The Tower Number Field Sieve

Razvan Barbulescu[1], Pierrick Gaudry[2], and Thorsten Kleinjung[3]

[1] CNRS and IMJ-PRG(UPMC/CNRS), France,
[2] CNRS and Loria(Inria/Univ. Lorraine/CNRS), France
[3] EPFL IC LACAL, Station 14, CH-1015 Lausanne, Switzerland

razvan.barbaud@imj-prg.fr, pierrick.gaudry@loria.fr, thorsten.kleinjung@epfl.ch

Abstract. The security of pairing-based crypto-systems relies on the difficulty to compute discrete logarithms in finite fields $\mathbb{F}_{p^n}$ where n is a small integer larger than 1. The state-of-art algorithm is the number field sieve (NFS) together with its many variants. When p has a special form (SNFS), as in many pairings constructions, NFS has a faster variant due to Joux and Pierrot. We present a new NFS variant for SNFS computations, which is better for some cryptographically relevant cases, according to a precise comparison of norm sizes. The new algorithm is an adaptation of Schirokauer's variant of NFS based on tower extensions, for which we give a middlebrow presentation.

Keywords: discrete logarithm, number field sieve, pairings.

1 Introduction

The discrete logarithm problem (DLP) in finite fields is a central topic in public key cryptography. The case of $\mathbb{F}_{p^n}$ where p is prime and n is a small integer greater than 1, albeit less studied than the prime case, is at the foundation of pairing-based cryptography.

The number field sieve (NFS) started life as a factoring algorithm but was rapidly extended to compute discrete logarithms in $\mathbb{F}_p$ [33,19,20] and has today a large number of variants. In 2000 Schirokauer [34] proposed the tower number field sieve (TNFS), as the first variant of NFS to solve DLP in fields $\mathbb{F}_{p^n}$ with $n > 1$. When n is fixed and the field cardinality $Q = p^n$ tends to infinity, he showed that TNFS has the heuristic complexity $L_Q(1/3, \sqrt[3]{64/9})$, where

$$L_Q(\alpha, c) = \exp\Big((c + o(1))(\log Q)^{\alpha}(\log\log Q)^{1-\alpha}\Big).$$

Schirokauer explicitly suggested that his algorithm might be extended to arbitrary fields $\mathbb{F}_{p^n}$ with $p = L_{p^n}(\alpha, c)$ and $\alpha > 2/3$, while maintaining the same complexity. Another question that he raised was whether his algorithm could take advantage of a situation where the prime p has a special SNFS shape, namely if it can be written $p = P(u)$ for an integer $u \approx N^{1/d}$ and a polynomial $P \in \mathbb{Z}[x]$ of degree d, with coefficients bounded by an absolute constant. By that time, even for prime fields the answer was not obvious.

In 2006 Joux, Lercier, Smart and Vercauteren [21] presented a new variant of NFS which applies to all finite fields $\mathbb{F}_{p^n}$ with $p = L_Q(\alpha, c)$ for some $\alpha \geq 1/3$ and $c > 0$, the JLSV algorithm. When $\alpha > 2/3$, their variant has complexity $L_Q(1/3, \sqrt[3]{64/9})$. The question of extending TNFS to arbitrary finite fields became obsolete, because, in case of a positive answer, it would have the same complexity as the JLSV algorithm.

In 2013 Joux and Pierrot designed another variant of NFS which applies to non-prime fields $\mathbb{F}_{p^n}$ where p is an SNFS prime. Their algorithm has complexity $L_Q(1/3, \sqrt[3]{32/9})$, which is the same as that of Semaev's SNFS algorithm for prime fields [35]. It shows that the pairing-based crypto-systems which use primes of a special form are more vulnerable to NFS attacks than the general ones. With this SNFS algorithm, the second question of Schirokauer lost its appeal as well, because this is the complexity that one can expect if Schirokauer's algorithm can be adapted when p is SNFS.

In 2014 Barbulescu, Gaudry, Guillevic and Morain improved the algorithm in [21] and set a record computation in a field $\mathbb{F}_{p^2}$ of 180 decimal digits. However, since their improvements do not apply to SNFS fields and since the algorithm of Joux and Pierrot was never implemented, it is important to find a practical algorithm for this case.

In this work, we wish to rehabilitate Schirokauer's TNFS algorithm. First, we show that indeed, the heuristic complexity carries over to the expected range of finite fields. In order to make this analysis, we restate the original TNFS with less technicalities than in the original presentation, taking advantage of tools that were invented later, like Schirokauer maps and virtual logarithms.

We also show that for extension fields based on SNFS primes, the complexity of TNFS drops as expected to $L_Q(1/3, \sqrt[3]{32/9})$.

Finally, going beyond the asymptotic formulae, we compute estimates that strongly suggest that TNFS is currently the most efficient algorithm for solving discrete logarithms in small degree extensions of SNFS prime fields, like the ones arising naturally in several pairing constructions.

Outline. After a brief description of Schirokauer's TNFS algorithm in Section 2, we present it with sufficiently many details to get a proper asymptotic analysis in Section 3. In Section 4, several variants are described and analyzed, in particular the SNFS variant. This is followed, in Section 5 by more precise estimates for cryptographically relevant sizes and comparisons with other methods. Further technicalities about TNFS are given in an appendix; these are mostly details that could be useful for an implementation but which do not change the complexities.

2 Overview of TNFS

To fix ideas, we consider the case of "large" characteristic, so that we target fields $\mathbb{F}_Q$ with $Q = p^n$ so that $p = L_Q(\alpha, c)$ for some constants $\alpha > 2/3$ and $c > 0$.

Pohlig and Hellman explained how to retrieve the discrete logarithm modulo the group order N from the value of the discrete logarithms modulo each prime factor ℓ of N. Furthermore, Pollard's rho algorithm allows to compute discrete logarithms for small primes. Hence it is enough to explain how to use NFS to compute discrete logarithms modulo prime factors ℓ of $\#\mathbb{F}_{p^n}^*$ larger that $L_{p^n}(1/3, c)$ for some $c > 0$.

A classical NFS algorithm would involve two irreducible polynomials f and g in $\mathbb{Z}[x]$ which have a common irreducible factor of degree n modulo p. Here, in TNFS, we consider two irreducible polynomials f and g defined over a ring R which is of the form $R = \mathbb{Z}[t]/h(t)$ for a monic irreducible polynomial h of degree n. We ask furthermore that h remains irreducible modulo p, so that there is a unique ideal $\mathfrak{p}$ above p in R. Finally, we require that f and g have a common root modulo $\mathfrak{p}$ in R.

In the rest of the article, we denote by K_f the number field K_f defined by f, and by K_g the one defined by g. Also we write $\mathbb{Q}(\iota)$ for the number field defined by h, so that K_f and K_g are as in the figure aside.

$$\begin{tikzcd}
K_f & & K_g \\
& \mathbb{Q}(\iota) \arrow[ul, dash, "f"] \arrow[ur, dash, "g"'] & \\
& \mathbb{Q} \arrow[u, dash, "h"'] &
\end{tikzcd}$$

The conditions imposed on f, g and h are such that there exist two ring homomorphisms from $R[x]$ to $R/\mathfrak{p} = \mathbb{F}_{p^n}$, one going through $R[x]/f(x)$, and the other through $R[x]/g(x)$, and for any polynomial in $R[x]$, the resulting values in $\mathbb{F}_{p^n}$ coincide, so that we get a commutative diagram as in the classical NFS algorithm (see Figure 1).

$$\begin{tikzcd}
& R[x] \arrow[dl] \arrow[dr] & \\
K_f \supset R[x]/\langle f(x)\rangle \arrow[dr, "\substack{\alpha_f \mapsto m \\ R \mapsto R \bmod p}"'] & & R[x]/\langle g(x)\rangle \subset K_g \arrow[dl, "\substack{\alpha_g \mapsto m \\ R \mapsto R \bmod p}"] \\
& R/pR = \mathbb{F}_{p^n} &
\end{tikzcd}$$

Fig. 1. Commutative diagram of TNFS for discrete logartihm in $\mathbb{F}_{p^n}$. In the classical case, $R = \mathbb{Z}$; here $R = \mathbb{Z}[\iota]$ is a subring of a number field of degree n where p is inert.

In our work, we did not find any gain by taking the coefficients of f or g as coefficients in R itself: in the constructions below, we will always consider polynomials with coefficients in $\mathbb{Z}$. Therefore, K_f and K_g can also be seen as compositum of two fields. However, it is still interesting to consider f and g as polynomials in $R[x]$, since this makes it easier to follow the analogy with the classical NFS.

Once this setting is done, the TNFS algorithm proceeds as usual. For many polynomials $a(\iota) - b(\iota)x$ in $R[x]$, we consider their two images in $R[x]/f(x)$ and $R[x]/g(x)$, and test them for smoothness as ideals. Each time the images are simultaneously smooth, we can write a relation: modulo the usual complications with principality defects and units that can be handled with the help of Schirokauer maps, it is possible to convert a relation into a linear relation between virtual logarithms of the factor base elements. Then follows a sparse linear algebra step to deduce the values of these virtual logarithms. And finally, the logarithm of an individual element of $\mathbb{F}_{p^n}$ can be computed using a descent step.

In the next section, we will enter into details, define more precisely the factor base elements and the associated smoothness notion, and estimate the size of the objects involved in the computation.

3 Detailed description and analysis

3.1 Polynomial selection

In the overview of the previous section, nothing is said about the respective degrees of f and g. In fact, there is some freedom here, and we could in principle have balanced degrees and use for instance the algorithm of [20] or we can use a linear polynomial g, both methods leading to the same asymptotic complexity. The only difference comes in the individual logarithm stage. In order to give the exposition short, we will only present this stage in the case where g is linear, but in practice one must take the one which minimizes the overall time.

To fix ideas, we take a linear polynomial g and a polynomial f with a degree of the form

$$\deg f = d = \delta\ (\log Q / \log\log Q)^{1/3},$$

where the constant δ is to be fixed later, so that f and g have a common root modulo p. They can be obtained by a simple base-m algorithm, yielding coefficients for f and g of size

$$\|f\|_\infty \approx \|g\|_\infty \approx p^{1/(d+1)}.$$

In practice, instead of a naïve base-m approach, one can use any of the methods known for the polynomial selection of NFS, when tackling prime fields or integer factorization [13,23,24,3,4].

