Computing discrete logarithms in $GF(p^6)$

Laurent Grémy, Aurore Guillevic, François Morain, Emmanuel Thomé

Computing discrete logarithms in $\mathbb{F}_{p^6}$

Laurent Grémy[1], Aurore Guillevic[1], François Morain[2], and Emmanuel Thomé[1]

[1] Université de Lorraine, CNRS, Inria, LORIA, F-54000 Nancy
[2] École Polytechnique/LIX, CNRS UMR 7161, Palaiseau, France

Abstract. The security of torus-based and pairing-based cryptography relies on the difficulty of computing discrete logarithms in small degree extensions of finite fields of large characteristic. It has already been shown that for degrees 2 and 3, the discrete logarithm problem is not as hard as once thought. We address the question of degree 6 and aim at providing real-life timings for such problems. We report on a record DL computation in a 132-bit subgroup of $\mathbb{F}_{p^6}$ for a 22-decimal digit prime, with p^6 having 422 bits. The previous record was for a 79-bit subgroup in a 240-bit field. We used NFS-DL with a sieving phase over degree 2 polynomials, instead of the more classical degree 1 case. We show how to improve many parts of the NFS-DL algorithm to reach this target.

1 Introduction

Since the 1970s and the first key-echange protocol, the security of the vast majority of asymmetric cryptosystems has relied on the hardness of two main number theory problems: the factorization of large integers and the computation of discrete logarithms. Given a finite cyclic group $(G, \cdot)$ of order ℓ, a generator g of this group, and an element $a \in G$, the goal of the discrete logarithm problem (DLP) is to solve $g^x = a$ for $x \in \mathbb{Z}/\ell\mathbb{Z}$. In this paper, we focus on discrete logarithms in finite fields of the form $\mathbb{F}_{p^6}$, where p is a prime. This corresponds to the medium characteristic situation studied in [30]. Breaking discrete logarithms in such a field can affect torus-based cryptography [34,43] (XTR and its generalization CEILIDH) and pairing-based [16] cryptography.

1.1 XTR and torus-based cryptography

The XTR setting considers the cyclotomic subgroup of a small degree extension $\mathbb{F}_{p^2}$ or $\mathbb{F}_{p^6}$. It was generalized to higher extensions, and led to torus-based cryptography. When these settings were proposed in 2000, computing a discrete logarithm in a non-prime field was supposed to be much harder than in a prime field. The cost is usually given in terms of the L-notation:

⋆ Experiments presented in this paper were carried out using the Grid'5000 testbed, supported by a scientific interest group hosted by Inria and including CNRS, RENATER and several Universities as well as other organizations (see `https://www.grid5000.fr`).

$L_{p^n}[\alpha, c] = \exp\big((c + o(1))\log(p^n)^\alpha \log\log(p^n)^{1-\alpha}\big)$. In 2005, Granger and Vercauteren estimated the cost of computing discrete logarithms in the torus of $\mathbb{F}_{p^6}$ to be in $L_{p^n}[1/2]$ rather than in $L_p[1/3]$ for prime fields [21]. One year later, in 2006, an $L_{p^n}[1/3, c = 2.43]$ variant of NFS was proposed [30]. Since then, the constant c was improved from 2.43 to 2.21 (see [5]) and now 1.93 (1.74 in favorable case) with the so-called exTNFS [32] in the specific case of composite extension degree n (e.g. $n = 6$). Multiple-field variants (MNFS) could allow to reduce even further the constant c.

The record computation of a discrete logarithm in a field $\mathbb{F}_{p^6}$ is held by Zajac for a 240-bit field [49], done in less than 38 days on a single 2 GHz computer. The relation collection was realized in about 24 days with a generalized line sieve algorithm: this was clearly the dominating part. The recent records are focused on improving this costly relation collection: the same numerical example of [49] was done again with a dedicated algorithm for dimension three, in about the same timing by Hayasaka et al. [27] and in less than one day by Gaudry, Grémy and Videau [18]. They also performed a relation collection for a 389-bit field in less than 800 days. One part of our experimental data finishes their work: we describe the linear algebra and one individual logarithm computation in Section 5.

1.2 Pairing-friendly curves of small embedding degree

The Weil and Tate pairings on elliptic curves were proposed as a constructive building block in asymmetric cryptography in 2000 for key exchange [28], short digital signatures [10] and identity-based encryption [31,9]. A pairing is a map $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ where the three groups are of large prime order ℓ, $\mathbb{G}_1$ and $\mathbb{G}_2$ are two distincts subgroups (of same order) of a *pairing-friendly* elliptic curve, and $\mathbb{G}_T$, the target group, is a multiplicative subgroup of a finite field.

$$\begin{array}{rccccc} & E(\mathbb{F}_q)[\ell] & & E(\mathbb{F}_{q^k})[\ell] & & \mathbb{F}_{q^k} \\ & \cup & & \cup & & \cup \\ e: & \mathbb{G}_1 & \times & \mathbb{G}_2 & \to & \mathbb{G}_T \\ & & (P,Q) & & \mapsto & e(P,Q) \end{array}$$

To ensure a good level of security for a pairing-friendly curve, one needs to estimate the complexity of computing a discrete logarithm in the prime order subgroup $E(\mathbb{F}_q)[\ell]$ of the curve on the one hand, and in the multiplicative subgroup of order ℓ of the embedding field $\mathbb{F}_{q^k} = \mathbb{F}_{p^n}$ on the other hand (and when q is a prime power, make sure that the embedding field is not actually a strict subfield of $\mathbb{F}_{q^k}$). The state of the art for the former is $O(\sqrt{\ell})$. For the latter, the quasi-polynomial-time, Function Field Sieve or Number Field Sieve algorithms apply, each to a certain type of fields.

A degree six extension field $\mathbb{F}_{p^6}$ is used in XTR and the cyclotomic subgroup of order $p^2 - p + 1$ is considered. It is also the field where a pairing takes its values, the elliptic curve being supersingular, defined over $\mathbb{F}_{p^2}$ and of order $p^2 - p + 1$. The hardness of a discrete logarithm computation on the curve, of prime order subgroup ℓ, has exponential growth $O(\sqrt{\ell})$, compared to subexponential growth

$L_{p^6}[1/3, c]$ in the target field $\mathbb{F}_{p^6}$. For this reason, for p above some threshold, the weakness against a discrete logarithm computation attack switches from the curve to the finite field. Since $\ell \approx p^2 - p + 1$ by construction, the complexity is actually in $O(p)$. For the size we target: p of 71 bits and ℓ of 132 bits, the computation will be already much faster in $\mathbb{F}_{p^6}$.

This is the contrary for an MNT curve (introduced by Miyaji, Nakabayashi and Takano in 2001 [39]). An MNT curve is defined over a prime field $\mathbb{F}_p$ and has prime order ℓ, hence a complexity in $O(\sqrt{\ell}) \sim O(\sqrt{p})$. This is easier than a computation in $\mathbb{F}_{p^6}$ for a 422-bit finite field. Because of the small size of our experiment, we expect the threshold for an MNT curve to be significantly larger than the prime p that is targeted in this work. We decided to focus on supersingular curves of order $p^2 - p + 1$ in this paper.

Supersingular curves. The supersingular curves are equipped with an easy-to-compute *distortion map* $\phi : E(\mathbb{F}_{q^k}) \to E(\mathbb{F}_{q^k})$. It can be turned into an isomorphism $\phi : \mathbb{G}_1 \to \mathbb{G}_2$, which is not available for ordinary curves. Many pairing-based cryptosystems can now be re-stated with an asymmetric pairing [35], where there is no straighforward isomorphism $\mathbb{G}_1 \to \mathbb{G}_2$. However in certain cases this is not possible, so that efficient symmetric pairings are still desired. The earliest "fast" symmetric pairings are now completely broken since they used supersingular curves over fields of characteristic 2 or 3: the target group is then a sugroup of $\mathbb{F}_{2^{4n}}$ or $\mathbb{F}_{3^{6m}}$, and the quasi-polynomial-time algorithm [6] is particularly devastating [20,1]. Since this algorithm does not apply to large characteristic, three constructions of supersingular curves survived. The first two are defined over a (large) prime field $\mathbb{F}_p$, and their embedding field is $\mathbb{F}_{p^2}$. The computation of a discrete logarithm in $\mathbb{F}_{p^2}$ was studied in [5]. The third construction uses supersingular curves defined over a quadratic field $\mathbb{F}_{p^2}$, of embedding degree 3, their embedding field being $\mathbb{F}_{p^6}$. This is the practical application of our discrete logarithm computation. An efficient Ate pairing computation on these curves was proposed in [12], and is competitive compared to supersingular curves of embedding degree 2. Numerical examples are provided in Section 5.

