LMS J. Comput. Math. 17 (Special issue A) (2014) 230–246
doi:10.1112/S1461157014000369

The multiple number field sieve for medium- and high-characteristic finite fields

Razvan Barbulescu and Cécile Pierrot

Abstract

In this paper we study the discrete logarithm problem in medium- and high-characteristic finite fields. We propose a variant of the number field sieve (NFS) based on numerous number fields. Our improved algorithm computes discrete logarithms in $\mathbb{F}_{p^n}$ for the whole range of applicability of the NFS and lowers the asymptotic complexity from $L_{p^n}(1/3, (128/9)^{1/3})$ to $L_{p^n}(1/3, (2^{13}/3^6)^{1/3})$ in the medium-characteristic case, and from $L_{p^n}(1/3, (64/9)^{1/3})$ to $L_{p^n}(1/3, ((92+26\sqrt{13})/27)^{1/3})$ in the high-characteristic case.

1. *Introduction*

Since 1976, many popular public key cryptosystems have been based on discrete exponentiation, including not only key exchange [8] but also signature [9], identification-based protocols [11, 19] and encryption [17]. More recently, the introduction of pairing-based cryptography [3, 12] has enlarged the scope of cryptographic schemes related to the discrete logarithm problem (DLP).

One very important challenge is to evaluate the security of these protocols, which requires estimation of the complexity of the DLP in the relevant groups. We focus in this article on the DLP in the multiplicative group of invertible elements in a finite field. We recall that, given a finite field $\mathbb{F}_{p^n}$, a generator g of $\mathbb{F}_{p^n}^*$ and an element $h \in \mathbb{F}_{p^n}^*$, we say that we solve the DLP in $\mathbb{F}_{p^n}$ if we recover the smallest positive integer x such that $g^x = h$.

Current discrete logarithm algorithms for finite fields $\mathbb{F}_{p^n}$ vary with the relative sizes of the characteristic p and the extension degree n. More precisely, finite fields split into three groups and each one corresponds to a given algorithm: when p is small compared to p^n, we use the quasi-polynomial algorithm [4]; when p is medium we apply the high-degree variant of the number field sieve (NFS-HD), and when p is large we consider the classical number field sieve (NFS), both presented in [14]. In prime fields, or in the case of finite fields of constant extension degree, one can apply other variants of the NFS [13, 16, 20]. Some particularities also appear for small characteristic, but they are not covered here. Yet, the various complexities that come out are all heuristic and sub-exponential. In this article we focus on the two cases which presently offer the best security: the medium-characteristic case (which is currently the hardest part of the DLP in finite fields) and the high-characteristic case. Our goal is to devise a variant of the NFS which has a smaller asymptotic complexity in both cases.

The algorithm we propose is inspired by two variants of the NFS. The first one, Coppersmith's modification [6], was designed for integer factorization, whereas the second

Received 27 February 2014; revised 23 May 2014.

2010 Mathematics Subject Classification 11T99 (primary), 11Y16 (secondary).

Contributed to the Algorithmic Number Theory Symposium XI, GyeongJu, Korea, 6–11 August 2014.

The first author's work was started when working at LORIA (CNRS, INRIA, Univ. Lorraine) and completed as a visiting researcher at LIX (CNRS, INRIA, École Polytechnique).

The second author's work was hosted by Ouragan Team, Institut Mathématique de Jussieu, and was funded by CNRS, Direction Générale de l'Armement, and INRIA.

one was put forward by Matyukhin to compute discrete logarithms in prime fields [16]. Both variants rely on the idea that, instead of a pair, one should use a larger set of number fields. Although Matyukhin's improvement has not been translated into a real-life speedup, it enables a reduction in the complexity of the DLP in prime fields. Without going into details, a drawback of Matyukhin's variant is due to the heterogenous roles played by the two number fields in the NFS. Basing his variant on the classical base-m method of polynomial selection, Matyukhin selects as first polynomial (related to the first number field) a polynomial g of degree 1, whereas the set of other polynomials f_i $(i = 1, 2, \ldots)$, defining the other number fields, are of higher degree. He then collects relations between elements that are smooth (in some sense that will be detailed later) in two number fields: the one defined by g and the other defined by a polynomial f_i.

Our aim in this article is twofold. First, we extend the scope of Matyukhin's variant from prime fields to all high-characteristic finite fields, simply by recalling a way to select polynomials that was not known in 2003 when Matyukhin began to adapt Coppersmith's modification to discrete logarithms. Numerous methods for the polynomial selection have been proposed since 2006, but they all produce unbalanced norms: the specific role of the first number field persists. This lack of homogeneity explains why we obtain (only) a little improvement in the complexity, going down from $L_{p^n}(1/3, (64/9)^{1/3})$, where $(64/9)^{1/3} \approx 1.92$, to

$$L_{p^n}(1/3, ((92 + 26\sqrt{13})/27)^{1/3}),$$

where $((92+26\sqrt{13})/27)^{1/3} \approx 1.90$. This second constant was exactly the one obtained in [16] in the case of prime fields.

Second, we propose a variation for medium-characteristic finite fields which leads to a better proportional improvement. The main idea in this case is to rely on the polynomial selection of NFS-HD that permits to balance both the degrees of the two polynomials and the sizes of their coefficients. Making linear combinations of these two polynomials, we obtain a large set of polynomials (thus of number fields) which play the same role. As a consequence, we are able to collect elements that are smooth in any pair of number fields. This allows to lower the asymptotic complexity of NFS-HD, from $L_{p^n}(1/3, (128/9)^{1/3})$ where $(128/9)^{1/3} \approx 2.42$, to:

$$L_{p^n}(1/3, (2^{13}/3^6)^{1/3}),$$

where $(2^{13}/3^6)^{1/3} \approx 2.24$. Considering both the medium- and high-characteristic cases, we suggest naming this algorithm the *multiple number field sieve* (MNFS) as a shorthand form of the name used in [10]. Note that our algorithm ranges in the same category as Coppersmith's and Matyukhin's variants but, as explained by Bernstein, they are different from the multiple *polynomial* sieving technique [2].

As a complement, this new algorithm is an opportunity to give a detailed analysis of the fact that the runtime of the individual logarithm phase is negligible with respect to the total runtime of the NFS. As far as we know, even though this is a classical result, the precise analysis has not been done in the literature.

The paper is organized as follows. In § 2 we introduce our tools and notation. In § 3 we present a refresher on the number field sieve. In § 4 we introduce the multiple number field sieve: we detail the medium- and high-characteristic cases and explain the particular benefit obtained for the medium case. We continue in § 5 with the asymptotic complexity analysis. Finally, in § 6, we give a toy example of our algorithm.

2. *Tools and notation*

If $f \in \mathbb{Z}[x]$ is an irreducible polynomial, K the number field of f and θ a complex root of f, then, for any polynomial $\phi \in \mathbb{Z}[x]$, the norm $\mathrm{N}(\phi(\theta))$ satisfies $\mathrm{Res}(\phi, f) = \pm f_d^{\deg \phi} \mathrm{N}(\phi(\theta))$,

https://doi.org/10.1112/S1461157014000369 Published online by Cambridge University Press

where f_d is the leading coefficient of f. Since we treat f_d together with small primes, we make no distinction in smoothness estimate between norms and resultants. Let $\|f\|_\infty$ be the largest coefficients of f in absolute value. We use the upper bound on the resultant:

$$|\mathrm{Res}(f,\phi)| \leqslant (\deg f + \deg \phi)! \cdot \|f\|_\infty^{\deg \phi} \cdot \|\phi\|_\infty^{\deg f}. \tag{2.1}$$

When dealing with index calculus algorithms it is handy to use the following notation:

$$L_q(\alpha, c) = \exp((c + o(1)) \cdot (\log q)^\alpha (\log\log q)^{1-\alpha}),$$

where α and c are constants such that $0 \leqslant \alpha \leqslant 1$ and $c > 0$, log denotes the natural logarithm and $o(1)$ implies $q \to \infty$. The notation $L_q(\alpha)$ is also used when the constant c is not explicitly specified. This notation appears in the complexity analysis and comes from the need to estimate the smoothness of integers. We recall that, given an integer y, an integer x is called y-smooth if it can be written as a product of factors less than y. The main tool when estimating the time needed to collect smooth numbers is the following theorem.

THEOREM 2.1 (Canfield, Erdős and Pomerance [5]). *Let $\psi(x, y)$ denote the number of positive integers up to x which are y-smooth. If $\epsilon > 0$ and $3 \leqslant u \leqslant (1-\epsilon)\log x/\log\log x$, then $\psi(x, x^{1/u}) = xu^{-u+o(u)}$.*

The previous notation permits us to simplify the use of this theorem. If we write the two integers x and y with the L_q-notation, we obtain a helpful corollary.