What is left is to select a polynomial h of degree n with small coefficients which is irreducible modulo p. This is done by testing polynomials with small coefficients and, heuristically, we succeed after n trials, on average, because the proportion of irreducible polynomials modulo p is $\approx 1/n$. As we will explain later, rather than having the polynomial h with the smallest coefficients, we might prefer some polynomial with slightly larger coefficients but with the additional property that the Galois group of h is cyclic of order n. For this, we test polynomials in families with a cyclic Galois group; for example Foster [17] gives a list of such families when $\deg h = 2, 3, 4, 5$ or 6.

If one is interested in rigorous results and not in the most efficient polynomials, then one can give a proof of existence based on Corollary 10 given in the Appendix. Indeed, by testing all the degree n polynomials h with coefficients upper bounded by $(An^B \log(pn)^C)^n$ for some effective constants A, B and C, one is guaranteed to find one which is irreducible modulo p in a time that is polynomial in $\log Q$.

3.2 Relation collection

In the top of the diagram of Figure 1 one usualy takes $a - bx$ with $a, b \in R[x]$. However, in its most general version NFS one considers polynomials in $R[x]$ of arbitrary degrees; this is in particular necessary for the medium characteristic case [21]. In our study, we did not find any case where it was advantageous to consider polynomials of degree more than 1. Therefore we stick to the traditional (a, b)-pairs terminology for designating a linear polynomial $a(\iota) - b(\iota)x$ in $R[x]$ that we consider as a candidate for producing a relation.

Ideals of degree 1. In our case, just like in the classical NFS, only ideals of degree 1 can occur in the factorizations of the elements in the number rings (except maybe for a finite number of ideals dividing the discriminants). This is, of course only true when thinking in the relative extensions; we formalize this in the following proposition that holds for f, but is also true for g if it happens to be non-linear.

Proposition 1 *Let $\mathbb{Q}(\iota)$ be a number field and let $\mathcal{O}_\iota$ be its ring of integers. Let f be a monic irreducible polynomial in $\mathbb{Q}(\iota)[x]$, and denote by α one of its roots. We denote by $K_f = \mathbb{Q}(\iota, \alpha)$ the corresponding extension field, and $\mathcal{O}_f$ its ring of integers.*

If $\mathfrak{q}$ is a prime ideal of $\mathcal{O}_\iota$ not dividing the index-ideal $[\mathcal{O}_f : \mathcal{O}_\iota[\alpha]]$, then the following statements hold.

i) The prime ideals of $\mathcal{O}_f$ above $\mathfrak{q}$ are all the ideals of the form

$$\mathfrak{Q} = \langle \mathfrak{q}, T(\alpha) \rangle,$$

where $T(x)$ are the lifts to $\mathcal{O}_\iota[x]$ of the irreducible factors of f in $\mathcal{O}_\iota/\mathfrak{q}[x]$. Moreover $\deg \mathfrak{Q} = \deg T$.

ii) If $a(t)$, $b(t) \in \mathbb{Z}[t]$ are such that $\mathfrak{q}$ divides $\mathrm{N}_{K_f/\mathbb{Q}(\iota)}(a(\iota) - b(\iota)\alpha)$ and $a(\iota)\mathcal{O}_\iota + b(\iota)\mathcal{O}_\iota = \mathcal{O}_\iota$, then the unique ideal of $\mathcal{O}_f$ above $\mathfrak{q}$ which divides $a(\iota) - b(\iota)\alpha$ is $\mathfrak{Q} = \langle \mathfrak{q}, \alpha - r(\iota) \rangle$ with $r \equiv a(\iota)/b(\iota) \pmod{\mathfrak{q}}$.

Proof. i) This is Proposition 2.3.9 of [14].

ii) Let $\mathfrak{Q} = \langle \mathfrak{q}, T(\alpha) \rangle$ be a prime ideal of K above $\mathfrak{q}$ that divides $a(\iota) - b(\iota)\alpha$. If $\mathfrak{Q}$ divides $b(\iota)$ then it also divides $a(\iota)$, and therefore we have a contradiction with the condition $a(\iota)\mathcal{O}_\iota + b(\iota)\mathcal{O}_\iota = \mathcal{O}_\iota$. Therefore we can simplify $\mathrm{val}_{\mathfrak{Q}}(a(\iota) - b(\iota)\alpha)$ by dividing out by $b(\iota)$:

$$\mathrm{val}_{\mathfrak{Q}}(a(\iota) - b(\iota)\alpha) = \mathrm{val}_{\mathfrak{Q}}(b(\iota)) + \mathrm{val}_{\mathfrak{Q}}(a(\iota)/b(\iota) - \alpha) = \mathrm{val}_{\mathfrak{Q}}(\alpha - r(\iota)).$$

This expression is non-zero only when $\mathfrak{Q} = \langle \mathfrak{q}, \alpha - r(\iota)\rangle$, which proves the statement.

Note that the coprimeness condition is similar to the one we have in the classical case, and the proportion of coprime pairs is

$$\prod_{\mathfrak{q} \text{ prime ideal in } \mathbb{Q}(\iota)} \left(1 - \frac{1}{\mathrm{N}(\mathfrak{q})^2}\right) = \frac{1}{\zeta_{\mathbb{Q}(\iota)}(2)},$$

replacing $1/\zeta_{\mathbb{Q}}(2)$ in the classical variant.

Factor base. The consequence of this result is that we keep only the degree 1 ideals in the factor bases for each side. With the same notations as above, and for a smoothness bound B, we define the factor base for f by

$$\mathcal{F}_f(B) = \left\{ \begin{array}{c} \text{prime ideals of } \mathcal{O}_f, \text{ coprime to } \mathrm{Disc}(K_f), \text{ of norm less than } B, \\ \text{whose inertia degree over } \mathbb{Q}(\iota) \text{ is one} \end{array} \right\}.$$

We define $\mathcal{F}_g(B)$ similarly; if g is linear this is just the set of prime ideals of $\mathcal{O}_\iota \cong \mathcal{O}_g$ of norm less than B. Prime ideals that divide the ideal-index $[\mathcal{O}_f : \mathcal{O}_\iota[\alpha]]$ are not covered by Proposition 1, and can still occur in the factorization of $(a(\iota) - b(\iota)\alpha)$. Moreover, since the index-ideal cannot be computed effectively, we consider together all the ideals above $\mathrm{Disc}(f)$ and above the leading coefficient of f. We denote them by $\mathcal{D}_f$ on the f-side, and $\mathcal{D}_g$ on the g-side. The cardinalities of these sets are bounded by a polynomial in $\log Q$. Since Proposition 1 cannot be used for detecting which elements of $\mathcal{D}_f$ divide $(a(\iota) - b(\iota)\alpha)$, we have to use general algorithms, and again, we refer to [14].

Finally, we join the two factor bases and these exceptional ideals in the global factor base defined by

$$\mathcal{F} = \mathcal{F}_f(B) \cup \mathcal{F}_g(B) \cup \mathcal{D}_f \cup \mathcal{D}_g.$$

We note that, as usual, the parameter B will be chosen of the form $B = L_Q(1/3, \beta)$, for a constant β to be fixed later.

By the prime ideal theorem, the number of prime ideals in $\mathbb{Q}(\iota)$ of norm less than B is $\frac{B}{\log B}(1 + o(1))$. Using Chebotarev's density theorem, the average number of roots of f (resp. g) modulo a random prime ideal $\mathfrak{q}$ is one. Hence the cardinality of the factor base is

$$\#\mathcal{F} = \frac{B}{\log B}(2 + o(1)),$$

which is similar to its value in the classical variant of NFS. As usual, in the complexity analysis, we approximate $\#\mathcal{F}$ by the quantity $L_Q(1/3, \beta)$, since polynomial-time factors are, in the end, hidden in the $o(1)$ added to the exponent constant.

Finding doubly-smooth (a, b)-pairs. Among various choices for the shape of the $a(t)$ and $b(t)$ polynomials that we tried, the one giving the smallest norms is when a and b are of maximal degree $n-1$ and for which their coefficients are all of more or less the same size.

Let us denote by A a bound on these coefficients of $a(t)$ and $b(t)$. In the end, it will be chosen to be just large enough so that we get enough relations to get a full-rank system by browsing through all the possible coprime (a, b)-pairs of degree at most $n-1$ fitting this bound.

In order to estimate the probability that an (a, b)-pair gives a relation, the first step is to bound the size of the absolute norms on the f- and the g-side. The main tool is the following bound on the resultant.

Theorem 2 *[10, Thm 7] If f and g are polynomials with complex coefficients, of degree d_f and d_g, then*

$$|\operatorname{Res}(f, g)| \le \|f\|_\infty^{d_g} \|g\|_\infty^{d_f} \, (d_f+1)^{d_g/2}(d_g+1)^{d_f/2}.$$

We can now give the formula for the bound on the norm. We write it with the notations of the f-side, but it applies also to the g-side, after replacing the degree d by 1.

Theorem 3 *Let h and f be monic irreducible polynomials over $\mathbb{Z}$ of respective degrees n and d. Let K be the compositum of the number fields defined by h and f, and let ι and α_f be roots in K of h and f, respectively.*

Let $a(t)$ and $b(t)$ be two polynomials of degree less than n and with coefficients bounded by A. Then, the absolute norm of the element $a(\iota) - b(\iota)\alpha_f$ of K is bounded by

$$|\operatorname{N}_{K/\mathbb{Q}}\left(a(\iota) - b(\iota)\alpha_f\right)| < A^{nd}\|f\|_\infty^{n}\|h\|_\infty^{d(n-1)}\, C(n, d), \tag{1}$$

where $C(n, d) = (n+1)^{(3d+1)n/2}(d+1)^{3n/2}$.

Proof. Since f is monic, we have

$$\operatorname{N}_{K/\mathbb{Q}}\left(a(\iota) - b(\iota)\alpha_f\right) = \operatorname{N}_{\mathbb{Q}(\iota)/\mathbb{Q}}\Big(F(a, b)\Big),$$

where $F(a, b) = \sum_{i\in[0,d]} f_i a(t)^i b(t)^{d-i}$. The i-th term of this sum is a product of f_i and of d factors that are polynomials of degree less than n. Each term of the sum is therefore a polynomial of degree less than or equal tp $d(n-1)$ with coefficients bounded by $\|f\|_\infty A^d n^d$. Therefore, we have

$$\big\|F(a, b)\big\|_\infty \le (d+1)\|f\|_\infty A^d n^d.$$

Finally, since h is monic, we have

$$\operatorname{N}_{\mathbb{Q}(\iota)/\mathbb{Q}}(F(a, b)) = \operatorname{Res}\big(h, F\,(a, b)\big)\,,$$

and we can apply Theorem 2 to get the following upper bound:

$$\begin{aligned}\operatorname{N}_{\mathbb{Q}(\iota)/\mathbb{Q}}(F\,(a, b)) &\le \big\|F\,(a, b)\big\|_\infty^{n}\|h\|_\infty^{d(n-1)}\,(n+1)^{d(n-1)/2}(d(n-1)+1)^{n/2}\\ &< \|h\|_\infty^{d(n-1)}\, A^{nd}\|f\|_\infty^{n}\,(d+1)^{\frac{3}{2}n}(n+1)^{\frac{(3d+1)n}{2}}\end{aligned}$$

If the polynomials f, g or h are not monic, the theorem does not apply, since the element $a(\iota) - b(\iota)\alpha_f$ is not an integer anymore. However, the denominators, that are powers of the primes dividing the leading coefficients are under control in term of smoothness (it suffices to add a few prime ideals in the factor bases). And in fact, the quantity based on resultants computed in the proof of the theorem is the one that is really used for smoothness testing. Therefore, the monic hypothesis is not a restriction, and is just there to avoid technicalities.