Our contributions

To attack the DLP over $\mathbb{F}_{p^6}$, we needed to improve several parts of NFS. A key ingredient to our computation is the use of sieving in dimension 3, which follows [18] and is explained in Section 2, as opposed to traditional sieving in dimension 2 (that is, "(a, b) pairs" encoding $a - bx$ become "(a_0, a_1, a_2) triples" encoding $a_0 + a_1 x + a_2 x^2$). To lower the impact of using ideals of degree 2, we were able to use nice families of cyclic degree 6 extensions, in which these ideals have a virtual logarithm equal to zero, see Section 3. Last, the individual logarithm computation had to be optimized: we were able to decrease the initial sizes of the boots needed, and we used a descent in dimension three in §2.5.

Our article is organized as follows. Section 2 contains a succinct description of NFS-DL and insists on the algebraic part, some of which is reused in Section 3 that

justifies our choice of degree 6 cyclic extensions to solve the problem. Section 4 builds on this and explains the selection of polynomials. Section 5 contains a list of discrete logarithm computations we were able to perform.

2 A crash course on NFS-DL

We start with an overview of NFS-DL, and then give technical details on the actual algebraic factorization of ideals in number fields, relevant to our computation.

Our goal is to compute discrete logarithms in the order ℓ subgroup of $\mathbb{F}_{p^n}^*$, where ℓ is a prime divisor of $\Phi_n(p)$, coprime to $\Phi_c(p)$ for all $c \mid n$. (This assumption matches the definition of embedding field of the pairing, mentioned in §1.2.)

2.1 Overview

The first step is the *polynomial selection* phase, where we find two irreducible (over $\mathbb{Q}$) polynomials f_0 and f_1 with integer coefficients, and such that $\varphi = \gcd(f_0, f_1) \bmod p$ is a degree n irreducible polynomial. We build $\mathbb{F}_{p^n}$ as $\mathbb{F}_p[X]/(\varphi)$.

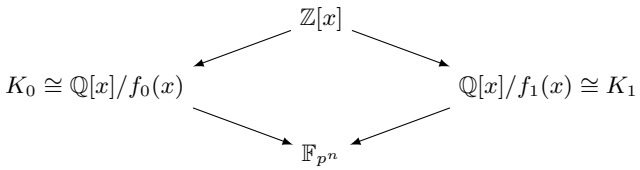

Fig. 1. The NFS diagram to compute discrete logarithms in $\mathbb{F}_{p^n}^*$.

We write $K_i = \mathbb{Q}(\alpha_i)$ for some root α_i of f_i for $i \in \{0, 1\}$. In the *relation collection* phase, we look for polynomials of degree $t-1$, say $A(x) = a_0 + a_1 x + \cdots + a_{t-1}x^{t-1}$, with integer coefficients, so that the integral pseudonorm

$$\mathrm{Res}_x(f_i(x), A(x))$$

factors over a factor basis $\mathcal{B}_i \subset \mathbb{Z}$ (for $i \in \{0, 1\}$). If this is achieved, then the algebraic numbers $A(\alpha_0)$ and $A(\alpha_1)$ factor as a product of prime ideals above prime elements in their factor bases. Applying reduction from K_i to $\mathbb{F}_{p^n}$, we get an additive relation between virtual logarithms of elements in the factor bases.

Once enough relations are collected, the *linear algebra* step aims to solve the relevant system and get the virtual logarithms of the primes.

In a last step, and perhaps the most significant from a cryptanalytic point of view, we compute *individual logarithms* using a method called descent. It should be remarked that this last step validates all the preceding computations.

2.2 Relation collection

The relation collection examines a subset $\mathcal{S}$ of the whole set of polynomials $A(x)$ of degree $t-1$. The subset $\mathcal{S}$ is called the *search space* and is made of the polynomials $A(x)$ of bounded coefficients. This search space is chosen so as to contain sufficiently many polynomials A to get a complete set of relations, that is, more than $\#(\mathcal{B}_0 \cup \mathcal{B}_1)$. A way to estimate the relations yield for a given $\mathcal{S}$ is to use the Murphy-E quantity [40,18].

The cost of factoring of the integral pseudonorms and testing if the factors are in the corresponding factor basis for each polynomial A on both sides is prohibitive. This is why we use sieving algorithms to partially factor the integral pseudonorm of all the polynomials in $\mathcal{S}$, in order to detect promising candidates that have a good chance to have a complete factorization involving only elements of the factor basis. Sieving algorithms have a major drawback: their memory consumption is proportional to the size of $\mathcal{S}$. All modern record computations of discrete logarithms in finite fields required $\mathcal{S}$ to be far too large to fit in memory (the 596-bit record of [11] needed more than 2^{60} elements, and [17] needed $2^{61.5}$).

To palliate these drawbacks, Pollard [42] suggested to divide the search space into many subsets of $\mathcal{S}$ using the special-$\mathfrak{q}$-method: all the elements A of a subset share the property that the factorization of $A(\alpha_0)$ (or resp. $A(\alpha_1)$) involves the ideal $\mathfrak{q}$, if the special-$\mathfrak{q}$ is forced on side 0 (resp. 1). If the special-$\mathfrak{q}$s are large enough, there is a small number of duplicated elements in the different subsets. The number of elements per subset, called *sieving region*, is adapted to fit into memory (typically 2^{31} elements per special-$\mathfrak{q}$) and the sieve algorithm in each subset can be processed independently. The special-$\mathfrak{q}$-method was extended to polynomials of any degree by Hayasaka et al. [26]. Enumerating the elements inside a special-$\mathfrak{q}$-subset can be performed using the algorithms proposed in [27,18]: we used in our practical computations an implementation of the three types of sieve described in [18]. The implementation is available in CADO-NFS [48].

2.3 Algebraic factorization

Let $f(x) = c_d x^d + \cdots + c_0$, and denote by K the associated number field $K = \mathbb{Q}[X]/(f(X)) = \mathbb{Q}(\alpha)$ and O_K its ring of integers (maximal order). We wish to factor the principal ideal $\langle A(\alpha)\rangle = A(\alpha)O_K$ where $A(x) = a_0 + \cdots + a_{t-1}x^{t-1}$ into prime ideals. To overcome the problem that this ideal might be fractional (non integral), it is customary to consider the ideal $\langle J_f^{\deg A} A(x)\rangle$ instead, where

$$J_f = \langle 1, \alpha\rangle^{-1} = \langle c_d, c_d\alpha + c_{d-1}, \ldots, c_d\alpha^{d-1} + c_{d-1}\alpha^{d-2} + \cdots + c_1\rangle$$

(see [15, §9]). Then $\langle J_f^{\deg A} A(\alpha)\rangle$ is an integral ideal, which factors as

$$\langle J_f^{\deg A} A(\alpha)\rangle = \prod_i \mathfrak{q}_i^{u_i} \tag{1}$$

for integers u_i and prime ideals $\mathfrak{q}_i$ (over some finite range for the index i).

Computing the valuations in (1) might require some careful work for a few $\mathfrak{q}$'s, as detailed in [13, chap. 4 and 6]. We start from the factorization of the norm

$$\mathcal{R} = \mathrm{Res}_x(A(x), f(x)) = \prod_j q_j^{v_j}$$

where q_j is a rational prime which is the norm of one or several of the $\mathfrak{q}_i$'s. $\mathcal{R}$ is precisely the norm of the integral ideal $\langle J_f^{\deg A} A(\alpha)\rangle$. In great generality, we have a direct relation between q_j and only one $\mathfrak{q}_i$, but in a few cases, telling apart which of the $\mathfrak{q}$ appear above a given q is not straightforward. Computer algebra software such as Magma or PARI/GP comes to help. Fortunately, only finitely many of these non straightforward cases may exist, so that some precomputation ahead of time is possible, and useful.

Since the first task is to compute the factorization of the norm, the factor basis is first and foremost the set of rational primes q for which $f(x) \bmod q$ has roots. While enumerating this set, some exceptional (yet non exclusive) events can be detected: when $q \mid c_d$, we have a *projective ideal*; when q divides $\mathrm{disc}(f)$ to some high power, or when f has multiple roots mod q we have a *bad ideal*. A nice degree 1 ideal is simply $\langle q, \alpha - r\rangle$ where r is a simple root of $f(X) \bmod q$, in such a way that the ideal is completely characterized by (q, r). On the contrary, a bad ideal cannot be so simply described; to differentiate these ideals, limited lifting in the q-adic field $\mathbb{Q}_q$ is useful.

Post-sieving and Schirokauer maps. For this experiment, valuations at prime ideals were computed with Magma. The rest of the computation, namely all the filtering and linear algebra, was done with CADO-NFS. The final computation of individual logarithms requires some care, since higher dimensional sieving is used again.

