)	$2^{65}M_r$
Gröbner bases (4 to 3)	$2^{47} \cdot$ (0.03135 seconds)	$2^{64}M_r$
Gröbner bases (3 to 2)	$2^{55.5} \cdot$ (0.002532 seconds)	$2^{69}M_r$

TABLE 4. Estimated costs of the main steps of the new DLP algorithm for computing discrete logarithms in $\mathbb{F}_{3^{12 \cdot 509}}$ ($q = 3^6$). M_r and M_{q^2} denote the costs of a multiplication modulo the 804-bit prime $r = (3^{509} - 3^{255} + 1)/7$ and a multiplication in $\mathbb{F}_{3^{12}}$. We use the cost ratio $M_r/M_{q^2} = 2^8$, and also assume that 2^{22} multiplications modulo r can be performed in 1 second (cf. §4.8).

4.2. **Finding logarithms of linear polynomials.** The factor base $\mathcal{B}_1$ has size $3^{12} \approx 2^{19}$. The cost of relation generation is approximately $2^{30}M_{q^2}$, whereas the cost of the linear algebra is approximately $2^{48}M_r$. Note that relation generation can be effectively parallelized, unlike the linear algebra where parallelization on conventional computers provides relatively small benefits.

4.3. **Finding logarithms of irreducible quadratic polynomials.** For each $u \in \mathbb{F}_{3^{12}}$, the expected cost of computing logarithms of all quadratics in $\mathcal{B}_{2,u}$ is $2^{39}M_{q^2}$ for the computation of $\mathcal{H}(Q)$, and $2^{48}M_r$ for the linear algebra. Note that the number of columns in $\mathcal{H}(Q)$ can be halved since the logarithms of all reducible quadratics are known. Since the expected number of relations obtainable is

$$\frac{N_{q^2}(1,6)}{q^{12}} \cdot (q^3 + q) \approx \frac{1}{719.98} \cdot (q^3 + q) \approx q^2 + 6659,$$

one can expect that the matrix $\mathcal{H}(Q)$ will have full rank.

This step is somewhat parallelizable on conventional computers since each set $\mathcal{B}_{2,u}$ can be handled by a different processor.

4.4. **Continued-fraction descent.** For the continued-fraction descent, we selected $m = 30$. The expected cost of this descent is $2^{79}M_{q^2}$. This descent can be effectively parallelized. The expected number of distinct irreducible factors of degree (at most) 30 obtained is $2A_{3^{12}}(30, 254) \approx 30$.

4.5. **Classical descent.** Two classical descent stages are employed. In the first stage, we have $D = 30$ and select $m = 15$, $s = 3$, $d_1 = 5$, $d_2 = 5$, and $\delta = 4$. The set of $D + \delta$ pairs

(i,j) selected was

$$([0,3]\times[0,5])\cup\{(4,0),(4,1),(4,2),(4,3),(4,4),(5,0),(5,1),(5,2),(5,3),(5,4)\},$$

yielding $t_1 = 138$ and $t_2 = 143$. Note that inequality (13) is satisfied. The expected cost of the descent for each of the 30 degree-30 polynomials is approximately $2^{52}\cdot S_{q^2}(15,108)$. The expected total number of distinct irreducible polynomials of degree (at most) 15 obtained is approximately 870.

In the second classical descent stage, we have $D = 15$ and select $m = 11$, $s = 3$, $d_1 = 3$, $d_2 = 4$, and $\delta = 4$. The set of $D+\delta$ pairs (i,j) selected was

$$([0,2]\times[0,4])\cup\{(3,0),(3,1),(3,2),(3,3)\},$$

yielding $t_1 = 110$ and $t_2 = 87$. Note that inequality (13) is satisfied. The expected cost of the descent for each of the 870 degree-15 polynomials is approximately $2^{50}\cdot S_{q^2}(11,87)$. The expected total number of distinct irreducible polynomials of degree (at most) 11 obtained is approximately 23,490.

Note that both classical descent stages can be effectively parallelized.

4.6. **QPA descent.** The QPA descent method is then applied to each of the 23,490 degree-11 polynomials Q obtained from the classical descent stage. We have $D = 11$ and $m = 7$. For each Q, the expected number of rows in $\mathcal{H}(Q)$ is 570,172, so we can expect this matrix to have full column rank (namely, $q^2 = 531,441$). For each Q, the expected cost of relation generation is $2^{29}\cdot S_{q^2}(7,33)$ and the cost of the linear algebra is $2^{48}M_r$. Also for each Q, the expected number of distinct polynomials of degree at most 7 obtained is expected to be $A_{q^2}(7,33)\cdot q^2 \approx 2^{22}$. Thus, the total number of distinct polynomials of degree at most 7 obtained after the QPA descent stage is approximately 2^{37}.

4.7. **Gröbner bases descent.** The Gröbner bases descent method is applied to each of the 2^{37} polynomials of degree (at most) 7 obtained after QPA descent. Our experiments were run using Magma v2.19-7 [39] on a 2.9 GHz Intel core i7-3520M.