It remains to plug-in $\|h\| = O(1)$ and the bounds for $\|f\|_\infty$ and $\|g\|_\infty$ coming from our choice of polynomial selection and we get:

$$N_{K_f/\mathbb{Q}}(a - b\alpha_f) \leq (A^{nd}\|f\|_\infty^n)^{1+o(1)} = (E^d Q^{1/(d+1)})^{1+o(1)}, \tag{2}$$

and

$$N_{K_g/\mathbb{Q}}(a - b\alpha_g) \leq (A^{n}\|g\|_\infty^n)^{1+o(1)} = (E Q^{1/(d+1)})^{1+o(1)}, \tag{3}$$

where we have set $E = A^n$, so that the quantity of pairs that are tested is E^2, just like in the classical NFS analysis. It is to be noted that the contribution of $C(n, d)$ remains negligible. Indeed, it would reach a value of the form $L_Q(2/3)$, only when n gets larger than an expression of the form $(\log Q/\log\log Q)^{1/3}$, which is not the case, since we ask that p is larger than any expression of the form $L_Q(2/3)$.

It is worth noticing that the expressions for the norms are the same as for the prime field case, where $Q = p$.

3.3 Writing and solving linear equations

Mapping a factorization of ideals to a linear combination of logarithms is not immediate unless the ring is principal and there are no units other than ± 1; both things are highly unlikely since the fields K_f and K_g have large degrees over $\mathbb{Q}$. Therefore, we have to resort to the notion of virtual logarithms, just like in the classical case.

For this, it is easier to work with absolute extensions. Then, we can use the same strategy as in Section 4.3 of [21], that we summarize in the following theorem which can be applied to K_f and K_g.

Theorem 4 *([21, Section 4.3]) Let $K = \mathbb{Q}(\theta)$ be a number field and $\mathfrak{P}$ a non-ramified ideal of its ring of integers $\mathcal{O}_K$, with residual field isomorphic to $\mathbb{F}_{p^n}$ in which we fix a generator t. Let ℓ be a prime factor of $p^n - 1$ and let $U = \{x \in K \mid \forall \mathfrak{L}$ above $\ell, \mathrm{val}_{\mathfrak{L}}(x) = 0\}$.*

We assume that there exists a Schirokauer function, i.e. an injective group homomorphism $\lambda = (\lambda_1, \ldots, \lambda_r) : (U/U^\ell, \cdot) \to (\mathbb{Z}/\ell\mathbb{Z}, +)^r$, where r is the unit rank of $\mathcal{O}_K$.

Assuming furthermore that ℓ does not divide the class number of K nor its discriminant, the following holds.

There exists a map $\log : \{$ideals of $\mathcal{O}_K$ coprime to $\mathfrak{P}\} \to \mathbb{Z}/\ell\mathbb{Z}$ and a map $\chi : \{1, \ldots, r\} \to \mathbb{Z}/\ell\mathbb{Z}$ called virtual logarithms, so that, for all $\phi \in \mathbb{Z}[x]$, such *that $\phi(\theta)$ is in U and coprime to $\mathfrak{P}$, we have*

$$\log_t \overline{\phi(\theta)}^{\mathfrak{P}} = \sum_{\mathfrak{Q} \text{ prime ideal}} \mathrm{val}_{\mathfrak{Q}}(\phi(\theta)) \log \mathfrak{Q} + \sum_{j=1}^{r} \lambda_j(\phi(\theta))\chi_j, \tag{4}$$

where $\overline{\phi(\theta)}^{\mathfrak{P}}$ is the projection of $\phi(\theta)$ in the residual field $\mathbb{F}_{p^n}$ of $\mathfrak{P}$.

In [33], Schirokauer explained how to construct an explicitly and efficiently computable map λ as in the theorem and brought heuristics to support the assumptions. These heuristics and the fact that the other hypothesis of the theorem are expected to be true rely on the condition that ℓ is not too small. These are the main reasons why we asked that ℓ grows at least like $L_Q(1/3)$ in the beginning.

For each (a, b)-pair that gives two smooth ideals in K_f and K_g, the element $a(\iota) - b(\iota)\alpha_f$ can be expressed in the absolute representation of $K_f = \mathbb{Q}(\theta_f)$ by a polynomial form $\phi_f(\theta_f)$, and similarly $a(\iota) - b(\iota)\alpha_g$ can be written $\phi_g(\theta_g)$ in $K_g = \mathbb{Q}(\theta_g)$. We refer for instance to [14] for algorithms to manipulate relative extensions as absolute extensions. Then, applying Theorem 4 to ϕ_f in K_f and ϕ_g in K_g, we obtain two linear expressions that must be equal, since they both correspond to the logarithm of the same element in $\mathbb{F}_{p^n}$.

As a consequence, each relation is rewritten as a linear equation between the virtual logarithms of the elements of the factor base and the χ_j for each field. We make the now classical heuristic that collecting roughly the same number of relations as the size of the factor base (say, a polynomial factor times more), then the linear system obtained in such a manner has a kernel of dimension one. A vector of this kernel is computed using Wiedemann's algorithm [36] in a quasi-quadratic time $B^{2+o(1)}$. This gives the logarithms of all the ideals in the factor base.

3.4 Overall complexity of the main phase

From the previous sections, we can now conclude about the complexity of the main steps of the algorithm. In fact, with our choice for the polynomial selection, and the kind of (a, b)-pairs that we test for smoothness, we have obtained exactly the same expressions for the sizes of the norms as in the usual NFS complexity analysis for prime fields, and in particular the same probability Prob that the product of the norms is smooth. Also, since the linear algebra step is also similar, the final complexity is the same: we have then to minimize $B^2 + E^2$ subject to the condition $E \cdot \mathrm{Prob} \geq B^{1+o(1)}$, and we refer for example to Conjecture 11.2 of [13]. Hence, the optimal values of the parameters are $E = B = L_Q(1/3, \sqrt[3]{\frac{8}{9}})$ and $d = \sqrt[3]{3}(\frac{\log Q}{\log \log Q})^{1/3}$, and the heuristic complexity of the main phase of TNFS is $L_Q(1/3, \sqrt[3]{\frac{64}{9}})$.

3.5 Individual logarithms

Let s be an element of $\mathbb{F}_{p^n}^*$ for which we want to compute the discrete logarithm. If s is very small, then it factors into ideals of the factor base, and its logarithm is easily retrieved. However, in general, this requires a 2-phase process that is not so trivial, although negligible compared to the other steps.

First, in what we call a smoothing phase, the element s is randomized and tested for B_1-smoothness with the ECM algorithm. The bound B_1 will be of the form $L_Q(2/3)$, so that the cost of the smoothing test is in $L_Q(1/3)$.

Thereafter, each prime ideal $\mathfrak{Q}$ which is not in the factor base is considered as a special-q and we search for a relation involving $\mathfrak{Q}$ and other smaller ideals. Continuing recursively, we get a special-q descent tree, from which the logarithm of s can be deduced.

Smoothing. The randomization is simple: we compute $z = s^e$ in $\mathbb{F}_{p^n}$ for random values e, and test z for smoothness. The logarithm of s is just the logarithm of z divided by e modulo ℓ.

To be more precise, the smoothness is not tested for the element z as an element of the finite field, but as the corresponding element in K_g. Indeed, in our construction, $z \in \mathbb{F}_{p^n}$ is represented by a polynomial of degree less than n with coefficients modulo p. Lifting these coefficients to integers, we obtain a polynomial which makes sense modulo $h(t)$, therefore an element of $\mathbb{Q}(\iota)$ which is isomorphic to K_g, because g is a linear polynomial. As usual, to test the smoothness of z as an element of $\mathbb{Q}(\iota)$, we test the smoothness of its norm as an integer. Using again the estimate of Theorem 3, the size S of this norm is $Q^{1+o(1)}$.

The bound B_1 can then be optimized w.r.t. this only step, like in the classical NFS: if this is too small, the probability of being smooth is too small, while if it is too large, the cost of testing the smoothness by ECM is prohibitive. The analysis is the same as in [15] and gives a value $B_1 = L_Q(2/3, (\frac{1}{3})^{1/3})$; the corresponding cost for the smoothing phase is $L_Q(1/3, 3^{1/3})$.

After the smoothing phase, the logarithm of s has been rewritten in terms of the logarithms of small prime ideals of K_g for which the logarithm is already known, and some largish prime ideals of K_g, of norm bounded by B_1. The next step is to compute the logarithms of these largish ideals.

Descent by special-q. As in NFS, the algorithm is recursive: if $\mathfrak{Q}$ is a prime ideal of degree one in K_f (respectively K_g), then we write $\log \mathfrak{Q}$ as a formal sum of virtual logs of ideals $\mathfrak{Q}'$ of K_f and K_g with norm less than $\mathrm{N}(\mathfrak{Q})^c$, for a positive parameter $c < 1$. For this, we consider the lattice of (a, b)-pairs for which $\mathfrak{Q}$ divides the element $a - b\alpha_f$ (resp. $a - b\alpha_g$). A basis for this lattice can be constructed and LLL-reduced. Small combinations of these basis vectors are then formed and the norms of the corresponding (a, b) pairs are tested for $\mathrm{N}(\mathfrak{Q})^c$-smoothness. We refer to Appendix 7.1 for the description of this special-q lattice technique, that is also used in practice during the collection of relations in the main stage. When a relation is found, this gives a new node in the descent tree, the children of it being the ideals of the relations that are still too large to be in the factor base. The total number of nodes is quasi-polynomial.