Schirokauer maps are defined as follows. We assume that ℓ does not ramify in K, and let m_i be the inertia degrees of prime ideals above ℓ. We let $\epsilon = \mathrm{lcm}(\{\ell^{m_i} - 1\})$. Let $\mathcal{T}$ denote the set of number field elements with zero valuation at all prime ideals above ℓ. Let $a = A(\alpha) \in \mathcal{T}$. The ℓ-adic expansion of $a^\epsilon - 1$ writes as $\ell L(a)(\alpha) + O(\ell^2)$, with $L(a) \in \mathbb{Z}/\ell\mathbb{Z}[x]$ and $\deg L(a) < n$. We let the Schirokauer maps be the r-coordinate vector $\Lambda(a)$ formed by coefficients of degree $n - r$ to $n - 1$ of $L(a)$, where r is the unit rank of K. The map Λ is a homomorphism from $(\mathcal{T}/\mathcal{T}^\ell, \times)$ to $((\mathbb{Z}/\ell\mathbb{Z})^r, +)$. We conjecture, following [46], that its restriction to units is surjective. In fact, fairly little is canonical with L (and hence with Λ), as it depends on the choice of the generating element α. We do however note, as it plays an important role in this paper, that the constant coefficient of $L(a)$ is special: if $\deg(L(a)) = 0$, so is $\deg(L(a^\sigma))$ for any field automorphism σ (this also extends to subfields).[3]

Virtual logarithms of the r coordinates of the Schirokauer map vector Λ are denoted by $(\mathrm{vlog}(SM_i))_{1\le i\le r}$, or $(\mathrm{vlog}(SM_{s,i}))_{1\le i\le r}$ when emphasis on the side $s \in \{0, 1\}$ is desired.

[3] We mention here an oversight in [4, Lemma 3.2], where Λ and L are mistakenly confused for one another.

Numbering ideals in a sensible way. In CADO-NFS, the output of the sieve is a list of rational primes dividing the norm of some $\langle J_f^{\deg A} A(\alpha)\rangle$. Let q be one such prime. Most often, prime ideals above q are written as $\mathfrak{q} = \langle q, \alpha - r\rangle$, for r a root of $f \bmod q$. The ideal $\mathfrak{q}$ contributes to the factorization of $\langle J_f^{\deg A} A(\alpha)\rangle$ if $A(r) = 0 \mod q$. If A and f have several roots in common modulo q, extra work is needed to separate the contribution of the ideals. Extra work is also needed for the exceptional cases of prime ideals whose two-element form can only be written as $\langle q, q_0 + q_1\alpha + \cdots + q_{d-1}\alpha^{d-1}\rangle$. To ensure consistent numbering, we keep a conversion table from prime ideals to column indices in the relation matrix.

2.4 Linear algebra

Once all valuations are computed, we get relations

$$(\deg A)\,\mathrm{vlog}(J_{f_0}) + \sum_{\mathfrak{q}_0 \in \mathcal{B}_0} u_{\mathfrak{q}_0}\,\mathrm{vlog}(\mathfrak{q}_0) + \sum_{i=1}^{r} \mathrm{vlog}(SM_{0,i})$$

$$\equiv (\deg A)\,\mathrm{vlog}(J_{f_1}) + \sum_{\mathfrak{q}_1 \in \mathcal{B}_1} u_{\mathfrak{q}_1}\,\mathrm{vlog}(\mathfrak{q}_1) + \sum_{i=1}^{r} \mathrm{vlog}(SM_{1,i}) \bmod \ell$$

in which the virtual logarithms are the unknowns.

A large matrix is built, each row corresponding to a relation and each column to a prime ideal, or the ideals J_{f_0} and J_{f_1}, or Schirokauer maps. Then, we enter the classical process of filtering, whose aim is to reduce the size of the matrix via elementary operations on rows and columns. Once a smaller (but still sparse) matrix is obtained, we used the distributed Block Wiedemann implementation from CADO-NFS to find the kernel of the matrix. Reconstructing all logarithms from the kernel is done using Magma.

2.5 Computing individual logarithms

To complete our work, we compute individual discrete logarithms of random-looking targets generated from the decimals of π. A target is an element of $\mathbb{F}_{p^6}$, and when it is an output of a pairing (see §1.2) or of XTR (§1.1), we firstly apply the isomorphism to the target to get our target in $\mathbb{F}_p[x]/(\varphi(x))$, that is, the degree 6 extension $\mathbb{F}_{p^6}$ is defined by $\varphi(x)$ given by the polynomial selection. Computing this isomorphism has insignificant computational cost.

Initial Splitting step (*a.k.a.* smoothing or boot). The first step is *initial splitting* and we refer to [24,25] for a complete description. Given a target $T_0 \in \mathbb{F}_p[x]/(\varphi(x))$, the strategy is to randomize it as $g^i T_0$ where g is the generator of the order-ℓ subgroup of $\mathbb{F}_{p^6}$, and try many exponents $i \in [1, \ldots, \ell - 1]$ until the resultant of f_0 and a preimage of $g^i T_0$ in $\mathbb{Z}[x]$, is B_{init}-smooth. Details are provided in §5.1.

Decreasing the norms: descent. The initial splitting step outputs a degree 2 polynomial $T = b_0 + b_1 x + b_2 x^2$ whose resultant with f_0 is B_{init}-smooth, that is $\text{Res}_x(f_0, T) = \prod q_i^{e_i}$, where the q_i are prime numbers smaller than B_{init}. Each q_i is treated as a special-$\mathfrak{q}$ and a sieving step in dimension 3 for the largest q_i is performed as in §2.2.

This forms a descent tree, where each node is a large prime, for which a relation involving only smaller primes is sought with a special-$\mathfrak{q}$ search. The smaller primes obtained in the relation form the children of the node.

Lemma 1 ([30, Lemma 2]). *Let $K = \mathbb{Q}[\theta]$ and $(a_0, \ldots, a_{t-1})$ a t-tuple of coprime integers, then any prime ideal $\mathfrak{p}$ that divides $\sum_{i=0}^{t-1} a_i \theta^i$ either divides the index $f_\theta = [\mathcal{O}_K : \mathbb{Z}[\theta]]$ or is of degree $< t$.*

In the relation collection, the degree of the polynomial $A(x)$ that gives a relation is fixed to $t-1$, which is usually 1 for prime fields, and 2 in our case. We have more freedom during the descent step: the degree can be different, typically larger than $t-1$. Higher degree sieving for the descent was already analyzed in [17, §5.4] for prime fields, but it did not provide a notable practical advantage. In our present case, we do need to perform the descent phase with polynomials of degree at least 2. Further details are given in Section 5.

Final recombination. When the factor basis is reached, that is we have a complete set of relations that starts from $g^i T_0$ and finally is expressed in terms of ideals of small norm and known virtual discrete logarithm, then we recombine everything to obtain $\log(g^i T_0)$, and eventually $\log_g T_0$.

3 Cyclic extensions in degree 6

Cyclic extensions improve both relation collection and linear algebra, as already remarked in [30, §4.3]. The article [4] compiles many results and properties of virtual logarithms of elements in Galois extensions, including cases where logarithms of units vanish. In the same spirit, we add Lemma 2 and Theorem 1. The most striking result is that ideals of degree 2 have virtual logarithm equal to zero. This eases the linear algebra step in a minor way, but is still good to have.

3.1 A cyclic degree 6 family

For convenience, we use the cyclic family of polynomials of degree six given in [22], parameterized by s:

$$C_s(x) = x^6 - 2sx^5 - (5s+15)x^4 - 20x^3 + 5sx^2 + (2s+6)x + 1 \ .$$

Since $C_{-(s+3)}(x) = x^6 C_s(1/x)$, we only consider $s > 0$. We compute

$$\text{disc}(C_s) = 2^6 \cdot 3^6 (s^2 + 3s + 9)^5.$$

For $s \notin \{0, 5\}$, C_s is irreducible, has 6 real roots and is equipped with a degree 6 cyclic automorphism $\sigma : x \mapsto -(2x+1)/(x-1)$. We note that $\sigma^2(x) = -(x+1)/x$ is of order 3, and $\sigma^3(x) = -(x+2)/(2x+1)$ is of order 2. The number field $K = \mathbb{Q}[x]/(C_s(x))$ has a quadratic subfield K^+ defined by the polynomial $h_s(y) = y^2 - 2sy - 3s - 9$. Over K^+, C_s splits as $(x^3 - yx^2 - (y+3)x - 1)(x^3 - \bar{y}x^2 - (\bar{y}+3)x - 1)$ where $\bar{y}$ is the conjugate of y in K^+. Generically, one has:

$$\mathrm{N}_{K/\mathbb{Q}}(x-1) = \mathrm{N}_{K/\mathbb{Q}}(2x+1) = \mathrm{N}_{K/\mathbb{Q}}(x+2) = -3^3$$
$$\mathrm{N}_{K/\mathbb{Q}}(x) = \mathrm{N}_{K/\mathbb{Q}}(x+1) = 1.$$

3.2 Cancellations of virtual logarithms

When we use NFS-DL with both polynomials from the family $C_s(x)$, we observe the following consequence of $C_s(x)$ having six real roots.