First, one descends from 7 to 4, i.e., $D = 7$ and $m = 4$. For each degree-7 polynomial Q, we have to solve a system of 26 quadratic polynomial equations in 32 variables over $\mathbb{F}_q$ (cf. (20)). Since the ideal generated by these polynomials typically has dimension greater than 0, we randomly fix some of the variables in the hope of obtaining a 0-dimensional ideal. (More precisely, we added some linear constraints involving pairs of variables, one variable from k_1 and the other from k_2.) Each degree-5 R obtained from the variety of the resulting ideal is tested for 4-smoothness. If no 4-smooth R is obtained, we randomly fix some other subset of variables and repeat. We ran 17,510 Gröbner bases descent experiments with randomly-selected degree-7 polynomials Q. On average, we had to find 1.831 Gröbner bases for each Q. The average number of R's tested for 4-smoothness for each Q was 1.252, which agrees with the expected number $q^{10}/N_{q^2}(4,5) \approx 1.25$. The average time to find each Gröbner basis was 42.0 seconds, and the memory consumption was 64 Mbytes. In total, the expected number of polynomials of degree at most 4 obtained is $2^{37}(q+1+A_{q^2}(4,5)) \approx 2^{47}$.

Next, one descends from 4 to 3, i.e., $D = 4$ and $m = 3$. For each degree-4 polynomial Q, we have to solve a system of 20 quadratic polynomial equations in 28 variables over $\mathbb{F}_q$. We proceed as above, by fixing some of the 28 variables. We ran 1,230,000 Gröbner bases descent experiments with randomly-selected degree-4 polynomials Q. On average, we had

to find 2.361 Gröbner bases for each Q. The average number of R's tested for 3-smoothness for each Q was 1.815, which agrees with the expected number $q^{10}/N_{q^2}(3,5) \approx 1.818$. The average time to find each Gröbner basis was 0.01328 seconds, and the memory consumption was 32 Mbytes. In total, the expected number of polynomials of degree at most 3 obtained is $2^{47}(q+1+A_{q^2}(3,5)) \approx 2^{57}$.

Finally, one descends from 3 to 2, i.e., $D = 3$ and $m = 2$. Since the total number of irreducible monic cubics over $\mathbb{F}_{q^2}$ is approximately $2^{55.5}$, which is less than 2^{57}, we perform the 3 to 2 descent for all irreducible monic cubics. For each such polynomial Q, we have to solve a system of 14 quadratic polynomial equations in 20 variables over $\mathbb{F}_q$. We proceed as above, by fixing some of the 20 variables. We ran 8,100,000 Gröbner bases descent experiments with randomly-selected degree-3 polynomials Q. On average, we had to find 2.026 Gröbner bases for each Q. The average number of R's tested for 2-smoothness for each Q was 1.499, which agrees with the expected number $q^6/N_{q^2}(2,3) \approx 1.5$. The average time to find each Gröbner basis was 0.00125 seconds, and the memory consumption was 32 Mbytes.

4.8. **Overall running time.** The second column of Table 4 gives the running time estimates for the main steps of the new DLP algorithm in three units of time: M_r, M_{q^2}, and seconds. In order to assess the overall time, we make some assumptions about the ratios of these units of time.

First, we shall assume that $M_r/M_{q^2} = 2^8$. To justify this, we observe that two 64-bit unsigned integers can be multiplied using a **Mul** operation on an Intel Core i7 machine in 3 clock cycles, and an integer modulo r can be accommodated in 13 64-bit words. Two such integers can be multiplied using a two-level Karatsuba algorithm using 121 **Mul** operations, and a reduction modulo r can be accomplished using Barrett's algorithm [10] in 222 **Mul** operations, for a total cost of 343 **Mul** operations or 1029 clock cycles. Unlike for 64-bit integer multiplication, there is no native support for $\mathbb{F}_{3^{12}}$ multiplication on an Intel Core i7 machine. However, we expect that a specially designed multiplier could be built to achieve a multiplication cost of 4 clock cycles. While building such a native multiplier would certainly be costly, this expense can be expected to be within the budget of a well-funded adversary who is contemplating implementing the new DLP algorithm. This gives us an M_r/M_{q^2} ratio of approximately 2^8.

Next, since a multiplication modulo r can be done in 1029 clock cycles, we will transform one second on a 2.9 GHz machine (on which the Gröbner bases descent experiments were performed) into $2^{22}M_r$.

Using these estimates, we see from the third column of Table 4 that the overall running time of the new algorithm is approximately $2^{73.7}M_r$. We note that the relation generation, continued-fraction descent, classical descent, and Gröbner bases descent steps, and also the relation generation portion of QPA descent, are effectively parallelizable in the sense that one can essentially achieve a factor-C speedup if C processors are available. On the other hand, the linear system of equations for finding logarithms of linear polynomials, the $3^{12} \approx 2^{19}$ linear systems of equations for finding logarithms of irreducible quadratic polynomials, and the $23,490 \approx 2^{15}$ linear systems of equations in QPA descent enjoy relatively modest benefits from parallelization on conventional computers.

Remark 1. *(caveat emptor)* Although our analysis is concrete rather than asymptotic, it must be emphasized that the analysis makes several heuristic assumptions and approximations. For example, there are the usual heuristic assumptions that certain polynomials encountered are uniformly distributed over the set of all polynomials of the same degree. Furthermore, we have assumed that the matrix $\mathcal{H}(Q)$ in QPA descent indeed has full column rank. Also, our run time analysis only accounts for the number of multiplications and ignores other operations such as additions and memory accesses. Thus, further analysis and experimentation is needed before one can conclude with certainty that the $2^{73.7}M_r$ running time estimate is an accurate measure of the efficiency of the new DLP algorithm for computing logarithms in the order-r subgroup of $\mathbb{F}^*_{3^{6\cdot 509}}$.