The cost of each step is determined by the size of $\mathrm{N}(a(\iota) - \alpha_f b(\iota))$ (resp. $\mathrm{N}(a(\iota) - \alpha_f b(\iota)))$ which are tested during the computations. The matrix $M_{\mathfrak{Q}}$ of the basis of the lattice has determinant $\det M_{\mathfrak{Q}} = \mathrm{N}(\mathfrak{Q})$, so a short vector in the LLL-reduced basis has coordinates of size $\approx \mathrm{N}(\mathfrak{Q})^{1/(2n)}$. We make the heuristic assumption that all the vectors of the reduced basis, $(a^{(k)}, b^{(k)})$ for $k = 1, \ldots, 2n$, have coordinates of the same size. The pairs (a, b) tested for smoothness are linear combinations $(a, b) = \sum_{k=1}^{2n} i_k (a^{(k)}, b^{(k)})$ where i_k are rational integers with absolute value less than a parameter E'. By Theorem 3, the size of the norms tested for smoothness is

$$N_{K_f/\mathbb{Q}}(a - b\alpha_f) \leq (\max(\|a\|_\infty, \|b\|_\infty)^{nd} \|f\|_\infty^n)^{1+o(1)} = (\mathrm{N}(\mathfrak{Q})^{d/2} (E')^d Q^{1/d})^{1+o(1)},$$

$$N_{K_g/\mathbb{Q}}(a - b\alpha_g) \leq (\max(\|a\|_\infty, \|b\|_\infty)^{n} \|g\|_\infty^n)^{1+o(1)} = (\mathrm{N}(\mathfrak{Q})^{1/2} E' Q^{1/d})^{1+o(1)}.$$

These expressions coincide with the ones in the analogous stage of the classical variant (for example in Equation (7.11) in [5]) and we obtain a complexity of $L_Q(1/3, 1.1338...)$ which is the same as in the classical case [15]. We conclude that the overall complexity of individual logarithm is dominated by the $L_Q(1/3, 3^{1/3})$ complexity of the smoothing test.

4 Variants

Note on the boundary case. TNFS can be applied to the boundary case $p = L_Q(2/3, c_p)$, $c_p > 0$, where one obtains a complexity $L_Q(1/3, c)$. The constant c is strictly larger then $\sqrt[3]{64/9}$ as the factor $C(n, d)$ in Equation (1) is not negligible any more. Yet, for some values of c_p, TNFS overcomes the method of [21], which was state-of-art until recently. Using the generalized Joux-Lercier method, the authors of [6,7] reduced the constant c to $(64/9)^{1/3} \approx 1.92$ and Pierrot [31] showed that a multiple fields variant allows to further reduce c to ≈ 1.90. Therefore, we do not reproduce here the tedious computations of the complexity in the boundary case.

The case of primes of special form (SNFS) Given a positive integer d, an integer p, not necessarily prime, is said to be a d-SNFS integer if it can be written as $p = P(u)$ for some integer $u \approx p^{1/d}$ and a polynomial $P \in \mathbb{Z}[x]$ such that $\|P\|_\infty$ is small (say, bounded by a constant). We remark that when a number is SNFS, then there can be several valid choices for d and P. This is typically the case for integers of the form $2^k + \varepsilon$, for tiny ε.

When solving DLP in fields $\mathbb{F}_{p^n}$ for d-SNFS primes p, we can follow the classical SNFS construction [27] and set

$$f(x) = P(x) \text{ and } g(x) = x - u.$$

When evaluating the sizes of the norms, Equation (2) can be restated with $\|f\|_\infty = O(1)$, so we obtain the following bound:

$$\mathrm{N}_{K_f/\mathbb{Q}}(a - b\alpha_f)\,\mathrm{N}_{K_g/\mathbb{Q}}(a - b\alpha_g) \leq (E^{d+1}Q^{1/d})^{1+o(1)}. \tag{5}$$

Following the analysis of Semaev [35], we obtain that if the degree d can be chosen to grow precisely as $d = \sqrt[3]{\frac{3}{2}}\left(\frac{\log Q}{\log\log Q}\right)^{1/3}$, then the overall complexity of SNFS is the same as that of factoring numbers from the Cunningham project, namely $L_Q\left(1/3, \sqrt[3]{\frac{32}{9}}\right)$.

Using multiple number fields (MNFS) Given a choice of polynomials f and g selected as in the first step of TNFS, one can construct a large number of polynomials f_i which share with f and g the root m modulo p. The idea goes back to Coppersmith's variant of NFS for factorization [16] and has been used again in [28], [8] and [31]. Let V be a parameter of size $L_Q(1/3, c_v)$ for some constant $c_v > 0$. For all $\mu(t)$ and $\nu(t) \in \mathbb{Z}[t]$ so that $\deg\mu, \deg\nu \leq n-1$ and $\|\mu\|_\infty, \|\nu\|_\infty \leq V^{1/(2n)}$, we set

$$f_{\mu,\nu} = \mu(\iota)f + \nu(\iota)g, \tag{6}$$

keeping only those polynomials that are irreducible (most of them are, so we expect that the correcting factor on the bound for $\|\mu\|_\infty$ and $\|\nu\|_\infty$ are only marginally adjusted). Let us denote by $K_{f_{\mu,\nu}}$ the number field generated by $f_{\mu,\nu}$ over $\mathbb{Q}(\iota)$, and call $\alpha_{\mu,\nu}$ a root of $f_{\mu,\nu}$ in its number field. For any pair (μ, ν) as above and (a, b) in the sieving domain, by Theorem 3 we have

$$\mathrm{N}_{K_{\mu,\nu}}(a - \alpha_{\mu,\nu}b) \leq A^{nd}(V^{1/(2n)}\|f\|_\infty)^n \|h\|_\infty^{nd} C(n,d) = (V^{1/2}E^dQ^{1/d})^{1+o(1)}. \tag{7}$$

In the multiple number field sieve a relation is given by a pair (a, b) in the sieving domain and by a polynomial $f_{\mu,\nu}$ from the set constructed above so that $\mathrm{N}_{K_g/\mathbb{Q}}(a - b\alpha_g)$ is B-smooth and $\mathrm{N}_{K_{f_{\mu,\nu}}}(a - b\alpha_{\mu,\nu})$ is B/V-smooth. We use as factor base the set

$$\mathcal{F} = \Big(\bigcup_{\mu,\nu} \mathcal{F}_{f_{\mu,\nu}}(B/V)\Big) \bigcup \mathcal{F}_g(B).$$

We collect relations as in Coppersmith's modification: collect pairs (a, b) in the sieving domain and keep only those for which $\mathrm{N}_{K_g/\mathbb{Q}}(a-\alpha_g b)$ is B-smooth. Then, for each surviving pair (a, b) we use ECM to collect polynomials $f_{\mu,\nu}$ such that $\mathrm{N}_{K_{f_{\mu,\nu}}/\mathbb{Q}}(a - \alpha_{\mu,\nu}b)$ is B/V-smooth.

We choose parameter E so that the number of collected pairs exceeds $2B$, which is an upper bound on $\#\mathcal{F}$. The same considerations as in [16] allow us to find the optimal parameters: $V = L_Q(1/3, 1 - (\frac{\sqrt{13}-1}{3})^{1/3})$, $E = B = L_Q(1/3, (\frac{46+13\sqrt{13}}{108})^{1/3})$ and $d = \delta(\log Q/\log\log Q)^{1/3}$ where $\delta = (\frac{32-2\sqrt{13}}{9})^{1/3}$, and the complexity of the multiple polynomial TNFS is $L_Q(1/3, (\frac{92+26\sqrt{13}}{27})^{1/3})$.

Automorphisms. Joux, Lercier, Smart and Vercauteren [21] proposed an improvement based on the field automorphisms of the number fields occurring in NFS. A recent preprint proves (a reformulation of) the following result:

Theorem 5 (Theorem 3.5 of [6]) *Let σ be a field automorphism of $K/\mathbb{Q}$. Assume that $\mathfrak{P}$ is a prime ideal of K above p such that $\sigma\mathfrak{P} = \mathfrak{P}$. Fix a prime ℓ dividing $\mathrm{N}(\mathfrak{P}) - 1$, coprime to the class number and the discriminant of K, and assume that the Schirokauer maps associated to ℓ have the properties in Theorem 4. Fix a generator t of the residual field of $\mathfrak{P}$ and, for any prime ideal $\mathfrak{Q}$, denote by $\log \mathfrak{Q}$ the virtual logarithm of $\mathfrak{Q}$ with respect to t and the Schirokauer maps associated to K and ℓ. Then, there exists a constant $\kappa \in [0, \mathrm{ord}(\sigma) - 1]$ such that for any $\mathfrak{Q}$ we have*

$$\log(\sigma\mathfrak{Q}) \equiv p^{\kappa} \log(\mathfrak{Q}) \mod \ell.$$

In Section 3.1 we noted that one might find ι so that $\mathbb{Q}(\iota)/\mathbb{Q}$ has n automorphisms over $\mathbb{Q}$. All these automorphisms can be used to speed-up computations, using the following result.

Corollary 6 *Let σ be an automorphism of $\mathbb{Q}(\iota)/\mathbb{Q}$ and call $\tilde{\sigma}$ the unique field automorphism of K_f such that $\tilde{\sigma}(\iota) = \sigma(\iota)$ and $\tilde{\sigma}(\alpha_f) = \alpha_f$. Then, there exists $\kappa \in [0, \mathrm{ord}(\sigma) - 1]$ such that, for all prime ideals $\mathfrak{Q}$ of K_f, we have*

$$\log(\tilde{\sigma}\mathfrak{Q}) \equiv p^{\kappa} \log \mathfrak{Q} \mod \ell.$$

Proof. The only condition which needs to be tested is $\tilde{\sigma}\mathfrak{P}_f = \mathfrak{P}_f$. We compute

$$\tilde{\sigma}\mathfrak{P}_f = \tilde{\sigma}\langle p\mathbb{Z}[\iota], \alpha_f - m\rangle = \langle \tilde{\sigma}(p)\mathbb{Z}[\iota], \tilde{\sigma}(\alpha_f) - \tilde{\sigma}(m)\rangle = \langle p\mathbb{Z}[\iota], \alpha_f - m\rangle = \mathfrak{P}_f.$$

According to [7], automorphisms allow us to sieve n times faster and to speed-up the linear algebra stage by a factor n^2. Note that, contrary to the classical variant of NFS where automorphisms were available only for certain values of n, TNFS has no restrictions.

