HAL
open science

Extended Tower Number Field Sieve

Taechan Kim, Razvan Barbulescu

To cite this version:

Taechan Kim, Razvan Barbulescu. Extended Tower Number Field Sieve. CRYPTO 2016, International association of cryptologic research, Aug 2016, Santa Barbara, United States. pp.543-571. hal-01281966

HAL Id: hal-01281966

https://hal.science/hal-01281966v1

Submitted on 3 Mar 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Extended Tower Number Field Sieve: A New Complexity for the Medium Prime Case*

Taechan Kim[1] and Razvan Barbulescu[2]

[1] NTT Secure Platform Laboratories, Japan
taechan.kim@lab.ntt.co.jp
[2] CNRS, Univ Paris 6, Univ Paris 7, France
razvan.barbulescu@imj-prg.fr

Abstract. We introduce a new variant of the number field sieve algorithm for discrete logarithms in $\mathbb{F}_{p^n}$ called exTNFS. The most important modification is done in the polynomial selection step, which determines the cost of the whole algorithm: if one knows how to select good polynomials to tackle discrete logs in $\mathbb{F}_{p^\kappa}$, exTNFS allows to use this method when tackling $\mathbb{F}_{p^{\eta\kappa}}$ whenever $\gcd(\eta, \kappa) = 1$. This simple fact has consequences on the asymptotic complexity of NFS in the medium prime case, where the complexity is reduced from $L_Q(1/3, \sqrt[3]{96/9})$ to $L_Q(1/3, \sqrt[3]{48/9})$, $Q = p^n$, respectively from $L_Q(1/3, 2.15)$ to $L_Q(1/3, 1.71)$ if multiple number fields are used. On the practical side, exTNFS can be used when $n = 6$ and $n = 12$ and this requires to update the keysizes used for the associated pairings-based cryptosystems.

Keywords: Discrete Logarithm Problem; Number Field Sieve; Finite Fields; Cryptanalysis

1 Introduction

The discrete logarithm problem (DLP) is at the foundation of a series of public key cryptosystems. Over a generic group of cardinality N, the best known algorithm to solve the DLP has an exponential running time of $O(\sqrt{N})$. However, if the group has a special structure one can design better algorithms, as it is the case for the multiplicative group of finite fields $\mathbb{F}_Q = \mathbb{F}_{p^n}$ where the DLP can be solved much more efficiently than in the exponential time.

When the characteristic p is small compared to the extension degree n, the best known algorithms have quasi-polynomial time complexity [6,17].

DLP over fields of medium and large characteristic Recall the usual L_Q-notation,

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

* This work is a merged version of two consecutive works [20] and [4].

for some constants $0 \leq \ell \leq 1$ and $c > 0$. We call the characteristic $p = L_Q(\ell_p, c_p)$ medium when $1/3 < \ell_p < 2/3$ and large when $2/3 < \ell_p \leq 1$. We are in the boundary case if $\ell_p = 2/3$.

For medium and large characteristic, in particular when Q is prime, all the state-of-art attacks are variants of the number field sieve (NFS) algorithm. Initially used for factoring, NFS was rapidly introduced in the context of DLP [16,26] to target prime fields. One had to wait almost one decade before the first constructions for $\mathbb{F}_{p^n}$ with $n > 1$ were proposed [27], known today [7] as the tower number field sieve (TNFS). This case is important because it is used to choose the key sizes for pairings based cryptosystems. Since 2006 one can cover the complete range of large and medium characteristic finite fields [18]. This latter approach that we denote by JLSV has the advantage to be very similar to the variant used to target prime fields, except for the first step called polynomial selection where two new methods were proposed: JLSV_1 and JLSV_2.

In the recent years NFS in fields $\mathbb{F}_{p^n}$ with $n > 1$ has become a laboratory where one can push NFS to its limits and test new ideas which are ineffective or impossible in the factorization variant of NFS. Firstly, the polynomial selection methods were supplemented with the generalized Joux-Lercier (GJL) method [22,5], with the Conjugation (Conj) method [5] and the Sarkar-Singh (SS) method [25]. One can see Table 1 for a summary of the consequences of these methods on the asymptotic complexity. In particular, in all these algorithms the complexity for the medium prime case is slightly larger than that of the large prime case.

Table 1: The complexity of each algorithms in the medium and large prime cases. Each cell indicates c if the complexity is $L_Q(1/3, (c/9)^{\frac{1}{3}})$.

$p = L_Q(\ell_p)$	$1/3 < \ell_p < 2/3$	best $\ell_p = 2/3$	$2/3 < \ell_p < 1$
TNFS [27,7]	none	none	64
NFS-JLSV [18]	128	64	64
NFS-(Conj and GJL) [5]	96	48	64
NFS-SS [25]	96	48	64
exTNFS (this article)	48	48	64

Secondly, a classical idea which was introduced in the context of factorization is to replace the two polynomials f and g used in NFS by a polynomial f and several polynomials g_i, $i = 1, 2, \ldots$ which play the role of g. All the currently known variants of NFS admit variants with multiple number fields (MNFS) which have a slightly better asymptotic complexity, as shown in Table 2. The discrete logarithm problem allows to have a case with no equivalent in the factorization context: instead of having a distinguished polynomial f and many sides g_i all the polynomials are interchangeable [8].

Table 2: The complexity of each algorithms using multiple number fields. Each cell indicates an approximation of c if the complexity is $L_Q(1/3, (c/9)^{\frac{1}{3}})$

$p = L_Q(\ell_p)$	$1/3 < \ell_p < 2/3$	best $\ell_p = 2/3$	$2/3 < \ell_p < 1$
MTNFS [7]	none	none	61.93
MNFS-JLSV [8]	122.87	61.93	61.93
MNFS-(Conj and GJL) [24]	89.45	45.00	61.93
MNFS-SS [25]	89.45	45.00	61.93
MexTNFS (this article)	45.00	45.00	61.93

Thirdly, when the characteristic p has a special form, as it is the case for fields in several pairings-based cryptosystems, one might speed-up the computations by variants called special number field sieve (SNFS). In Table 3 we list the asymptotic complexity of each algorithm. Once again, the medium characteristic case is harder than the large characteristic one.

Table 3: The complexity of each algorithms used when the characteristic has a special form (SNFS) Each cell indicates an approximation of c if the complexity is $L_Q(1/3, (c/9)^{\frac{1}{3}})$

$p = L_Q(\ell_p)$	$1/3 < \ell_p < 2/3$	$2/3 < \ell_p < 1$
JP [19]	64	32
STNFS [7]	none	32
SexTNFS (this article)	32	32

Our contributions Let us place ourselves in the case when the extension degree is composite with relatively prime factors, $n = \eta\kappa$ with $\gcd(\eta, \kappa) = 1$. The basic idea is to use the trivial equality

$$\mathbb{F}_{p^n} = \mathbb{F}_{(p^\eta)^\kappa}.$$

In the JLSV algorithm, $\mathbb{F}_{p^n}$ is constructed as $\mathbb{F}_p[x]/k(x)$ for an irreducible polynomial $k(x)$ of degree n. In the TNFS algorithm $\mathbb{F}_{p^n}$ is obtained as R/pR where R is a ring of integers of a number field where p is inert. In our construction $\mathbb{F}_{p^\eta} = R/pR$ as in TNFS and $\mathbb{F}_{p^n} = (R/pR)[x]/(k(x))$ where k is a degree κ irreducible polynomial over $\mathbb{F}_{p^\eta}$.

Interestingly, this construction can be integrated in an algorithm, that we call the extended number field sieve (exTNFS), in which we can target $\mathbb{F}_{p^{\eta\kappa}}$ with the

same complexity as $\mathbb{F}_{P^\kappa}$ for a prime P of the same bitsize as p^η. Hence we obtain complexities for composite extension degrees which are similar in the medium characteristic case to the large characteristic case. Since the previous algorithms have an "anomaly" in the case $\ell_p = 2/3$, where the complexity is better than in the large prime case, when n is composite we obtain a better complexity for the medium prime case than in the large prime case.

Overview We introduce the new algorithm in Section 2 and analyse its complexity in Section 3. The multiple number field variant and the one dedicated to fields of SNFS characteristic are discussed in Section 4. In Section 5 we make a precise comparison to the state-of-art algorithms at cryptographic sizes before concluding with the consequences on the key size estimations for pairing-based construction.

2 Extended TNFS

2.1 Setting

Throughout this paper, we target fields $\mathbb{F}_Q$ with $Q = p^n$ where $n = \eta\kappa$ such that $\gcd(\eta, \kappa) = 1$ and the characteristic p is medium or large, i.e. $\ell_p > 1/3$.

First we select a polynomial $h(t) \in \mathbb{Z}[t]$ of degree η which is irreducible modulo p. We put $R := \mathbb{Z}[t]/h(t)$ and note that $R/pR \simeq \mathbb{F}_{p^\eta}$. Then we select two polynomials f and g with integer coefficients whose reductions modulo p have a common factor $k(x)$ of degree κ which is irreducible over $\mathbb{F}_{p^\eta}$. Our algorithm is unchanged if f and g have coefficients in R because in all the cases we use the number fields K_f (resp. K_g) defined by f (resp. g) above the fraction field of R but this generalization is not needed for the purpose of this paper, except in a MNFS variant.

The conditions on f, g and h yield two ring homomorphisms from $R[x]/f(x)$ (resp. $R[x]/g(x)$) to $(R/pR)/k(x) = \mathbb{F}_{p^{\eta\kappa}}$: in order to compute the reduction of a polynomial in $R[x]$ modulo p then modulo $k(x)$ one can start by reducing modulo f (resp. g) and continue by reducing modulo p and then modulo $k(x)$. The result is the same if we use f as when we use g. Thus one has the commutative diagram in Figure 1 which is a generalization of the classical diagram of NFS.

After the polynomial selection, the exTNFS algorithm proceeds as all the variants of NFS, following the same steps: relations collection, linear algebra and individual logarithm. Most of these steps are very similar to the TNFS algorithms as we shall explain below.