COROLLARY 2.2. *Let $(\alpha_1, \alpha_2, c_1, c_2) \in [0,1]^2 \times [0,\infty)^2$ be four reals such that $\alpha_1 > \alpha_2$ or such that $\alpha_1 = \alpha_2$ and $c_1 > c_2$. Let $\mathcal{P}$ denote the probability that a random positive integer below $x = L_q(\alpha_1, c_1)$ splits into primes less than $y = L_q(\alpha_2, c_2)$. Then*

$$\mathcal{P}^{-1} = L_q(\alpha_1 - \alpha_2, (\alpha_1 - \alpha_2)c_1 c_2^{-1}).$$

3. *A short refresher on discrete logarithms in medium- and high-characteristic finite fields*

In the sequel, $Q = p^n$ denotes the cardinality of the finite field being considered, p its characteristic and n the extension degree relatively to the base field. The number field sieve is the state-of-art algorithm for discrete logarithm in both medium- and high-characteristic finite fields, in its high-degree and classical variants, respectively.

3.1. *The medium-characteristic case: $p = L_Q(l_p, c_p)$ with $1/3 \leqslant l_p < 2/3$*

We first recall the high-degree variant (NFS-HD) as proposed in [14].

Setup: general setting. In order to compute discrete logarithms in $\mathbb{F}_{p^n}$, a degree-n extension of the base field $\mathbb{F}_p$, we start by choosing two polynomials f_1 and f_2 in $\mathbb{Z}[X]$ with a common root m in $\mathbb{F}_{p^n}$. In other words, we choose f_1 and f_2 such that the greatest common divisor of these two polynomials has an irreducible factor of degree n over $\mathbb{F}_p$. As a consequence, we can draw the commutative diagram in Figure 1.

Let $\mathbb{Q}(\theta_1)$ denote $\mathbb{Q}[X]/(f_1(X))$ and $\mathbb{Q}(\theta_2)$ denote $\mathbb{Q}[X]/(f_2(X))$, the two number fields defined by f_1 and f_2, *that is*, θ_1 and θ_2 are roots of these polynomials in $\mathbb{C}$.

Choice of polynomials. The first option put forward in [14] is to choose $f_1 \in \mathbb{Z}[X]$ as a degree-n polynomial, with small coefficients and irreducible over $\mathbb{F}_p$, while f_2 is defined as the polynomial $f_1 + p$.

https://doi.org/10.1112/S1461157014000369 Published online by Cambridge University Press

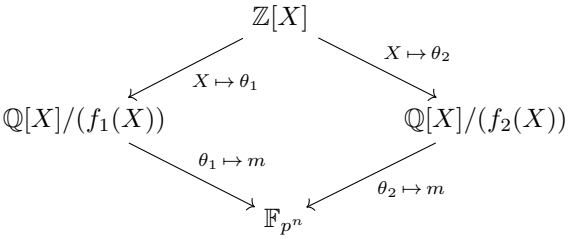

FIGURE 1. *Commutative diagram of NFS.*

In order to balance the size of the norms computed during the algorithm, another approach is also mentioned. This variant uses continued fractions and involves changing the polynomial selection such that the coefficients of both polynomials are of size $O(\sqrt{p})$. This is developed in § 4.2.

Relation collection[†]. The first phase creates relations in the finite field by sieving on polynomials of degree $t-1$. More precisely, we first set the following two bounds: S the sieve limit and B the smoothness bound. Then we consider all t-tuples of coprime integers $(a_0, \ldots, a_{t-1}) \in [1, S] \times [-S, S]^{t-1}$ such that, for $\phi(x) = \sum_{j=0}^{t-1} a_i x^i$, the norms $\mathrm{N}(\phi(\theta_1))$ and $\mathrm{N}(\phi(\theta_2))$ are both B-smooth. After some post-processing described in [14], each such t-tuple yields a linear equation between 'logarithms of ideals' coming from both number fields and belonging to the smoothness base.

Linear algebra. Once the relation collection phase is complete, we solve the resulting sparse system of equations modulo $p^n - 1$, the cardinality of $\mathbb{F}_{p^n}^*$, and recover 'logarithms of ideals' in the smoothness base. To be more precise, the linear algebra is done modulo a large factor of this cardinality, while small prime factors are considered separately, using a combination of Pollard rho and Pohlig–Hellman algorithms.

Individual discrete logarithms. Once the relation collection and the linear algebra phase have been performed, we know the logarithms of all the ideals in the smoothness base. In the last stage, also called the descent phase, we compute the discrete logarithm of an arbitrary element in the finite field. The approach proposed in [14] is based on a 'special-**q**' descent as follows.

If z belongs to $\mathbb{F}_{p^n} \simeq \mathbb{F}_p[X]/(f_1(X))$ and is represented by the polynomial $z(X) \in \mathbb{Z}[X]$ with coefficients in $[-p/2, p/2]$, we recall that we denote by $\bar{z}$ the element $z(\theta_1)$ of the number field $\mathbb{Q}(\theta_1)$. In a nutshell, in order to compute the discrete logarithm of an arbitrary element s of $\mathbb{F}_{p^n}$, we proceed in two steps: continued fraction descent (or smoothing) and special-**q** descent. In the continued fraction descent we search for an integer e such that, for $z = s^e$, the norm of $\bar{z}$ is C-smooth and square-free. The second condition implies that only degree-one ideals appear in the factorization of $(\bar{z})$. After finding such a z, we factor the principal ideal generated by $\bar{z}$ into degree-one prime ideals of small norms. Some of them are not in the smoothness base (those whose norm is smaller than C but larger than B). To compute the logarithm of such an ideal $\mathbf{q}$ we start a 'special-**q**' descent, progressively lowering the norm of ideals until we reach B. Finally, we backtrack to recover the logarithm of $\bar{z}$ and consequently the logarithm of s.

[†]This first step is also called the sieving phase in the case of the number field sieve.

https://doi.org/10.1112/S1461157014000369 Published online by Cambridge University Press

Asymptotic complexity. The smoothness base is composed of the 'logarithms' of prime ideals of degree 1 and norm less than a certain smoothness bound of size $L_Q(1/3)$. The sieving space consists of degree-$(t-1)$ polynomials of $\mathbb{Z}[X]$ with bounded coefficients, such that the cardinality of the sieving space is also of size $L_Q(1/3)$. With such parameters, the heuristic asymptotic complexity of NFS-HD as obtained in [14] is

$$L_Q(1/3, (128/9)^{1/3}).$$

This asymptotic complexity is valid for the whole range of finite fields of characteristic $p = L_Q(l_p, c_p)$, with $1/3 \leqslant l_p < 2/3$. Indeed, the extended version given in [15] permits such an analysis even in the (smaller p) boundary case $l_p = 1/3$, when c_p is larger than $(16/9)^{1/3}$.

3.2. *The high-characteristic case: $p = L_Q(l_p, c_p)$ with $l_p > 2/3$*

The classical NFS used for high-characteristic finite fields works like the NFS-HD except for two points. On one hand, we can use a smaller sieving space: it suffices to sieve on linear polynomials. On the other hand, the higher value of the characteristic p requires us to change the setup in favor of a polynomial selection based on lattice reduction. Without giving the whole construction, we just recall that the two polynomials proposed in [14] have respectively degree n and degree d and both coefficient sizes bounded by $p^{n/(d+1)}$, with d a parameter that depends on p^n. The linear algebra and the individual logarithm phase remain the same. Finally, the asymptotic heuristic complexity of NFS as obtained in [14] is

$$L_Q(1/3, (64/9)^{1/3}).$$

3.3. *The boundary case: $p = L_Q(2/3, c_p)$*

Nothing forbids either the NFS-HD or the NFS to be applied in this configuration. Yet, this boundary case is particular since we have to consider both the NFS-HD algorithm and the classical NFS from [14]. Figure 2 shows how complexities vary with c_p in this case and gives the best algorithm to choose, depending on the value of c_p. Moreover, the analysis of the algorithm in the NFS-HD part remains particular because for a fixed value of c_p, the parameter t is constant when Q grows to infinity (this has to be compared with the analysis in the general medium case where t grows with Q).

4. *The multiple number field sieve*

4.1. *From two number fields to numerous number fields*

Let $\mathbb{F}_{p^n}$ denote the finite field in which we want to compute discrete logarithms. As previously shown in Figure 1, the NFS is based on two different paths on a commutative diagram that both lead to $\mathbb{F}_{p^n}$. The modification we propose in this paper involves constructing a diagram with a larger number of paths to the finite field.