5 Comparison for cryptographically relevant sizes

The complexity of NFS and its many variants is written in the form $C^{1+o(1)}$, which can hide large factors, and therefore we cannot decide which variant to implement based only on asymptotic complexity. We follow the methodology of [7, Section 4.4] and do a more precise comparison by evaluating the upper bound on the size of the integers which are tested for smoothness: the product of the norms with respect to the two sides.

5.1 The case of general primes

When p is not an SNFS number, we compare TNFS to the algorithm of Joux, Lercier, Smart and Vercauteren(JLSV) [21]. From Equations (2) and (3) we find a formula for the logarithm of the product of the norms in TNFS:

$$C_{\mathrm{NFS}} = (d+1)\log_2 E + \frac{2}{d+1}\log_2 Q,$$

where $d = \deg f$ can be chosen as desired (unlike the SNFS variant of the algorithm where d might be imposed by the shape of p). Note that we prefered the notation C_{NFS} to C_{TNFS} which emphasizes that the NFS formula.

Since the JLSV algorithm comes with a variety of methods to select polynomials, we cannot give a unified formula for the size of norms' product, so we use the minimum of the formulae in [7]; therefore, in the following, when we say JLSV, this covers both variants explained in [21] as well as the Conjugation and Generalized Joux-Lercier methods. The choice of the parameter E depends on the size of the norms, but for a first comparison we can use the default values of CADO-NFS [7, Table 2].

In Figure 2 we compare TNFS to JLSV when p is a general prime (not SNFS), for a range $300 \leq \log_2 Q \leq 1000$. We conclude that in this range, when $n \geq 5$, TNFS is competitive and must be kept for an even more accurate comparison.

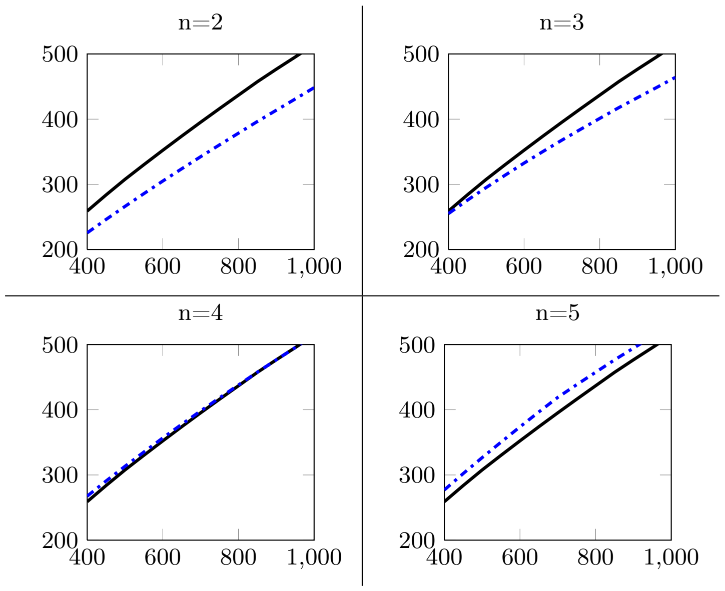

Fig. 2. Comparison of TNFS (in black) and the best variant of JLSV algorithm (in dashdotted blue).

5.2 The case of primes of special shape (SNFS)

The importance of the *d* parameter. If we want to compute discrete logarithms in a field $\mathbb{F}_{p^n}$ such that p is d-SNFS for a parameter d, then the first question to ask is whether to use a general algorithm like TNFS and JLSV or a specialized variant of these two, namely the SNFS variant of TNFS or the Joux-Pierrot algorithm.

When $d = 6$ we can rely on a real-life example: Aoki et al. [2] factored a 1039-bit integer with SNFS, using sextic polynomials, i.e. $d = 6$. The current record, hold by Kleinjung et al. [26], was obtained with a MNFS variant and targeted d-SNFS integers for $d = 8$. Their computations were much faster than the evaluated time to factor a 1024-bit RSA modulus, so it is safe to say that SNFS is the best option when $\log_2 Q \approx 1024$ and $d = 6$ or when $d = 8$ for slightly larger targets. However, the value of d is fixed in most cases and can take very different values among curves used in pairing-based crypto-systems, going from $d = 2$ for MNT curves [29] to $d = 56$ in other constructions [18, Table 6.1],[30].

If the polynomial P such that $p = P(u)$ has a special shape, one can try to reduce the value of d using techniques from the Cunningham project records. On the one hand, if $P = T(x^a)$ with $T \in \mathbb{Z}[x]$ and $a \in \mathbb{N}$, we can also write $p = T(v)$ with $v = u^a$, so p is $(\deg T)$-SNFS. This technique can be used for example in the construction of Brezing-Weng [12, Section 3, item 3(b)] where $P(x) = \mu a^2 + \nu b^2$ for some small constants μ and ν and where $a, b \in \mathbb{Z}[x^5]$ have degree 5 and respectively 15; we replace P of degree 30 by a polynomial of degree 6. On the other hand, if P has even degree and is reciprocal, i.e. its coefficients form a palindrome, we can divide the degree of P by 2. Indeed, if P is reciprocal it can be written as $P(x) = x^{(\deg P)/2} T(x + \frac{1}{x})$ with $T \in \mathbb{Z}[x]$; we select $f = T(x)$ and $g = ux - (u^2 + 1)$, which share the root $u + \frac{1}{u}$ modulo p and are so that $\|f\|_\infty = O(1)$ and $\|g\|_\infty = p^{1/\deg f}$. This technique can be used for a construction of Freeman, Scott and Teske [18, Construction 6.6].

Modeling. A good comparison requires a precise estimation of the norms. However, several factors in Equation (1) can be negligible in some cases but can also be very large in the others:

$$\text{negligible factors} = C(n, d) \|f\|_\infty^n \|h\|_\infty^d .$$

The factor $C(n, d)$ is itself a bad estimation of the number of terms in the Sylvester discriminant, which can vary between 6 bits for $n = 2$ and $d = 3$ to 15 bits for $n = 5$ and $d = 6$. This determines us to restrict to $n \leq 5$ and $d \leq 6$. The factor $\|f\|_\infty^n$ equals 1 if $\|f\|_\infty = 1$ but can be as large as 2^{62} when $n = 12$ and $\|f\|_\infty = 36$. Hence, it is impossible to plot the size of the norms for all SNFS numbers, independently of the polynomial f.

For our modeling, we consider the case $\|f\|_\infty = \|h\|_\infty = 1$ and neglect the combinatorial factor $C(n, d)$ for small values of n and d. From Equation (5) the dominant factor in the product of the norms for TNFS, when tackling SNFS primes, is

$$C_{\text{SNFS}}(n, d) = \log(E^{d+1}) + \log(Q^{1/d}).$$

Note again that this formula is the same as that of the complexity of the factoring variant of SNFS.

The product of the norms in the Joux-Pierrot algorithm is bounded by $(n + 1)^{2t} (\log n)^{nd}\ E^{2n(d+1)/t}\ Q^{(t-1)/(nd)}$ (discussion preceding Equation (5) in [22]), and for the comparison we keep only the logarithm of most important factors:

$$C_{\mathrm{JP}}(n,d,t) = \frac{2n}{t}\log(E^{d+1}) + \frac{t-1}{n}\log(Q^{1/d}).$$

Let us see two examples in which we tackle fields of about one kilobit, for which we use the approximation $\log_2 E = 30.4$, as in [2].

A first example. We target a 1024-bit field $\mathbb{F}_{p^2}$ for a 6-SNFS prime p and we set the parameters equal to their value in the computation of Aoki et al. If one chooses to forget that p has a special shape and uses JLSV with conjugation method, then the product of the norms has bitsize ≈ 439. If instead one uses the special shape of p, the product of the norms for TNFS has bitsize $C_{\mathrm{SNFS}}(n = 2, d = 6) \approx 386$, while the best parameters for the Joux-Pierrot algorithm yield $C_{\mathrm{JP}}(n = 2, d = 6, t = 3) \approx 457$. A probabilistic experiment suggests that our model is quite precise, as the negligible factors do not add more than 6 bits.

Barreto-Naehrig. The elliptic curves proposed by Barreto and Naehrig [9] correspond to finite fields of parameters $n = 12$ and $d = 4$. We tackle a field of 1024-bit cardinality and we will use a value of E close to the one in the factorization record, i.e. $\log_2 E = 30.4$. If we forget that p is SNFS, then we can choose the value of d in TNFS and we find $C_{\mathrm{NFS}}(n = 12, d = 7) = 500$. If instead we use the special shape of p we obtain $C_{\mathrm{SNFS}}(n = 12, d = 4) = 408$ and $C_{\mathrm{JP}}(n = 12, d = 4, t = 12) = 539$. A probabilistic experiment showed that the product of the norms for TNFS is 60 to 80 bits larger when $f = 36x^4 + 12x^3 + 16x^2 + 2x + 1$ and $h = x^{12} - x - 1$.

There are however examples when the specialized algorithms do not apply.

Fact 7 *When $d = 2$, our SNFS algorithms are not better than the general ones, i.e.*

$$C_{\mathrm{JLSV}} \leq \min(C_{\mathrm{JP}}, C_{\mathrm{SNFS}}),$$

where C_{JLSV} is the complexity of the JLSV algorithm with conjugation method.

To see this, note first that the Joux-Pierrot algorithm keeps unchanged the stages of JLSV once finished the the polynomial selection. In the Joux-Pierrot algorithm one constructs polynomials f and g such that $\deg(f) = nd$, $\deg(g) = n$, $\|f\|_\infty = O(1)$ and $\|g\|_\infty = Q^{1/(nd)}$. However, when $n = 2$, they have the same characteristics as the polynomials constructed by the Conjugation method, which applies to arbitrary primes.

Also note that the SNFS variant of TNFS uses a polynomial g with coefficients of size $p^{1/d}$. When $d = 2$ the norm of the g-side has bitsize larger than $\frac{1}{2}\log_2 Q$, which is typical for algorithms of complexity $L_Q(1/2)$ and is larger than the norms considered in the JLSV algorithm in the range $\log_2 Q \leq 1000$ and $n \leq 5$.

Plots. Let us plot the modelled bitsize of the norms product for TNFS and Joux-Pierrot in the range which is currently feasible or might become in the near future: $300 \leq \log_2 Q \leq 1000$. Allong with the two competitive algorithms, Joux-Pierrot (C_{JP}) and the SNFS variant of TNFS (C_{SNFS}) we include the bitsize of the integers that have to be smooth when factoring RSA numbers (C_{NFS}).