2.2 Detailed Descriptions

Polynomial Selection

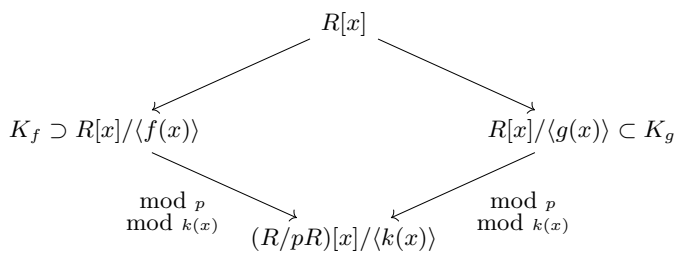

Fig. 1: Commutative diagram of exTNFS. When $R = \mathbb{Z}$ this is the diagram of NFS for non-prime fields. When $k(x) = x - m$ for some $m \in R$ this is the diagram of TNFS. When both $R = \mathbb{Z}$ and $k(x) = x - m$ this is the diagram of NFS.

Choice of h We have to select a polynomial $h(t) \in \mathbb{Z}[x]$ of degree η which is irreducible modulo p and whose coefficients are as small as possible. As in TNFS we try random polynomials h with small coefficients and factor them in $\mathbb{F}_p[t]$ to test irreducibility. Heuristically, one succeeds after η trials and since $\eta \leq 3^\eta$ we expect to find h such that $\|h\|_\infty = 1$. For a more rigorous description on the existence of such polynomials one can refer to [7].

Next we select f and g in $\mathbb{Z}[x]$ which have a common factor $k(x)$ modulo p of degree κ which remains irreducible over $\mathbb{F}_{p^\eta}$. It is here that we use the condition $\gcd(\eta, \kappa) = 1$ because an irreducible polynomial $k(x) \in \mathbb{F}_p[x]$ remains irreducible over $\mathbb{F}_{p^\eta}$ if and only if $\gcd(\eta, \kappa) = 1$. If one has an algorithm to select f and g in $R[x]$ one might drop this condition, but in this paper f and g have integer coefficients. Thus it is enough to test the irreducibility of $k(x)$ over $\mathbb{F}_p$ and we have the same situation as in the classical variant of NFS for non-prime fields (JLSV): JLSV_1, JLSV_2, Conjugation method, GJL and Sarkar-Singh. Let us present two of these methods which are important for results of asymptotic complexity.

$JLSV_2$ method We briefly describe the polynomial selection introduced in Section 3.2 of [18]. One first chooses a monic polynomial $f_0(x)$ of degree κ with small coefficients, which is irreducible over $\mathbb{F}_p$ (and automatically over $\mathbb{F}_{p^\eta}$ because $\gcd(\eta, \kappa) = 1$). Set an integer $W \approx p^{1/(D+1)}$, where D is a parameter determined later subject to the condition $D \geq \kappa$. Then we define $f(x) := f_0(x + W)$. Take the coefficients of $g(x)$ as the shortest vector of an LLL-reduced basis of the lattice L defined by the columns:

$$L := (p \cdot \mathbf{x^0}, \ldots, p \cdot \mathbf{x}^\kappa, \mathbf{f(x)}, \mathbf{xf(x)}, \ldots, \mathbf{x^{D+1-\kappa} f(x)}).$$

Here, $\mathbf{f(x)}$ denotes the vector formed by the coefficients of a polynomial f. Finally, we set $k = f$ then we have

- $\deg(f) = \kappa$ and $\|f\|_\infty = O(p^{\frac{\kappa}{D+1}})$;
- $\deg(g) = D \geq \kappa$ and $\|g\|_\infty = O(p^{\frac{\kappa}{D+1}})$.

Conjugation method We recall the polynomial selection method in Algorithm 4 of [5]. First, one chooses two polynomials $g_1(x)$ and $g_0(x)$ with small coefficients such that $\deg g_1 < \deg g_0 = \kappa$. Next one chooses a quadratic, monic, irreducible polynomial $\mu(x) \in \mathbb{Z}[x]$ with small coefficients. If $\mu(x)$ has a root δ in $\mathbb{F}_p$ and $g_0 + \delta g_1$ is irreducible over $\mathbb{F}_p$ (and automatically over $\mathbb{F}_{p^\eta}$ because $\gcd(\eta, \kappa) = 1$), then set $k = g_0 + \delta g_1$. Otherwise, one repeats the above steps until such g_1, g_0, and δ are found. Once it has been done, find u and v such that $\delta \equiv u/v \pmod p$ and $u, v \leq O(\sqrt{p})$ using rational reconstruction. Finally, we set $f = \mathrm{Res}_Y(\mu(Y), g_0(x) + Yg_1(x))$ and $g = vg_0 + ug_1$. By construction we have

- $\deg(f) = 2\kappa$ and $\|f\|_\infty = O(1)$;
- $\deg(g) = \kappa$ and $\|g\|_\infty = O(\sqrt{p}) = O(Q^{\frac{1}{2\eta\kappa}})$.

The bound on $\|f\|_\infty$ depends on the number of polynomials $g_0 + \delta g_1$ tested before we find one which is irreducible over $\mathbb{F}_p$. Heuristically this happens on average after 2κ trials. Since there are $3^{2\kappa} > 2\kappa$ choices of g_0 and g_1 of norm 1 we have $\|f\|_\infty = 1$.

Relation Collection The elements of $R = \mathbb{Z}[x]/h(x)$ can be represented uniquely as polynomials of $\mathbb{Z}[x]$ of degree less than $\deg h$.

We proceed as in TNFS and enumerate all the pairs $(a, b) \in \mathbb{Z}[t]^2$ of degree $\leq \eta - 1$ such that $\|a\|_\infty$, $\|b\|_\infty \leq A$ for a parameter A to be determined. We say that we obtain a relation for the pair (a, b) if

$$N_f(a, b) := \mathrm{Res}_t(\mathrm{Res}_x(a(t) - b(t)x, f(x)), h(t)) \text{ and}$$
$$N_g(a, b) := \mathrm{Res}_t(\mathrm{Res}_x(a(t) - b(t)x, g(x)), h(t))$$

are B-smooth for a parameter B to be determined (an integer is B-smooth if all its prime factors are less than B). If ι denotes a root of h in R our enumeration is equivalent to putting linear polynomials $a(\iota) - b(\iota)x$ in the top of the diagram of Figure 1. One can generalize exTNFS to the case where one puts non-linear polynomials $r(x) \in R[x]$ in the diagram but this is not necessary in this paper.

For each pair (a, b) we can write a linear equation and this part, although computationally negligible, demands some mathematical details.

Factor base Let α_f (resp. α_g) be a root of f in K_f (resp. of g in K_g), the number field it defines over the fraction field of R. Then the norm of $a(\iota) - b(\iota)\alpha_f$ (resp. $a(\iota) - b(\iota)\alpha_g$) over $\mathbb{Q}$ is $\mathrm{Res}_t(\mathrm{Res}_x(a(t) - b(t)x, f(x)), h(t))$ (resp. $\mathrm{Res}_t(\mathrm{Res}_x(a(t) - b(t)x, g(x)), h(t))$) up to a power of $l(f)$ (resp. $l(g)$), the leading coefficient of f (resp. g). We call factor base the set of prime ideals of K_f and K_g which can occur in the factorization of $a(\iota) - b(\iota)\alpha_f$ and $a(\iota) - b(\iota)\alpha_g$ when both norms are B-smooth. By Proposition 1 in [7] we can give an explicit description of the factor base as $\mathcal{F}(B) := \mathcal{F}_f(B) \bigcup \mathcal{F}_g(B)$ where

$$\mathcal{F}_f(B) = \left\{ \langle \mathfrak{q}, \alpha - \gamma \rangle : \begin{array}{l} \mathfrak{q} \text{ is a prime in } \mathbb{Q}(\iota) \text{ lying over a prime} \\ p \leq B \text{ and } f(\gamma) \equiv 0 \pmod{\mathfrak{q}} \end{array} \right\}$$
$$\bigcup \{\text{prime ideals of } K_f \text{ dividing } l(f)\mathrm{Disc}(f)\}.$$

and similarly for $\mathcal{F}_g(B)$.

Schirokauer maps If $\langle a(\iota) - b(\iota)\alpha_f \rangle = \prod_{\mathfrak{q} \in \mathcal{F}_f(B)} \mathfrak{q}^{\mathrm{val}_{\mathfrak{q}}(a(\iota)-b(\iota)\alpha_f)}$ and $\langle a(\iota) - b(\iota)\alpha_g \rangle = \prod_{\mathfrak{q} \in \mathcal{F}_g(B)} \mathfrak{q}^{\mathrm{val}_{\mathfrak{q}}(a(\iota)-b(\iota)\alpha_g)}$ we write

$$\sum_{\mathfrak{q} \in \mathcal{F}_f(B)} \mathrm{val}_{\mathfrak{q}}(a(\iota)-b(\iota)\alpha_f) \log \mathfrak{q} + \epsilon_f(a,b) = \sum_{\mathfrak{q} \in \mathcal{F}_g(B)} \mathrm{val}_{\mathfrak{q}}(a(\iota)-b(\iota)\alpha_g) \log \mathfrak{q} + \epsilon_g(a,b)$$

where the log sign denotes virtual logarithms in the sense of [26] and [18] and ϵ_f and ϵ_g are correction terms called Schirokauer maps which were first introduced in [26].

The novelty for TNFS and exTNFS with respect to JLSV is that K_f and K_g are constructed as tower extensions instead of absolute extensions. On the other hand, it is more convenient to work on absolute extensions when we compute Schirokauer maps. We solve this problem by computing primitive elements θ_f (resp. θ_g) of $K_f/\mathbb{Q}$ (resp. $K_g/\mathbb{Q}$). For a proof we refer to Section 4.3 in [18].

Linear algebra and individual logarithm These two steps are unchanged with respect to the classical variant of NFS. The linear algebra step, comes after relation collection and consists in solving the linear system over $\mathbb{F}_l$ for some prime factor l of the order of $\mathbb{F}_Q^*$. Using Wiedemann's algorithm this has a quasi-quadratic complexity in the size of the linear system, which is equal to the cardinality of the factor base. In [7] it is shown that the factor base has $(2 + o(1))B/\log B$ elements, so the cost of the linear algebra is $B^{2+o(1)}$.

In the individual logarithm step one writes any desired discrete logarithm as a sum of virtual logarithms of elements in the factor base. Since the step is very similar to the corresponding step in NFS we keep the description for the Appendix.

