1-1-2019

A unified polynomial selection method for the (Tower) number field sieve algorithm

Palash Sarkar
Indian Statistical Institute, Kolkata

Shashank Singh
Indian Institute of Science Education and Research Bhopal

Advances in Mathematics of Communications
Volume 13, No. 3, 2019, 435–455
doi:10.3934/amc.2019028

A UNIFIED POLYNOMIAL SELECTION METHOD FOR THE (TOWER) NUMBER FIELD SIEVE ALGORITHM

Palash Sarkar

Indian Statistical Institute
Kolkata 700108, West Bengal, India

Shashank Singh*

Indian Institute of Science Education and Research Bhopal
Bhopal 462066, Madhya Pradesh, India

(Communicated by Alfred Menezes)

Abstract. At Eurocrypt 2015, Barbulescu et al. introduced two new methods of polynomial selection, namely the Conjugation and the Generalised Joux-Lercier methods, for the number field sieve (NFS) algorithm as applied to the discrete logarithm problem over finite fields. A sequence of subsequent works have developed and applied these methods to the multiple and the (extended) tower number field sieve algorithms. This line of work has led to new asymptotic complexities for various cases of the discrete logarithm problem over finite fields. The current work presents a unified polynomial selection method which we call Algorithm $\mathcal{D}$. Starting from the Barbulescu et al. paper, all the subsequent polynomial selection methods can be seen as special cases of Algorithm $\mathcal{D}$. Moreover, for the extended tower number field sieve (exTNFS) and the multiple extended TNFS (MexTNFS), there are finite fields for which using the polynomials selected by Algorithm $\mathcal{D}$ provides the best asymptotic complexity. Suppose $Q = p^n$ for a prime p and further suppose that $n = \eta\kappa$ such that there is a $c_\theta > 0$ for which $p^\eta = L_Q(2/3, c_\theta)$. For $c_\theta > 3.39$, the complexity of exTNFS-$\mathcal{D}$ is lower than the complexities of all previous algorithms; for $c_\theta \notin (0, 1.12) \cup [1.45, 3.15]$, the complexity of MexTNFS-$\mathcal{D}$ is lower than that of all previous methods.

1. Introduction

One of the important problems in cryptography is to compute discrete logarithms over the multiplicative group of a finite field. Let p be a prime, $n \geq 1$ be an integer and $Q = p^n$. Suppose that $\mathfrak{g}$ is a generator of the non-zero elements of the finite field $\mathbb{F}_Q$. The discrete logarithm problem in $\mathbb{F}_Q^*$ (loosely speaking, one talks about the discrete logarithm problem in $\mathbb{F}_Q$) is the following. Given $\mathfrak{g}$ and a non-zero element $\mathfrak{h}$ of $\mathbb{F}_Q$, compute a such that $\mathfrak{g}^a = \mathfrak{h}$. Here a is called the discrete logarithm of $\mathfrak{h}$ to the base $\mathfrak{g}$.

There are two known general approaches to this problem which lead to sub-exponential run-times. These are the function field sieve (FFS) [1, 2, 18, 20] and

2010 *Mathematics Subject Classification:* Primary: 11Y16; Secondary: 94A60.
Key words and phrases: Finite fields, discrete logarithm, number field sieve, tower number field sieve, multiple tower number field sieve.

* Corresponding author: Shashank Singh.

the number field sieve (NFS) [11, 19, 21] algorithms. Suppose that $p = L_Q(a, c_p)$ where

$$L_Q(a, c_p) = \exp\left((c_p + o(1))(\ln Q)^a (\ln \ln Q)^{1-a}\right).$$

Depending on the value of a, fields $\mathbb{F}_Q$ are classified into the following types: small characteristic, if $a < 1/3$; medium characteristic, if $1/3 \leq a < 2/3$; boundary, if $a = 2/3$; and large characteristic, if $a > 2/3$. The case $a = 2/3$ has been singled out as a boundary case in [4] since it is possible to show that the best complexity for this case is lower than the best complexities for the medium characteristic and the large characteristic cases. For $a = 1/3$, on the other hand, no such complexity improvement is known.