Note that, together with C_{SNFS} (SNFS variant of TNFS) and C_{JP} (Joux-Pierrot), we also plot C_{NFS} which represents the bitsize of the product of the norms in NFS when factoring RSA numbers. We make separate graphs for each pair (n, d) where n is the degree of the target field and d is the parameter such that p is d-SNFS, as those parameters are unique (in general) for each finite field: Figure 3 (n=2), Figure 4 (n=3), Figure 5 (n=4) and Figure 6 (n=5). Albeit the value of E depends on the size of the norms, in a first approximation we can use the formula $E = c \cdot L_Q(1/3, (4/9)^{1/3})$ where c is a constant chosen such that the formula fits the value of E in the example of Aoki et al.

We emphasize that our comparisons are imprecise since they are based only on the product of the norms. Nevertheless, one might make a series of remarks:

– when $d \geq 3$, the two algorithms specialized in fields of SNFS characteristic have smaller norms than those of NFS when factoring RSA numbers;
– when $d \geq 4$, TNFS is an important challenger for the Joux-Pierrot algorithm.

6 Cryptographic consequences

The number field sieve algorithm is still far from being fully understood, in particular for extension fields that are so important for pairing-based cryptography. In the past few years, several improvements have been made in the asymptotic complexities in various scenarios, leading in particular to an $L(1/3, \sqrt[3]{32/9})$ complexity for small degree extensions of SNFS-prime fields, that are common in pairing-friendly constructions.

We have shown, that in this setting, an old NFS variant due to Schirokauer could compete and probably overcome the algorithm by Joux-Pierrot. We acknowledge that the comparison is not perfect since it is based on a model where the efficiency is directly linked to the size of product of the norms of the elements that have to be tested for smoothness. Still, in some cases, the difference is large enough (a few dozens of bits), so that we are confident that this should translate into a significant practical difference.

Of course, only a careful implementation of both algorithms could confirm this. Unfortunately, this goes way beyond the scope of this paper. As far as we know, Joux-Pierrot's algorithm has not been used so far for a record-setting computation, and Schirokauer's TNFS would require even more implementation work to handle the sieve in higher dimension. And since doing experiments with non-optimized implementations and small field sizes could lead to highly misleading conclusions, we preferred to keep this for future work.

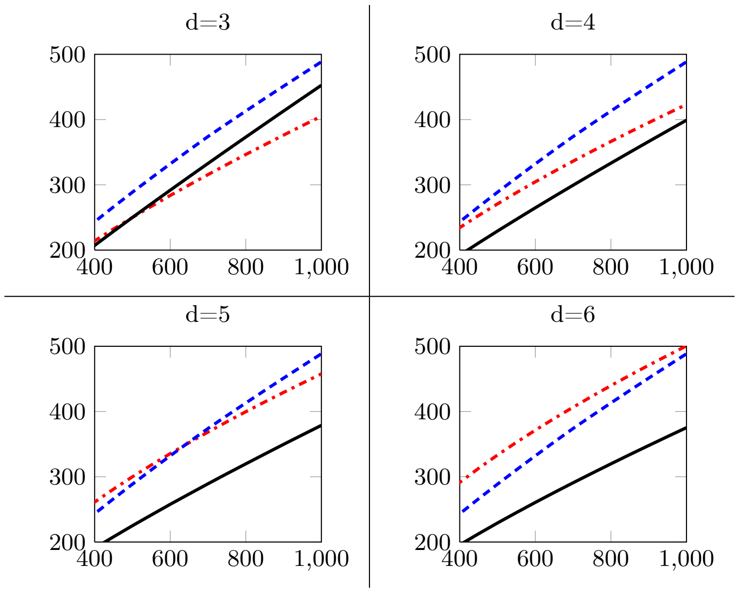

Fig. 3. Comparison of C_{NFS} (in dashed blue), C_{SNFS} (in black) and C_{JP} (in dasdotted red) in $\mathbb{F}_{p^n}$ with $n = 2$, for d-SNFS primes.

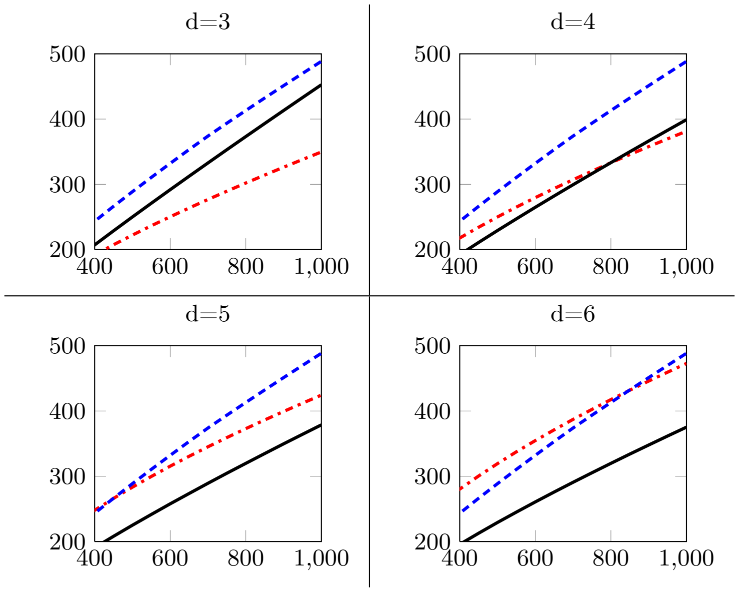

Fig. 4. Comparison of C_{NFS} (in dashed blue), C_{SNFS} (in black) and C_{JP} (in dashdotted red) in $\mathbb{F}_{p^n}$ with $n = 3$, for d-SNFS primes.

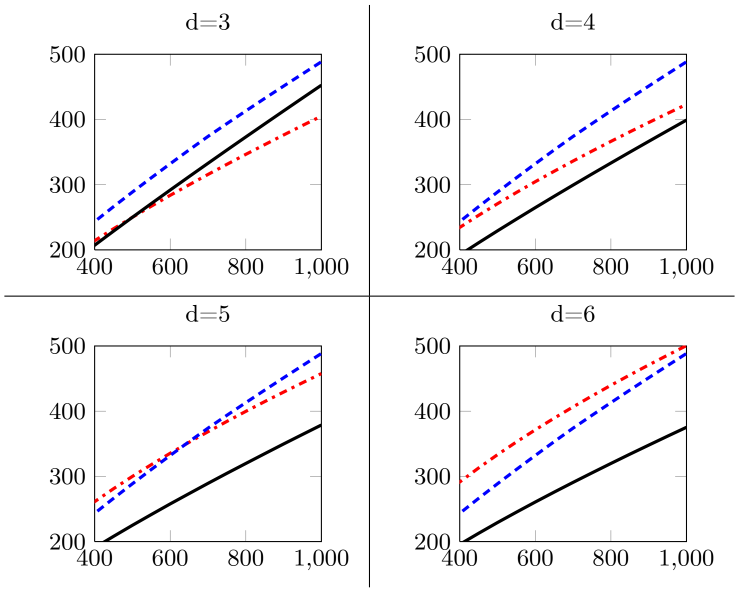

Fig. 5. Comparison of C_{NFS} (in dashed blue), C_{SNFS} (in black) and C_{JP} (in dashdotted red) in $\mathbb{F}_{p^n}$ with $n = 4$, for d-SNFS primes.

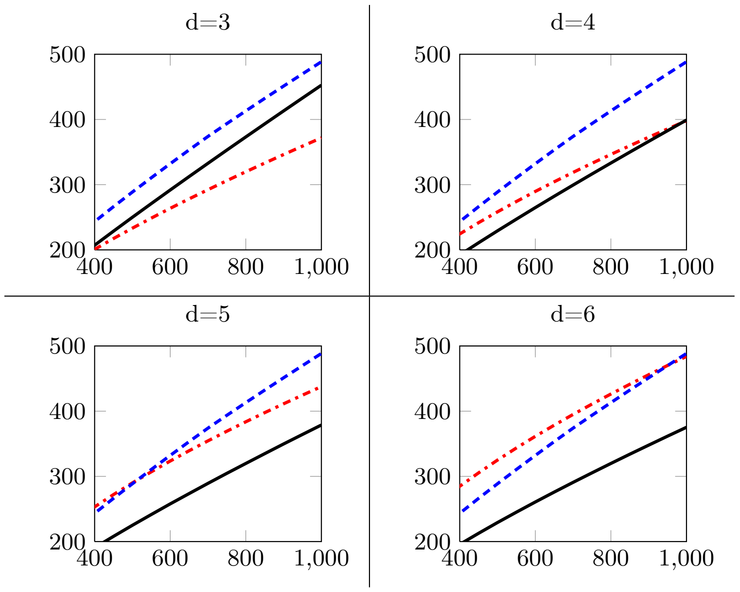

Fig. 6. Comparison of C_{NFS} (in dashed blue), C_{SNFS} (in black) and C_{JP} (in dashdotted red) in $\mathbb{F}_{p^n}$ with $n = 5$, for d-SNFS primes.

7 Appendix: technicalities

7.1 Special-q sieving

In practice for prime fields the relation collection phase is split in subtasks following the so-called special-q sieving strategy. It is expected, but no so obvious, that this technique can be adapted to the case of TNFS.

The general case. Given a prime ideal $\mathfrak{Q}$ of K_f (resp. of K_g), the special-q algorithm collects (most of) the coprime pairs $(a,b) \in \mathbb{Z}[\iota]^2$ which satisfy

- $a - b\alpha_f \equiv 0 \mod \mathfrak{Q}$;
- $\mathrm{N}_{K_f/\mathbb{Q}}(a-b\alpha_f)/\,\mathrm{N}_{K_f/\mathbb{Q}}(\mathfrak{Q})$ and $\mathrm{N}_{K_g/\mathbb{Q}}(a-b\alpha_g)$ are B-smooth,

and which have coefficients bounded by $\mathrm{N}_{K_f/\mathbb{Q}}(\mathfrak{Q})^{1/2n}I$ for a parameter I.