3 Complexity

The complexity analysis of exTNFS follows the steps of the analysis of NFS in the case of prime fields. It is expected that the stages of the algorithm other than the relation collection and the linear algebra are negligible, hence we select parameters to minimize their cost and afterwards we check that the other stages are indeed negligible.

Let us call T the time spent in average for each polynomial $r \in R[x]$ enumerated in the relation collection stage (in this paper $r = a(\iota) - b(\iota)x$), and let P_f (resp. P_g) be the probability that the norm N_f (resp. N_g) of r with respect to f (resp. g) is B-smooth. The number of polynomials that we test before finding each new relation is on average $1/(P_f P_g)$, so the cost of the relations collection is $\#\mathcal{F}(B)T/(P_f P_g)$.

We make the usual heuristic that the proportion of smooth norms is the same as the proportion of arbitrary positive integers of the same size, so $P_f = \mathrm{Prob}(N_f, B)$ (resp $P_g = \mathrm{Prob}(N_g, B)$) where $\mathrm{Prob}(x, y)$ is the probability that an arbitrary integer less than x is y-smooth. The value of T depends on whether we use a sieving technique or we consider each value and test smoothness with

ECM [21]; if we use the latter variant we obtain $T = L_B(1/2, \sqrt{2})(\log Q)^{O(1)}$, so $T = B^{o(1)}$. Using the algorithm of Wiedemann [28] the cost of the linear algebra is $(\#\mathcal{F}(B))^{2+o(1)} = B^{2+o(1)}$. Hence, up to an exponent $1 + o(1)$, we have

$$\text{complexity(exTNFS)} = \frac{B}{\text{Prob}(N_f, B)\text{Prob}(N_g, B)} + B^2. \tag{1}$$

This equation is the same for NFS, TNFS, exTNFS and the corresponding SNFS variants. The differences begin when we look at the size of N_f and N_g which depend on the polynomial selection method. In what follows we instantiate Equation (1) with various cases and obtain equations which have already been analysed in the literature.

Lemma 1. *Let h and f be irreducible polynomials over $\mathbb{Z}$ and call $\eta := \deg h$ and $\kappa := \deg(f)$. Let $a(t), b(t) \in \mathbb{Z}[t]$ be polynomials of degree at most $\eta - 1$ with $\|a\|_\infty, \|b\|_\infty \leq A$. We put $N_f(a, b) := \text{Res}_t(\text{Res}_x(a(t) - b(t)x, f(x)), h(t))$. Then we have*

1. $$|N_f(a, b)| < A^{\eta\cdot\kappa}\|f\|_\infty^\eta \|h\|_\infty^{\kappa\cdot(\eta-1)} C(\eta, \kappa), \tag{2}$$
   where $C(\eta, \kappa) = (\eta + 1)^{(3\kappa+1)\eta/2}(\kappa + 1)^{3\eta/2}$.
2. *Assume in addition that $\|h\|_\infty$ is bounded by an absolute constant H and that $p = L_Q(\ell_p, c)$ for some $\ell_p > 1/3$ and $c > 0$. Then*
   $$N_f(a, b) \leq E^\kappa \|f\|_\infty^\eta L_Q(2/3, o(1)), \tag{3}$$
   where $E = A^\eta$

Proof. 1. This is proven in Theorem 3 in [7].
2. The overhead is bounded as follows

$$\begin{aligned}\log(\|h\|_\infty^{\kappa(\eta-1)} C(\eta, \kappa)) &\leq \kappa\eta \log H + 3\kappa\eta \log \eta + 3\eta \log \kappa \\ &= O(\log(Q)^{1-\ell_p}(\log\log Q)^{\ell_p}) \\ &= o(1)\log(Q)^{2/3}(\log\log Q)^{1/3}.\end{aligned}$$

□

If $N_f = L_Q(2/3)$ then we can forget the overhead $L_Q(2/3, o(1))$ as the Canfield-Erdös-Pomerance theorem states that the smoothness probability satisfies, uniformly on x and y in the validity domain,

$$\text{Prob}(x^{1+o(1)}, y) = \text{Prob}(x, y)^{1+o(1)}.$$

The next statement summarizes our results.

Theorem 1. *(under the classical NFS heuristics) If $Q = p^n$ is a prime power such that*

- *$p = L_Q(\ell_p, c_p)$ with $1/3 < \ell_p$;*

algorithm	C	conditions
exTNFS-JLSV$_2$	$(64/9)^{\frac{1}{3}}$	$\kappa = o\left((\frac{\log Q}{\log\log Q})^{\frac{1}{3}}\right)$
exTNFS-GJL	$(64/9)^{\frac{1}{3}}$	$\kappa \leq (\frac{8}{3})^{-\frac{1}{3}}(\frac{\log Q}{\log\log Q})^{\frac{1}{3}}$
exTNFS-Conj	$(48/9)^{\frac{1}{3}}$	$\ell_p < 2/3$ or $\ell_p = 2/3$ and $c_p < 12^{\frac{1}{3}}$ $\kappa = 12^{-\frac{1}{3}}(\frac{\log Q}{\log\log Q})^{\frac{1}{3}}$
SexTNFS	$(32/9)^{\frac{1}{3}}$	$\kappa = o\left((\frac{\log Q}{\log\log Q})^{\frac{1}{3}}\right)$ p is d-SNFS with $d = \frac{(2/3)^{\frac{1}{3}}+o(1)}{\kappa}(\frac{\log Q}{\log\log Q})^{\frac{1}{3}}$
MexTNFS-JLSV$_2$	$(\frac{92+26\sqrt{13}}{27})^{\frac{1}{3}}$	$\kappa = o\left((\frac{\log Q}{\log\log Q})^{\frac{1}{3}}\right)$
MexTNFS-GJL	$(\frac{92+26\sqrt{13}}{27})^{\frac{1}{3}}$	$\kappa \leq (\frac{7+2\sqrt{13}}{6})^{-1/3}(\frac{\log Q}{\log\log Q})^{\frac{1}{3}}$
MexTNFS-Conj	$\frac{3+\sqrt{3(11+4\sqrt{6})}}{\left(18(7+3\sqrt{6})\right)^{1/3}}$	$\ell_p < 2/3$ or $\ell_p = 2/3$ and $c_p < (\frac{56+24\sqrt{6}}{12})^{1/3}$ $\kappa = ((\frac{56+24\sqrt{6}}{12})^{-1/3} + o(1))(\frac{\log Q}{\log\log Q})^{\frac{1}{3}}$

Table 4: Complexity of exTNFS variants

– $n = \eta\kappa$ *such that* $\gcd(\eta, \kappa) = 1$

then the discrete logarithm over $\mathbb{F}_Q$ *can be solved in* $L_Q(1/3, C)$ *where* C *and the additional conditions are listed in Table 4.*

In the rest of this section we prove this statement. In any case in the table, one shares the conditions $\kappa = o\left((\frac{\log Q}{\log\log Q})^{\frac{1}{3}}\right)$ or $\kappa \leq c(\frac{\log Q}{\log\log Q})^{\frac{1}{3}}$ for some constant $c > 0$. These are equivalent to say that $P = p^\eta = L_Q(\ell_P)$ for some $\ell_P \geq 2/3$.

3.1 exTNFS-JLSV$_2$

In this section we assume that n has a factor κ such that

$$\kappa = o\left(\left(\frac{\log(Q)}{\log\log(Q)}\right)^{1/3}\right).$$

Let us introduce $\|h\|_\infty = O(1)$ and the values of $\|f\|_\infty, \|g\|_\infty \approx p^{\kappa/(D+1)}$ coming from the JLSV$_2$ method (Section 2.2) in Equation (2). Then we get

$$|N_f(a,b)| < \left(A^{\eta\kappa}(p^{\frac{\kappa}{D+1}})^\eta\right)^{1+o(1)} = \left(E^\kappa P^{\frac{\kappa}{D+1}}\right)^{1+o(1)}, \tag{4}$$

$$|N_g(a,b)| < \left(A^{\eta D}(p^{\frac{\kappa}{D+1}})^\eta\right)^{1+o(1)} = \left(E^D P^{\frac{\kappa}{D+1}}\right)^{1+o(1)}, \tag{5}$$

where we set $E := A^\eta$ and $P := |R/pR| = p^\eta$.

One recognizes the expressions for the norms in the large prime case [18, Appendix A.3.], where $P = p$ and $\kappa = n$. We conclude that we have the same complexity:

$$\text{complexity(exTNFS with JLSV}_2) = L_Q(1/3, \sqrt[3]{64/9}).$$

3.2 exTNFS-GJL

We relax a bit the condition from the previous section: we assume that n has a factor κ such that

$$\kappa \leq (8/3)^{-\frac{1}{3}} \left(\frac{\log(Q)}{\log\log(Q)} \right)^{1/3}.$$

Recall the characteristics of our polynomials: $\|h\|_\infty = O(1)$ and $\deg h = \eta$; $\|f\|_\infty = O(1)$ and $\deg f = d+1$ for a parameter $d \geq \kappa$; $\|g\|_\infty \approx p^{\kappa/(d+1)}$ and $\deg g = d$. We inject these values in Equation (2) and we get

$$|N_f(a,b)| < E^{d+1} L_Q(2/3, o(1)), \tag{6}$$

$$|N_g(a,b)| < E^d Q^{1/(d+1)} L_Q(2/3, o(1)), \tag{7}$$

where we set $E := A^\eta$ and $P := |R/pR| = p^\eta$. We recognize the expression in the first equation of Section 4.2 in [5], so

$$\text{complexity(exTNFS with GJL)} = L_Q(1/3, \sqrt[3]{64/9}).$$

3.3 exTNFS-Conj

We propose here a variant of NFS which combines exTNFS with the Conjugation method of polynomial selection.