Basic idea. We suppose that we are able to represent $\mathbb{F}_{p^n}$ in V different (but compatible) ways such that we can draw the commutative diagram shown in Figure 3. Again, for each $i \in [1, \ldots, V]$, θ_i is a root in $\mathbb{C}$ of the polynomial f_i and $\mathbb{Q}(\theta_i)$ denotes $\mathbb{Q}[X]/(f_i(X))$. To guarantee the commutativity of the diagram, we impose the requirement of a common root $m \in \mathbb{F}_{p^n}$ for all polynomials. We give in § 4.2 a possible method to obtain convenient f_i that are compatible with such a representation of the finite field. At first glance, one might think that we obtain a linear system for each pair of polynomials f_i and f_j with $1 \leqslant i < j \leqslant V$. Yet our idea goes beyond this. Basically, we take advantage of the fact that the linear equations deal with logarithms of elements of $\mathbb{F}_{p^n}$ and thus 'forget' the numerous number fields used to produce them. Hence, one can use multiple fields in order to produce a single matrix.

https://doi.org/10.1112/S1461157014000369 Published online by Cambridge University Press

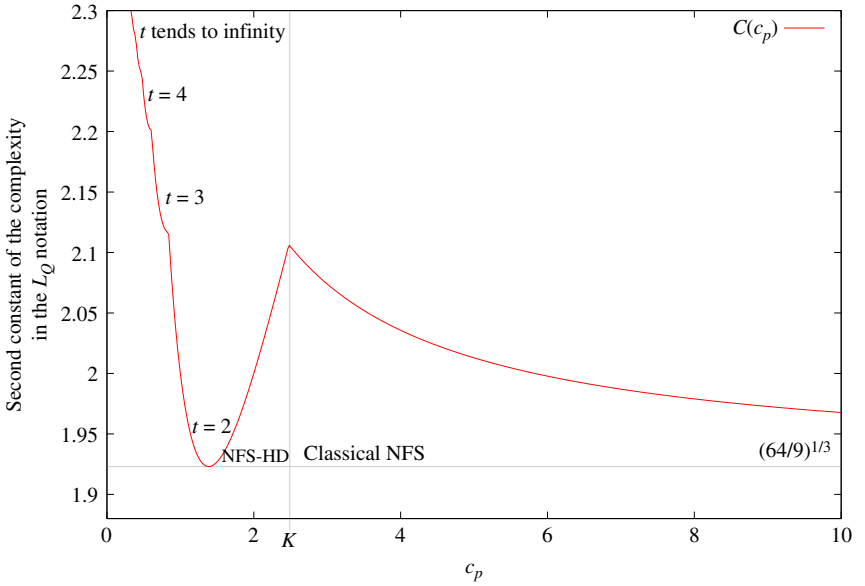

FIGURE 2. *Asymptotic complexities $L_Q(1/3, C(c_p))$ in the boundary case, as a function of c_p with $p = L_Q(2/3, c_p)$. The line $c_p = K \approx 2.5$ is the boundary between domains of NFS-HD and classical NFS. The degree $t - 1$ of the polynomials we sieve on is also indicated on the NFS-HD part of the graph.*

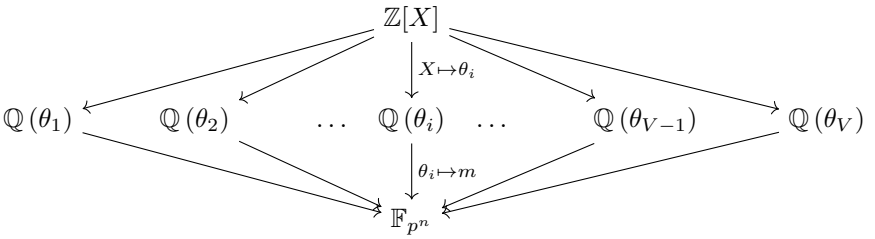

FIGURE 3. *Commutative diagram for the multiple number field sieve.*

MNFS algorithm for medium-characteristic finite fields. With this new diagram in hand, we design an algorithm as follows. As in NFS-HD (presented in § 3), t denotes the number of terms of the polynomials we sieve on, S the sieve limit and B the smoothness bound. We emphasize that we have the same smoothness bound for all number fields. The new smoothness base consists then of the union of all degree-one prime ideals whose norms are smaller than B in each of the V number fields.

We consider all t-term polynomials $\phi = a_0 + \ldots + a_{t-1}x^{t-1}$ with coefficients of size bounded by S. In practice, we sieve only for coprime t-tuples $(a_0, \ldots, a_{t-1})$ for which a_0 is positive. Yet, these two latest restrictions have no effect on the asymptotic complexity and we will not consider them in the sequel. We then collect the polynomials ϕ for which there exists a pair $(i, j) \in [1, \ldots, V]^2$ with $i \neq j$ such that $\mathrm{N}(\phi(\theta_i))$ and $\mathrm{N}(\phi(\theta_j))$ are both B-smooth. In the following, we say that such polynomials are *doubly smooth*. After the same post-processing as in [14], each such polynomial ϕ yields a linear equation between 'logarithms of ideals' coming from the two number fields $\mathbb{Q}(\theta_i)$ and $\mathbb{Q}(\theta_j)$. Hence, we have a linear equation between (a few) logarithms of ideals in the smoothness base.

https://doi.org/10.1112/S1461157014000369 Published online by Cambridge University Press

The linear algebra phase is exactly the same as in § 3. Concerning the individual logarithm phase, although in practice we advise using only two number fields as in § 3, from a theoretical viewpoint, it is important to adapt the descent to our multiple variant of the NFS. Essentially, we proceed in two steps: continued fraction descent, unchanged from § 3, and special-**q** descent. The special-**q** descent differs from § 3 in a single point. The logarithm of each **q** is expressed as a combination of degree-one prime ideals of norm smooth in any of the number fields (and not only the first two). More details on the impact of this individual logarithm phase are given in the Appendix, showing that the individual logarithm phase takes a negligible time.

MNFS algorithm for high-characteristic finite fields. As for the NFS, two modifications appear in high characteristic: the relation collection is simpler since we sieve on linear polynomials, but, in the meantime, polynomial selection has to be redesigned (see § 4.2). As in Matyukhin's variant, we are forced to consider the specific role of the first number field and a third change comes up. We collect polynomials that are B-smooth in the first number field, for some bound B, and B'-smooth in any of the $V-1$ other number fields, where B' is the second smoothness bound. The linear algebra and the individual discrete logarithm phase remain unchanged compared to the medium case.

4.2. *Choice of polynomials*

We explain in this subsection how to choose V polynomials $f_1, \ldots, f_V$ that allow a commutative diagram as in Figure 3. In a nutshell, the main idea is to use two polynomials f_1 and f_2 that have a common root in $\mathbb{F}_{p^n}$ and to create new polynomials by linear combinations.

The medium-characteristic case. (i) Choice of f_1 and f_2. We use the continued fraction polynomial selection method of [**14**] in order to choose our first two polynomials. Proposed initially as a practical improvement, this method allows us to balance both degrees and sizes of coefficients. With this consideration, norms in each number field can be upper-bounded by the same value, which is of key importance for the multiple variant we propose. The selection of f_1 and f_2 works as follows. As in § 3.1, let $f_1 \in \mathbb{Z}[X]$ be a polynomial of degree n, irreducible modulo p, written as

$$f_1 = g + c \cdot h,$$

where g and h are polynomials with small coefficients and c is of size $O(\sqrt{p})$. Thanks to the ontinued fraction algorithm we can write $c \equiv a/b \pmod{p}$ with a and b also of the order of $\sqrt{p}$. We now define f_2 by

$$f_2 \equiv bf_1 \pmod{p}.$$

With this selection f_1 and f_2 have both degree n and coefficients of size $O(\sqrt{p})$.

(ii) Constructing the remaining polynomials. We construct each f_i for $3 \leqslant i \leqslant V$ by linear combination of the two first polynomials:

$$f_i = \alpha_i f_1 + \beta_i f_2.$$

Since we want V such polynomials, the coefficients α_i and β_i are of the size of $\sqrt{V}$. Moreover, we check in § 5 that V is negligible[†] compared to the characteristic p. Thus for $3 \leqslant i \leqslant V$, the polynomial f_i also has coefficients of size $O(\sqrt{p})$.