Remark 2. *(looseness of our upper bound on the running time)* Remark 1 notwithstanding, our analysis is quite conservative and there are several possible ways in which the upper bound on the running time could be improved. (i) In our estimates for the number of branches in a descent step, we assume that each distinct irreducible polynomial obtained has degree exactly m, whereas in practice many of these polynomials will have degree significantly less than m. Thus, it would appear that our upper bound on the number of nodes in the descent tree is quite loose. (ii) The Gröbner bases descent running times reported in §4.7 can be expected to be significantly improved by a native implementation of the F4 [20] or F5 [21] Gröbner basis finding algorithms optimized for characteristic-three finite fields. (Magma implements the F4 algorithm, but is not optimized for characteristic-three finite fields.) (iii) An optimized Gröbner basis implementation might be successful in performing the descent from $D = 11$ to $D = 6$, thereby replacing the QPA descent from $D = 11$ to $D = 7$ and significantly reducing the number of nodes in the descent tree. (iv) Bernstein's smoothness testing method [11] might be faster in practice than the basic method described in §2.2. (v) Sieving can be expected to significantly speedup the continued-fraction descent stage [6]. (vi) If Wiedemann's algorithm for solving the linear systems of equations is employed, then the M_r unit of measure can be replaced by A_r (where A_r denotes the cost of an *addition* modulo r).

4.9. **Comparisons with Joux-Lercier.** Shinohara et al. [42] estimated that the running time of the Joux-Lercier algorithm [35] for computing discrete logarithms in $\mathbb{F}_{3^{6\cdot 509}}$ is $2^{111.35}$ for the relation generation stage, and $2^{102.69}$ for the linear algebra stage. Since relation generation is effectively parallelizable on conventional computers whereas linear algebra is not, a conservative estimate for the running time of the Joux-Lercier algorithm is $2^{102.69}M_r$. Further justification for discounting the running time of relation generation is that Joux's pinpointing technique [31] can be employed to significantly reduce the relation generation time, while not having a noticeable impact on the linear algebra time.

Thus, we see that the new algorithm with an overall running time of $2^{73.7}M_r$, is about 2^{29} times faster for computing logarithms in $\mathbb{F}_{3^{6\cdot 509}}$ than the previous fastest algorithm. Moreover, if one had access to a massive number of processors (e.g., 2^{30} processors), then the bottleneck of the new algorithm is a linear algebra computation with a run time of $2^{48}M_r$, whereas the bottleneck of the Joux-Lercier algorithms remains the linear algebra computation with a run time of $2^{102.69}M_r$.

We believe that these comparisons justify the claim made in the abstract about the weakness of the field $\mathbb{F}_{3^{6\cdot 509}}$, and thereby also the supersingular elliptic curve over $\mathbb{F}_{3^{509}}$ with embedding degree 6.

5. Concluding remarks

Our concrete analysis of the new algorithm of Joux and Barbulescu et al. has shown that the supersingular elliptic curve over $\mathbb{F}_{3^{509}}$ with embedding degree 6 is significantly less resistant to attacks on the elliptic curve discrete logarithm problem than previously believed. Consequently, this elliptic curve is not suitable for implementing pairing-based cryptosystems. Our analysis applies equally well to the supersingular elliptic curve over $\mathbb{F}_{3^{5\cdot 97}}$ with embedding degree 6 that has been proposed for compact hardware implementation of pairing-based cryptosystems by Estibals [19].

An important open question is whether the new algorithm or its implementation can be improved to the extent that the discrete logarithm problem in $\mathbb{F}_{3^{6\cdot 509}}$ can be feasibly solved using existing computer technology or special-purpose hardware [38, 25].

Another important question is whether the new attack is effective for finding discrete logarithms in other small-characteristic finite fields of interest in pairing-based cryptography. Our preliminary analysis suggests that the new algorithm is ineffective for computing discrete logarithms in $\mathbb{F}_{2^{4\cdot 1223}}$. Moreover, the new algorithm only appears advantageous over Joux-Lercier for discrete logarithm computations in $\mathbb{F}_{2^{12\cdot 367}}$ and $\mathbb{F}_{2^{12\cdot 439}}$ in the situation where a massive number of processors (e.g., 2^{30} processors) are available; see Appendix A.

Acknowledgements

We would like to thank Pierre-Jean Spaenlehauer for answering our questions about Gröbner basis finding algorithms.

References

[1] J. Adikari, M. Anwar Hasan and C. Negre, "Towards faster and greener cryptoprocessor for eta pairing on supersingular elliptic curve over $\mathbb{F}_{2^{1223}}$", *Selected Areas in Cryptography – SAC 2012*, LNCS 7707 (2013), 166–183.

[2] L. Adleman and M.-D. Huang, "Function field sieve method for discrete logarithms over finite fields", *Information and Computation*, 151 (1999), 5–16.

[3] O. Ahmadi, D. Hankerson and A. Menezes, "Software implementation of arithmetic in $\mathbb{F}_{3^m}$", *International Workshop on Arithmetic of Finite Fields – WAIFI 2007*, LNCS 4547 (2007), 85–102.