Let us consider the case when $n = \eta\kappa$ with

$$\kappa = \left(\frac{1}{12^{1/3}} + o(1) \right) \left(\frac{\log(Q)}{\log\log(Q)} \right)^{1/3}.$$

As before, evaluating the values coming from the Conjugation method (Section 2.2) in Equation (2), we have

$$|N_f(a,b)| < E^{2\kappa} L_Q(2/3, o(1)), \tag{8}$$

$$|N_g(a,b)| < E^\kappa (p^{\kappa\eta})^{1/(2\kappa)} L_Q(2/3, o(1)). \tag{9}$$

When we combine Equations (8) and (9) we obtain

$$|N_f(a,b)| \cdot |N_g(a,b)| < E^{3\kappa} Q^{(1+o(1))/(2\kappa)}.$$

But this is Equation (5) in [5] when $t = 2$. The rest of the computations are identical as in point 3. of Theorem 1 in [5], so

$$\text{complexity(exTNFS-Conj)} = L_Q(1/3, (48/9)^{1/3}).$$

4 Variants

4.1 The case when p is has a special form (SexTNFS)

In some pairings-based constructions p has a special form, e.g. in the Barreto-Naehrig curves [9] $p = 36u^4 + 36u^3 + 24u^2 + 6u + 1$ of embedding degree 12 and

in the Freeman pairings construction of embedding degree 10 [14, Section 5.3] $p = 25u^4 + 25u^3 + 25u^2 + 10u + 3$. For a given integer d, an integer p is d-SNFS if there exists an integer u and a polynomial $\Pi(x)$ with integer coefficients so that

$$p = \Pi(u),$$

$\deg \Pi = d$ and $\|\Pi\|_\infty$ is bounded by an absolute constant.

We consider the case when $n = \eta\kappa$, $\gcd(\eta, \kappa) = 1$ with $\kappa = o\left(\left(\frac{\log Q}{\log\log Q}\right)^{1/3}\right)$ and p is d-SNFS. In this case exTNFS is unchanged: we select h, f and g three polynomials with integer coefficients so that

- h is irreducible modulo p, $\deg h = \eta$ and $\|h\|_\infty = O(1)$;
- f and g have a common factor $k(x)$ modulo p which is irreducible of degree κ.

Choice of f and g using the method of Joux and Pierrot Find a polynomial S of degree $\kappa - 1$ with coefficients in $\{-1, 0, 1\}$ so that $k(x) = x^\kappa + S(x) - u$ is irreducible modulo p. Since the proportion of irreducible polynomials in $\mathbb{F}_p$ of degree κ is $1/\kappa$ and there are 3^κ choices we expect this step to succeed. Then we set

$$\begin{cases} g = x^\kappa + S(x) - u \\ f = \Pi(x^\kappa + S(x)). \end{cases}$$

If f is not irreducible over $\mathbb{Z}[x]$, which arrives with negligible probability, start over. Note that g is irreducible modulo p and that f is a multiple of g modulo p. By construction we have:

- $\deg(g) = \kappa$ and $\|g\|_\infty = u = p^{1/d}$;
- $\deg(f) = \kappa d$ and $\|f\|_\infty = 2^d \|\Pi\|_\infty = O(2^d)$.

Let us compute the analysis of this particular case of exTNFS. We inject these values in Equations (2) and obtain

$$|N_f(a, b)| \leq E^{\kappa d} L_Q(2/3, o(1))$$
$$|N_g(a, b)| \leq E^\kappa P^{1/d} L_Q(2/3, o(1)),$$

where $E := A^\eta$ and $P := |R/pR| = p^\eta$. We recognize the size of the norms in the analysis by Joux and Pierrot [19, Section 6.3.], so we obtain the same complexity as in their paper:

$$\text{complexity(exTNFS for SNFS primes)} = L_Q(1/3, (32/9)^{1/3}).$$

4.2 The multiple polynomial variants (MexTNFS)

Virtually every variant of NFS can be accelerated using multiple polynomials and exTNFS makes no exception. The multiple variant of exTNFS is as follows: choose f and g which have a common factor $k(x)$ modulo p which is irreducible of degree κ using any of the methods given in Section 2.2. Next we set $f_1 = f$

and $f_2 = g$ and select other $V-2$ irreducible polynomials $f_i := \mu_i f_1 + \nu_i f_2$ where $\mu_i = \sum_{j=0}^{\eta-1} \mu_{i,j}\iota^j$ and $\nu_i = \sum_{j=0}^{\eta-1} \nu_{i,j}\iota^j$ are elements of $R = \mathbb{Z}[t]/h\mathbb{Z}[t]$ such that $\|\mu_i\|_\infty, \|\nu_i\|_\infty \leq V^{\frac{1}{2\eta}}$ where $V = L_Q(1/3, c_v)$ is a parameter which will be selected later. Denote α_i a root of f_i for $i = 1, 2, \ldots, V$.

Once again the complexity depends on the manner in which the polynomials f and g are selected.

MexTNFS-JLSV$_2$ Barbulescu and Pierrot [8, Section 5.3.] analysed the complexity of MNFS with JLSV$_2$, so we only need to check that the size of the norm is the same for NFS and exTNFS for each polynomial f_i with $1 \leq i \leq V$. By construction we have:

- $\deg(f_1) = \kappa$ and $\|f_1\|_\infty = p^{\frac{\kappa}{D+1}}$;
- $\deg(f_i) = D \geq \kappa$ and $\|f_i\|_\infty = V^{\frac{1}{2\eta}} p^{\frac{\kappa}{D+1}}$ for $2 \leq i \leq V$.

As before, we inject these values in Equations (2) and obtain

$$|N_{f_1}(a,b)| < E^\kappa (p^{\kappa\eta})^{\frac{1}{D+1}} L_Q(2/3, o(1))$$
$$|N_{f_i}(a,b)| < E^D (p^{\kappa\eta})^{\frac{1}{D+1}} L_Q(2/3, o(1)) \text{ for } 2 \leq i \leq V.$$

We emphasize that $(V^{\frac{1}{2\eta}})^\eta = V^{\frac{1}{2}} = L_Q(1/3, c_v/2) = L_Q(2/3, o(1))$ which is true without any condition on η. Hence we obtain

$$\text{complexity(MexTNFS-JLSV}_2) = L_Q\left(1/3, \left(\frac{92+26\sqrt{13}}{27}\right)^{1/3}\right).$$

MexTNFS-Conj and GJL Pierrot [24] studied the multiple polynomial variant of NFS when the Conjugation method or GJL are used. To show that we obtain the same complexities we need to show that the norm with respect to each polynomial is the same as in the classical NFS, except for a factor $L_Q(2/3, o(1))$, which boils down to testing again that $(V^{\frac{1}{2\eta}})\eta = L_Q(2/3, o(1))$ which is always true. When $P = p^\eta = L_Q(2/3, c_P)$ such that $c_P > (\frac{7+2\sqrt{13}}{6})^{1/3}$ and t is the number of coefficients of the enumerated polynomials r, then the complexity obtained is $L_Q(1/3, C(t, c_P))$ where

$$C(t, c_P) = \frac{2}{c_P t} + \sqrt{\frac{20}{9(c_P t)^2} + \frac{2}{3} c_P (t-1)}.$$

The best case is when $c_P = (\frac{56+24\sqrt{6}}{12})^{1/3}$ and $t = 2$ (linear polynomials):

$$\text{complexity(best case of MexTNFS-Conj)} = L_Q\left(1/3, \frac{3+\sqrt{3(11+4\sqrt{6})}}{\left(18(7+3\sqrt{6})\right)^{1/3}}\right),$$

where the second constant being approximated by 1.71.

5 Comparison and examples

NFS, TNFS and exTNFS have the same main lines:

- we compute a large number of integer numbers;
- we factor these numbers to test if they are B-smooth for some parameter B;
- we solve a linear system depending on the previous steps.

If we reduce the size of the integers computed in the algorithm we reduce the work needed to find a subset of integers which are B-smooth, which further allows us to adapt the other parameters so that the linear algebra is also cheap. A precise analysis is complex because in some variants one tests smoothness using ECM while in others one can sieve (which is faster). Nevertheless, as a first comparison we use the criterion in which one must minimize the bitsize of the product of the norms.

5.1 Precise comparison when p is arbitrary

Each method of polynomial selection has a different expression of the norm bitsize, which depends on the number t of coefficients of the polynomials $r(x)$ that are enumerated during the relation collection. Let us reproduce Table 2 in [25], which we extend with TNFS and exTNFS:

Method	norms product	conditions
NFS-JLSV$_1$	$E^{\frac{4n}{t}} Q^{\frac{t-1}{n}}$	
NFS-JLSV$_2$	$E^{\frac{2(n+D)}{t}} Q^{\frac{t-1}{D+1}}$	$D \geq n$
NFS-GJL	$E^{\frac{2(2r+1)}{t}} Q^{\frac{t-1}{r+1}}$	$r \geq n$
NFS-Conj	$E^{\frac{6n}{t}} Q^{\frac{t-1}{2n}}$	
NFS-SS	$E^{\frac{2\eta(2r+1)}{t}} Q^{\frac{t-1}{\eta(r+1)}}$	$n = \eta\kappa, r \geq \kappa$
TNFS	$E^{\frac{2(d+1)}{t}} Q^{\frac{2(t-1)}{d+1}}$	n small
exTNFS-JLSV$_1$	$E^{\frac{4\kappa}{t}} Q^{\frac{t-1}{\kappa}}$	$n = \eta\kappa, \gcd(\eta, \kappa) = 1, \eta$ small
exTNFS-JLSV$_2$	$E^{\frac{2(\kappa+D)}{t}} Q^{\frac{t-1}{D+1}}$	$n = \eta\kappa, \gcd(\eta, \kappa) = 1, \eta$ small, $r \geq \kappa$
exTNFS-GJL	$E^{\frac{2(2r+1)}{t}} Q^{\frac{t-1}{r+1}}$	$n = \eta\kappa, \gcd(\eta, \kappa) = 1, \eta$ small, $r \geq \kappa$
exTNFS-Conj	$E^{\frac{6\kappa}{t}} Q^{\frac{(t-1)}{2\kappa}}$	$n = \eta\kappa, \gcd(\eta, \kappa) = 1, \eta$ small
exTNFS-SS	$E^{\frac{2d(2r+1)}{t}} Q^{\frac{t-1}{d(r+1)}}$	$n = d\eta\kappa, \gcd(\eta, \kappa) = 1, \eta$ small, $r \geq \kappa$

Table 5: Comparison of norm sizes.

Note that the method of Sarkar and Singh requires that n is composite. The settings based on TNFS (TNFS, exTNFS-GJL etc) have an overhead due to the combinatorial factor which is not written in this table, so we add the condition that the degree of the intermediate number field must be small. Finally, exTNFS requires the additional condition that κ and η are relatively prime.