There has been tremendous progress in the FFS algorithm leading to a quasi-polynomial time algorithm [5, 23] for the small characteristic case. Using algorithms given in [17, 5], a record computation of discrete log in the binary extension field $\mathbb{F}_{2^{9234}}$ was reported by Granger et al. [12]. The FFS algorithm also applies to the medium prime case and this has been reported in [20, 16, 29].

The application of NFS to compute discrete logarithms over finite fields was first proposed by Gordon [11] for prime order fields, i.e., for $n = 1$. Application to composite order fields, i.e., for $n > 1$, was shown by Schirokauer [33]. Important improvements to the NFS for prime order fields were given by Joux and Lercier [19]. Joux, Lercier, Smart and Vercauteren [21] showed that the NFS algorithm is applicable to all finite fields. When the prime p is of a special form, Joux and Pierrot [22] showed the application of the special number field sieve algorithm to obtain improved complexity. Use of multiple number fields to obtain faster asymptotic complexity was proposed by Barbulescu and Pierrot in [7].

In an influential work, Barbulescu et al. [4] introduced two new methods for polynomial selection for NFS, namely, the generalised Joux-Lercier (GJL) and the Conjugation methods. Using these algorithms, for the boundary case, the best complexity obtained was $L_Q(1/3, (48/9)^{1/3})$; the best complexities obtained for the medium and the large prime cases were $L_Q(1/3, (96/9)^{1/3})$ and $L_Q(1/3, (64/9)^{1/3})$ respectively. Pierrot [28] derived the asymptotic complexity of the multiple NFS (MNFS) for the GJL and the Conjugation methods and in all cases obtained lower values of the second term in the corresponding sub-exponential expressions. Sarkar and Singh [31] provided an algorithm for polynomial selection which both generalised and subsumed the GJL and the Conjugation methods. The asymptotic complexities for both NFS and MNFS were derived. For $p = L_Q(2/3, c_p)$, there are ranges of values for c_p, where complexities obtained by the method of [31] is lower than those obtained in [4] and [28].

The tower number field sieve algorithm had initially been proposed by Schirokauer [33]. Barbulescu et al. [6] provided a detailed analysis of this algorithm. Subsequently, Kim and Barbulescu [24] combined previous polynomial selection methods with the extended TNFS (exTNFS) algorithm to obtain improved complexities for the medium prime case when the extension degree n is composite and not a prime-power. Complexities of the variants using multiple tower number fields and special number fields were derived. Sarkar and Singh [30] proposed an algorithm for polynomial selection which provided improved complexities for the medium prime case for all composite n. This was followed by a paper by Jeong and Kim [25] who showed how to combine the Conjugation method with exTNFS to cover all composite extension degrees and improve the complexity for the medium prime case over what was obtained in [30].

Practical issues in relation collection were considered in Gaudry et al. [10]. Using the Conjugation method for selecting polynomials, Guillevic et al. [14] reported a computation of discrete logarithm on an 170-bit MNT curve.

The improved complexities of the various versions of the tower number field sieve algorithm impacts the choice of key sizes in pairing based cryptography. Menezes, Sarkar and Singh [27] provide a concrete analysis of the various methods with the goal of determining key sizes suitable for implementing bilinear pairings at the 128-bit and the 192-bit security levels. Updated estimates of secure key sizes for pairings have been proposed by Barbulescu and Duquesne [3].

Our contributions. In the (tower) NFS algorithm, there are two number fields which are represented by two polynomials $f(x)$ and $g(x)$ satisfying certain restrictions. The generalisation of using multiple number fields requires a set of polynomials to represent the different number fields. The overall complexity of the algorithm is determined by the choice of the polynomials, in particular their degrees and their infinity norms (i.e., the maximum of the absolute values of the coefficients). The lower the value of the degrees and the infinity norms, the faster the overall time complexity. It is, however, difficult to ensure that both the degrees and the infinity norms are simultaneously low. A number of algorithms have been proposed in the literature for polynomial selection providing various trade-offs between the degrees and the infinity norms.