In the main lines, the sieving is done by Algorithm 1, where a key role is played by the lattice $L_{\mathfrak{Q}}$ of (a,b)-pairs such that $\mathfrak{Q} \mid a - b\alpha_f$:

$$L_{\mathfrak{Q}} = \Big\{(a_0,\dots,a_{n-1},b_0,\dots,b_{n-1}) \in \mathbb{Z}^{2n} \mid \big(\sum_{k=0}^{n-1} a_k\iota^k\big) - \alpha_f\big(\sum_{k=0}^{n-1} b_k\iota^k\big) \equiv 0 \mod \mathfrak{Q}\Big\}.$$

Algorithm 1 Special-q task

```
1: Compute an LLL-reduced basis of $L_{\mathfrak{Q}}$, $u^{(1)},\dots,u^{(2n)}$, and for each $k$ define the
   pair $(a^{(k)},b^{(k)})$ by $a^{(k)} = \sum_{i=0}^{n-1} u_i^{(k)}\iota^i$ and $b^{(k)} = \sum_{i=n}^{2n-1} u_i^{(k)}\iota^i$.
2: Initialize an array indexed by $(i_1,\dots,i_{2n}) \in \prod_{k=1}^{2n}[-I,I]$ with the value of
   $\log_2 \mathrm{N}_{K_f/\mathbb{Q}}(a-b\alpha_f)$ where
       $$a = \sum_{k=1}^{2n} i_k a^{(k)} \text{ and } b = \sum_{k=1}^{2n} i_k b^{(k)}.$$
3: For each $\mathfrak{L}$ in $\mathcal{F}_f$ update the entries of the array such that $a - b\alpha_f \in \mathfrak{L}$.
4: Collect yield(f), the coprime pairs $(a,b)$ associated to entries of the array with
   value less than a given threshold.
5: Repeat Steps 2-4 with $f$ replaced by $g$, and collect yield(g).
6: return yield(f) $\bigcap$ yield(g)
```

In more detail, if $\mathfrak{Q} = \langle \mathfrak{q}, \alpha_f - \rho_{\mathfrak{Q}}(\iota)\rangle$ and $\mathfrak{q} = \langle q, \varphi_{\mathfrak{q}}(\iota)\rangle$, we can assume that $\varphi_{\mathfrak{q}}$ is monic and define the matrix

$$M_{\mathfrak{Q}} = \left(\begin{array}{c|ccc}
q & 0 & \cdots\ \cdots & 0 \\
\ddots & \vdots & & \vdots \\
q & & & \\
\boxed{\text{vector}(\varphi_{\mathfrak{q}})} & & & \\
\ddots & \vdots & & \vdots \\
\boxed{\text{vector}(\varphi_{\mathfrak{q}})} & 0 & \cdots\ \cdots & 0 \\
\hline
\text{vector}(\rho_{\mathfrak{Q}}(\iota)) & 1 & & \\
\text{vector}(\rho_{\mathfrak{Q}}(\iota)\iota) & & \ddots & \\
\vdots & & & \ddots \\
\text{vector}(\rho_{\mathfrak{Q}}(\iota)\iota^{n-1}) & & & 1
\end{array}\right).$$

One can check that the rows of $M_{\mathfrak{Q}}$ form a basis of $L_{\mathfrak{Q}}$, and that $\det(L_{\mathfrak{Q}}) = q^{\deg(\varphi_{\mathfrak{q}})} = \mathrm{N}_{\mathbb{Q}(\iota)/\mathbb{Q}}(\mathfrak{q}) = \mathrm{N}_{K_f/\mathbb{Q}}(\mathfrak{Q})$ and $\dim L_{\mathfrak{Q}} = 2n$. Then, the coefficients of the shortest vector in an LLL-reduced basis have size about $\mathrm{N}_{K_f/\mathbb{Q}}(\mathfrak{Q})^{1/(2n)}$. We make the heuristic assumption that for a large proportion of ideals $\mathfrak{Q}$, all the vectors in the reduced basis have coefficients of this size. Then, the coefficients of the (a,b) pairs visited during Steps 3-4-5 of Algorithm 1 are approximatively equal to $I\,\mathrm{N}_{K_f/\mathbb{Q}}(\mathfrak{Q})^{1/(2n)}$.

The critical part of Algorithm 1 is Step 4., where we need to solve a problem that Pollard [32] asked in the case $m = 2$.

Problem 1. Compute the intersection of a sub-lattice of $\mathbb{Z}^m$ with an interval product $\prod_{k=0}^{m-1} I_k$.

Since the dimension is fixed or small enough, we can use a generic lattice enumeration algorithm like the Kannan-Fincke-Pohst algorithm. In the case $m = 2$, Franke and Kleinjung [25, Appendix A] gave an elegant algorithm that proved very efficient in practice. Extending this algorithm to higher dimension is still an open problem.

The particular case of Gaussian integers When $h = x^2 + 1$, $\iota = i$ and we have a series of advantages. First of all, we have $\deg(h) = n = 2$, so the combinatorial overhead $C(n, d)$ in Theorem 3 is small.

Secondly, the ring $\mathbb{Z}[i]$ is Euclidean, so that we can speed-up Step 1 of Algorithm 1.

Lemma 8 *Let q and r be two elements of $\mathbb{Z}[i]$ such that q is irreducible and r is not divisible by q. Assume that $\mathfrak{Q} = \langle q, \alpha_f - r\rangle$ is a prime ideal of K_f. Let $(u_j, v_j, d_j)_{j\geq 0}$ be the sequence of Bezout coefficients*

$$u_j q + v_j r = d_j$$

obtained during the Extended Euclidean Algorithm(EEA). Let $j \geq 0$ be an integer. For $k = 1, 2, 3, 4$ we set

$$\begin{aligned} (a^{(1)}, b^{(1)}) &= (d_j, v_j), & (a^{(2)}, b^{(2)}) &= (id_j, iv_j), \\ (a^{(3)}, b^{(3)}) &= (d_{j+1}, v_{j+1}), & (a^{(4)}, b^{(4)}) &= (id_{j+1}, iv_{j+1}), \end{aligned}$$

and define $u^{(k)} = (\mathrm{Re}(a^{(k)}), \mathrm{Im}(a^{(k)}), \mathrm{Re}(b^{(k)}), \mathrm{Im}(b^{(k)}))$. Then the vectors $u^{(1)}$, $u^{(2)}$, $u^{(3)}$, $u^{(4)}$ form a basis of the lattice $L_{\mathfrak{Q}}$.

Proof. Note first that if two elements e_1, e_2 form a basis for a $\mathbb{Z}[i]$-module M, then the set $\{e_1, ie_1, e_2, ie_2\}$ is a basis of M seen as a $\mathbb{Z}$-module. We apply this fact to $M = \{(a, b) \in \mathbb{Z}[i] \times \mathbb{Z}[i] \mid a - br \equiv 0 \mod q\}$, so it is sufficient to show that (d_j, v_j) and (d_{j+1}, v_{j+1}) form a basis of M when seen as a $\mathbb{Z}[i]$-module.

By construction of the sequence $(u_j, v_j, d_j)_{j\geq 0}$, there exist invertible matrices $I_1, I_2, \ldots \in \mathrm{GL}(\mathbb{Z}[i], 2)$ so that, for all $j \geq 1$,

$$\begin{pmatrix} u_{j+1} & v_{j+1} & d_{j+1} \\ u_j & v_j & d_j \end{pmatrix} = I_j \begin{pmatrix} u_j & v_j & d_j \\ u_{j-1} & v_{j-1} & d_{j-1} \end{pmatrix}.$$

Therefore, for all j, the pairs (d_j, v_j) and (d_{j+1}, v_{j+1}) span the same $\mathbb{Z}[i]$-module. In particular, for $j = 0$, we have

$$(d_0, v_0) = (q, 0) \quad \text{and} \quad (d_1, v_1) = (r, 1),$$

which is a basis of M, so that any pair in the sequence spans M. Finally, a pair $(a, b) \in \mathbb{Z}[i]\times\mathbb{Z}[i]$ is in M if and only if the vector $u = (\mathrm{Re}(a), \mathrm{Im}(a), \mathrm{Re}(b), \mathrm{Im}(b))$ is in the lattice $L_{\mathfrak{Q}}$, which completes the proof.

We interrupt the execution of EEA at its middle point, i.e. for the least index j where $\mathrm{N}_{\mathbb{Q}(i)/\mathbb{Q}}(d_j) < \sqrt{\mathrm{N}_{\mathbb{Q}(i)/\mathbb{Q}}(q)}$. As in the classical variant of NFS, we make the heuristic that

$$\forall k \in [1, 4] \qquad \left\| (a^{(k)}, b^{(k)}) \right\|_\infty \approx \sqrt{|q|}.$$

Hence, we replaced Step 1 in Algorithm 1 by EEA in $\mathbb{Z}[i]$.

Another advantage of $\mathbb{Z}[i]$ is that we can easily deal with the roots of unity. Indeed, the roots of unity have a bad effect on the sieve since, for any pairs (a, b) found during the sieve, one will also find (ua, ub) for all roots of unity u. For a practical implementation one might prefer to choose h so that there are no roots of unity other than ± 1.

In the case $h = x^2 + 1$, we can impose that we have no duplicates due to roots of unity. For this, we modify Step 2 of Algorithm 1 so that the indices run in

$$(i_1, i_2, i_3, i_4) \in [0, I] \times [0, I] \times [-I, I] \times [-I, I]$$

instead of $[-I, I]^4$. By doing so we divide by four the number of pairs (a, b) sieved in the special-q task associated to $\mathfrak{Q}$. Indeed, if a pair (a, b) is written as $(a, b) = \sum_{k=1}^4 i_k (a^{(k)}, b^{(k)})$, then when we multiply (a, b) by roots of unity we use indices:

$$\begin{array}{ll} (a, b) & \leftrightarrow (i_1, i_2, i_3, i_4) \\ (-a, -b) & \leftrightarrow (-i_1, -i_2, -i_3, -i_4) \\ (ia, ib) & \leftrightarrow (-i_2, i_1, -i_4, i_3) \\ (-ia, -ib) & \leftrightarrow (i_2, -i_1, i_4, -i_3). \end{array}$$

Exactly one of the pairs has $i_1, i_2 \geq 0$.

7.2 Using a cyclotomic field for $\mathbb{Q}(\iota)$

Although we found no practical advantage for cyclotomic fields other than $\mathbb{Q}(i)$, they allow us to give a poof of existence for the polynomial h, as required in the TNFS construction of Section 3.1.