Extrapolation E The parameter E depends on the implementation of NFS and might be different for one variant to another. Let us take for example three computations with NFS which tackle various problems of the same bitsize:

- Danilov and Popovyan [13] factored a 180-digit RSA modulus using $\log_2 E \approx 30$ (although the size of the pairs (a, b) in theirs computations is not written explicitly, one can compute E using the range of special-q's and the default cardinality of the sieving space per special-q, which is 2^{30});
- Bouvier et al. [11] computed discrete logarithms in a 180-digit field $\mathbb{F}_p$ using $\log_2 E \approx 30$ (computed from other parameters).
- Barbulescu et al. [5] computed discrete logarithms in a 180-digit field $\mathbb{F}_{p^2}$ using $\log_2 E \approx 29$.

We see that in the first approximation E depends only on the bitsize of the field that we target and has the same value as in the factoring variant of NFS. Let us extrapolate E from the pair $(\log_2 Q = 600, \log_2 E = 30)$ using the formula

$$E = cL_Q(1/3, (8/9)^{1/3}).$$

Since exTNFS requires that $\gcd(\eta, \kappa) = 1$, the first case to study is $n = 6$.

The case of fields $\mathbb{F}_{p^6}$ When $n = 6$ we can use the general methods

- NFS-JLSV$_1$ (bitsize $E^{\frac{24}{t}} Q^{\frac{t-1}{6}}$, best values of t are 3 and 2)
- NFS-GJL with r equal to its optimal value, 6 (bitsize $E^{\frac{26}{t}} Q^{\frac{t-1}{7}}$, best values of t are 3 and 2)
- TNFS with $\deg f = 5$, its optimal value for this range of fields (bitsize $E^{\frac{12}{t}} Q^{\frac{t-1}{3}}$, best value of t is 2)

as well as the methods which exploit the fact that n is composite

- Sarkar-Singh (NFS-SS) with $\eta = 2$ and $r = 3$, best value so that $r \geq n/\eta$ for this range of fields, $(E^{\frac{28}{t}} Q^{\frac{t-1}{8}})$ respectively $\eta = 3$ and $r = 2$, best value so that $r \geq n/\eta$ for this range of fields, (bitsize $E^{\frac{30}{t}} Q^{\frac{t-1}{9}}$, best t are 4 and 3)
- exTNFS with $\eta = 2$ or $\eta = 3$ and one of two methods for selecting f and g
  - exTNFS-GJL with $\eta = 3$, $r = 2$ its best value so that $r \geq n/\eta$, (bitsize $E^{\frac{10}{t}} Q^{\frac{t-1}{3}}$, best value of t is 2)
  - exTNFS-GJL with $\eta = 2$, $r = 3$ its best value so that $r \geq n/\eta$, $(E^{\frac{14}{t}} Q^{\frac{t-1}{4}}$, best values of t are 3 and 2)
  - exTNFS-Conj with $\eta = 2$ (bitsize $E^{\frac{18}{t}} Q^{\frac{t-1}{6}}$, best values of t is 2).
  - exTNFS-Conj with $\eta = 3$ (bitsize $E^{\frac{12}{t}} Q^{\frac{t-1}{4}}$, best values of t are 3 and 2).

We plot the values of the norms product in Figure 2. Note that exTNFS with the Conjugation method seems to be the best choice for fields between 300 and 1000 bits.

For even more insight enter into details on a specific field.

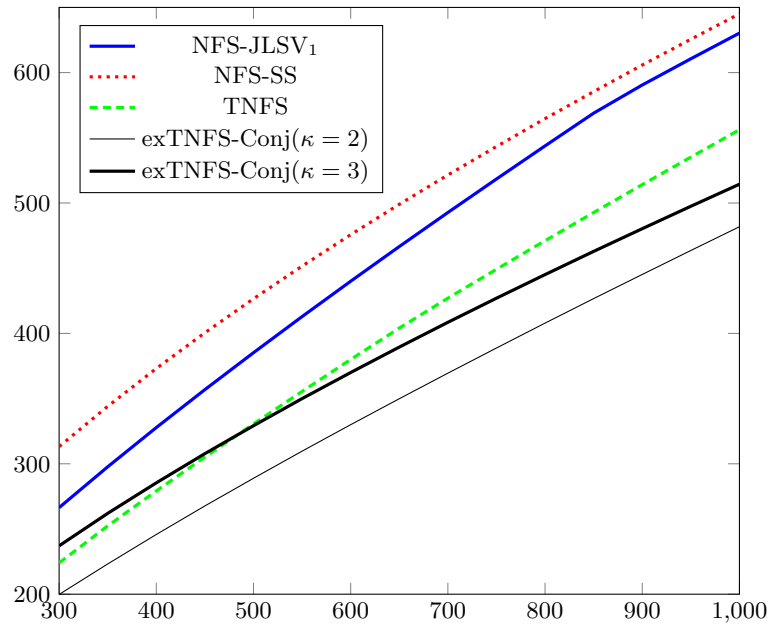

Fig. 2: Plot of the norms bitsize for several variants of NFS

Example 1: Let us consider the field $\mathbb{F}_{p^6}$ when

$$p = 31415926535897932384626433832 89.$$

The bitsize of $Q = p^6$ is 608 and its number of decimal digits is 182. Since the parameter E can only be chosen after an effective computation we are bound to make the hypothesis that it will have a similar value as in a series of record computations with NFS having the same input size:

In the following $\log_2 E = 30$. Let us make a list with the norm sizes obtained with each version of NFS:

1. NFS-JLSV$_1$. We take for example $f = x^6 - 1772453850905517$ and $g = 1772453850905515x^6 + 96769484157334$. The sieving space contains polynomials of degree two $r(x) = a + xb + cx^2$, i.e. $t = 3$, and the upper bound on the norms' product is

$$\text{norms bitsize(NFS-JLSV}_1) = 8 \log_2 E + \frac{1}{3} \log_2 Q \approx 440.$$

2. TNFS. We take $f = x^5 + 727139x^3 + 538962x^2 + 513716x + 691133$, $g = x - 1257274$ and $h = x^6 + x^4 + x + 1$. This time $t = 2$. Note that the parameter $d = \deg f$ is equal to 5, so that we have

$$\text{norms bitsize(TNFS)} = 6 \log_2 E + \frac{1}{3} \log_2 Q \approx 380.$$

3. exTNFS-Conj with $\eta = 2$ and $\kappa = 3$. We take $f = x^6 - 3$, $g = 309331385734750x^3 - 1851661516636217$ and $h = x^2 + 2$. Here $t = 2$. Hence we obtain

$$\text{norms bitsize(exTNFS } \eta = 2) = 9 \log_2 E + \frac{1}{6} \log_2 Q \approx 370.$$

4. exTNFS-Conj with $\eta = 3$ and $\kappa = 2$. We take $f = x^4 - 2x^3 + x^2 - 3$, $g = x^2 + 314159265358979323846264338358 8x + 26075443773076496496160262641 83$ and $h = x^3 + x + 1$. Again $t = 2$. This leads to

$$\text{norms bitsize(exTNFS } \kappa = 2) = 6 \log_2 E + \frac{1}{4} \log_2 Q \approx 330.$$

We conclude that in this example the best choice is exTNFS with $\kappa = 2$.

The condition $\gcd(\eta, \kappa) = 1$ restricts the values of n where exTNFS applies to $n = 6$, 10, 12, 14, 18, 20, 24 etc, but we do not discuss them in detail.

5.2 Precise comparison when p is SNFS

To compare precise norm sizes when p is a d-SNFS prime, let us consider Table 6.

Method	**condition**	**norms product**
STNFS	$E^{\frac{2(d+1)}{t}} Q^{\frac{t-1}{d}}$	
SNFS-JP	$E^{\frac{2n(d+1)}{t}} Q^{\frac{t-1}{nd}}$	
SexTNFS	$E^{\frac{2\kappa(d+1)}{t}} Q^{\frac{t-1}{\kappa d}}$	$n = \eta\kappa$ $\gcd(\kappa, \eta) = 1$ $2 \leq \eta < n$

Table 6: Comparison of norm sizes when p is d-SNFS prime.

Note that SexTNFS encompass SNFS-JP when $\eta = 1$, and STNFS when $\eta = n$, so we only call it SexTNFS when $2 \leq \eta < n$.

As in the case when p is arbitrary, we do not have precise estimations of E, especially in the large range of fields $\log_2 Q \in [1000, 10000]$. We are going to extrapolate from the pair $(\log_2 Q = 1039, \log_2 E = 30.38)$, due to the record of [1], using the formula

$$E = cL_Q(1/3, (4/9)^{\frac{1}{3}}).$$

Let us introduce a notation for the bitsize of SexTNFS, for any integers $\kappa \geq 1$ and $t \geq 2$:

$$C_{norm}(t, \kappa) = \frac{2\kappa(d+1)}{t} \log E + \frac{t-1}{\kappa d} \log Q.$$

For each κ, $C_{norm}(t, \kappa)$ has a minimum at the integer $t \geq 2$ which best approximates $\left(\frac{2\kappa^2 d(d+1) \log E}{\log Q}\right)^{1/2}$.

The case of 4*-SNFS primes* . To fix ideas, we restrict at the case $d = 4$. When $\kappa = 1$, i.e. STNFS, the norm size has its minimum at $t = 2$ as soon as $\frac{\log Q}{\log E} \geq 40/2^2 = 10$. In our range of interest ($300 \leq \log_2 Q \leq 10000$), the ratio $\log Q/\log E$ is always larger than 19. So, we only take care of sieving linear polynomials in the case of STNFS with $d = 4$. Similarly, it suffices to consider sieving linear polynomials in the case of SexTNFS with $\kappa = 2$ (resp. $\kappa = 3$) whenever $\log Q/\log E \geq 40$ (resp. $\log Q/\log E \geq 90$). It is satisfied when Q is of at least 1450 bits (resp. 6300 bits).

Let us compare the norm sizes of STNFS and SexTNFS when we sieve only linear polynomials ($t = 2$) in both cases. The value $C_{norm}(2, \kappa)$ has a minimum at $\kappa = \left(\frac{\log Q}{d(d+1)\log E}\right)^{1/2}$. In the case of $d = 4$, this value has minimum at $\kappa = 2$ or $\kappa = 3$ whenever $20 \leq \log Q/\log E \leq 180 = 20 \cdot 3^2$. Thus, in fields with large size, SexTNFS with $\kappa = 2$ or $\kappa = 3$ is better than STNFS.