Moreover, we recall that with this notation f_1 and f_2 have a common root in $\mathbb{F}_{p^n}$. Since each polynomial f_i is a linear combination of these two polynomials, all the V polynomials share a common root in $\mathbb{F}_{p^n}$. This allows us to represent the finite field as in Figure 3, with

[†]Which is reasonable considering that the NFS chooses $V = 2$. In our complexity analysis, we take V of size $L_Q(1/3)$ which is negligible compared to a medium characteristic.

https://doi.org/10.1112/S1461157014000369 Published online by Cambridge University Press

the extra advantage that all the polynomials have the same degree, namely n, and the same size of coefficients, namely $O(\sqrt{p})$.

The high-characteristic case. Even though alternative techniques exist, we select the first two polynomials as in [14]. Given a parameter d, we construct f_1 and f_2 with lattice reduction such that:

$$\deg f_1 = n, \quad \|f_1\|_\infty = Q^{1/(d+1)},$$

$$\deg f_2 = d, \quad \|f_2\|_\infty = Q^{1/(d+1)}.$$

We define the $V-2$ other polynomials $f_i = \alpha_i f_1 + \beta_i f_2$, for α_i and β_i integers of size $\sqrt{V}$. We choose V of size $L_Q(1/3)$ so that it remains negligible compared to $\|f_1\|_\infty$. Hence, all the polynomials have the same size of coefficients, although they do not have the same degree.

4.3. *Benefit of numerous number fields in the medium case*

The main idea of the MNFS is to take advantage of the various paths of our new commutative diagram. As we increase the number of number fields, the cardinality of the smoothness base grows from approximately $2B$ to VB elements. At first glance, one might imagine that this is not a good idea, as enlarging the size of the smoothness base increases the runtime of the linear algebra stage and thus the final asymptotic complexity. In fact, in the medium case, this side effect is more than counterbalanced by a higher probability of an element being doubly smooth. More precisely, call $\mathcal{P}$ the probability of smoothness of an arbitrary integer of the size of $N_{\mathbb{Q}(\theta_i)}(\phi)$, where ϕ is a polynomial in the sieving domain and $N_{\mathbb{Q}(\theta_i)}(\phi)$ the norm of the related element in the number field $\mathbb{Q}(\theta_i)$. With NFS-HD, the probability of ϕ giving a good relation is $\mathcal{P}^2$. In our multiple variant, it suffices to have a pair $(i,j) \in [1,\dots,V]$ with $i \neq j$ for which the norms $N_{\mathbb{Q}(\theta_i)}(\phi)$ and $N_{\mathbb{Q}(\theta_j)}(\phi)$ are smooth in order to get a good relation. Hence, the probability of an element of the sieving space being doubly smooth is $\mathcal{P}^2 V(V-1)/2$. For a V-fold increase in the number of number fields, and thus in the size of the smoothness base, we increase the probability of an element being doubly smooth by a factor of roughly V^2.

This improvement is most welcome since it allows us to lower simultaneously the smoothness bound and the runtime of the relation collection. In simple words, we expect a speedup of the same order of magnitude as V. By choosing a number of number fields of the same order of magnitude as the actual complexity of the discrete logarithm problem in the medium-characteristic case, namely by choosing $V = L_Q(1/3)$, we expect a change of the second constant in the complexity formula $L_Q(1/3, (128/9)^{1/3})$. We leave the details of the asymptotic complexity for the next section.

5. *Asymptotic heuristic complexity*

We first write the relations between p, n, and $Q = p^n$ in the following form:

$$p = \exp(c_p(\log Q)^{l_p}(\log\log Q)^{1-l_p}) \quad \text{and} \quad n = \frac{1}{c_p}\left(\frac{\log Q}{\log\log Q}\right)^{1-l_p}.$$

The analysis of the asymptotic heuristic complexity of the MNFS depends on the interaction of the following parameters: the number of number fields V, the smoothness bound B (or the two smoothness bounds B and B' in the high-characteristic case), the sieving bound S on the coefficients of the elements in the sieving domain and the degree $t-1$ of these elements.

https://doi.org/10.1112/S1461157014000369 Published online by Cambridge University Press

5.1. *The medium-characteristic case*

We deal here with $\mathbb{F}_{p^n}$ where the characteristic can be written as $p = L_Q(l_p, c_p)$ with $1/3 < l_p < 2/3$. We recall that each of the V polynomials has degree n and coefficients of size $O(\sqrt{p})$. We assume that we can express the four parameters V, B, S and t as:

$$V = L_Q(1/3, c_v), \quad B = L_Q(1/3, c_b), \quad S = L_Q(l_p - 1/3, c_s c_p), \quad t = \frac{c_t}{c_p}\left(\frac{\log Q}{\log\log Q}\right)^{2/3-l_p},$$

where c_v, c_b, c_s and c_t will be determined later on in order to minimize the complexity. With these parameters, the total sieving space and the size of the smoothness base are respectively $S^t = L_Q(1/3, c_s c_t)$ and $VB = L_Q(1/3, c_v c_b)$.

We minimize the time of the relation collection and linear algebra stages. For the optimal choice of parameters, the individual logarithm phase is negligible. Thus, we consider that the individual logarithm stage is less important, but the curious reader can find its complexity analysis in the Appendix. To minimize the complexity of the two main phases, we balance the complexities of sieving and of linear algebra. Using Wiedemann's algorithm [**22**], the cost of the linear algebra stage is $(VB)^{2+o(1)}$, so we require that V, B, S and t satisfy $S^t = (VB)^2$. This leads to the first condition:

$$c_s c_t = 2c_v + 2c_b. \tag{5.1}$$

Moreover, since we need to have enough good relations after sieving, we also want to have $S^t\mathcal{P} = VB$, where $\mathcal{P}$ denotes the probability of an element of the sieving space being doubly smooth. Together with the previous remark, this means that we require

$$VB \approx 1/\mathcal{P}. \tag{5.2}$$

Let us give an evaluation of the probability $\mathcal{P}$. For each $1 \leqslant i \leqslant V$, let us note N_i the norm of $\phi(\theta_i)$. Since, for all i we have $\|f_i\|_\infty \approx \sqrt{p}$ and $\deg f_i = n$, we obtain that each norm is bounded by the same value, namely $S^n p^{t/2}$. Hence $\mathcal{P}$ is the probability that an integer less than $S^n p^{t/2}$ splits into primes less than B in at least two number fields. If we call $\mathcal{P}'$ the probability of smoothness of an arbitrary element in one given number field then:

$$\mathcal{P} = \frac{V(V-1)}{2} \cdot \mathcal{P}'^2. \tag{5.3}$$

We now make the usual heuristic hypothesis, and assume that the probability $\mathcal{P}'$, which is equal to the probability of $S^n p^{t/2}$ being B-smooth, follows the theorem of Canfield, Erdős and Pomerance. Besides, the computation of $S^n p^{t/2}$ with the L_Q notation gives $L_Q(2/3, c_s + c_t/2)$. As a result, after plugging our values into Corollary 2.2, we find $\mathcal{P}' = L_Q(1/3, -(c_s + c_t/2))/(3c_b))$. Finally, thanks to (5.3), we have:

$$2\mathcal{P} = L_Q\left(\frac{1}{3}, 2c_v - \frac{2c_s + c_t}{3c_b}\right).$$

Equation (5.2) means that we want $VB/2 = L_Q(1/3, (c_s + c_t/2)/(3c_b) - 2c_v)$. Since the factor 2 is here negligible, this leads to $c_v + c_b = (2c_s + c_t)/(3c_b) - 2c_v$. Hence the second condition we require is

$$9c_v c_b + 3c_b^2 = 2c_s + c_t. \tag{5.4}$$

To simplify our notation, we write $c_t = xc_s$ with $x \geqslant 0$. Together with (5.4) this gives $(9/2)(2c_v + 2c_b)c_b - 6c_b^2 = (2 + x)c_s$. Then, thanks to the equality $2c_b + 2c_v = xc_s^2$ coming from (5.1) we obtain

$$12c_b^2 - 9xc_s^2 c_b + (4 + 2x)c_s = 0. \tag{5.5}$$

https://doi.org/10.1112/S1461157014000369 Published online by Cambridge University Press

The final complexity is $L_Q(1/3, xc_s^2)$, thus we want to minimize xc_s^2 under the constraint (5.5). The discriminant of this quadratic equation in c_b is $\Delta = 3c_s(27x^2c_s^3 - 32\cdot(2+x))$. There exists a solution for c_b if $27x^2c_s^3 \geqslant 32\cdot(2+x)$. The minimal value of c_s that can be taken is thus $[(32(2+x))/27x^2]^{1/3}$. With this choice of c_s the second constant of the complexity becomes $[(32(2+x)^2)/27x^2]^{1/3}$. A short computation with the derivative shows that the complexity is minimal for $x = 2$. This leads to $c_s = 2^{5/3}/3$, $c_b = 2^{4/3}/3$, $c_v = 2^{4/3}/3^2$, and $c_t = 2^{8/3}/3$. Finally, the asymptotic complexity of the MNFS is

$$L_Q\left(\frac{1}{3}, \left(\frac{2^{13}}{3^6}\right)^{1/3}\right).$$

Compare this with the complexity of NFS-HD given in [14], which is $L_Q(1/3, (128/9)^{1/3})$. Our second constant is $(2^{13}/3^6)^{1/3} \approx 2.24$, while $(128/9)^{1/3} \approx 2.42$.