In this paper, we present a new polynomial selection algorithm which we call Algorithm $\mathcal{D}$. We show that the following previously proposed methods can be seen as special cases of Algorithm $\mathcal{D}$.

- The Conjugation and the Generalised Joux-Lercier (GJL) algorithms by Barbulescu et al. [4]
- Algorithm $\mathcal{A}$ by Sarkar and Singh [31]. (Algorithm $\mathcal{A}$ subsumes the Conjugation and the GJL methods.)
- The generalised Conjugation method of Jeong and Kim [25]. (The method of Jeong and Kim subsumes the polynomial selection algorithm by Kim and Barbulescu [24].)

Additionally, we show that Algorithm $\mathcal{D}$ can be instantiated to obtain a variant of the GJL algorithm which works in the setting of extended TNFS. Algorithm $\mathcal{D}$ provides polynomials to represent two number fields. Using previously proposed ideas [7, 28], we show how to extend this to obtain multiple polynomials so that the multiple number fields can be used.

In terms of complexity[1], since the best previous polynomial selection algorithms are special cases of Algorithm $\mathcal{D}$, the complexities achieved using polynomials obtained from Algorithm $\mathcal{D}$ are never greater than what has been previously achieved. Below we highlight the cases where improved complexities for certain medium prime cases are obtained using polynomials selected by Algorithm $\mathcal{D}$.

Let $p = L_Q(a, c_p)$ with $1/3 < a \leq 2/3$ and $n = \eta\kappa$ be such that there is a $c_\theta > 0$ with $p^\eta = L_Q(2/3, c_\theta)$. The expression for the complexity obtained in [24, 25] is a function of c_θ. The minimum value of the complexity (as a function of c_θ) is the best known complexity for the medium prime case and is achieved only for a particular value of c_θ.

1. Using two number fields, the best known complexity for the medium prime case is $L_Q(1/3, (48/9)^{1/3})$ [24, 25] This complexity is achieved only for $c_\theta =$

[1]As in previous works, the running time estimates given in this paper are heuristic.

$12^{1/3}$. Algorithm $\mathcal{D}$ does not lower this complexity. Rather, we are able to show that for $c_\theta > 3.39$, the complexity achieved using polynomials obtained from Algorithm $\mathcal{D}$ is lower than the complexity of all previous algorithms using two number fields.

2. In the case of multiple number fields, the best known complexity for the medium prime case is $L_Q(1/3, 1.71)$ [24, 25] and is achieved only for $c_\theta = 2.123$. Again, Algorithm $\mathcal{D}$ does not lower this complexity and instead we are able to show that for $c_\theta \notin (0, 1.12) \cup [1.45, 3.15]$, the complexity achieved using polynomials obtained from Algorithm $\mathcal{D}$ is lower than that of all previous methods.

2. Basics of the tower number field sieve algorithm

Index calculus algorithms for computing discrete logarithms in a group have a general structure. A small subset of elements of the group is identified and called the factor basis. The first phase of the algorithm consists of finding linear relations between the discrete logarithms of the elements of the factor basis. This provides a system of linear equations among the discrete logarithms of the elements of the factor basis. Usually this system of linear equations turn out to be quite sparse. The second phase consists of using linear algebra techniques to solve the system of linear equations. This phase provides the discrete logarithms of the elements of the factor basis. In the third phase, the target element is decomposed over the factor basis. Using the already computed values of the discrete logarithms of the factor basis elements, this decomposition allows computing the discrete logarithm of the target element. The relation collection and the individual logarithm phases depend on the underlying group and the particular index calculus algorithm. On the other hand, the linear algebra phase is the same for all index calculus algorithms.

Typically, the linear algebra phase is performed modulo one (or, a few) large prime factor of the order of the group. This yields the discrete logarithm of the target element modulo this prime. The discrete logarithm of the target element modulo the smaller factors of the order of the group is computed using the Pohlig-Hellman and Pollard rho algorithms. The Chinese Remainder Theorem is used to combine the discrete logarithms modulo the different factors to obtain the discrete logarithm modulo the order of the group.