[4] D. Aranha, J. Beuchat, J. Detrey and N. Estibals, "Optimal eta pairing on supersingular genus-2 binary hyperelliptic curves", *Topics in Cryptology – CT-RSA 2012*, LNCS 7178 (2012), 98–115.

[5] R. Barbulescu, C. Bouvier, J. Detrey, P. Gaudry, H. Jeljeli, E. Thomé, M. Videau and P. Zimmermann, "Discrete logarithm in $GF(2^{809})$ with FFS", available at http://eprint.iacr.org/2013/197.

[6] R. Barbulescu and P. Gaudry, personal communication, August 12, 2013.

[7] R. Barbulescu, P. Gaudry, A. Joux and E. Thomé, "A quasi-polynomial algorithm for discrete logarithm in finite fields of small characteristic: Improvements over FFS in small to medium characteristic", available at http://eprint.iacr.org/2013/400.

[8] P. Barreto, S. Galbraith, C. Ó hÉigeartaigh and M. Scott, "Efficient pairing computation on supersingular abelian varieties", *Designs, Codes and Cryptography*, 42 (2007), 239–271.

[9] P. Barreto, H. Kim, B. Lynn and M. Scott, "Efficient algorithms for pairing-based cryptosystems", *Advances in Cryptology – CRYPTO 2002*, LNCS 2442 (2002), 354–368.

[10] P. Barrett, "Implementing the Rivest Shamir and Adleman public key encryption algorithm on a standard digital signal processor", *Advances in Cryptology – CRYPTO '86*, LNCS 263 (1987), 311–323.

[11] D. Bernstein, "How to find small factors of integers", manuscript, 2002; available at http://cr.yp.to/papers/sf.pdf.

[12] J. Beuchat, J. Detrey, N. Estibals, E. Okamoto and F. Rodríguez-Henríquez, "Fast architectures for the η_T pairing over small-characteristic supersingular elliptic curves", *IEEE Transactions on Computers*, 60 (2011), 266–281.

[13] J. Beuchat, E. López-Trejo, L. Martínez-Ramos, S. Mitsunari and F. Rodríguez-Henríquez, "Multi-core implementation of the Tate pairing over supersingular elliptic curves", *Cryptology and Network Security – CANS 2009*, LNCS 5888 (2009), 413–432.

[14] I. Blake, R. Fuji-Hara, R. Mullin and S. Vanstone, "Computing logarithms in finite fields of characteristic two", *SIAM Journal on Algebraic and Discrete Methods*, 5 (1984), 276–285.

[15] D. Boneh, B. Lynn, and H. Shacham, "Short signatures from the Weil pairing", *Journal of Cryptology*, 17 (2004), 297–319.

[16] S. Chatterjee, D. Hankerson and A. Menezes, "On the efficiency and security of pairing-based protocols in the Type 1 and Type 4 settings" *International Workshop on Arithmetic of Finite Fields – WAIFI 2010*, LNCS 6087 (2010), 114–134.

[17] Q. Cheng, D. Wan and J. Zhuang, "Traps to the BGJT-algorithm for discrete logarithms", available at http://eprint.iacr.org/2013/673.

[18] D. Coppersmith, "Fast evaluation of logarithms in fields of characteristic two", *IEEE Transactions on Information Theory*, 30 (1984), 587–594.

[19] N. Estibals, "Compact hardware for computing the Tate pairing over 128-bit-security supersingular curves", *Pairing-Based Cryptography – Pairing 2010*, LNCS 6487 (2010), 397–416.

[20] J. Faugère, "A new efficient algorithm for computing Gröbner bases (F_4)", *Journal of Pure and Applied Algebra*, 139 (1999), 61–88.

[21] J. Faugère, "A new efficient algorithm for computing Gröbner bases without reduction to zero (F_5)", *Proceedings of the 2002 International Symposium on Symbolic and Algebraic Computation* (ISSAC 2002), 2002, 75–83.

[22] S. Galbraith, "Supersingular curves in cryptography", *Advances in Cryptology – ASIACRYPT 2001*, LNCS 2248 (2001), 495–513.

[23] S. Galbraith, K. Harrison and D. Soldera, "Implementing the Tate pairing", *Algorithmic Number Theory – ANTS 2002*, LNCS 2369 (2002), 324–337.

[24] S. Galbraith, K. Paterson and N. Smart, "Pairings for cryptographers", *Discrete Applied Mathematics*, 156 (2008), 3113–3121.

[25] W. Geiselmann, A. Shamir, R. Steinwandt and E. Tromer, "Scalable hardware for sparse systems of linear equations, with applications to integer factorization", *Cryptographic Hardware and Embedded Systems – CHES 2005*, LNCS 3659 (2005), 131–146.

[26] F. Göloğlu, R. Granger, G. McGuire and J. Zumbrägel, "On the function field sieve and the impact of higher splitting probabilities: Application to discrete logarithms in $\mathbb{F}_{2^{1971}}$", *Advances in Cryptology – CRYPTO 2013*, LNCS 8043 (2013), 109–128.

[27] F. Göloğlu, R. Granger, G. McGuire and J. Zumbrägel, "Solving a 6120-bit DLP on a desktop computer", available at http://eprint.iacr.org/2013/306.

[28] R. Granger, D. Page and M. Stam, "Hardware and software normal basis arithmetic for pairing based cryptography in characteristic three", *IEEE Transactions on Computers*, 54 (2005), 852–860.