In Figure 3 we plot the norm sizes of SNFS-JP, STNFS, and SexTNFS for $n = 12$ and $d = 4$ for Q is of from 300 bits to 5000 bits. We also compare these values with the best choice for general prime cases (exTNFS with Conjugation when $\kappa = 3$). From the plots we remark that STNFS could be a best choice for small Q otherwise SexTNFS with small κ becomes an important challenger against any other methods as the size of Q grows.

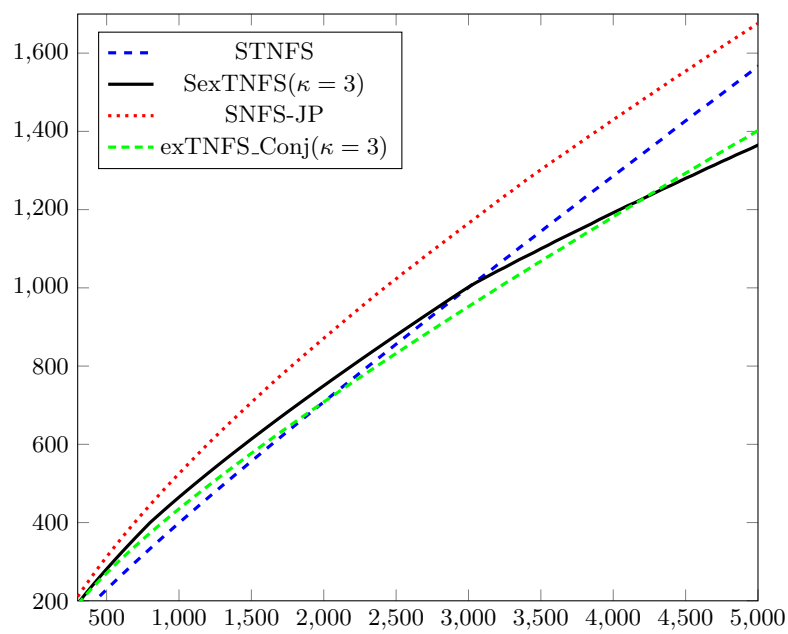

Fig. 3: Comparison when $n = 12$ and $d = 4$ for $300 \leq \log_2 Q \leq 5000$

To get a better intuition, let us see in detail a specific field.

Example 2: We consider the prime $p = P_4(u_4)$ where

$$P_4(x) = 36x^4 + 36x^3 + 24x^2 + 6x + 1 \text{ and } u_4 = 2^{158} - 2^{128} - 2^{68} + 1$$

(Section 6 in [2]), and note that p is 4-SNFS. The bitsize of p^{12} is 7647 for which we predict by extrapolation that $\log_2 E = 76.15$.

Let us make a list with the norm sizes obtained with each version of NFS:

1. STNFS. The size of the norms is $E^{2(d+1)/t}Q^{(t-1)/d}$ and has its minimum for $t = 2$. Take for example $h = x^{12} + x^{10} + x^9 - x^6 - 1$, $f = P_4$ and $g = x - u_4$.

$$\text{norms bitsize(STNFS)} = 5\log_2 E + \frac{1}{4}\log_2 Q \approx 2292.$$

2. SNFS-JP. The size of the norms is $E^{2n(d+1)/t}Q^{(t-1)/(nd)}$ and has its minimum when $t = 8$. Take for example $f = P_4(x^{12} + x^6 + x^3 + 1)$ and $g = (x^{12} + x^6 + x^3 + 1) - u_4$.

$$\text{norms bitsize(SNFS-JP)} = \frac{120}{7}\log_2 E + \frac{1}{8}\log_2 Q \approx 2257.$$

3. SexTNFS-JP $\eta = 4$. In this case the norm size is $E^{2\kappa(d+1)/t}Q^{\frac{(t-1)}{\kappa d}}$ and has its minimum when $t = 2$. Take for example $h = x^4 - x - 1$, $f = P_4(x^3 - x^2)$ and $g = x^3 - x^2 - u_4$.

$$\text{norms bitsize(SexTNFS)} = 15\log_2 E + \frac{1}{12}\log_2 Q \approx 1779.$$

One can do a similar analysis in the cases $d = 5$, $d = 6$ etc, but we do not present the details here.

6 Cryptologic consequences

The keysizes used in pairings-based cryptosystems are computed under the hypothesis that *DLP in* $\mathbb{F}_{p^n}$ *with* $2 \leq n \leq 12$ is at least as difficult as factoring an integer of the same size as p^n (see for example [15]). This hypothesis has been invalidated for $n = 2$ by the record computations presented in [5] where DLP in $GF(p^2)$ was 260 times faster than in $GF(p)$, and similar estimations were given for $\mathbb{F}_{p^3}$. The precise estimation in the same paper concluded however that the security of fields $\mathbb{F}_{p^6}$ was much less affected and there was nothing said about $\mathbb{F}_{p^{12}}$

Thanks to exTNFS we addressed the case of $\mathbb{F}_{p^6}$ and by a precise estimation concluded that it has norm sizes approximatively equal to those in the case of $\mathbb{F}_{p^2}$. This invalidates the keysizes which are currently used for $\mathbb{F}_{p^6}$. In order to extrapolate the new keysizes it is necessary to use the complexity $L_Q(1/3, \sqrt[3]{48/9})$ instead of the old $L_Q(1/3, \sqrt[3]{64/9})$ (a MNFS variant is also available of complexity

$L_Q(1/3, 1.71)$). The same is true when $n = \kappa\eta$ with $\kappa = 2$ or 3 and $\gcd(\kappa, \eta) = 1$, e.g. n is in the list

$$10, 12, 14, 18, 21, 22, 24.$$

When p is of special form, as in the Barreto-Naehrig construction, a precise estimations using STNFS and SexTNFS invalidated the current keysizes. In order to extrapolate the new keysizes it s necessary to use the new complexity $L_Q(1/3, \sqrt[3]{32/9})$ instead of the old $L_Q(1/3, \sqrt[3]{64/9})$.

A Non-linear polynomials

In all the variants of exTNFS that we have discussed, one puts linear polynomials $r(x) \in R[x]$ in the diagram of Figure 1. This is justified by the fact that exTNFS is a way of copying the setting from large characteristic to the medium prime case. Since in the large characteristic, the best choice is to take linear polynomials in all the variants, NFS, MNFS, SNFS, we have done the same thing in exTNFS, MexTNFS and SexTNFS.

The estimation of the norms sizes given in Lemma 1 is central in the analysis of exTNFS. For completion reasons we generalize this result to arbitrary degrees.

Lemma 2. *Let h be an irreducible polynomial over $\mathbb{Z}$ of degree η and f be an irreducible polynomial over $\mathbb{Z}[\iota]$ of degree κ. Let ι (resp. α) be a root of h (resp. f) in its number field and set $K_f := \mathbb{Q}(\iota, \alpha)$. Let $A > 0$ be a real number and T an integer such that $2 \leq T \leq \kappa$. For each $i = 0, \ldots, \kappa - 1$, let $a_i(t) \in \mathbb{Z}[t]$ be polynomials of degree $\leq \eta - 1$ with $\|a_i\|_\infty \leq A$. Then we have*

$$\left|N_{K_f/\mathbb{Q}}\Big(\sum_{i=0}^{T-1} a_i(\iota)\alpha^i\Big)\right| < A^{\eta\kappa}\|f\|_\infty^{(T-1)\eta}\|h\|_\infty^{(T+\kappa-1)(\eta-1)} D(\eta, \kappa),$$

where $D(\eta, \kappa) = \big((2\kappa-1)(\eta-1)+1\big)^{\eta/2}(\eta+1)^{(2\kappa-1)(\eta-1)/2}\big((2\kappa-1)!\eta^{2\kappa}\big)^{\eta}$. The above formula remains the same when we restrict the coefficients of f to be integers.

Proof. By abusing the notation, we write $f(t, x) := \sum_i f_i(t)x^i$ with $\deg_t(f_i) \leq \kappa - 1$ for $f(x) = \sum_i f_i(\iota)x^i \in \mathbb{Z}[\iota][x]$. Write $A(t, x) := \sum_i a_i(t)x^i$ and $r(t) := \mathrm{Res}_x\big(A(t, x), f(t, x)\big)$, then we have

$$N_{K_f/\mathbb{Q}(\iota)}(A(\iota, \alpha)) = r(\iota).$$

By Theorem 8 and Theorem 10 in [10], the degree of $r(t)$ is given by $(\kappa + T - 1)(\eta - 1)$ and

$$\|r(t)\|_\infty \leq (T + \kappa - 1)!\eta^{T+\kappa-2}A^\kappa\|f\|_\infty^{T-1}.$$

Then by Theorem 7 in the same article, we have

$$|N_{\mathbb{Q}(\iota)/\mathbb{Q}}(r(\iota))| \leq (\deg r + 1)^{\deg h/2}(\deg h + 1)^{\deg r/2}\|r\|_\infty^{\deg h}\|h\|_\infty^{\deg r}.$$

Combining all together, we obtain the desired result. □

This result allows to analyze MexTNFS-SS when $\kappa = \frac{1}{c_p}(\frac{\log Q}{\log\log Q})^3$ and $c_p < (\sqrt{78}/9 + 29/36)^{\frac{1}{3}} \approx 1.21$. Indeed, in this case one puts non-linear polynomials in the diagram, as indicated in Table 4 of [25].

Once again we check when $D(\eta, \kappa) = L_Q(2/3, o(1))$ and obtain the condition $\eta\kappa = o((\frac{\log Q}{\log\log Q})^{\frac{2}{3}})$. The factor $\|h\|_\infty^{(T+\kappa-1)(\eta-1)}$ is also negligible under the same condition. Hence the overhead is negligible for all range $\ell_p > 1/3$.

B Individual Logarithm

Let $s \in \mathbb{F}_{p^n}^* = \mathbb{F}_{p^{\eta\kappa}}^*$ be an element for which we want to compute the discrete logarithm. In general, the discrete logarithm of s can be found by following two steps: smoothing step and special-q descent.

In the smoothing step, the value s is randomized by $z := s^e$ for random value e and B_1-smoothness of z (for pre-determined value $B_1 > B$) is tested. Then, for each prime ideal $\mathfrak{D}$ which is not in the factor base, one finds a linear relation involving $\mathfrak{D}$ and other smaller ideals. This step is called special-q descent. We recursively produce the special-q descent tree, and finally deduce the desired discrete logarithm.