Lemma 2. *For all principal ideals of O_K, there exists a generator γ with Schirokauer maps $\Lambda(\gamma) = 0$. Furthermore, if the defining polynomial of K splits completely in $\mathbb{R}$, then for any automorphism σ of K, we have $\Lambda(\gamma^\sigma) = 0$.*

Proof. By the assumption that Λ is surjective on the units, we may find γ with $\Lambda(\gamma) = 0$. Since the defining polynomial splits completely in $\mathcal{R}$, the unit rank is $[K : \mathbb{Q}] - 1$. Hence $\Lambda(a)$ captures all but the first coordinates of $L(a)$, following the notations used in 2.3. Then $\Lambda(\gamma) = 0$ implies that $L(a)(\alpha)$ is a rational number, which is Galois invariant.

A consequence of this lemma is that virtual logarithms are very constrained.

Theorem 1. *Let p, ℓ, and the degree n be as in Section 2. Let K be a cyclic number field of degree n, whose defining polynomial splits completely in $\mathbb{R}$. Assume that ℓ is coprime to $\#\operatorname{Cl}(\mathcal{O}_K)$ as well as $p^c - 1$ for all proper divisors c of n. If $\mathfrak{q}$ is a prime ideal of O_K that has less than n distinct Galois conjugates (in particular, if its inertia degree is greater than 1, or if it is ramified), then* $\operatorname{vlog}(\mathfrak{q}) \equiv 0 \mod \ell$.

Proof. The virtual logarithm of $\mathfrak{q}$ is unequivocally defined as $h^{-1} \log_{\mathbb{F}_{p^n}} \gamma$, where $h = \#\operatorname{Cl}(\mathcal{O}_K)$ is the class number of K, and γ is a generator of $\mathfrak{q}^h$ as in Lemma 2. Let σ be the Frobenius automorphism of p (i.e. such that $\alpha^\sigma - \alpha^p \in pO_K$). Let $c < n$ be the number of distinct conjugate prime ideals of $\mathfrak{q}$. Because $\mathrm{Gal}(K/\mathbb{Q})$ is cyclic and p is inert, we have that $\tau = \sigma^c$ is such that $\tau(\mathfrak{q}) = \mathfrak{q}$ (i.e. τ is in the decomposition group of $\mathfrak{q}$). Per Lemma 2, we have $\Lambda(\gamma^\tau) = 0$, so that $\log_{\mathbb{F}_{p^n}}(\gamma^\tau) = p^c \log_{\mathbb{F}_{p^n}} \gamma$, whence $(p^c - 1) \operatorname{vlog} \mathfrak{q} = 0$. Given that c is a proper divisor of n and ℓ is coprime to $p^c - 1$, this concludes the proof.

4 Polynomial selection for $\mathbb{F}_{p^6}$

The polynomial selection is the first step of the NFS algorithm and its variants. Many methods were proposed in the last few years, and we can partition them in three types:

1. methods that define two number fields over a base field (originally $\mathbb{Q}$). These are (in historical order) base-m, Joux–Lercier (JL), JL–Smart–Vercauteren JLSV_0, JLSV_1, JLSV_2, generalized JL (GJL), Conjugation, and Sarkar–Singh [19,29,30,5,36,45];
2. methods to exploit the structure of the subfields: TNFS and exTNFS, which require an adaptation of one of the above methods since the base field is no longer $\mathbb{Q}$ [47,7,32,44,33];
3. multiple-field variants that can apply to any of the previous methods [2,41] (the prequels being [14] for factorization and [37] for prime fields).

Using an exTNFS variant for $\mathbb{F}_{p^6}$ would mean first to define a quadratic, resp. cubic number field as a base field, before running one of the type 1 polynomial selection methods, as if it were for $n = 3$, resp. $n = 2$. Because of this structure, an efficient sieve in dimension 4, resp. 6 would be required[4] In this paper we first investigate a sieve in dimension three without a tower structure for now. This is a mandatory step before being able to run an efficient sieve in dimension four, and then implement exTNFS for the first time in $\mathbb{F}_{p^6}$. We will compare the following polynomial selections, with a sieve in dimension 2 or 3: JLSV_1 [30], conjugation [5], (GJL) [5,36], and Sarkar–Singh [45] which is a combination of Conjugation and GJL that exploits the decomposition of n as 2×3 of 3×2 without needing a tower extension.

4.1 First comparison of polynomial selection methods

To choose the best method, we first compare the average size of the norms in the sieving phase. We wrote a prototype of polynomial selection in Magma, whose aim is first to select polynomials with smallest possible coefficients, without trying to improve the smoothness properties of the polynomials. Then with these polynomials, we compute the average of the pseudonorms of elements $a_0 + a_1 x$ for dimension two, and $a_0 + a_1 x + a_2 x^2$ for dimension three. We denote by S the size of the search space $\mathcal{S}$, that is, $S = \#\mathcal{S}$. For a sieving dimension t, $\mathcal{S}$ is defined by the inequalities $-E \leq a_i \leq E$ for $0 \leq i < t-1$, and $0 < a_{t-1} \leq E$, so that $2S \approx (2E)^t$. To get a rough idea of the largest norm, we set the $a_i = E \approx (2S)^{1/t}/2$, where $S = L_Q[1/3, c + o(1)]$. To be more precise, we fix the $o(1)$ in the formula for S such that it matches the previous relation collection record of 389 bits in $\mathbb{F}_{p^6}$ of [18] and set $\log_2 S = 53$ for $\log_2 p^6 = 389$ bits. Our estimates are presented in Figure 2. Clearly, the JLSV_1, Sarkar–Singh with $(\deg f_0, \deg f_1) = (8, 6)$, and GJL methods with a dimension 3 sieving provide much smaller norms than the conjugation method, which would be competitive with a dimension 4 sieving, that is not yet available. We continued our comparison between GJL, Sarkar–Singh $(8, 6)$ and JLSV_1 methods.

[4] $\mathbb{F}_{p^6}$ would be represented as a cubic extension of a quadratic field, or possibly the converse. We would sieve over polynomials A of either of the forms $(a_0 + a_1 y) + (b_0 + b_1 y)x$ or $(a_0 + a_1 y + a_2 y^2) + (b_0 + b_1 y + b_2 y^2)x$, that is dimension four or six.

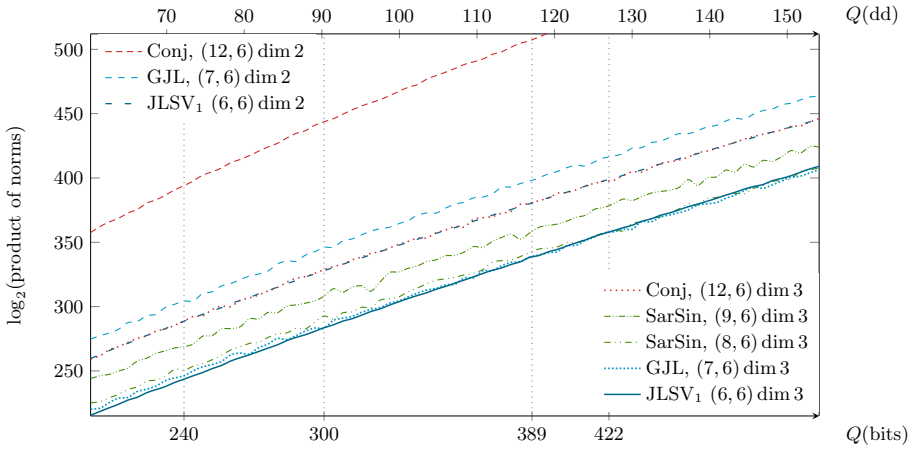

Fig. 2. Estimation of the sizes of the norms.

4.2 Refined comparison of polynomial selection methods

The size of the norms for a fixed size of $Q = p^6$ and a fixed bound on the coefficients of the polynomials A in the set $\mathcal{S}$ provides a first rough comparison of the polynomial selection methods. To refine the comparison, we start again from the same $\mathcal{S}$ and same estimation of the norms, given p^6 and polynomials f_0, f_1. Then we set a smoothness bound $B = S^{1/2}$ and approximate the probability of an integer of the same size as the norm to be B-smooth with the Dickman-ρ function [40]. We obtain an estimate of the total number of relations that we could get. Then we vary B to obtain at least $\#(\mathcal{F}_0 \cup \mathcal{F}_1)$ relations. We check it with the inequality, where $\mathrm{Li}(x) = \int_2^x \frac{dt}{\log t}$ is the offset logarithmic integral:

$$2\,\mathrm{Li}(B) \leq S \cdot \Pr(N_{K_0/\mathbb{Q}} \text{ is } B\text{-smooth}) \cdot \Pr(N_{K_1/\mathbb{Q}} \text{ is } B\text{-smooth}) \qquad (2)$$

We vary S again and adjust B accordingly in a bootstrapping process, to balance the expected time between relation collection and linear algebra: $S^{1/2} = \#(\mathcal{F}_0 \cup \mathcal{F}_1)$. Our estimates are summarized in Table 1. We considered each side separately to estimate the smoothness probability (instead of the product of the norms in the asymptotic formulas). Other things held constant, it is better to have balanced norms. We also estimated the average best expected $\alpha(f_0)$ and $\alpha(f_1)$. The α value is lower (i.e. better) for dimension three sieve.