[29] D. Hankerson, A. Menezes and M. Scott, "Software implementation of pairings", In M. Joye and G. Neven, editors, *Identity-Based Cryptography*, IOS Press, 2008.

[30] T. Hayashi, T. Shimoyama, N. Shinohara and T. Takagi, "Breaking pairing-based cryptosystems using η_T pairing over $GF(3^{97})$", *Advances in Cryptology – ASIACRYPT 2012*, LNCS 7658 (2012), 43–60.

[31] A. Joux, "Faster index calculus for the medium prime case: Application to 1175-bit and 1425-bit finite fields", *Advances in Cryptology – EUROCRYPT 2013*, LNCS 7881 (2013), 177–193.

[32] A. Joux, "A new index calculus algorithm with complexity $L(1/4+o(1))$ in very small characteristic", available at http://eprint.iacr.org/2013/095.

[33] A. Joux, "Discrete logarithm in $GF(2^{6128})$", Number Theory List, May 21 2013.

[34] A. Joux and R. Lercier, "The function field sieve is quite special", *Algorithmic Number Theory – ANTS 2002*, LNCS 2369 (2002), 431–445.

[35] A. Joux and R. Lercier, "The function field sieve in the medium prime case" *Advances in Cryptology – EUROCRYPT 2006*, LNCS 4004 (2006), 254–270.

[36] B. LaMacchia and A. Odlyzko, "Solving large sparse linear systems over finite fields", *Advances in Cryptology – CRYPTO '90*, LNCS 537 (1991), 109–133.
[37] A. Lenstra, "Unbelievable security: Matching AES security using public key systems", *Advances in Cryptology – ASIACRYPT 2001*, LNCS 2248 (2001), 67–86.
[38] A. Lenstra, A. Shamir, J. Tomlinson and E. Tromer, "Analysis of Bernstein's factorization circuit", *Advances in Cryptology – ASIACRYPT 2002*, LNCS 2501 (2002), 1–26.
[39] Magma v2.19-7, http://magma.maths.usyd.edu.au/magma/.
[40] Maple 17, http://www.maplesoft.com/products/maple/.
[41] D. Page, N. Smart and F. Vercauteren, "A comparison of MNT curves and supersingular curves", *Applicable Algebra in Engineering, Communication and Computing*, 17 (2006), 379–392.
[42] N. Shinohara, T. Shimoyama, T. Hayashi and T. Takagi, "Key length estimation of pairing-based cryptosystems using η_T pairing", *Information Security Practice and Experience – ISPEC 2012*, LNCS 7232 (2012), 228–244.
[43] D. Wiedemann, "Solving sparse linear equations over finite fields", *IEEE Transactions on Information Theory*, 32 (1986), 54–62.

Appendix A. Computing discrete logarithms in $\mathbb{F}_{2^{12\cdot 367}}$

We present a concrete analysis of the DLP algorithm described in §3 for computing discrete logarithms in $\mathbb{F}_{2^{12\cdot 367}}$. In fact, this field is embedded in the quadratic extension field $\mathbb{F}_{2^{24\cdot 367}}$, and it is the latter field where the DLP algorithm of §3 is executed. Thus, we have $q = 2^{12}$, $n = 367$, and $N = 2^{24\cdot 367} - 1$. Note that $2^{24\cdot 367} \approx 2^{8808}$. We wish to find $\log_g h$, where g is a generator of $\mathbb{F}^*_{2^{24\cdot 367}}$ and $h \in \mathbb{F}^*_{2^{24\cdot 367}}$.

As mentioned in §1, our main motivation for finding discrete logarithms in $\mathbb{F}_{2^{12\cdot 367}}$ is to attack the discrete logarithm problem in $\mathrm{Jac}_C(\mathbb{F}_{2^{367}})$, where C is the supersingular genus-2 curve $Y^2 + Y = X^5 + X^3$ with $\#\mathrm{Jac}_C(\mathbb{F}_{2^{367}}) = 13 \cdot 7170258097 \cdot r$, and where $r = (2^{734} + 2^{551} + 2^{367} + 2^{184} + 1)/(13 \cdot 7170258097)$ is a 698-bit prime. Note that $r^2 \nmid N$. The discrete logarithm problem in the order-r subgroup of $\mathrm{Jac}_C(\mathbb{F}_{2^{367}})$ can be efficiently reduced to the discrete logarithm problem in the order-r subgroup of $\mathbb{F}^*_{2^{12\cdot 367}}$. Thus, we will henceforth assume that h has order r and that we only need to find $\log_g h \bmod r$. An immediate consequence of this restriction is that all the linear algebra in the new algorithm has to be performed modulo the 698-bit r instead of modulo the 8808-bit N.

The parameters for each step of the algorithm were chosen in order to balance the running time of the steps. We also took into account the degree to which each step could be parallelized on conventional computers. A summary of the parameter choices for the descent is given in Figure 2. The costs of each step are given in Table 5.