Theorem 9 ([1], Prop. 3) *Assuming the Extended Riemann Hypothesis (ERH), there is a constant $c > 0$, such that for all $p, n \in \mathbb{N}$, p prime and $\gcd(n,p) = 1$, there exists a prime q such that $q \equiv 1 \pmod{n}$, $q < cn^4 \log(pn)^2$ and p is inert in the unique subfield K of $\mathbb{Q}(\zeta_q)$ with $[K : \mathbb{Q}] = n$.*

Corollary 10 *Under ERH, there exists a constant $c > 0$ such that, for any integer n and any prime $p > n$, there exists an effectively constructible polynomial $h \in \mathbb{Z}[x]$ such that:*

- *h is irreducible modulo p;*
- *$\|h\|_\infty < (2cn^4 \log(np)^2)^n$.*

Proof. Let c be the constant of the theorem above. Let q a prime associated with p and n and let ζ_q be a primitive qth root of unity and η a Gaussian period:

$$\eta = \sum_{x \in \mathbb{F}_q^* / (\mathbb{F}_q^*)^{(q-1)/n}} \zeta_q^x.$$

If $r_1, \ldots, r_n$ are a set of representatives of $\mathbb{F}_q^* / (\mathbb{F}_q^*)^{(q-1)/n}$, then the conjugates of η are its images by the morphisms $\sigma_i : \zeta_q \mapsto \zeta_q^{r_i}$. Hence, the minimal polynomial of η over $\mathbb{Q}$ is

$$h = \prod_{i=0}^{n-1} (x - \sigma_i(\eta)).$$

For $k \in [0, n]$, a crude estimate of the kth coefficient of f is $\binom{n}{k} |\eta|^k$, which is further upper bounded by $2^n (q-1)^n$, and finally by $(2cn^4 \log(np)^2)^n$. The coefficients of h add a factor $\|h\|_\infty^{n(d-1)}$ in Equation (1). It remains negligible, i.e. $L_Q(2/3)^{o(1)}$, when $n^2 = o(d)$ or equivalently when $p = L_Q(\alpha)$ for $\alpha > 5/6$.

7.3 The Waterloo improvement

At the beginning of the individual logarithm stage, the smoothing step can be sped up in practice using the continued fraction method, also called "Waterloo improvement"[4]. It allows to replace the probability of an integer of size S to be smooth by the probability of two numbers of size $\sqrt{S}$ to be simultaneously smooth. This does not change the complexity, unless we make the $o(1)$ expression explicit, but has a measurable practical impact. A TNFS equivalent for the continued-fraction method is to LLL-reduce the lattice generated by the rows of the matrix

$$M(z) = \left(\begin{array}{ccc|ccc} p & & & & & \\ & \ddots & & & 0 & \\ & & p & & & \\ \hline & z & & 1 & & \\ & \vdots & & & \ddots & \\ & \iota^{n-1} z & & & & 1 \end{array}\right),$$

[4] The name, coined by Coppesmith, makes reference to the group who first used this technique [11].

where z is a lift of the target element of the finite field, and z, …, $\iota^{n-1}z$ are written by their coordinates as elements of $\mathbb{Q}(\iota)$. Since $\det M(z) = p^n = Q$, a short vector $(u_0, \dots, u_{n-1}, v_0, \dots, v_{n-1})$ has coordinates of size $\approx Q^{1/2n}$. The quotient u/v where $u = \sum_{k=0}^{n-1} u_k \iota^k$ and $v = \sum_{k=0}^{n-1} v_k \iota^k$ is an element of $\mathbb{Q}(\iota)$ that reduces to the same element of $\mathbb{F}_{p^n}$ as z. Therefore, instead of testing for smoothness the norm of z, of size $S = Q$, we test whether the norms of u and v, both of size $\sqrt{Q}$, are smooth.

References

1. Leonard M Adleman and Hendrik W Lenstra. Finding irreducible polynomials over finite fields. In *Proceedings of the eighteenth annual ACM symposium on Theory of computing*, pages 350–355. ACM, 1986.
2. Kazumaro Aoki, Jens Franke, Thorsten Kleinjung, Arjen K Lenstra, and Dag Arne Osvik. A kilobit special number field sieve factorization. In *Advances in Cryptology–ASIACRYPT 2007*, pages 1–12. Springer, 2007.
3. Shi Bai. *Polynomial selection for the number field sieve.* PhD thesis, Australian National Univers., 2011.
4. Shi Bai, Cyril Bouvier, Alexander Kruppa, and Paul Zimmermann. Better polynomials for GNFS. Preprint, 2014.
5. Razvan Barbulescu. *Algorithmes de logarithmes discrets dans les corps finis.* PhD thesis, Université de Lorraine, 2013.
6. Razvan Barbulescu, Pierrick Gaudry, Aurore Guillevic, and François Morain. (Algebraic) improvements to the number field sieve for non-prime finite fields. Preprint available at `http://hal.inria.fr/hal-01052449`.
7. Razvan Barbulescu, Pierrick Gaudry, Aurore Guillevic, and François Morain. Improving NFS for the discrete logarithm problem in non-prime finite fields. In *Advances in Cryptology - EUROCRYPT 2015*, volume 9056 of *Lecture Notes in Comput. Sci.*, pages 129–155. Springer, 2015.
8. Razvan Barbulescu and Cécile Pierrot. The multiple number field sieve for medium- and high-characteristic finite fields. *LMS Journal of Computation and Mathematics*, 17:230–246, 1 2014.
9. Paulo S.L.M. Barreto and Michael Naehrig. Pairing-friendly elliptic curves of prime order. In *Selected areas in cryptography*, volume 3897 of *Lecture Notes in Comput. Sci.*, pages 319–331. Springer, 2006.
10. Yuval Bistritz and Alexander Lifshitz. Bounds for resultants of univariate and bivariate polynomials. *Linear Algebra and its Applications*, 432(8):1995–2005, 2010.
11. I. F. Blake, R. Fuji-Hara, R. C. Mullin, and S. A. Vanstone. Computing logarithms in finite fields of characteristic two. *SIAM Journal on Algebraic Discrete Methods*, 5(2):276–285, 1984.
12. Friederike Brezing and Annegret Weng. Elliptic curves suitable for pairing based cryptography. *Designs, Codes and Cryptography*, 37(1):133–141, 2005.
13. Joe P. Buhler, Hendrik W. Lenstra, Jr, and Carl Pomerance. Factoring integers with the number field sieve. In Arjen. Lenstra and Hendrik. Jr. Lenstra, editors, *The development of the number field sieve*, volume 1554 of *Lecture Notes in Math.*, pages 50–94. Springer Berlin Heidelberg, 1993.
14. Henri Cohen. *Advanced topics in computational number theory*, volume 193 of *Graduate Texts in Mathematics.* Springer, 2000.

15. An Commeine and Igor 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.
16. Don Coppersmith. Modifications to the number field sieve. *J. of Cryptology*, 6(3):169–180, 1993.
17. Kurt Foster. HT90 and "simplest" number fields. *Illinois J. Math.*, 55(4):1621–1655, 2011.
18. David Freeman, Michael Scott, and Edlyn Teske. A taxonomy of pairing-friendly elliptic curves. *J. of Cryptology*, 23(2):224–280, 2010.
19. Daniel M. Gordon. Discrete logarithms in GF(p) using the number field sieve. *SIAM Journal on Discrete Mathematics*, 6(1):124–138, 1993.
20. A. Joux and R. Lercier. The function field sieve is quite special. In *Algorithmic Number Theory–ANTS V*, volume 2369 of *Lecture Notes in Comput. Sci.*, pages 431–445. Springer, 2002.
21. Antoine Joux, Reynald Lercier, Nigel Smart, and Frederik Vercauteren. The number field sieve in the medium prime case. In *Advances in Cryptology–CRYPTO 2006*, volume 4117 of *Lecture Notes in Comput. Sci.*, pages 326–344. Springer, 2006.
22. Antoine Joux and Cécile Pierrot. The special number field sieve in $\mathbb{F}_{p^n}$, Application to pairing-friendly constructions. In *Pairing-Based Cryptography – Pairing 2013*, volume 8365 of *Lecture Notes in Comput. Sci.*, pages 45–61, 2013.
23. Thorsten Kleinjung. On polynomial selection for the general number field sieve. *Mathematics of Computation*, 75(256):2037–2047, 2006.
24. Thorsten Kleinjung. Polynomial selection, 2008. Slides at CADO workshop, available online at `http://cado.gforge.inria.fr/workshop/slides/kleinjung.pdf`.
25. Thorsten Kleinjung, Kazumaro Aoki, Jens Franke, K. Arjen Lenstra, Emmanuel Thomé, W. Joppe Bos, Pierrick Gaudry, Alexander Kruppa, L. Peter Montgomery, Dag Arne Osvik, Hermann te riele, Andrey Timofeev, and Paul Zimmermann. Factorization of a 768–bit RSA modulus. In *Advances in Cryptology–CRYPTO 2010*, volume 6223 of *Lecture Notes in Comput. Sci.*, pages 333–350. Springer, 2010.
26. Thorsten Kleinjung, Joppe W Bos, and Arjen K Lenstra. Mersenne factorization factory. In *Advances in Cryptology–ASIACRYPT 2014*, volume 8873 of *Lecture Notes in Comput. Sci.*, pages 358–377. Springer, 2014.
27. Arjen Lenstra, Hendrik Lenstra, Mark Manasse, and John Pollard. The number field sieve. In *The development of the number field sieve*, volume 1554 of *Lecture Notes in Math.*, pages 11–42. Springer, 1993.
28. Dmitri V. Matyukhin. On asymptotic complexity of computing discrete logarithms over GF(p). *Discrete Mathematics and Applications*, 13(1):27–50, 2003.
29. Atsuko Miyaji, Masaki Nakabayashi, and Shunzou Takano. New explicit conditions of elliptic curve traces for FR-reduction. *IEICE Trans. on Fundamentals of Electronics, Communic. and Computer Sci.*, 84(5):1234–1243, 2001.
30. Angela Murphy and Noel Fitzpatrick. Elliptic curves for pairing applications. Cryptology ePrint Archive, Report 2005/302, 2005. `http://eprint.iacr.org/`.
31. Cécile Pierrot. The multiple number field sieve with Conjugation and Generalized Joux-Lercier methods. In *Advances in Cryptology - EUROCRYPT 2015*, volume 9056 of *Lecture Notes in Comput. Sci.*, pages 156–170. Springer, 2015.
32. John M. Pollard. The lattice sieve. In *The development of the number field sieve*, volume 1554 of *Lecture Notes in Math.*, pages 43–49. Springer, 1993.
33. Oliver Schirokauer. Discrete logarithms and local units. *Philos. Trans. Roy. Soc. London Ser. A*, 345(1676):409–423, 1993.

34. Oliver Schirokauer. Using number fields to compute logarithms in finite fields. *Math. Comp.*, 69(231):1267–1283, 2000.
35. Igor Semaev. Special prime numbers and discrete logs in finite prime fields. *Math. Comp.*, 71(237):363–377, 2002.
36. Douglas H. Wiedemann. Solving sparse linear equations over finite fields. *IEEE Trans. Inform. Theory*, 32(1):54–62, 1986.