The TNFS algorithm is an index calculus algorithm for computing discrete logarithms over a finite field. The algorithm, though, does not directly work over a finite field. Instead, it starts with two (or more) appropriate number fields and uses suitable homomorphisms to map to the desired finite field. Consequently, the identification of the factor basis and the associated relation collection and individual logarithm phases are most conveniently expressed in terms of the background number fields.

Below we provide an overview of the TNFS algorithm. More detailed descriptions can be found in [24, 6]. The TNFS algorithm applies to fields $\mathbb{F}_Q$ where $Q = p^n$, p is a prime and $n = \eta\kappa$ is a factorisation of n. Depending on the values of η and κ, several variants are obtained.

- If $\eta = 1$ (and $\kappa = n$) then this is the classical NFS algorithm.
- If $\eta = n$ (and $\kappa = 1$) then the variant proposed by Barbulescu et al. in [6] is obtained where this case was called TNFS.
- If $1 < \eta, \kappa < n$ then the variant proposed in [24] is obtained where this case was called exTNFS.

Let $h(z)$ be a monic polynomial with integer coefficients and of degree η which is irreducible over both $\mathbb{Z}$ and $\mathbb{F}_p$. Let $R := \mathbb{Z}[z]/(h(z))$. Note that $\mathbb{F}_{p^\eta} = \mathbb{F}_p[z]/(h(z))$. Let $f(x)$ and $g(x)$ be polynomials in $R[x]$ satisfying the following properties.

1. Both $f(x)$ and $g(x)$ are irreducible over R.
2. Over $\mathbb{F}_{p^\eta}$, $f(x)$ and $g(x)$ have a common factor $\varphi(x)$ of degree κ.

The field $\mathbb{F}_{p^n}$ is realised as $\mathbb{F}_{p^\eta}[x]/(\varphi(x)) = (R/pR)[x]/(\varphi(x))$.

The description of TNFS algorithm involves a tower of number fields as shown in Figure 1, where $K_h := \mathbb{Q}[z]/(h(z))$ and K_f and K_g are the field extensions of K_h defined by the polynomials $f(x)$ and $g(x)$ respectively.

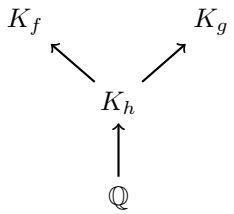


Figure 1. Tower of Number Fields

The polynomials $h(z) \in \mathbb{Z}[z]$ and $f(x)$, $g(x) \in R[x]$ provide two different homomorphisms to move from $R[x]$ to the field $\mathbb{F}_{p^n}$. This is shown in the commutative diagram of Figure 2.

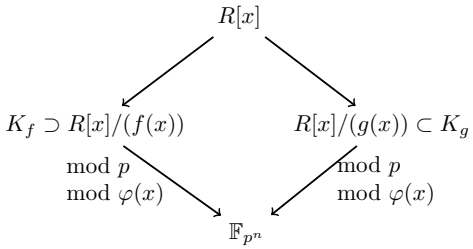


Figure 2. Commutative diagram for TNFS

Let α_f (resp. α_g) be a root of $f(x)$ (resp. $g(x)$) in K_f (resp. K_g). Let $\mathcal{O}_f$ (resp. $\mathcal{O}_g$) be the ring of integers of the number field K_f (resp. K_g). The factor basis consists of two parts, one corresponding to K_f and the other corresponding to K_g. Let $\phi(x) \in R[x]$ be of degree less than t. The case $t = 2$ was expliticly considered in [6, 24] and the appendix of [24] briefly considers the case $t > 2$. A single relation is obtained from the factorisations of the principal ideals $\phi(\alpha_f)\mathcal{O}_f$ and $\phi(\alpha_g)\mathcal{O}_g$ in $\mathcal{O}_f$ and $\mathcal{O}_g$ respectively when both the norms are B-smooth for a suitably chosen smoothness bound B. So, the factor basis consists of the prime ideals of K_f and K_g which can occur in the factorisations of $\phi(\alpha_f)\mathcal{O}_f$ and $\phi(\alpha_g)\mathcal{O}_g$ respectively when both of these norms are B-smooth.