If $f \in \mathbb{F}_q[X]$ has degree d, then $X^q \bmod f$ can be determined by first precomputing $X^4, X^8, \ldots, X^{4(d-1)} \bmod f$ by repeated multiplication by X. Thereafter, computing the fourth power of a polynomial modulo f can be accomplished by computing fourth powers of the coefficients of the polynomial, and then multiplying the precomputed polynomials by these fourth powers (and adding the results). In this way, we get a loose upper bound of $4d^2 + 11d^2 = 15d^2$ $\mathbb{F}_{2^{24}}$-multiplications of the cost to compute $X^{2^{24}} \bmod f$, and the total cost for testing m-smoothness of f (cf. §2.2) becomes

$$S_{2^{24}}(m, d) = 2d^2(d + m + 7.5) \quad \mathbb{F}_{2^{24}}\text{-multiplications}. \tag{22}$$

A.1. **Setup.** We chose the representations

$$\mathbb{F}_{2^{12}} = \mathbb{F}_2[U]/(U^{12} + U^7 + U^6 + U^5 + U^3 + U + 1)$$

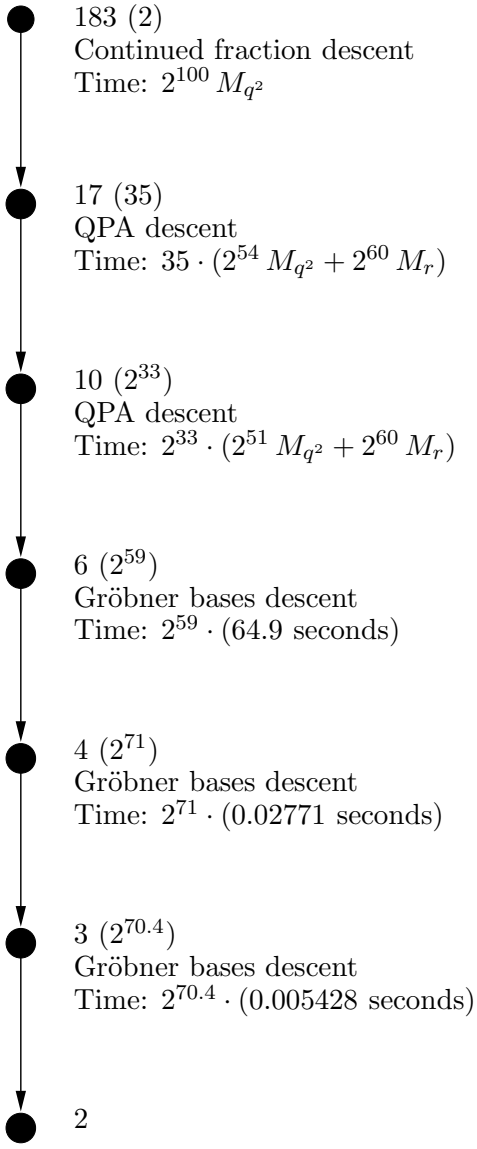

FIGURE 2. A typical path of the descent tree for computing an individual logarithm in $\mathbb{F}_{2^{24\cdot 367}}$ ($q = 2^{12}$). The numbers in parentheses next to each node are the expected number of nodes at that level. 'Time' is the expected time to generate all nodes at a level.

and

$$\mathbb{F}_{2^{24}} = \mathbb{F}_{2^{12}}[V]/(V^2 + U^{152}V + U^{3307}).$$

We selected

$$h_0 = (U^{2111}V + U^{2844})X^2 + (U^{428}V + U^{2059})X + (U^{1973}V + U^{827}) \in \mathbb{F}_{2^{24}}[X]$$

and

$$h_1 = X + U^{2904}V + U^{401} \in \mathbb{F}_{2^{24}}[X],$$

Finding logarithms of linear polynomials		
Relation generation	$2^{35}M_{q^2}$	$2^{29}M_r$
Linear algebra	$2^{60}M_r$	$2^{60}M_r$
Finding logarithms of irreducible quadratic polynomials		
Relation generation	$2^{24} \cdot 2^{44}M_{q^2}$	$2^{62}M_r$
Linear algebra	$2^{24} \cdot 2^{60}M_r$	$2^{84}M_r$
Descent		
Continued-fraction (183 to 17)	$2^{100}M_{q^2}$	$2^{94}M_r$
QPA (17 to 10)	$35 \cdot (2^{54}M_{q^2} + 2^{60}M_r)$	$2^{65}M_r$
QPA (10 to 6)	$2^{33} \cdot (2^{51}M_{q^2} + 2^{60}M_r)$	$2^{93}M_r$
Gröbner bases (6 to 4)	$2^{59} \cdot (64.9$ seconds$)$	$2^{87}M_r$
Gröbner bases (4 to 3)	$2^{71} \cdot (0.02771$ seconds$)$	$2^{88}M_r$
Gröbner bases (3 to 2)	$2^{70.4} \cdot (0.005428$ seconds$)$	$2^{85}M_r$

TABLE 5. Estimated costs of the main steps of the new DLP algorithm for computing discrete logarithms in $\mathbb{F}_{2^{24\cdot 367}}$ ($q = 2^{12}$). M_r and M_{q^2} denote the costs of a multiplication modulo the 698-bit prime $r = (2^{734} + 2^{551} + 2^{367} + 2^{184} + 1)/(13 \cdot 7170258097)$ and a multiplication in $\mathbb{F}_{2^{24}}$. We use the cost ratio $M_r/M_{q^2} = 2^6$, and also assume that 2^{22} multiplications modulo r can be performed in 1 second (cf. §A.8).

and $I_X \in \mathbb{F}_{2^{24}}[X]$ to be the monic degree-367 irreducible factor of $h_1X^{2^{12}} - h_0$. The other irreducible factors of $h_1X^{2^{12}} - h_0$ have degrees 23, 103, 162, 298 and 3144.