We assumed that a Galois automorphism of order six was available with the JLSV$_1$ method, of order two with Sarkar–Singh $(8, 6)$, but none with GJL. A Galois automorphism of order k provides a k-fold speedup for the relation collection. Unfortunately in our implementation, the linear algebra benefits at most from a two-fold speedup (for even k only).

For each size of finite field (240 bits to 422 bits), the JLSV$_1$ method produces the smallest norms, which are balanced, and has a Galois speed-up of order six. For all these reasons it seemed the most promising method.

$\log_2 p^6$	$\log_2 p$	$\log_2 S$	$\log_2 N_{K_0}$	$\log_2 N_{K_1}$	$\log_2(N_{K_0}N_{K_1})$	$\log_2 B$	relation collection	linear algebra
JLSV$_1$, $\deg f_0 = \deg f_1 = 6$, σ of order 6, $\alpha(f_0) = -3.0$, $\alpha(f_1) = -8.0$								
240	40	37	112	113	225	21	2^{35}	2^{35}
300	50	42	132	133	265	23	2^{39}	2^{40}
389	65	48	158	159	317	26	2^{45}	2^{46}
422	71	50	168	168	336	28	2^{47}	2^{48}
GJL, $\deg f_0 = 7, \deg f_1 = 6$, no Galois automorphism, $\alpha(f_0) = 0.0$, $\alpha(f_1) = -4.0$								
240	40	41	92	146	238	23	$2^{40.5}$	2^{40}
300	51	45	104	173	277	25	2^{45}	2^{45}
389	65	50.5	118	210	328	28.5	$2^{50.5}$	$2^{50.5}$
422	71	52.5	122	224	346	29.5	$2^{52.5}$	$2^{52.5}$
Sarkar–Singh, $\deg f_0 = 8, \deg f_1 = 6$, σ of order 2, $\alpha(f_0) = -2.0$, $\alpha(f_1) = -4.0$								
240	40	40	106	140	246	23	2^{39}	2^{39}
300	50	43	112	156	268	24.5	2^{42}	2^{42}
389	65	49	131	196	327	28	2^{48}	2^{48}
422	71	50	135	206	341	29	2^{50}	2^{50}

Table 1. Relation collection space and smoothness bound estimates, and approximation of the relation collection and linear algebra time.

4.3 Optimizing JLSV$_1$ pairs of polynomials

The next step is to run the JLSV$_1$ polynomial selection method for the given prime p, and to select polynomials that have good smoothness properties. For that we used the dimension three α and Murphy's E functions as defined in [18].

The JLSV$_1$ method outputs two polynomials of degree n and coefficients of size $p^{1/2}$. We used the cyclic degree 6 family C_s introduced in Section 3, allowing a six-fold speed-up in the relation collection[5]. We can enumerate all the parameters s such that $\sqrt{p}/2 < |s| < \sqrt{p}$, $C_s(x)$ is irreducible, and has a good α value, that is $\alpha(C_s) \leq -2.0$ in our case. We pre-selected about 4000 such polynomials C_s as good f_0 candidates. Given a $f_0 = C_{s_0}$ for a certain s_0, the second polynomial f_1 is built as follows: One computes a rational reconstruction of the parameter s_0 modulo p: $s_0 = u/v \mod p$, where $|u|, |v| \sim p^{1/2}$ and $|v| \neq 1$. Then one sets $f_1 = vC_{u/v}$. To improve $\alpha(f_1)$ without increasing the size of the largest coefficient of f_1 denoted by $\|f_1\|_\infty = \max_{0 \leq i \leq \deg f_1} |f_{1,i}|$, we can enumerate the linear combinations $f_1 + \lambda f_0$, where $0 < |\lambda| < \|f_1\|_\infty / \|f_0\|_\infty$ (by construction, we will have $\|f_1\|_\infty > \|f_0\|_\infty$ and we can choose to have $\|f_1\|_\infty / \|f_0\|_\infty$ of about 2^{10}). The improved polynomial $f_1 + \lambda f_0$ is still in the family C_s since it is linear in s. There is a large room for improving α in the JLSV$_1$ method, without increasing the size of the coefficients (neither the size of the norms), which is another reason why we have chosen it for our record computations.

[5] the Galois action does not produce more relations, it produces the same relations but six times faster.

5 Computations

We ran complete computations in $\mathbb{F}_{p^6}$ for different problem sizes. Three of them were already done, at least partially, in previous work: for these, we provide an experimental improvement. For the largest problem size, the experimental data we provide is new. Timings of all these different works are summarized in Table 4, see also [23]. We used computer clusters of various research institutes and universities to run our experiments. Computations for bitsizes 240, 300 and 389 all used `Intel Xeon E5520` CPUs, with clock speed 2.27 GHz, while for the 422-bit record, we used also a set of clusters from the `grid5000` platform. We give in Table 2 the primes and labels we will use to refer to them, for each bitsize. The `p6bd40` problem was covered in [49]. Relation collection was dramatically

name	p	seed for p	$\log_2 p$	$\log_2 p^6$	$\log_2 \ell$	ℓ
`p6bd40`	1081034284409	[49]	40	240	79	$(p^2-p+1)/3$
`p6bd50`	1043035802846857	[18]	50	300	100	p^2-p+1
`p6bd65`	31415926535897942161	[18]	65	389	130	p^2-p+1
`p6dd22`	1350664108659952233509	RSA1024	71	422	132	$(p^2-p+1)/651$

Table 2. Primes, bitsizes and labels

improved by [18], and that paper also completed relation collection for the `p6bd50` and `p6bd65` problems. For this reason, we refer to [18] for experimental data about relation collection for these three problems, as we merely based our work on the data set produced by [18]. We contributed new linear algebra computations and new individual logarithm computations for problems `p6bd40`, `p6bd50` and `p6bd65`, providing key improvements over the previous state of the art. We also report an entirely new computation for the larger challenge `p6dd22`.

Table 3 gives polynomial selection parameters, and relation collection parameters and results, for all experiments. The sieving region bounds are denoted by $H = (a_0, a_1, a_2)$, the precomputed factor basis bounds involved in the sieve by `lims=lim0,lim1` (a.k.a. `fbb0,fbb1`) and the large prime bounds, i.e. the smoothness bounds by `lpbs=lpb0,lpb1`. In the sieving process, the prime ideals in K_0, resp. K_1, of norm at most `lim0` bits, resp. `lim1` bits involved in a pseudo-norm are sieved. After the sieving process, if the remaining non-factorized part of a pseudo-norm is less than `threshold` bits, a cofactorization process with ECM tries to factor it further. This entails finding the prime ideals of norm between `lims` and `lpbs`. Details about the computation of the `p6dd22` are given in §5.3.

5.1 Individual logarithms

Initial Splitting step. Since $\mathbb{F}_{p^6}$ has three proper subfields $\mathbb{F}_p$, $\mathbb{F}_{p^2}$ and $\mathbb{F}_{p^3}$, we can apply the fast initial splitting technique of [25]. The target $T = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + a_4 x^4 + a_5 x^5 \in \mathbb{F}_{p^6}$ is expressed as

$$T = w_0(u_0 + U)(v_0 + v_1 V + V^2)(b_0 + b_1 x + b_2 x^2) \ , \tag{3}$$

	p6bd40 [18]	p6bd50 [18]	p6bd65 [18]	p6dd22 (new)
α-values	$-1.8, -11.5$	$-4.9, -12$	$-5.7, -11.5$	$-2.4, -14.3$
Murphy-E	$2^{-21.2}$	$2^{-23.7}$	$2^{-28.3}$	$2^{-29.0}$
Sieving region H	$6, 6, 6$	$7, 7, 7$	$9, 9, 8$	$9, 9, 8$
`lims (fbbs)`	$2^{19}, 2^{19}$	$2^{20.5}, 2^{20.5}$	$2^{21}, 2^{21}$	$2^{21}, 2^{21}$
Smoothness bounds (`lpbs`)	$2^{23}, 2^{23}$	$2^{25}, 2^{25}$	$2^{28}, 2^{28}$	$2^{29}, 2^{29}$
$\#\mathcal{S} = q_{\max} 2^{H_0+H_1+H_2}$	2^{41}	2^{46}	2^{54}	2^{55}
Special-$\mathfrak{q}$ side	1	1	1	0
Size of largest norms, after removing q (bits)	115, 117	128, 139	160, 173	151, 203
thresholds	65, 65	80, 80	90, 90	90, 110–123
q-range	$]2^{19}, 2^{21.2}[$	$]2^{20.5}, 2^{22.3}[$	$]2^{21}, 2^{25.1}[$	$]2^{21}, 2^{27.9}[$
# relations	1,445,094	5,857,098	29,679,203	100,778,132
unique	1,258,327	5,245,451	23,654,314	71,850,465
purged	246,236	621,360	5,440,780	18,335,401
filtered	72,749	201,601	1,661,759	5,218,599