The explicit form of the factor basis for the case of $t = 2$ has been provided in [6, 24] and we mention this below. The factor basis is $\mathcal{F}$ which can be written

as $\mathcal{F} = \mathcal{F}_f \cup \mathcal{F}_g$. The definition of $\mathcal{F}_f$ is given below and the definition of $\mathcal{F}_g$ is obtained by replacing f with g.

$$\mathcal{F}_f = \left\{ \langle \mathfrak{q}, \alpha_f - \gamma \rangle : \begin{array}{l} \mathfrak{q} \text{ is a prime in } K_h \text{ lying above a prime } q < B, \\ f(\gamma) \equiv 0 \bmod \mathfrak{q} \end{array} \right\} \bigcup \{\text{prime ideals of } K_f \text{ dividing } l(f)\mathrm{Disc}(f)\}$$

where $l(f)$ denotes the leading coefficient of $f(x)$ and $\mathrm{Disc}(f)$ denotes the discriminant of $f(x)$.

It has been shown in [6] that the cardinality of the factor basis is given as follows:

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

In the asymptotic analysis, this is taken to be $B^{1+o(1)}$.

For $t > 2$, a suitable factor basis can be defined and for a fixed value of t, the asymptotic expression for $\#\mathcal{F}$ remains $B^{1+o(1)}$.

We next consider the manner in which a relation is generated. Let $\phi(x)$ be a polynomial in $R[x]$ of degree at most $t-1$. Suppose that the principal ideals $\phi(\alpha_f)\mathcal{O}_f$ and $\phi(\alpha_g)\mathcal{O}_g$ factor over $\mathcal{F}_f$ and $\mathcal{F}_g$ respectively giving rise to the following relations:

$$\phi(\alpha_f)\mathcal{O}_f = \prod_{\mathfrak{l} \in \mathcal{F}_f} \mathfrak{l}^{\mathrm{val}_{\mathfrak{l}}(\phi(\alpha_f))} \text{ and } \phi(\alpha_g)\mathcal{O}_g = \prod_{\mathfrak{l} \in \mathcal{F}_g} \mathfrak{l}^{\mathrm{val}_{\mathfrak{l}}(\phi(\alpha_g))}. \tag{2}$$

These two relations can then be converted into a linear relation. The method for doing this is described in Section 4.3 of [21] which has later been summarised as Theorem 4 in [6]. As shown in Figure 2, starting from $\phi(x)$ there are two different homomorphisms which lead to the same element of the field $\mathbb{F}_{p^n}$. Combining these homomorphisms with the factorisations of the ideals $\phi(\alpha_f)\mathcal{O}_f$ and $\phi(\alpha_g)\mathcal{O}_g$ provides two different factorisations of an element of $\mathbb{F}_{p^n}$. The details of obtaining such a relation involves the notion of virtual logarithms [34, 21] and Schirokauer maps [32]. Following the exposition of the technique in [21, 6, 24] a linear relation of the following form is obtained:

$$\begin{aligned} &\sum_{\mathfrak{l} \in \mathcal{F}_f} \mathrm{val}_{\mathfrak{l}}(\phi(\alpha_f)) \log_f(\mathfrak{l}) + \sum_{j=1}^{r_1} \lambda_{f,j}(\phi(\alpha_f))\chi_{f,j} \\ &\equiv \sum_{\mathfrak{l} \in \mathcal{F}_g} \mathrm{val}_{\mathfrak{l}}(\phi(\alpha_g)) \log_f(\mathfrak{l}) + \sum_{j=1}^{r_2} \lambda_{g,j}(\phi(\alpha_g))\chi_{g,j} \pmod{\ell}. \end{aligned} \tag{3}$$

Here ℓ is a large prime factor of $p^n - 1$ such that the discrete logarithm is desired to be computed modulo ℓ; $\lambda_{f,j}$ and $\lambda_{g,j}$ arise from Schirokauer maps defined modulo the ℓ-th power of the units; r_1 and r_2 are the unit ranks of $\mathcal{O}_f$ and $\mathcal{O}_g$ respectively; $\log_f$ (resp. $\log_g$) is a map from ideals of $\mathcal{O}_f$ (resp. $\mathcal{O}_g$) to $\mathbb{Z}/\ell\mathbb{Z}$; and $\chi_{f,j} : \{1, \ldots, r_1\} \to \mathbb{Z}/\ell\mathbb{Z}$ and $\chi_{g,j} : \{1, \ldots, r_2\} \to \mathbb{Z}/\ell\mathbb{Z}$ are called virtual logarithms.