A.2. **Finding logarithms of linear polynomials.** The factor base $\mathcal{B}_1$ has size 2^{24}. The cost of relation generation is approximately $2^{35}M_{q^2}$, whereas the cost of the linear algebra is approximately $2^{60}M_r$.

A.3. **Finding logarithms of irreducible quadratic polynomials.** For each $u \in \mathbb{F}_{2^{24}}$, the expected cost of computing logarithms of all quadratics in $\mathcal{B}_{2,u}$ is $2^{44}M_{q^2}$ for the computation of $\mathcal{H}(Q)$, and $2^{60}M_r$ for the linear algebra.

A.4. **Continued-fraction descent.** For the continued-fraction descent, we selected $m = 17$. The expected cost of this descent is $2^{100}M_{q^2}$. The expected number of distinct irreducible factors of degree (at most) 17 obtained is $2A_{2^{24}}(17, 183) \approx 35$.

A.5. **Classical descent.** When applicable, classical descent is preferable to QPA descent since the former produces a far smaller number of branches when descending from a polynomial Q. However, in the field under consideration we have $q = 2^{12}$, so at least one of X^{2^s} and $X^{2^{12-s}}$ has degree at least 64. This means that at least one of the polynomials $R_1 = R(X, X^{2^s})$ and $R_2 = R'(X^{2^{12-s}}, h_0)$ (cf. §3.5) has very large degree, rendering classical descent ineffective.

A.6. **QPA descent.** The QPA descent method is applied to each of the 35 degree-17 polynomials Q obtained from the continued-fraction descent stage. We have $D = 17$ and $m = 10$. For each Q, the expected cost of relation generation is $2^{54}M_{q^2}$ and the cost of the linear algebra is $2^{60}M_r$. Also for each Q, the expected number of distinct polynomials of degree at most 6 obtained is expected to be $A_{q^2}(10, 51) \cdot q^2 \approx 2^{28}$. Thus, the total number

of distinct polynomials of degree at most 10 obtained after the first QPA descent stage is approximately 2^{33}.

The QPA descent method is then applied to each of these 2^{33} degree-10 polynomials Q. We have $D = 10$ and $m = 6$. For each Q, the expected cost of relation generation is $2^{51}M_{q^2}$ and the cost of the linear algebra is $2^{60}M_r$. Also for each Q, the expected number of distinct polynomials of degree at most 6 obtained is expected to be $A_{q^2}(6, 30) \cdot q^2 \approx 2^{26}$. Thus, the total number of distinct polynomials of degree at most 6 obtained after the second QPA descent stage is approximately 2^{59}.

A.7. **Gröbner bases descent.** The Gröbner bases descent method is applied to each of the 2^{59} polynomials of degree (at most) 6 obtained after QPA descent. Our experiments were run using Magma v2.19-7 [39] on a 2.9 GHz Intel core i7-3520M.

First, one descends from 6 to 4, i.e., $D = 6$ and $m = 4$. For each degree-6 polynomial Q, we have to solve a system of 26 quadratic polynomial equations in 34 variables over $\mathbb{F}_q$ (cf. (20)). After fixing some variables, each degree-6 R obtained from the variety of the resulting ideal is tested for 4-smoothness. If no 4-smooth R is obtained, we randomly fix some other subset of variables and repeat. We ran 11,810 Gröbner bases descent experiments with randomly-selected degree-6 polynomials Q. On average, we had to find 2.112 Gröbner bases for each Q. The average number of R's tested for 4-smoothness for each Q was 1.585, which agrees with the expected number $q^{12}/N_{q^2}(4, 6) \approx 1.579$. The average time to find each Gröbner basis was 30.74 seconds. In total, the expected number of polynomials of degree at most 4 obtained is $2^{59}(q + 1 + A_{q^2}(4, 6)) \approx 2^{71}$.

Next, one descends from 4 to 3, i.e., $D = 4$ and $m = 3$. For each degree-4 polynomial Q, we have to solve a system of 20 quadratic polynomial equations in 28 variables over $\mathbb{F}_q$. We proceed as above, by fixing some of the 28 variables. We ran 3,608,000 Gröbner bases descent experiments with randomly-selected degree-4 polynomials Q. On average, we had to find 2.362 Gröbner bases for each Q. The average number of R's tested for 3-smoothness for each Q was 1.817, which agrees with the expected number $q^{10}/N_{q^2}(3, 5) \approx 1.818$. The average time to find each Gröbner basis was 0.01173 seconds. In total, the expected number of polynomials of degree at most 3 obtained is $2^{71}(q + 1 + A_{q^2}(3, 5)) \approx 2^{83}$.

Finally, one descends from 3 to 2, i.e., $D = 3$ and $m = 2$. Since the total number of irreducible monic cubics over $\mathbb{F}_{q^2}$ is approximately $2^{70.4}$, which is less than 2^{83}, we perform the 3 to 2 descent for all monic irreducible cubics. For each such polynomial Q, we have to solve a system of 14 quadratic polynomial equations in 20 variables over $\mathbb{F}_q$. We proceed as above, by fixing some of the 20 variables. We ran 1,080,000 Gröbner bases descent experiments with randomly-selected degree-3 polynomials Q. On average, we had to find 2.024 Gröbner bases for each Q. The average number of R's tested for 2-smoothness for each Q was 1.5, which agrees with the expected number $q^6/N_{q^2}(2, 3) \approx 1.5$. The average time to find each Gröbner basis was 0.002682 seconds.