REMARK 5.1. We notice that minimizing the asymptotic complexity leads to a linear dependence not only between c_s and c_t (we have forced $c_t = 2c_s$) but also between c_b and c_v (since we finally get $c_b = 3c_v$). In other words, a natural balance between the smoothness bound and the number of finite fields used in the algorithm appears as

$$B = V^3.$$

5.2. *The boundary case $p = L_Q(2/3, c_p)$*

We consider in this case a family of algorithms indexed by the degree $t-1$ of the polynomials we are sieving on and we compute the asymptotic complexity of each algorithm. Our goal is to obtain the final complexity as a function of c_p and t. The analysis here parallels the previous one, except that the round-off error in t is no longer negligible.

Sieving on polynomials of degree $t-1$. We assume that t is fixed and that we can express V, B and S as $V = L_Q(1/3, c_v), B = L_Q(1/3, c_b)$ and $S = L_Q(1/3, c_s)$, where c_v, c_b and c_s will be determined later[†]. To minimize the final complexity, we want to balance the complexities of the sieving and linear algebra stages. Since the sieving space is $S^t = L_Q(1/3, c_s t)$ and the linear algebra costs approximately $(VB)^2 = L_Q(1/3, 2c_v + 2c_b)$ operations, we require that

$$c_s t = 2c_v + 2c_b. \tag{5.6}$$

We require also that $VB = 1/\mathcal{P}$, where, as before, $\mathcal{P}$ denotes the probability of an element of the sieving space being doubly smooth. Each norm is bounded by $S^n p^{(t-1)/2} = L_Q(2/3, c_s/c_p + c_p(t-1)/2)$. Applying Theorem 2.1 and (5.3), we get $\mathcal{P} = L_Q(1/3, 2c_v - (2c_s + c_p^2(t-1)))/(3c_bc_p))$. This means that we want to have $c_v + c_b = (2c_s + c_p^2(t-1))/(3c_bc_p) - 2c_v$. In other words, the second condition is

$$3c_pc_b^2 + 9c_vc_pc_b - 2c_s - c_p^2(t-1) = 0. \tag{5.7}$$

The final complexity will be $L_Q(1/3, 2(c_b + c_v))$. Our method differs now from the previous analysis. Since we want to minimize the sum $c_b + c_v$, we introduce two variables: $x = c_b + c_v$ and $y = c_b - c_v$. We then have $c_b = (x+y)/2, c_v = (x-y)/2$, and $c_s = 2x/t$. The constraint (5.7) becomes $(3/4)c_p(x+y)^2 + (9/4)c_p(x+y)(x-y) - 4x/t - c_p^2(t-1) = 0$ and, finally,

$$f(x,y) := 6c_ptx^2 - 3c_pty^2 + 3c_ptxy - 8x - 2c_p^2t(t-1) = 0.$$

[†]Here S is given by a different formula than in the medium-characteristic case.

https://doi.org/10.1112/S1461157014000369 Published online by Cambridge University Press

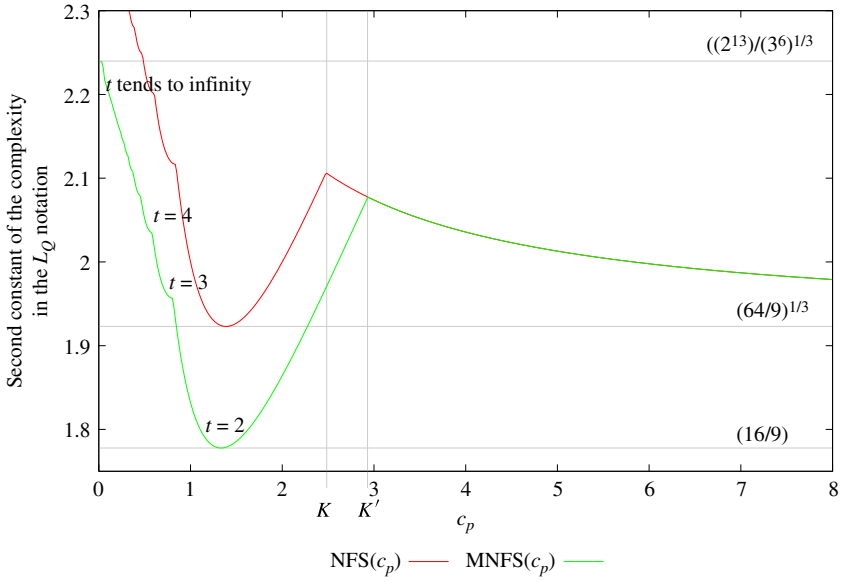

FIGURE 4. *Asymptotic complexities $L_Q(1/3, NFS(c_p))$ and $L_Q(1/3, MNFS(c_p))$ in the boundary case, as a function of c_p with $p = L_Q(2/3, c_p)$. The red curve corresponds to the classical variants of NFS and NFS-HD, while the green one corresponds to MNFS. The crossing point between NFS-HD and NFS, which is given by $c_p = K \approx 2.5$ in the classical NFS, switches to $c_p = K' \approx 2.9$ in MNFS. Here t is the number of terms (degree $t-1$) of the polynomials in the sieving space.*

We want to minimize $\varphi : (x, y) \mapsto x$ under this constraint. Since the gradient of the function f is $\bigtriangledown f(x, y) = (12c_p tx + 3c_p ty - 8, -6c_p ty + 3c_p tx)$, the method of Lagrange multipliers shows that we need $-6c_p ty + 3c_p tx = \partial\varphi/\partial y = 0$. This leads to $y = x/2$. We finally recover the linear dependence of the general case (see Remark 5.1): $c_b = 3c_v$. We then have $c_s = 8c_v/t$. Putting these values in (5.7), we get $54c_p tc_v^2 - 16c_v - c_p t^2(t-1) = 0$. Solving for c_v, we get $c_v = (8 + (64 + 54c_p^3 t^2(t-1))^{1/2})/(54c_p t)$. We recall that the second constant of the final complexity is $2(c_b + c_v) = 8c_v$. Consequently, the MNFS in this boundary case has complexity

$$L_Q\left(\frac{1}{3}, \frac{2}{3}\left(\frac{16}{9c_p t} + \sqrt{\left(\frac{16}{9c_p t}\right)^2 + \frac{8}{3}c_p(t-1)}\right)\right)$$

where $t-1$ is the degree of the polynomials we are sieving on. Compare this with the asymptotic complexity of the NFS in this case [14], which is $L_Q(1/3, (2/3) \cdot (2/(c_p t) + \sqrt{(2/(c_p t))^2 + 3c_p(t-1)}))$. Figure 4 compares the NFS and MNFS in this boundary case. Note that for the complexity of the NFS we use the formula given in Appendix A.3 of [14]. We do not discuss a multiple variant of the NFS using the lattice-based polynomial selection, but the first author will in future work analyze the case $c_p \in [3, \infty[$ in detail using a new polynomial selection in [1, § 8.3.1].

The optimal case: sieving on linear polynomials. We now consider c_p as a variable in order to get the value that minimizes the previous second constant of the complexity $C(c_p) = 16/(9c_p t) + (16^2(9c_p t)^{-2} + 8c_p(t-1)/3)^{1/2}$ for each algorithm. Then, looking at the minimal complexity of each algorithm as a function of t, we show that the optimal case in this boundary case is reached when we sieve on linear polynomials. $C(c_p)$ comes to a minimum $4/(3tc_p^2) = ((t-1) - 64 \cdot 27^{-1} t^{-2} c_p^{-3})/((16^2(9c_p t)^{-2} + 8c_p(t-1)3^{-1})^{1/2})$. This leads to

https://doi.org/10.1112/S1461157014000369 Published online by Cambridge University Press

$3^3t^2(t-1)c_p^3(3^3t^2(t-1)c_p^3-2^8)=0$. Thus we take the optimal $c'_p=(2^8(3^3t^2(t-1))^{-1})^{1/3}$ and compute the value of $C(c'_p)$:

$$L_Q\left(\frac{1}{3},\left(\frac{2^{13}(t-1)}{3^6t}\right)^{1/3}\right).$$

For $t=2$ we obtain the best of the NFS complexities in any of the medium-, boundary and high-characteristic cases:

$$L_Q\left(\tfrac{1}{3},\tfrac{16}{9}\right).$$

Splicing the boundary case with the medium-characteristic case. A family of finite fields whose characteristic p can be written as $p=L_Q(2/3,c_p)$, with a constant c_p converging to zero, can also be seen as having $p=L_Q(l_p,c'_p)$ with $l_p<2/3$, converging to $2/3$. If we consider that p is in the boundary case, when c_p tends to zero, the best choice is to force t to tend to infinity (see Figure 4). But the limit of the complexity given in (5.8) when $t\to\infty$ is $L_Q(1/3,(2^{13}/3^6)^{1/3})$. This is exactly the asymptotic complexity given by our multiple variant of the NFS in the medium-characteristic case.