Table 3. Properties of the polynomials, parameters and statistics of the relation collection with dimension two and dimension three sieving, see also [23].

where $\langle 1, U\rangle$ is a polynomial basis of $\mathbb{F}_{p^2}$, $\langle 1, V, V^2\rangle$ is a polynomial basis of $\mathbb{F}_{p^3}$, $w_i, u_i, v_i \in \mathbb{F}_p$ and $|b_i| \approx p^{2/3}$, so that the resultant of f_0 and $b_0 + b_1 x + b_2 x^2$ (where the b_i's are lifted in $\mathbb{Z}$) is bounded by $O(p^5)$ (assuming $\|f_0\|_\infty = p^{1/2}$ since we are in the JLSV$_1$ case). We observed that a representation as in (3) was found for 2/3 of the $g^i T_0$. If it is not, we skip that i and proceed to the next one. In the JLSV$_1$ case for $\mathbb{F}_{p^6}$, asymptotically the optimal B_{init} is $L_{p^6}[2/3, 0.614]$ and the number of trials to find a smooth resultant is $L_{p^6}[1/3, 1.357]$ [25].

The descent. The descent was not manageable with the classical dimension two sieving, so we opted for dimension three sieving. This was due to the large size of the norms involved in the descent. The JLSV$_1$ method does not have a preferred side for the descent: both polynomials have coefficients of size $p^{1/2}$.

Given a special-$\mathfrak{q}$ of norm $\pm q$, the set of degree-2 polynomials A such that $A(\alpha_0)$ (resp. $A(\alpha_1)$) involves $\mathfrak{q}$ in its ideal factorization is a dimension three lattice $\Lambda_\mathfrak{q}$ of volume q. Let $\boldsymbol{v}_0, \boldsymbol{v}_1, \boldsymbol{v}_2$ be a reduced basis, obtained for example by the LLL algorithm. The coefficients of the vectors are typically close to $q^{1/3}$. We enumerate linear combinations $\lambda_0 \boldsymbol{v}_0 + \lambda_1 \boldsymbol{v}_1 + \lambda_2 \boldsymbol{v}_2$, which form the polynomials $A(x) = \sum_{j=0}^{2} \sum_{i=0}^{2} \lambda_i \boldsymbol{v}_i[j] x^j$, by the same (sieving) procedure as the one of the relation collection. Given a search space volume S, we bound the λ_i's by $S^{1/36}$, so that the resultant of A and f_0 or f_1 is bounded by $O(S^2 q^2 p)$ [8]. When A is of degree 1, then $\Lambda_\mathfrak{q}$ becomes a two-dimensional lattice: the reduction of the lattice

[6] In fact, if one of the vectors $\boldsymbol{v}_i$ has coordinates shorter than the expected $q^{1/3}$, it suffices to set skew bounds on the λ_i's. Furthermore, having a short vector in the lattice allows us to expect more often a relation involving small ideals, which is better.

outputs two short vectors whose coefficients are typically close to $q^{1/2}$, and the resultants are bounded by $O(S^3q^3p^{1/2})$. The crossover point between dimension three and two sieving is roughly at $Sq = p^{1/2}$: when $Sq > p^{1/2}$, one should prefer dimension three, while for $Sq < p^{1/2}$ dimension two is better.

5.2 p6bd65

The polynomials are

$$\begin{aligned} f_0 &= x^6 - 218117072x^5 - 5452926954x^4 - 20x^3 + 5452926802x^2 + 218117078x + 1, \\ \text{and } \quad f_1 &= 2880648044402x^6 + 13810904846422x^5 - 8682458549954x^4 - 57612960888002x^3 \\ &\quad - 34527262116054x^2 + 3472983419982x + 2880648044402. \end{aligned}$$

The relation collection was done in [18]. We only report the linear algebra and individual logarithm timings.

Linear algebra. We used the Block Wiedemann implementation in CADO-NFS, with parameters $n = 10$ and $m = 20$. The cumulated numbers of core years for the various steps of the algorithm are 80 days for the Krylov sequences, 6 days for the linear generator computation, and 14 days for the final computation of the solution, which yielded the values of 19,805,202 logarithms of the factor bases.

Individual logarithm. Take $g = x + 3 \in \mathbb{F}_{p^6} = \mathbb{F}_p[x]/(f_0(x))$. From $\mathrm{N}_0(g) = 11 \cdot 23 \cdot 37 \cdot 1398037$, we get $\mathrm{vlog}(g) = 907665820983150820551985406251606874974$. The target is

$$\begin{aligned} z &= x^5 + 3141592653589793238x^4 + 462643383279502884 1x^3 \\ &\quad + 9716939937510582097x^2 + 4944592307816406286x + 2089986280348253421 \end{aligned}$$

and $g^{116775}z$ has a smooth norm:

$$\begin{aligned} \mathrm{N}(g^{116775}z) &= 11 \cdot 23 \cdot 97 \cdot 46073 \cdot 2958947 \cdot 1009479469 \cdot 6931176587051 \cdot 24379478228011 \\ &\quad \cdot 70817385294241 \cdot 199377274156547 \cdot 373976871809623 \end{aligned}$$

Descending all of these took approximately 19 hours. We get

$$\mathrm{vlog}(z) = 594727449023976898713456336273989724540.$$

5.3 p6dd22

The polynomials are

$$\begin{aligned} f_0 &= x^6 - 183758937422x^5 - 459397343702x^4 - 20x^3 \\ &\quad + 459397343552x^2 + 183758937482x + 1, \\ \text{and } \quad f_1 &= 1470039093602x^6 - 7380547581022x^5 - 4050195535655x^4 - 2940078187200x^3 \\ &\quad + 1845136895255x^2 + 16200782142622x + 1470039093602. \end{aligned}$$

Relation collection. For this computation, we selected the sieving region to be $2^{10} \times 2^{10} \times 2^8$ for each special-$\mathfrak{q}$. Both smoothness bounds were 2^{29} and sieving bounds were 2^{21}. We sieved the $2^{23.6}$ smallest special-$\mathfrak{q}$s on the f_0-side with norm larger than 2^{21}. More precisely, thanks to the order 6 Galois action, we only had to consider $2^{21.1}$ special-$\mathfrak{q}$ orbits.

We designed the polynomials with balanced coefficient sizes but unbalanced α: we were lucky and got $\alpha(f_1) = -14.4$, but $\alpha(f_0) = -2.2$ only. With the special-$\mathfrak{q}$ on side 0, the norm ranged from 142 to 191 bits, once the contribution of the special-$\mathfrak{q}$ was removed. On side 1, the norm ranged from 175 to 245 bits. Taking into account the offset $\alpha/\log 2$ (3.2 and 20.8 bits), the yield was better with this choice of special-$\mathfrak{q}$ than if we had put in on side 1, at least for the small special-$\mathfrak{q}$s. It was a closer call for larger special-$\mathfrak{q}$s. We increased the cofactorization threshold on side 1 from 110 to 115 then 121, allowing more room of the cofactorization process after the sieving. We found $\approx 72M$ unique relations, after removing the 28.8% duplicates, in about 8400 core-days.

Linear algebra. We used a combination of Intel Xeon E5-2630v3, E5-2650, E7-4850 v3 CPUs, connected with Infiniband FDR fabric. The block Wiedemann algorithm was used with parameters $m = 30$ and $n = 10$. The cumulated running times for the various steps of the algorithm were 2.67 core years for the computation of the Krylov sequences, 0.1 core years for the computation of the linear generator, and 0.3 core years for the computation of the solution vector.