A.8. **Overall running time.** In order to assess the overall time, we make some assumptions about the ratios of units of time used in Table 5, namely M_r, M_{q^2}, and seconds.

First, we shall assume that $M_r/M_{q^2} = 2^6$. To justify this, we use estimates similar to the ones in §4.8. An integer modulo r can be accommodated in 11 64-bit words. Two such integers can be multiplied using a one-level Karatsuba algorithm using 73 **Mul** operations, and a reduction modulo r can be accomplished using Barrett's algorithm in 166 **Mul** operations, for a total cost of 239 **Mul** operations or 717 clock cycles. Using the carry-less

multiplication instruction PCLMULQDQ, a multiplication in $\mathbb{F}_{2^{12}}$ can be performed at a price of 3-4 clock cycles, and we estimate that the cost of a field multiplication in $\mathbb{F}_{2^{24}}$ to be approximately 15 clock cycles. This gives us an M_r/M_{q^2} ratio of approximately 2^6.

Next, since a multiplication modulo r can be done in 717 clock cycles, we will transform one second on a 2.9 GHz machine (on which the Gröbner bases descent experiments were performed) into $2^{22}M_r$.

Using these estimates, we see from the third column of Table 4 that the overall running time of the new algorithm is approximately $2^{94.6}M_r$. As with the case of $\mathbb{F}_{3^{12\cdot 509}}$, the relation generation, continued-fraction descent, classical descent, and Gröbner bases descent steps, and also the relation generation portion of QPA descent, are effectively parallelizable on conventional computers, whereas the linear system of equations for finding logarithms of linear polynomials, the 2^{24} linear systems of equations for finding logarithms of irreducible quadratic polynomials, and the 2^{33} linear systems of equations in QPA descent enjoy relatively modest benefits from parallelization on conventional computers.

A.9. **Comparisons with Joux-Lercier.** The Joux-Lercier algorithm [35] with pinpointing [31] is an alternative method for computing discrete logarithms in the order-r subgroup of $\mathbb{F}^*_{2^{12\cdot 367}}$. The algorithm works with two polynomial representations of $\mathbb{F}_{2^{12\cdot 367}}$.

The factor base can be taken to be the set of all monic irreducible polynomials of degree at most 4 over $\mathbb{F}_{2^{12}}$ in each of the two representations. The action of the 2^{12}-power Frobenius is used to reduce the factor base size by a factor of 12, yielding a factor base of size $2^{43.4}$. Taking $d_1 = 37$ and $d_2 = 10$ (see Section 2 of [31] for the definitions of d_1 and d_2), the running time of relation generation is approximately $2^{94.0}M_q$, where M_q denotes the cost of a multiplication in $\mathbb{F}_{2^{12}}$ (cf. Section 4 of [31]). The (sparse) matrix in the linear algebra stage has $2^{43.4}$ rows and columns, and approximately 28 nonzero entries per row. Using standard techniques for solving sparse systems of linear equations [36], the expected cost of the linear algebra is approximately $2^{91.6}M_r$. Taking $M_r/M_q \approx 2^{7.5}$ gives a running time of $2^{86.5}M_r$ for relation generation and a total running time of $2^{91.6}M_r$.

Thus, we see that the total running time of the Joux-Lercier algorithm is less than the total running time of $2^{94.6}M_r$ of the new algorithm. However, when a very large number of conventional computers is available, the bottleneck in the new algorithm is the solution of 2^{33} independent linear systems each requiring time $2^{60}M_r$, whereas the bottleneck in the Joux-Lercier algorithm is the solution of a single linear system requiring time $2^{91.6}M_r$. Since the former is more amenable to parallelization on a very large network on conventional computers, a reasonable conclusion is that the new algorithm is more effective than Joux-Lercier.

To lend further weight to this conclusion, we observe that special-purpose hardware for solving the relatively-small linear systems of equations in the new algorithm can reasonably be expected to be built at a cost that is well within the budget of a well-funded organization. In 2005, Geiselmann et al. [25] estimated that the cost of special-purpose hardware for solving a linear system where the matrix has 2^{33} rows and columns, and approximately 2^7 nonzero entries (integers modulo 2) per row would be approximately U.S. $2 million; the linear system would be solvable in 2.4 months. For $\mathbb{F}_{2^{12\cdot 367}}$, each matrix in the new algorithm has 2^{24} rows and columns, and approximately 2^{12} nonzero entries (integers modulo r) per row. On the other hand, the cost of special-purpose hardware for solving the linear system encountered in the Joux-Lercier algorithm would be prohibitive.

Our conclusions about the relative weakness of $\mathbb{F}_{2^{12\cdot 367}}$ for discrete logarithm cryptography also apply to the field $\mathbb{F}_{2^{12\cdot 439}}$. All these conclusions are subject to the caveats in Remark 1 in §4.8.

Computer Science Department, CINVESTAV-IPN
E-mail address: gora.adj@gmail.com

Department of Combinatorics & Optimization, University of Waterloo
E-mail address: ajmeneze@uwaterloo.ca

Computer Science Department, CINVESTAV-IPN
E-mail address: thomaz.figueiredo@gmail.com

Computer Science Department, CINVESTAV-IPN
E-mail address: francisco@cs.cinvestav.mx