5.3. *The high-characteristic case*

We recall that the high-characteristic case contains all finite fields $\mathbb{F}_{p^n}$ for which the characteristic can be written as $p=L_Q(l_p,c_p)$ with $l_p>2/3$. In this case, the MNFS suffers from a chronology problem. Nothing forbids the analysis made by Matyukhin for the prime fields [**16**] being applied in high-characteristic finite fields. Unfortunately, the first polynomial selection that allowed NFS (and thus MNFS) to be extended to this case was published three years later in [**14**]. This is why we explain here in a few words how to connect the polynomial selection of Joux, Lercier, Smart, and Vercauteren to Matyukhin's results.

Let $f_1,\dots,f_V$ be as in § 4.2, and θ_i be a complex root of f_i for all i. We recall that f_1 plays a specific role here. As in Matyukhin's variant, we sieve for triples (a,b,i) with $|a|,|b|$ bounded by a sieving parameter S and $i\in[1,\dots,V]$ such that $\gcd(a,b)=1$, $\mathrm{N}(a-b\theta_1)$ is B-smooth and $\mathrm{N}(a-b\theta_i)$ is B'-smooth for a another smoothness bound.

For all i, if a and b are smaller than S we have the two limits on norms: $\mathrm{N}(a-b\theta_1)=|b^{\deg f_1}f_1(a/b)|\leqslant(n+1)\|f_1\|_\infty S^n$ and $\mathrm{N}(a-b\theta_i)=|b^{\deg f_i}f_i(a/b)|\leqslant(d+1)\|f_i\|_\infty S^d$.

As usual, we choose $n=c_p^{-1}(\log Q/(\log\log Q))^{1-l_p}$, $d=\delta(\log Q/(\log\log Q))^{1/3}$ and $S=L_Q(1/3,c_sc_p)$. Thus $S^n=L_Q(l)$, with $l\leqslant 2/3$. For $i\in[1,\dots,V]$, this leads to

$$|\mathrm{N}(a-b\theta_1)|\leqslant(Q^{1/(d+1)})^{1+o(1)}\quad\text{and}\quad|\mathrm{N}(a-b\theta_i)|\leqslant(Q^{1/(d+1)}S^d)^{1+o(1)}.$$

These two upper bounds are independent of n and thus coincide with conditions (34) and (35) in Matyukhin's work. By reproducing the computations done in [**16**] for prime field, we obtain that the optimal parameters are $B=S=L_Q(1/3,((46+13\sqrt{13})/108)^{1/3})$, $B'=\exp(\log B(\sqrt{13}-1)/3)$, $V=L_Q(1/3,1-((\sqrt{13}-1)/3)^{1/3})$, $d=((46+13\sqrt{13})(4\sqrt{13}-10)^3(2^23^6)^{-1}\log Q(\log\log Q)^{-1})^{1/3}$. We conclude that in the high-characteristic case the MNFS has a complexity of

$$L_Q\left(\frac{1}{3},\left(\frac{92+26\sqrt{13}}{27}\right)^{1/3}\right),$$

where $((92+26\sqrt{13})/27)^{1/3}\approx 1.902$. Compare this with the complexity of the former state-of-the-art algorithm in this case, the NFS, which is $L_Q(1/3,(64/9)^{1/3})$, where $(64/9)^{1/3}\approx 1.923$.

6. *A small numerical example*

Let us illustrate the different steps of MNFS with a simple example, which can be run in Sage [**18**]. We do not want to use Schirokauer maps which, despite being efficient in the record

https://doi.org/10.1112/S1461157014000369 Published online by Cambridge University Press

computations, are not very intuitive. This restricts us to small examples such as the field $\mathbb{F}_{p^n}$ with $p = 103$ and $n = 3$. We compute discrete logarithms modulo the prime factor 3571 of $p^n - 1$. In the sieving and linear algebra stages we use $V = 11$ number fields.

Polynomial selection. We use $z = \lceil\sqrt{103}\rceil = 11$, $g_1 = x^3 + 11$ and $g_2 = zx^3 + (z^2 - 103) = 11x^3 + 18$. Then we obtain the other polynomials as linear combinations of g_1 and g_2 with the smallest coefficients in absolute value: $f_1 = -9x^3 + 4$, $f_2 = -10x^3 - 7$, $f_3 = x^3 + 11$, $f_4 = -8x^3 + 15$, $f_5 = -18x^3 + 8$, $f_6 = 11x^3 + 18$, $f_7 = -19x^3 - 3$, $f_8 = -17x^3 + 19$, $f_9 = -20x^3 - 14$, $f_{10} = 2x^3 + 22$ and $f_{11} = -21x^3 - 25$. Note that all the polynomials are divisible by $x^3 + 11$ modulo 103. Hence we represent $\mathbb{F}_{103^3} = \mathbb{F}_{103}[x]/\langle x^3 + 11\rangle = \mathbb{F}_{103}(m)$, where m is a root of $x^3 + 11$ in $\mathbb{F}_{103^3}$.

Construction of the smoothness base. For every polynomial f_i, K_i is its associated number field, h_i the class number of K_i, $\mathcal{O}_i$ the ring of integers, θ_i a root of f_i in K_i and $\theta_i' = l(f_i)\theta_i$, where $l(f_i)$ is the leading coefficient of f_i. For each field K_i we compute a system of fundamental units. We set the smoothness bound B to 300. We compute the smoothness base by adding, for each polynomial f_i, the prime ideals of degree 1 and norm less than B, as well as the prime ideals of arbitrary degree which divide the index $[\mathcal{O}_i : \mathbb{Z}[\theta_i']]$. For instance, for $f_4 = -8x^3+15$ we add 59 prime ideals, including the degree-two ideal $I = \langle 2, \theta_4'^2/16 + \theta_4'/4 + 1\rangle$. The smoothness base has cardinality 697. For each prime ideal $\mathbf{q}$ in the smoothness base we choose an arbitrary generator $\gamma_{\mathbf{q}}$ of $\mathbf{q}^{h_i}$, where h_i is the class number of the field containing $\mathbf{q}$.

Relation collection. We set the parameter t to 3. For the sieve, we consider all the polynomials $\phi \in \mathbb{Z}[x]$ of degree 2 and with coefficients bounded in absolute value by $S = 50$. Without effect on complexity, we also use the polynomials ϕ of degree 1 and coefficients bounded by $S' = S\|f_1\|_\infty^{1/n} \approx 136$, which allows us to have algebraic numbers $\phi(\theta_i)$ of approximately same norm. We then collect polynomials ϕ which are B-smooth for at least two polynomials f_i. We obtain a file in which a typical line is

$$3x^2 + 5x + 3 : 4, 9, 10, 11 \tag{6.1}$$

which records that $\mathrm{Res}(\phi, f_i)$ is B-smooth for $\phi = 3x^2 + 5x + 3$ and $i = 4, 9, 10, 11$.

Matrix construction. Consider the line in equation (6.1). For each pair of indices we can obtain a relation among logarithms, but pair (i_1, i_3) is a duplicate for pairs (i_1, i_2) and (i_2, i_3). We can add three equations to the matrix, for example, $(4, 9)$, $(4, 10)$ and $(4, 11)$. With 2239 polynomials ϕ, we obtain a matrix of 3302 rows. In order to write the row of the pair $(4, 9)$, we factor $\phi(\theta_4)\mathcal{O}_4$ and $\phi(\theta_9)\mathcal{O}_9$. Then $\phi(\theta_4) = u_{\phi,4}\prod_{\mathbf{q}}\gamma_{\mathbf{q}}^{\mathrm{val}(\phi(\theta_4),\mathbf{q})}$ where $u_{\phi,4}$ is a unit. We write then $u_{\phi,4}$ as a product of powers of fundamental units. We proceed similarly for $\phi(\theta_9)$ and obtain the unit $u_{\phi,9}$. Then we write a row with the valuations $\mathrm{val}(\phi(\theta_4), \mathbf{q})$, the exponents of $u_{\phi,4}$, the valuations $\mathrm{val}(\phi(\theta_9))$ with opposite sign and the exponents of $u_{\phi,9}$ with opposite sign. The 708 columns of the matrix are indexed with the prime ideals in K_1, then the fundamental units in K_1, and so on with $K_2, \ldots, K_{11}$.