Individual discrete logarithm computation. Define $\mathbb{F}_{p^2} = \mathbb{F}_p[i]/(i^2+2)$. The curve $E/\mathbb{F}_{p^2} : y^2 = x^3 + b,\ b = i+2$ is supersingular of trace p, hence of order $p^2 - p + 1$. Define $\mathbb{F}_{p^6} = \mathbb{F}_{p^2}[j]/(j^3 - b)$. The embedding field of the curve E is $\mathbb{F}_{p^6}$. We take $G_0 = (6, 875904596857578874580 + 2210981389734019530 62i)$ as a generator of $E(\mathbb{F}_{p^2})$, and $G_1 = [651]G_0$ is a generator of $E(\mathbb{F}_{p^2})[\ell]$. The distortion map $\phi : (x, y) \mapsto (x^p/(jb^{(p-2)/3}), y^p/(b^{(p-1)/2}))$ gives a generator $G_2 = \phi(G_1)$ of the second dimension of the ℓ-torsion. We take the point $P_0 = (314159265358979323847 + 2643383279502884197 16i, 935658401868915145130 + 643077111364229171931i) \in E(\mathbb{F}_{p^2})$ from the decimals of π, and $P = 651P_0 \in E(\mathbb{F}_{p^2})[\ell]$ is our challenge. We aim to compute the discrete logarithm of P to base G_1. To do so, we transfer G_1 and P to $\mathbb{F}_{p^6}$, and obtain $g = \mathrm{e}_{\mathrm{Tate}}(G_1, \phi(G_1))$ and $t = \mathrm{e}_{\mathrm{Tate}}(P_1, \phi(G_1))$, or

$$\begin{aligned} t = {} & 265997258109245157592 + 3973907757729746440 09x + 84184346073477818 48x^2 \\ & + 131994088093768382310 3x^3 + 116091350004927737629 4x^4 + 775101705346231535180x^5, \\ g = {} & 118987624922477279445 9 + 375273593285154553828x + 426102368940555566443x^2 \\ & + 192100975135320642877x^3 + 871172323955942457570x^4 + 955501495504184789 96x^5. \end{aligned}$$

The initial splitting gave a 41-bit smooth generator $g^{545513} = uvw(-141849807327922 - 5453622801413x + 541464063196 59x^2)$ where $u \in \mathbb{F}_{p^2}, v \in \mathbb{F}_{p^3}, w \in \mathbb{F}_p$ so that their logarithm modulo ℓ is zero. The norm of the latter term is: $3^3 \cdot 7^2 \cdot 11^2 \cdot 17 \cdot 317 \cdot 35812537 \cdot 16941885101 \cdot 17450874689 \cdot 22088674079 \cdot 35134635829 \cdot 85053580259 \cdot 144278841431 \cdot$

1128022180423 · 2178186439939. We had 8 special-q to descend. The smallest special-q had 34-bit norm $q_{34} = 16941885101$. We used the same sieving implementation to find a relation involving this ideal, and smaller ones. We set the search space to 2^{31} and the smoothness bound to 29 bits. We were able to find in 836s on a `Core i5-6500 @ 3.2GHz` three relations involving q_{34} on the side 0, and other prime ideals of norm strictly smaller than 2^{29}.

We also got a 45-bit smooth challenge of norm 821·3877·6788447·75032879·292064093·257269999897 · 456432316517 · 1029313376969 · 3142696252889 · 4321280585357 · 18415984442663:

$$g^{58779}t = uvw(-137392843659670 - 349183027245 09x + 134011712202 12x^2)$$

We obtained $\mathrm{vlog}(g) = 14636111560202813908403410352551744199 92$ and $\mathrm{vlog}(t) = 18004302008056970405325216125240295266 11$, so that $\log_g(t) = \mathrm{vlog}(t)/\mathrm{vlog}(g) \mod \ell = 75207848026896577063286973539798946459 2$.

year	finite field	size of p^n	authors	algorithm	rel. col. c-days	lin. alg. c-days	total c-days	total c-years
2013	$\mathbb{F}_{p^{12}}$	203	HAKT	NFS-HD	10.5	0.28	11	0.03
2008	$\mathbb{F}_{p^6}$	240	Zajac	NFS-HD	24.16	13.44	38	0.10
2015	$\mathbb{F}_{p^6}$	240	HAKT	NFS-HD	21.94	–	–	–
2017	$\mathbf{F_{p^6}}$	**240**	**this work**	**NFS-HD**	**0.90**	**0.22**	**1.12**	**0.003**
2017	$\mathbf{F_{p^6}}$	**300**	**this work**	**NFS-HD**	**6.84**	**1.64**	**8.48**	**0.03**
2017	$\mathbb{F}_{p^5}$	324	GGM	NFS-HD	359	11.5	386	1.05
2017	$\mathbf{F_{p^6}}$	**389**	**this work**	**NFS-HD**	**790**	**100**	**890**	**2.44**
2015	$\mathbb{F}_{p^4}$	392	BGGM	NFS	114	390	510	1.40
2017	$\mathbf{F_{p^6}}$	**422**	**this work**	**NFS-HD**	**8400**	**1120**	**9520**	**26**
2015	$\mathbb{F}_{p^3}$	593	BGGM	NFS	3287	5113	8400	23
2016	$\mathbb{F}_{p^2}$	595	BGGM	NFS	157	18	175	0.48
2017	$\mathbb{F}_p$	768	KDLPS	NFS	1461000	401775	1935825	5300

Table 4. Comparison with other record computations in core-days, and total in core-years, including also the polynomial selection and individual logarithm computation if known. For references, see `https://gitlab.inria.fr/dldb/discretelogdb`.

6 Cryptographic implications

We demonstrated the practicality of sieving in higher dimension for computing discrete logarithms in finite fields of medium characteristic, with a record-breaking computation in a 422-bit field $\mathbb{F}_{p^6}$. Moreover our parameter comparisons of Section 4 can be extrapolated to estimate the cost of computing discrete logarithms in larger fields $\mathbb{F}_{p^6}$, and also be generalized for $\mathbb{F}_{p^{12}}$. To reach the next pairing frontier, that is $\mathbb{F}_{p^{12}}$, it seems necessary to combine these ideas and extend them so as to make new variants practical. This work will be a useful additional step to a precise estimation of the cost of computing discrete logarithms in the embedding

field $\mathbb{F}_{p^{12}}$ of Barreto-Naehrig (BN) curves, following Barbulescu–Duquesne [3] and Menezes, Sarkar and Singh [38].

Acknowledgments. The authors are grateful to Pierrick Gaudry and Paul Zimmermann for numerous discussions all along this work. Many thanks to the referees whose remarks helped us improve the presentation of our results.

References

1. Adj, G., Canales-Martínez, I., Cruz-Cortés, N., Menezes, A., Oliveira, T., Rivera-Zamarripa, L., Rodríguez-Henríquez, F.: Computing discrete logarithms in cryptographically-interesting characteristic-three finite fields. ePrint report (2016), http://eprint.iacr.org/2016/914, see also http://ecc2016.yasar.edu.tr/slides/ecc2016-gora.pdf
2. Barbulescu, R., Pierrot, C.: The multiple number field sieve for medium- and high-characteristic finite fields. LMS J. Comput. Math. 17, 230–246 (Jan 2014), http://journals.cambridge.org/article_S1461157014000369
3. Barbulescu, R., Duquesne, S.: Updating key size estimations for pairings. ePrint report (2017), http://eprint.iacr.org/2017/334
4. Barbulescu, R., Gaudry, P., Guillevic, A., Morain, F.: Improvements to the number field sieve for non-prime finite fields (Nov 2014), working paper, https://hal.inria.fr/hal-01052449
5. Barbulescu, R., Gaudry, P., Guillevic, A., Morain, F.: Improving NFS for the discrete logarithm problem in non-prime finite fields. In: Oswald, E., Fischlin, M. (eds.) EUROCRYPT 2015, Part I. LNCS, vol. 9056, pp. 129–155. Springer, Heidelberg (Apr 2015)
6. Barbulescu, R., Gaudry, P., Joux, A., Thomé, E.: A heuristic quasi-polynomial algorithm for discrete logarithm in finite fields of small characteristic. In: Nguyen, P.Q., Oswald, E. (eds.) EUROCRYPT 2014. LNCS, vol. 8441, pp. 1–16. Springer, Heidelberg (May 2014)
7. Barbulescu, R., Gaudry, P., Kleinjung, T.: The tower number field sieve. In: Iwata, T., Cheon, J.H. (eds.) ASIACRYPT 2015, Part II. LNCS, vol. 9453, pp. 31–55. Springer, Heidelberg (Nov / Dec 2015)
8. Bistritz, Y., Lifshitz, A.: Bounds for resultants of univariate and bivariate polynomials. Linear Algebra and its Applications 432(8), 1995 – 2005 (2010)
9. Boneh, D., Franklin, M.K.: Identity-based encryption from the Weil pairing. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 213–229. Springer, Heidelberg (Aug 2001)
10. Boneh, D., Lynn, B., Shacham, H.: Short signatures from the Weil pairing. In: Boyd, C. (ed.) ASIACRYPT 2001. LNCS, vol. 2248, pp. 514–532. Springer, Heidelberg (Dec 2001)
11. Bouvier, C., Gaudry, P., Imbert, L., Jeljeli, H., Thomé, E.: Discrete logarithms in GF(p) — 180 digits. NMBRTHRY archives, item 004703 (Jun 2014), https://listserv.nodak.edu/cgi-bin/wa.exe?A2=NMBRTHRY;615d922a.1406
12. Chen, B., Zhao, C.A.: Self-pairings on supersingular elliptic curves with embedding degree three. Finite Fields and Their Applications 28, 79 – 93 (2014), http://www.sciencedirect.com/science/article/pii/S1071579714000240
13. Cohen, H.: A course in algorithmic algebraic number theory, Graduate Texts in Mathematics, vol. 138. Springer–Verlag (2000), fourth printing