The complexity of the individual logarithm step differs by polynomial selection methods. In the following, to fix ideas, we consider only the JLSV$_2$ and Conjugation methods (exTNFS-JLSV$_2$ and exTNFS-Conj), but similar argument directly applies to any other polynomial selection method.

Smoothing. For each $z \in \mathbb{F}_{p^n}$ we compute an element $\bar{z} \in K_f = \mathbb{Q}(\iota, \alpha_f)$ which is sent to z when ι is mapped to a root of h in $\mathbb{F}_{p^\eta}$ and α_f in a root of f in $\mathbb{F}_{p^{\eta\kappa}}$. Then we test if $N_{K_f/\mathbb{Q}}(\bar{z})$ is B_1-smooth and squarefree. Let us discuss how to compute and what is the size of its norm.

JLSV$_2$ As before, we consider the target field $\mathbb{F}_{p^n}$ as an extension field $\mathbb{F}_{p^{\eta\kappa}} = \mathbb{F}_{p^\eta}(m) = \mathbb{F}_{p^\eta}[x]/k(x)$ over $\mathbb{F}_{p^\eta} = \mathbb{F}_p(\iota) = \mathbb{F}_p[t]/h(t)$. For a given z in $\mathbb{F}_{p^n}^*$, we write $z = \sum_i z_i(\iota)m^i$, where the coefficients of z_i are non-negative intergers bounded by p. We set

$$\bar{z} = \sum_{i=0}^{\kappa-1} z_i(\iota)\alpha_f^i$$

and, by Lemma 2 for $T = \kappa$, we obtain

$$|N_{K_f/\mathbb{Q}}(\bar{z})| \leq \left(p^n(p^{\kappa/(D+1)})^{n-\eta}\right)^{1+o(1)} \leq Q^{2-2/(\kappa+1)+o(1)},$$

where, in the last inequality, we used the condition that $D \geq \kappa$.

Conjugation In this case, a direct lift would make that $\bar{z}$ has degree κ instead of $2\kappa = \deg K_f$, and the coefficients $z_i(t)$ have norm bounded by p. In order to "spread" the coefficients, i.e. compute another polynomial with the same image in $\mathbb{F}_{p^n}$ of degree 2κ and coefficients of norm $p^{1/2}$, we need to use the LLL algorithm. With no extra cost we can obtain a further improvement: use the Waterloo improvement which consists in replacing the smoothness condition of integers of a given size X by the smoothness condition of two integers of size $X^{1/2}$.

The Waterloo improvement for exTNFS-Conj is as follows: we find two bivariate polynomials $u(t,x) = \sum_{i=0}^{2\kappa-1} u_i(t)x^i$ and $v(t,x) = \sum_{i=0}^{2\kappa-1} v_i(t)x^i \in \mathbb{Z}[t,x]$ such that z is the image in $\mathbb{F}_{p^n}$ of

$$\bar{z} := \frac{u(\iota, \alpha_f)}{v(\iota, \alpha_f)}$$

where $\|u_i\|_\infty, \|v_j\|_\infty \leq 2^n p^{1/4}$. For this we LLL-reduce the lattice of dimension $4n$ defined by the lines of the matrix

$$L = \left(\begin{array}{cccc|cccc}
p & & & & & & & \\
& \ddots & & & & & & \\
& & p & & & & & \\
\boxed{\quad \mathrm{vec}(k) \quad} & & & & & & & \\
& & & \ddots & & & & \\
& & & \boxed{\quad \mathrm{vec}(k) \quad} & & & & \\
\hline
\multicolumn{4}{c|}{\mathrm{vec}(z \bmod (h,f))} & 1 & & & \\
\multicolumn{4}{c|}{\vdots} & & \ddots & & \\
\multicolumn{4}{c|}{\mathrm{vec}(t^i x^j z \bmod (h,f))} & & & \ddots & \\
\multicolumn{4}{c|}{\vdots} & & & & \ddots \\
\multicolumn{4}{c|}{\mathrm{vec}(t^{\eta-1} x^{2\kappa-1} z \bmod (h,f))} & & & & 1
\end{array}\right)$$

the first n rows contain only the diagonal coefficient equal to p and where, for all bivariate polynomial $w(t,x) = \sum_{i=0}^{2\kappa-1} w_i(t)x^i$ with $w_i(t) = \sum_{j=0}^{\eta-1} w_{i,j} t^{\eta-1-j}$, $\mathrm{vec}(w) = (w_{0,0}, \ldots, w_{0,\eta-1}, \ldots, w_{2\kappa-1,0}, \ldots, w_{2\kappa-1,\eta-1})$ of dimension $2n$. In particular, $k \in \mathbb{F}_{p^n}[x]$ has been seen as a two-variate polynomial.

By dividing if necessary by the leading coefficient, we can assume that $k(x)$ is monic, hence the right-most coordinate of $\mathrm{vec}(k)$ is 1. Then $\det L = p^n$ and we have u, v with $\|u_i\|_\infty, \|v_j\|_\infty \leq 2^{(4n-1)/4} Q^{\frac{1}{4n}} \leq 2^n Q^{\frac{1}{4n}}$. By Lemma 2 we obtain that

$$|N_{K_f/\mathbb{Q}}(u(\iota,\alpha_f)) N_{K_f/\mathbb{Q}}(v(\iota,\alpha_f))| \leq 2^{n^2} Q \left(\|f\|_\infty^{(2\kappa-1)\eta} \|h\|_\infty^{(3\kappa-1)} (\eta-1) D(\eta, 2\kappa) \right)^2.$$

The term in the later bracket is $L_Q(2/3, o(1))$ and 2^{n^2} is negligible compared to Q if and only if $\ell_p > 1/2$. We conclude that when $\ell_p > 1/2$

$$|N_{K_f/\mathbb{Q}}(u(\iota,\alpha_f)) N_{K_f/\mathbb{Q}}(v(\iota,\alpha_f))| = Q^{1+o(1)}.$$

Once the lift $\bar{z}$ has been computed, the smoothing step is carried out as usual: one tests that the norm of $\bar{z}$ (or u and v) is squarefree and B_1-smooth where $B_1 = L_Q(2/3, \beta_1)$ for some constant $\beta_1 > 0$. We recognize the complexity analysis done in [12] in the case of prime fields: the complexity of the smoothing step is $L_Q(1/3, c_{\text{smooth}})$ with

- $c_{\text{smooth}} = 6^{\frac{1}{3}}$ for exTNFS-JLSV$_2$;
- $c_{\text{smooth}} = 3^{\frac{1}{3}}$ for exTNFS-Conj.

Descent by special-*q* Recall how the special-q descent is done in the large characteristic case of NFS (for example NFS-JLSV$_2$). Due to the condition that $N_{K_f/\mathbb{Q}}(\bar{z})$ is squarefree the ideal generated by $\bar{z}$ factors only into prime ideals of degree 1. For a prime ideal $\mathfrak{q}$ of degree 1 in K_f that appears in the factorization of the principal ideal $(\bar{z})$, we write the logarithm of $\mathfrak{q}$ as a formal sum of virtual logarithms of ideals in K_f and K_g of norm less than $N(\mathfrak{q})^c$ for a constant $c < 1$. For this, we enumerate pairs $(a, b) \in \mathbb{Z} \times \mathbb{Z}$ such that $\mathfrak{q}$ divides $(a - b\alpha_f)$ to find one pair such that

- $(a - b\alpha_f)/\mathfrak{q}$ factors into prime ideals of norm less than $N(\mathfrak{q})^c$, and
- the ideal $(a - b\alpha_g)$ factors into prime ideals of norm less than $N(\mathfrak{q})^c$.

To do this we find two pairs $(a^{(1)}, b^{(1)})$ and $(a^{(2)}, b^{(2)})$ of euclidean norm less than a constant times $N(\mathfrak{q})^{\frac{1}{2}}$, using LLL. Then we enumerate the pairs $i_1 + i_2$ for all rational integers with $|i_1|, |i_2| \leq E'$. The complexity of the descent is mainly determined by the size of the norms:

$$|N_{K_f/\mathbb{Q}}(a - b\alpha_f)| \leq \left((E')^{\kappa} N(\mathfrak{D})^{\kappa/2} Q^{1/(D+1)}\right)^{1+o(1)},$$

$$|N_{K_g/\mathbb{Q}}(a - b\alpha_g)| \leq \left((E')^{D} N(\mathfrak{D})^{D/2} Q^{1/(D+1)}\right)^{1+o(1)}.$$

In our two cases, exTNFS-JLSV$_2$ and exTNFS-Conj, we enumerate $a(\iota), b(\iota) \in R \subset \mathbb{Q}(\iota)$ where $a(t), b(t) \in \mathbb{Z}[x]$ of degree $\leq \eta - 1$ and $\|a\|_\infty, \|b\|_\infty \leq (E')^{\frac{1}{\eta}}$ so that $a(\iota) - b(\iota)\alpha_f \equiv 0 \bmod \mathfrak{q}$. This can be done in the following manner (cf Appendix 7.1 in [7]). First, we construct the lattice

$$L(\mathfrak{q}) := \{(a, b) = (a_0, \ldots, a_{\eta-1}, b_0, \ldots, b_{\eta-1}) \in \mathbb{Z}^{2\eta} : a(\iota) - b(\iota)\alpha_f \equiv 0 \bmod \mathfrak{q}\},$$

which has determinant $N(\mathfrak{q})$. Let $(a^{(k)}, b^{(k)})$, $k = 1, 2, \ldots, 2\eta$, be the LLL-reduced basis of this lattice. Then we test the above smoothness conditions for pairs $(a, b) = \sum_{k=1}^{2\eta} i_k (a^{(k)}, b^{(k)})$, where i_k are rational integers with absolute value less than $I := (E')^{\frac{1}{\eta}}$. By Lemma 1, in the case of exTNFS-JLSV$_2$ the size of the norms is

$$|N_{K_f/\mathbb{Q}}(a - b\alpha_f)| \leq \left((E')^{\kappa} N(\mathfrak{q})^{\kappa/2} Q^{1/(D+1)}\right)^{1+o(1)},$$

$$|N_{K_g/\mathbb{Q}}(a - b\alpha_g)| \leq \left((E')^{D} N(\mathfrak{q})^{D/2} Q^{1/(D+1)}\right)^{1+o(1)}.$$

Then, the rest of the analysis is similar to that of Chapter 7.3. in [3] and we conclude that in exTNFS-JLSV$_2$ the special-q descent is negligible compared to the smoothing step.