Linear algebra stage. Using Sage, we obtain the right kernel of the matrix, which is a vector space of dimension 9. Although in theory we need a kernel of dimension 1, this is a reassuring result because it coincides with what happens for the classical variants of the NFS and NFS-HD. Indeed, it is known that this is due to the ideals, in our case seven in number, which do not occur in any relation. Then we erase the corresponding (empty) columns from the matrix and compute the new kernel, whose dimension is 2. It contains a vector corresponding to the logarithms and a parasite vector because the degree-two ideal I presented above always occurs

https://doi.org/10.1112/S1461157014000369 Published online by Cambridge University Press

paired with $\langle 2, \theta_4'/4+1\rangle$. For example, when setting $\mathbf{q}_3 = \langle 3, \theta_1'/3\rangle$, $\mathbf{q}_5 = \langle 5, \theta_1'/3-2\rangle$ and $\mathbf{q}_{11} = \langle 11, \theta_1'/3+1\rangle$, we have $\log \mathbf{q}_3 \equiv 1 \bmod 3751$, $\log \mathbf{q}_5 \equiv 681 \bmod 3751$ and $\log \mathbf{q}_{11} \equiv 160 \bmod 3751$.

Solution verification. Using explicit units instead of Schirokauer maps allows us to check the output of linear algebra. We consider the generator $g = m+4$ of $(\mathbb{F}_{103^3})^*$, but any generator can be used. By Pollard's rho method we compute $\log_g \gamma_{\mathbf{q}_3}(m) \equiv 599 \bmod 3751$, $\log_g \gamma_{\mathbf{q}_5}(m) \equiv 825 \bmod 3751$ and $\log_g \gamma_{\mathbf{q}_{11}}(m) \equiv 2994 \bmod 3751$. These values are proportional to $h_1 \log \mathbf{q}_3$, $h_1 \log \mathbf{q}_5$ and $h_1 \log \mathbf{q}_{11}$.

Smoothing. Consider a random element of $\mathbb{F}_{103^3}$, say $s = 55m^2+17m+26$. We try random values of $e \in [0, 3571-1]$ until we find $e = 989$. We have that $z = s^e = 64m^2+98m+79$ and $\bar{z} = 64\theta_1^2+98\theta_1+79$ is C-smooth for $C = 500 > B$. Then we factor $\bar{z}$: $\bar{z}\mathcal{O}_1 = \langle 3, \theta_1'/3\rangle^{-4}\langle 13, \theta_1'/3-3\rangle\langle 17, \theta_1'/3+6\rangle\langle 71, \theta_1'/3-7\rangle\langle 353, \theta_1'+17\rangle$. All the ideals above are in the smoothness base, except for $\mathbf{q} := \langle 353, \theta_1'+17\rangle$.

Descent by special-$\mathbf{q}$. We reduce the lattice generated by the matrix

$$\begin{pmatrix} 1 & 17 & 0 \\ 0 & 1 & 17 \\ 0 & 0 & 353 \end{pmatrix}.$$

The three vectors obtained correspond to polynomials $\phi_1 = 5x^2+2x+1$, $\phi_2 = x^2-4x-4$ and $\phi_3 = -2x^2+7x-9$. Note that, by construction, $\phi_i(\theta_1')$ is divisible by $\mathbf{q}$ for all $i \in \{1,2,3\}$. Then we enumerate the linear combinations of ϕ_1, ϕ_2 and ϕ_3 with coefficients in $[-A, A]$ for $A = 100$. We find $\phi = x^2-4x-4$ which is such that $\phi(\theta_i')$ is B-smooth for $i \in \{1,2,3\}$, but not for $i \in [4, 11]$. Using the relation of logarithms corresponding to $\phi(\theta_1')$ and $\phi(\theta_2')$, we retrieve the logarithm of $\mathbf{q}$ and hence the logarithm of s.

It is important to note that the continued fractions descent (smoothing) and the special-$\mathbf{q}$ descent steps can be done before or after the main phase. Also, the number of number fields in the individual logarithm stage is independent of that in the main phase.

7. *Conclusion*

In this paper we have focused on the currently hardest case of the discrete logarithm problem in finite fields: medium-characteristic finite fields. Initially proposed in 1993 for integer factorization, the idea of enlarging the number of number fields in the NFS finds its most favorable case in discrete logarithms for medium-characteristic finite fields, that is, when $1/3 < l_p < 2/3$, where $p = L_{p^n}(l_p, c_p)$. More precisely, instead of having a rational side and many algebraic sides as for factorization, in our MNFS, all the polynomials play the same role. Our algorithm has an asymptotic heuristic complexity of $L_{p^n}(1/3, (2^{13}/3^6)^{1/3})$, surpassing that of NFS-HD, $L_{p^n}(1/3, (128/9)^{1/3})$. In the boundary case where $l_p = 2/3$, the second constant remains below the constant of the NFS in the same case. Finally, in high-characteristic finite field, that is, when $2/3 < l_p \leqslant 1$, we have the same complexity as in factorization or prime fields: it is reduced from $L_{p^n}(1/3, (64/9)^{1/3})$ to $L_{p^n}(1/3, ((92+26\sqrt{13})/27)^{1/3})$.

Appendix. *Individual logarithm phase*

The individual logarithm phase is negligible in the effective computations which can presently be done. Hence we recommend implementing it as in the NFS, using only two number fields.

https://doi.org/10.1112/S1461157014000369 Published online by Cambridge University Press

Moreover, one can extrapolate that this stage is negligible at cryptographic scales. Nevertheless, a naive approach in this stage leads to an asymptotic complexity of $L_Q(1/3, c)$ with a constant c which dominates that of the main phase. We propose a modification which allows us to overcome this theoretical difficulty.

We propose to modify the algorithm and start it with the individual logarithm phase. For this phase we use $W = L_Q(1/3, c_w)$, $c_w > 0$, number fields $\mathbb{Q}(\theta_i)$ constructed as in § 4.2, where W is possibly larger than the number V of fields used in the main phase. Let $s \in \mathbb{F}_Q$ be the element whose discrete logarithm is required. As in the classical variant of the NFS, the individual logarithm phase has two steps. First we express the logarithm of s as a linear combination of logarithms of degree-one ideals $\mathbf{q}$ in $\mathbb{Q}(\theta_1)$ of norm less than a constant $C > B$ made explicit below. Then we use the special-$\mathbf{q}$ technique to express the logarithm of each $\mathbf{q}$ as a linear combination of degree-one prime ideals in any of the W number fields, whose norm is less than B. We will check that only $\exp((\log\log Q)^2)$ prime ideals are effectively used in the process. Therefore they belong to a set E of number fields whose cardinality is negligible with respect to V. This allows us to extend E to a set of V number fields as in § 4.2. We then go on to the relation collection and linear algebra stages. We obtain the discrete logarithms of the smoothness base, in particular of the subset of ideals which belong to number fields of E. These allow us to backtrack and find the discrete logarithm of s. Let us now turn to the complexity analysis.

Continued fraction descent or smoothing step. Let $s \in \mathbb{F}_{p^n}$ be the element whose discrete logarithm is required. Recall that to each $z = \sum_{j=0}^{n-1} z_j m^j \in \mathbb{F}_{p^n} = \mathbb{F}_p(m)$ we associate $\bar{z} = \sum_{j=0}^{n-1} z_j \theta_1^j \in \mathbb{Q}(\theta_1)$. In the smoothing step we try random exponents $e \in [0, Q-1]$ until, for $z = s^e$, the norm of $\bar{z}$ is square-free and C-smooth, where $C = L_Q(2/3, c)$ for a constant c to be optimized. Note that, by equation (2.1), the norm of $\bar{z}$ is bounded by $(2n)! p^n \sqrt{p}^n = Q^{3/2+o(1)} = L_Q(1, 3/2)$. By Theorem 2.1, the smoothness probability of each norm is $1/L_Q(1/3, 1/(2c))$. Since the time of a C-smoothness test using ECM is $L_C(1/2, \sqrt{2})(\log Q)^{O(1)}$, each test requires a time $L_Q(1/3, 2\sqrt{c/3})$. We optimize for $c = (3/4)^{1/3}$ and we find a complexity of

$$L_Q(\tfrac{1}{3}, (\tfrac{9}{2})^{1/3}).$$

Since $(9/2)^{1/3} \approx 1.65$ is smaller than 2.24, this step is negligible with respect to the main phase. Note that one can search for a better constant using the admissibility technique in [**1**, Chapter 4].