14. Coppersmith, D.: Modifications to the number field sieve. Journal of Cryptology 6(3), 169–180 (1993)
15. Elkenbracht-Huizing, R.M.: An implementation of the number field sieve. Experiment. Math. 5(3), 231–253 (1996)
16. Freeman, D., Scott, M., Teske, E.: A taxonomy of pairing-friendly elliptic curves. Journal of Cryptology 23(2), 224–280 (Apr 2010)
17. Fried, J., Gaudry, P., Heninger, N., Thomé, E.: A kilobit hidden SNFS discrete logarithm computation. In: Coron, J., Nielsen, J.B. (eds.) EUROCRYPT 2017, Part I. LNCS, vol. 10210, pp. 202–231. Springer, Heidelberg (May 2017)
18. Gaudry, P., Grémy, L., Videau, M.: Collecting relations for the number field sieve in $GF(p^6)$. LMS Journal of Computation and Mathematics 19, 332 – 350 (2016), https://hal.inria.fr/hal-01273045
19. Gordon, D.M.: Discrete logarithms in GF(p) using the number field sieve. SIAM Journal on Discrete Mathematics 6(1), 124–138 (1993)
20. Granger, R., Kleinjung, T., Zumbrägel, J.: Breaking '128-bit secure' supersingular binary curves - (or how to solve discrete logarithms in $\mathbb{F}_{2^{4\cdot 1223}}$ and $\mathbb{F}_{2^{12\cdot 367}}$). In: Garay, J.A., Gennaro, R. (eds.) CRYPTO 2014, Part II. LNCS, vol. 8617, pp. 126–145. Springer, Heidelberg (Aug 2014)
21. Granger, R., Vercauteren, F.: On the discrete logarithm problem on algebraic tori. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 66–85. Springer, Heidelberg (Aug 2005)
22. Gras, M.N.: Special units in real cyclic sextic fields. Math. Comp. 48(177), 179–182 (1987), http://dx.doi.org/10.2307/2007882
23. Grémy, L.: Sieve algorithms for the discrete logarithm in medium characteristic finite fields. Ph.D. thesis, Université de Lorraine (2017)
24. Guillevic, A.: Computing individual discrete logarithms faster in GF(p^n) with the NFS-DL algorithm. In: Iwata, T., Cheon, J.H. (eds.) ASIACRYPT 2015, Part I. LNCS, vol. 9452, pp. 149–173. Springer, Heidelberg (Nov / Dec 2015)
25. Guillevic, A.: Faster individual discrete logarithms with the QPA and NFS variants. HAL archive (Aug 2017), 2nd version, https://hal.inria.fr/hal-01341849
26. Hayasaka, K., Aoki, K., Kobayashi, T., Takagi, T.: An experiment of number field sieve for discrete logarithm problem over GF(p^{12}). In: Fischlin, M., Katzenbeisser, S. (eds.) Number Theory and Cryptography. LNCS, vol. 8260, pp. 108–120. Springer (2013)
27. Hayasaka, K., Aoki, K., Kobayashi, T., Takagi, T.: A construction of 3-dimensional lattice sieve for number field sieve over $\mathbb{F}_{p^n}$. Cryptology ePrint Archive, Report 2015/1179 (2015), http://eprint.iacr.org/2015/1179
28. Joux, A.: A one round protocol for tripartite Diffie-Hellman. In: Bosma, W. (ed.) ANTS-IV. Lecture Notes in Comput. Sci., vol. 1838, pp. 385–394. Springer (2000)
29. Joux, A., Lercier, R.: Improvements to the general number field sieve for discrete logarithms in prime fields. A comparison with the Gaussian integer method. Math. Comp. 72(242), 953–967 (2003)
30. Joux, A., Lercier, R., Smart, N., Vercauteren, F.: The number field sieve in the medium prime case. In: Dwork, C. (ed.) CRYPTO 2006. LNCS, vol. 4117, pp. 326–344. Springer, Heidelberg (Aug 2006)
31. Kasahara, M., Ohgishi, K., Sakai, R.: Cryptosystems based on pairing. In: The 2000 Symposium on Cryptography and Information Security. vol. SCIS2000-C20 (Jan 2000)
32. Kim, T., Barbulescu, R.: Extended tower number field sieve: A new complexity for the medium prime case. In: Robshaw, M., Katz, J. (eds.) CRYPTO 2016, Part I. LNCS, vol. 9814, pp. 543–571. Springer, Heidelberg (Aug 2016)

33. Kim, T., Jeong, J.: Extended tower number field sieve with application to finite fields of arbitrary composite extension degree. In: Fehr, S. (ed.) PKC 2017, Part I. LNCS, vol. 10174, pp. 388–408. Springer, Heidelberg (Mar 2017)
34. Lenstra, A.K., Verheul, E.R.: The XTR public key system. In: Bellare, M. (ed.) CRYPTO 2000. LNCS, vol. 1880, pp. 1–19. Springer, Heidelberg (Aug 2000)
35. Lewko, A.B.: Tools for simulating features of composite order bilinear groups in the prime order setting. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 318–335. Springer, Heidelberg (Apr 2012)
36. Matyukhin, D.: Effective version of the number field sieve for discrete logarithms in the field GF(p^k) (in Russian). Trudy po Discretnoi Matematike 9, 121–151 (2006), http://m.mathnet.ru/php/archive.phtml?wshow=paper&jrnid=tdm&paperid=144&option_lang=eng
37. Matyukhin, D.V.: On asymptotic complexity of computing discrete logarithms over GF(p). Discrete Mathematics and Applications 13(1), 27–50 (2003)
38. Menezes, A., Sarkar, P., Singh, S.: Challenges with Assessing the Impact of NFS Advances on the Security of Pairing-Based Cryptography. In: Phan, R., Yung, M. (eds.) Mycrypt 2016. LNCS, vol. 10311, pp. 83–108. Springer (2017)
39. Miyaji, A., Nakabayashi, M., Takano, S.: Characterization of elliptic curve traces under FR-reduction. In: Won, D. (ed.) ICISC 00. LNCS, vol. 2015, pp. 90–108. Springer, Heidelberg (Dec 2001)
40. Murphy, B.A.: Polynomial selection for the number field sieve integer factorisation algorithm. Ph.D. thesis, Australian National Univers. (1999), http://maths-people.anu.edu.au/~brent/pd/Murphy-thesis.pdf
41. Pierrot, C.: The multiple number field sieve with conjugation and generalized Joux-Lercier methods. In: Oswald, E., Fischlin, M. (eds.) EUROCRYPT 2015, Part I. LNCS, vol. 9056, pp. 156–170. Springer, Heidelberg (Apr 2015)
42. Pollard, J.: The lattice sieve. In: Lenstra, A.K., Lenstra, Jr., H.W. (eds.) The development of the number field sieve, LNM, vol. 1554, pp. 43–49. Springer (1993)
43. Rubin, K., Silverberg, A.: Torus-based cryptography. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 349–365. Springer, Heidelberg (Aug 2003)
44. Sarkar, P., Singh, S.: A general polynomial selection method and new asymptotic complexities for the tower number field sieve algorithm. In: Cheon, J.H., Takagi, T. (eds.) ASIACRYPT 2016, Part I. LNCS, vol. 10031, pp. 37–62. Springer, Heidelberg (Dec 2016)
45. Sarkar, P., Singh, S.: New complexity trade-offs for the (multiple) number field sieve algorithm in non-prime fields. In: Fischlin, M., Coron, J.S. (eds.) EUROCRYPT 2016, Part I. LNCS, vol. 9665, pp. 429–458. Springer, Heidelberg (May 2016)
46. Schirokauer, O.: Discrete logarithms and local units. Philos. Trans. Roy. Soc. London Ser. A 345(1676), 409–423 (1993)
47. Schirokauer, O.: Using number fields to compute logarithms in finite fields. Math. Comp. 69(231), 1267–1283 (2000), http://www.ams.org/journals/mcom/2000-69-231/S0025-5718-99-01137-0/
48. The CADO-NFS development team: CADO-NFS, an implementation of the number field sieve algorithm (2017), http://cado-nfs.gforge.inria.fr/, development version
49. Zajac, P.: Discrete Logarithm Problem in Degree Six Finite Fields. Ph.D. thesis, Slovak University of Technology (2008), http://www.kaivt.elf.stuba.sk/kaivt/Vyskum/XTRDL