In the case of exTNFS-Conj, we use again Lemma 1 and obtain:

$$|N_{K_f/\mathbb{Q}}(a - b\alpha_f)| \leq \left((E')^{2\kappa} N(\mathfrak{q})^{\kappa}\right)^{1+o(1)},$$

$$|N_{K_g/\mathbb{Q}}(a - b\alpha_g)| \leq \left((E')^{\kappa} N(\mathfrak{q})^{\kappa/2} Q^{1/(2\kappa)}\right)^{1+o(1)}.$$

We make an usual heuristic argument that a number x is y-smooth with the probability of $\rho(\log x/\log y)$ for Dickman function ρ. So, the probability of the pair (a, b) to be descended is given by

$$\text{Prob}[(a,b) \text{ descends}] \geq \rho\left(\frac{3\kappa \log E' + (3\kappa/2)\log \nu + (1/(2\kappa))\log Q}{c \log \nu}\right)^{1+o(1)}, \tag{10}$$

where $\nu := N(\mathfrak{q})$.

In the case when ν is large, i.e. $\nu = L_Q(2/3, \beta_1)$, where β_1 is imposed by the smoothing step described above, the inverse of the probability can be approximated by

$$\rho\left(\frac{3\kappa}{2c}\right)^{-1+o(1)} = L_Q\left(\frac{1}{3}, \frac{c_\kappa}{2c}\right)^{1+o(1)},$$

where $c_\kappa = \kappa/(\frac{\log Q}{\log\log Q})^{\frac{1}{3}} = 12^{-\frac{1}{3}}$. Multiplying this by the time for ν^c-smoothness test the total cost becomes

$$L_Q\left(1/3, \frac{c_\kappa}{2c} + 2\sqrt{\frac{c\beta_1}{3}}\right)^{1+o(1)}.$$

This value is minimized by $L_Q(1/3, (9\beta_1 c_\kappa/2)^{1/3})$ when $c = \left(\frac{3c_\kappa^2}{4\beta_1}\right)^{1/3}$. When we use that $\beta_1 = (1/3)^{1/3}$ and $c_\kappa = 12^{-\frac{1}{3}}$, we deduce the complexity

$$L_Q\left(1/3, (81/32)^{\frac{1}{9}}\right)$$

that is less than the complexity of the smoothing step.

In the case of small ν, i.e. $\nu = L_Q(1/3)$, the hardest descent step corresponds to the case when $\nu^c = B$ (the smoothness bound for the factor base). In this case, again by Equation (10), we have the probability of the descent,

$$L_Q\left(1/3, \frac{c_\kappa}{2c} + \frac{c_\kappa \epsilon}{\beta} + \frac{1}{6c_\kappa \beta}\right)^{-1+o(1)}.$$

The complexity is minimized when the size of sieving space equals to the inverse of the above probability. This translates to

$$2\epsilon = \frac{c_\kappa}{2c} + \frac{c_\kappa \epsilon}{\beta} + \frac{1}{6c_\kappa \beta}.$$

This shows that the optimal value for c can be any value close but not equal to 1, e.g. $c = 0.999$, and the optimal complexity of descent step for small ν is $L_Q(1/3, 2\epsilon)$ where

$$\epsilon = \left(\frac{c_\kappa}{2} + \frac{1}{6\beta c_\kappa}\right) / \left(2 - \frac{c_\kappa}{\beta}\right) = 12^{-1/3} \approx 0.44,$$

where we used $\beta = (2/3)^{1/3}$ and $c_\kappa = 12^{-1/3}$. This complexity is negligible to the smoothing step.

algorithm	rels collection +lin. algebra	smoothing	special-q descent	extra conditions
exTNFS-JLSV$_2$	$(64/9)^{\frac{1}{3}}$	$(54/9)^{\frac{1}{3}}$	negligible	-
exTNFS-Conj	$(48/9)^{\frac{1}{3}}$	$(27/9)^{\frac{1}{3}}$	negligible	$\ell_p > 1/2$

Table 7: Complexity of individual logarithm

For medium ν, i.e. $\nu = L_Q(\ell)$ with $1/3 < \ell < 2/3$, it is obviously faster than the case of large ν. So, we omit detailed analysis for this case and refer to Chapter 7.3. in [3].

We conclude this section of the Appendix with a summary of our results in Table 7.

References

1. K. Aoki, J. Franke, T. Kleinjung, A. K. Lenstra, and D. A. Osvik. A kilobit special number field sieve factorization. In *Advances in Cryptology – ASIACRYPT 2007*, volume 4833 of *Lecture Notes in Comput. Sci.*, pages 1–12. Springer, 2007.
2. D. F. Aranha, L. Fuentes-Castaneda, E. Knapp, A. Menezes, and F. Rodrıguez-Henrıquez. Implementing pairings at the 192-bit security level. *Pairing-Based Cryptography–Pairing 2012*, 7708:177, 2012.
3. R. Barbulescu. *Algorithms of discrete logarithm in finite fields.* Theses, Université de Lorraine, Dec. 2013.
4. R. Barbulescu. An appendix for a recent paper of kim. *IACR Cryptology ePrint Archive*, 2015:1076, 2015.
5. R. Barbulescu, P. Gaudry, A. Guillevic, and F. Morain. Improving NFS for the discrete logarithm problem in non-prime finite fields. In Oswald and Fischlin [23], pages 129–155.
6. R. Barbulescu, P. Gaudry, A. Joux, and E. Thomé. A heuristic quasi-polynomial algorithm for discrete logarithm in finite fields of small characteristic. In *Advances in Cryptology - EUROCRYPT 2014 - 33rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Copenhagen, Denmark, May 11-15, 2014. Proceedings*, pages 1–16, 2014.

7. R. Barbulescu, P. Gaudry, and T. Kleinjung. The Towed Number Field Sieve. In T. Iwata and J. H. Cheon, editors, *ASIACRYPT 2015, to appear*, LNCS. Springer, 2015.
8. R. Barbulescu and C. Pierrot. The multiple number field sieve for medium- and high-characteristic finite fields. *LMS Journal of Computation and Mathematics*, 17:230–246, 2014. The published version contains an error which is corrected in version 2 available at https://hal.inria.fr/hal-00952610.
9. P. S. L. M. Barreto and M. Naehrig. Pairing-friendly elliptic curves of prime order. In *Selected Areas in Cryptography, 12th International Workshop, SAC 2005, Kingston, ON, Canada, August 11-12, 2005, Revised Selected Papers*, pages 319–331, 2005.
10. Y. Bistritz and A. Lifshitz. Bounds for resultants of univariate and bivariate polynomials. *Linear Algebra and its Applications*, 432(8):1995 – 2005, 2010. Special issue devoted to the 15th {ILAS} Conference at Cancun, Mexico, June 16-20, 2008.
11. C. Bouvier, P. Gaudry, L. Imbert, H. Jeljeli, and E. Thom. Discrete logarithms in GF(p) — 180 digits, 2014. Announcement available at the NMBRTHRY archives, item 004703.
12. A. Commeine and I. Semaev. An algorithm to solve the discrete logarithm problem with the number field sieve. In *Public Key Cryptology–PKC 2006*, volume 3958 of *Lecture Notes in Comput. Sci.*, pages 174–190. Springer, 2006.
13. S. Danilov and I. Popovyan. Factorization of rsa-180, 2010. http://eprint.iacr.org/2010/270.
14. D. Freeman. Constructing pairing-friendly elliptic curves with embedding degree 10. In *Algorithmic Number Theory, 7th International Symposium, ANTS-VII, Berlin, Germany, July 23-28, 2006, Proceedings*, pages 452–465, 2006.
15. D. Freeman, M. Scott, and E. Teske. A taxonomy of pairing-friendly elliptic curves. 23(2):224–280, 2010.
16. D. M. Gordon. Discrete logarithms in gf(p) using the number field sieve. *SIAM J. Discret. Math.*, 6(1):124–138, Feb. 1993.
17. R. Granger, T. Kleinjung, and J. Zumbrägel. On the powers of 2. Cryptology ePrint Archive, Report 2014/300, 2014. http://eprint.iacr.org/.
18. A. Joux, R. Lercier, N. P. Smart, and F. Vercauteren. The number field sieve in the medium prime case. In C. Dwork, editor, *Advances in Cryptology - CRYPTO 2006, 26th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 2006, Proceedings*, volume 4117 of *Lecture Notes in Computer Science*, pages 326–344. Springer, 2006.
19. A. Joux and C. Pierrot. The special number field sieve in $\mho_{p^n}$ - application to pairing-friendly constructions. In *Pairing-Based Cryptography - Pairing 2013 - 6th International Conference, Beijing, China, November 22-24, 2013, Revised Selected Papers*, pages 45–61, 2013.
20. T. Kim. Extended tower number field sieve: A new complexity for medium prime case. *IACR Cryptology ePrint Archive*, 2015:1027, 2015.
21. H. W. Lenstra Jr. Factoring integers with elliptic curves. *Annals of Mathematics*, pages 649–673, 1987.
22. D. V. Matyukhin. Effective version of the number field sieve for discrete logarithm in a field GF(pˆk). *Trudy po Diskretnoi Matematike*, 9:121–151, 2006.
23. E. Oswald and M. Fischlin, editors. *Advances in Cryptology - EUROCRYPT 2015 - 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part I*, volume 9056 of *Lecture Notes in Computer Science*. Springer, 2015.

24. C. Pierrot. The multiple number field sieve with conjugation and generalized joux-lercier methods. In Oswald and Fischlin [23], pages 156–170.
25. P. Sarkar and S. Singh. New complexity trade-offs for the (multiple) number field sieve algorithm in non-prime fields. Cryptology ePrint Archive, Report 2015/944, 2015. http://eprint.iacr.org/.
26. O. Schirokauer. Discrete logarithms and local units. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 345(1676):409–423, 1993.
27. O. Schirokauer. Using number fields to compute logarithms in finite fields. *Math. Comput.*, 69(231):1267–1283, 2000.
28. D. Wiedemann. Solving sparse linear equations over finite fields. *IEEE Trans. Inform. Theory*, 32(1):54–62, 1986.