Special-$\mathbf{q}$ descent. Let $\mathbf{q} = \langle q, \theta_i - \rho \rangle$ be a degree-one prime ideal in one of the number fields $\mathbb{Q}(\theta_i)$. For simplicity of notation we assume that $i = 1$. For a parameter d, we consider the lattice of degree-$(d-1)$ polynomials ϕ such that $\phi(\theta_1)$ is divisible by $\mathbf{q}$. Using the LLL algorithm [**21**], we obtain d polynomials $\phi_1, \ldots, \phi_d$ of degree $d-1$ and coefficient size $q^{1/d}$ such that $\phi_1(\theta_1), \ldots, \phi_d(\theta_1)$ is divisible by $\mathbf{q}$. Next, we enumerate the polynomials $\phi = \sum_{j=1}^{d} \alpha_j \phi_j$ with $\alpha_j \in [-A, A]$, for a parameter A, and test using ECM whether the two conditions below are simultaneously satisfied:

- the norm of $\phi(\theta_1)$ writes as q times a $q^{0.99}$-smooth and square-free number;
- for at least one index $i \in [1, W]$, the norm of $\phi(\theta_i)$ is $q^{0.99}$-smooth and square-free.

Note that the constant 0.99 can be replaced by any other value in the interval $(0, 1)$. We stop the enumeration when such a polynomial ϕ is smooth, allowing us to express $\log \mathbf{q}$ as a sum of logarithms of smaller ideals. This completes what is called a *descent step*. We continue the process recursively with the new ideals introduced by $\phi(\theta_i)$. We end the descent when $q^{0.99}$ is less than B.

The bit-size $\log q$ of the ideals $\mathbf{q}$ involved is multiplied by 0.99 at each step, so the descent tree has height $\log_{0.99}(\log B)/(\log C) = O(1)\log_{1/2}(\log 2)/(\log Q) = O(\log\log Q)$. Each step

https://doi.org/10.1112/S1461157014000369 Published online by Cambridge University Press

introduces at most $2\log \mathrm{Res}(\phi, f_1) \leqslant 2\log Q$ new ideals, so the width of the tree is at most $O(\log Q)$. In total, there are

$$\exp(O(1)(\log\log Q)^2)$$

nodes in the descent. So the complete recursive process takes time $T^{1+o(1)}$, where T is the time of each descent step. In order to evaluate T, the time of each descent step, we set

$$d = \frac{c_d}{c_p}\left(\frac{\log Q}{\log\log Q}\right)^{2/3-l_p} \quad \text{and} \quad A = \text{ absolute constant.}$$

This allows us to have norms of size $L_Q(2/3)$ and $L_Q(1/3)$ polynomials to be tested in each descent step, as in the classical variant of the NFS [7]. Let $\mathbf{q} = \langle q, \theta_1' - \rho\rangle$ be a prime ideal with $B < q < C$, which must be descended. According to Theorem 2.1, the probability that a polynomial ϕ satisfies the two conditions above is $Wu^{-u+o(u)}$, where $u = 2\log \mathrm{Res}(f,\phi)(0.99\log q)^{-1}$. The largest value of u is obtained at the end of the descent, when $q^{0.99} = B$. The success probability is $L_Q(1/3, (2/c_d + c_d/c_b + 2c_a/(c_d c_b))/3 - c_w)^{-1}$.

The number of polynomials to be tested is $O(1)$. Each polynomial is tested for W fields, so the descent time is $W^{1+o(1)} = L_Q(1/3, c_w)$. We minimize this time by imposing $3c_w = 2/c_d + c_d/c_b$. We set $c_d = \sqrt{2c_b}$ and we obtain $c_w = 2\sqrt{2}/(3\sqrt{c_b}) = 2^{5/6}3^{-1/2} \approx 1.03$.

We conclude that the overall time of the individual logarithm phase is $L_Q(1/3, (9/2)^{1/3})$, which is negligible with respect to the main phase.

References

1. R. Barbulescu, 'Algorithmes de logarithmes discrets dans les corps finis'. PhD Thesis, Université de Lorraine, 2013.
2. D. J. Bernstein, 'The multiple-lattice number field sieve', Technical report, University of California, Berkeley, 1991.
3. D. Boneh and M. K. Franklin, 'Identity-based encryption from the Weil pairing', *SIAM J. Comput.* 32 (2003) no. 3, 586–615.
4. R. Barbulescu, P. Gaudry, A. Joux and E. Thomé, 'A heuristic quasi-polynomial algorithm for discrete logarithm in finite fields of small characteristic', *Advances in cryptology – EUROCRYPT 2014*, Lecture Notes in Computer Science 8441 (Springer, 2014) 1–16.
5. E. R. Canfield, P. Erdős and C. Pomerance, 'On a problem of Oppenheim concerning factorisatio numerorum', *J. Number Theory* 17 (1983) 1–28.
6. D. Coppersmith, 'Modifications to the number field sieve', *J. Cryptology* 6 (1993) no. 3, 169–180.
7. A. Commeine and I. Semaev, 'An algorithm to solve the discrete logarithm problem with the number field sieve', *Public key cryptology – PKC 2006*, Lecture Notes in Computer Science 3958 (Springer, 2006) 174–190.
8. W. Diffie and M. E. Hellman, 'New directions in cryptography', *IEEE Trans. Inf. Theory* 22 (1976) no. 6, 644–654.
9. T. ElGamal, 'A public key cryptosystem and a signature scheme based on discrete logarithms', *IEEE Trans. Inform. Theory* 31 (1985) no. 4, 469–472.
10. R. M. Elkenbracht-Huizing, 'A multiple polynomial general number field sieve', *Algorithmic number theory*, Lecture Notes in Computer Science 1122 (ed. H. Cohen; Springer, 1996) 99–114.
11. A. Fiat and A. Shamir, 'How to prove yourself: practical solutions to identification and signature problems', *Advances in cryptology – CRYPTO '86*, Lecture Notes in Computer Science 263 (Springer, 1986) 186–194.
12. A. Joux, 'A one round protocol for tripartite Diffie–Hellman', *J. Cryptology* 17 (2004) no. 4, 263–276.
13. A. Joux and R. Lercier, 'Improvements to the general number field sieve for discrete logarithms in prime fields. A comparison with the Gaussian integer method', *Math. Comput.* 72 (2003) no. 242, 953–967.
14. A. Joux, R. Lercier, N. P. Smart and F. Vercauteren, 'The number field sieve in the medium prime case', *Advances in cryptology – CRYPTO 2006*, Lecture Notes in Computer Science 4117 (Springer, 2006) 326–344.
15. A. Joux and C. Pierrot, 'The special number field sieve in $\mathbb{F}_{p^n}$, application to pairing-friendly constructions', *Pairing-based cryptography – Pairing 2013*, Lecture Notes in Computer Science 8365 (Springer, 2013) 45–61.
16. D. V. Matyukhin, 'On asymptotic complexity of computing discrete logarithms over $GF(p)$', *Discrete Math. Appl.* 13 (2003) no. 1, 27–50.

https://doi.org/10.1112/S1461157014000369 Published online by Cambridge University Press

17. P. Paillier, 'Public-key cryptosystems based on composite degree residuosity classes', *Advances in cryptology – Eurocrypt '99*, Lecture Notes in Computer Science 1592 (Springer, 1999) 223–238.
18. W. A. Stein *et al.*, Sage Mathematics Software (Version 5.8). The Sage Development Team, 2013, http://www.sagemath.org.
19. C.-P. Schnorr, 'Efficient identification and signatures for smart cards', *Advances in cryptology – CRYPTO '89*, Lecture Notes in Computer Science 435 (Springer, 1990) 239–252.
20. O. Schirokauer, 'Using number fields to compute logarithms in finite fields', *Math. Comput.* 69 (2000) no. 231, 1267–1283.
21. J. von zur Gathen and J. Gerhard, *Modern computer algebra*, 3rd edn (Cambridge University Press, 2013).
22. D. Wiedemann, 'Solving sparse linear equations over finite fields', *IEEE Trans. Inform. Theory* 32 (1986) no. 1, 54–62.

Razvan Barbulescu
Université de Lorraine
France
razvan.barbulescu@inria.fr

Cécile Pierrot
Laboratoire d'Informatique de Paris 6
UPMC
France
Cecile.Pierrot@lip6.fr

https://doi.org/10.1112/S1461157014000369 Published online by Cambridge University Press