The main task of the relation collection phase is to obtain polynomials $\phi(x)$ which give rise to relations of the type given by (2). For this, it is sufficient to ensure that the norms $N(f, \phi) := \mathrm{N}_{K_f/\mathbb{Q}}(\phi(\alpha_f))$ and $N(g, \phi) := \mathrm{N}_{K_g/\mathbb{Q}}(\phi(\alpha_g))$ are both B-smooth. As mentioned in [24], the norms can be expressed in terms of

resultants as follows:

$$N(f,\phi) = \mathrm{Res}_z(\mathrm{Res}_x(\phi(x), f(x)), h(z))$$
$$N(g,\phi) = \mathrm{Res}_z(\mathrm{Res}_x(\phi(x), g(x)), h(z)) \tag{4}$$

where Res denotes the resultant.

The polynomials $\phi(x) \in R[x]$ are chosen with the restriction $\|\phi\|_\infty = E^{2/(\eta t)}$ for an appropriate choice of E. This ensures that the number of polynomials $\phi(x)$ considered in the relation collection phase is E^2. The time spent per polynomial depends on whether a sieving technique is used or whether each norm is tested for smoothness using Elliptic Curve factoring Method (ECM). Asymptotically, the second cost would be greater and from the discussion in Section 3 of [24], the time spent per polynomial turns out to be $B^{o(1)}$. So, the total cost of the relation collection phase is $E^2 B^{o(1)}$. Choosing E to be equal to B, this cost comes out to be $B^{2+o(1)}$. For more details on sieving, we refer to [10, 15, 36].

A little more than $\#\mathcal{F} + r_1 + r_2 = B^{1+o(1)}$ relations of type (3) are collected. The resulting system of linear equations is solved using the block Wiedemann [35] algorithm to obtain the logarithms of the factor basis elements modulo the prime ℓ. The cost of the linear algebra stage is $B^{2+o(1)}$. Due to the choice $E = B$, this cost is equal to the cost of relation collection.

The discrete logarithm of the target element is computed in the individual discrete logarithm phase of the algorithm. The idea is the following. The target element is lifted to one of the number fields, which is then written in terms of the element of factor basis using the recursive procedures outlined in [21, 24, 13]. The discrete logarithm of the target element can then be expressed in terms of logarithms of the factor basis elements. Substituting these values, which have already been obtained from the linear algebra step, the discrete logarithm of the desired element is obtained. Asymptotically, the cost of the individual discrete logarithm phase is dominated by the costs of the linear algebra and the relation collection phases and so this phase is not considered in the asymptotic complexity analysis of the algorithm.

The sizes of the norms $N(f,\phi)$ and $N(g,\phi)$ can be upper bounded using known upper bounds on resultants [8]. Let $\mathfrak{f}(z,x)$ be a bivariate polynomial with integer coefficients where $\mathfrak{f}_{i,j}$ is the coefficient of $x^i z^j$. Then $\|\mathfrak{f}\|_\infty = \max|\mathfrak{f}_{i,j}|$. Recall that $h(z)$ is a monic polynomial of degree η and let $H = \|h\|_\infty$. Let $\mathfrak{f}(x) \in R[x]$ so that $\deg_z f = \eta - 1$. Also, $\phi(x) \in R[x]$ is of degree $t-1$ implying that $\deg_z \phi = \eta - 1$, $\deg_x \phi = t - 1$. Further, as mentioned above $\|\phi\|_\infty = E^{2/(\eta t)}$. Then, using the bounds on resultants from [8] the following upper bound on $|\mathrm{Res}_z(\mathrm{Res}_x(\phi(x), \mathfrak{f}(x)), h(z))|$ is obtained in Appendix **A** of [27]:

$$|\mathrm{Res}_z(\mathrm{Res}_x(\phi(x), \mathfrak{f}(x)), h(z))| \leq \mathfrak{C}(\eta, t, \deg_x \mathfrak{f}, H) \times E^{(2\deg_x \mathfrak{f})/t} \times \|\mathfrak{f}\|_\infty^{\eta(t-1)} \tag{5}$$

where

$$\begin{aligned}\mathfrak{C}(\eta,t,s,H) &= ((\eta-1)(t+s-1)+1)^{\eta/2}(\eta+1)^{(\eta-1)(t+s-1)/2} H^{(\eta-1)(t+s-1)} \\ &\quad \times \left((t+s-1)!\,\eta^{t+s-2}\right)^{\eta}.\end{aligned} \tag{6}$$

Applying (5) with $\mathfrak{f} = f$ and $\mathfrak{f} = g$ in succession and noting that $N(f,\phi)$ and $N(g,\phi)$ are given by (4), we obtain

$$|N(f,\phi)| \leq \mathfrak{C}(\eta, t, \deg_x f, H) \times E^{(2\deg_x f)/t} \times \|f\|_\infty^{\eta(t-1)}; \tag{7}$$

$$|N(g,\phi)| \leq \mathfrak{C}(\eta, t, \deg_x g, H) \times E^{(2\deg_x g)/t} \times \|g\|_\infty^{\eta(t-1)}. \tag{8}$$

For a given B, ensuring the B-smoothness of norms (given by (4)), depend on the values of the norms. For the complexity analysis instead of the actual values of the norms, the upper bounds given by (7) and (8) are used. These values depend on the properties of polynomials $h(z)$, $f(x)$ and $g(x)$. In particular, $N(f,\phi)$ and $N(g,\phi)$ are determined by the infinity norms and the degrees of the polynomials $h(z)$, $f(x)$ and $g(x)$. The degree of $h(z)$ is η and it is usually possible to choose its infinity norm H to be small. So, the main task of reducing computational complexity boils down to choosing $f(x)$ and $g(x)$ having low infinity norms and low degrees. Simultaneously achieving both of these goals is difficult. As mentioned earlier, various algorithms have been proposed in the literature for choosing $f(x)$ and $g(x)$ offering different trade-offs between degrees and infinity norms.

3. A new polynomial selection method for exTNFS

The work [4] provides two methods for selecting polynomials for the classical NFS algorithm. These are called the generalised Joux-Lercier (GJL) and the Conjugation method. The GJL method is based on an earlier method due to Joux and Lercier [19] and uses the LLL algorithm to select polynomials.

The GJL matrix: Given a monic polynomial $\varphi(x) = \varphi_0 + \varphi_1 x + \cdots + \varphi_{k-1}x^{k-1} + x^k$ with integer coefficients and $r \geq k$, define an $(r+1)\times(r+1)$ matrix in the following manner:

$$M_{\varphi,r} = \begin{bmatrix} p & & & & & & \\ & \ddots & & & & & \\ & & \ddots & & & & \\ & & & p & & & \\ \varphi_0 & \varphi_1 & \cdots & \varphi_{k-1} & 1 & & \\ & \ddots & \ddots & & & \ddots & \\ & & \varphi_0 & \varphi_1 & \cdots & \varphi_{k-1} & 1 \end{bmatrix} \tag{9}$$

Apply the LLL algorithm to $M_{\varphi,r}$ and let the first row of the resulting LLL-reduced matrix be $[\psi_0, \ldots, \psi_r]$. This vector is taken to represent a polynomial $\psi(x) = \psi_0 + \psi_1 x + \cdots + \psi_r x^r$ and we write

$$\mathrm{LLL}(M_{\varphi,r}) = \psi(x) = \psi_0 + \psi_1 x + \cdots + \psi_r x^r \tag{10}$$

to denote the polynomial $\psi(x)$. The determinant of $M_{\varphi,r}$ is p^k and so by the LLL-reduced property [26], $\|\varphi\|_\infty = O(p^{k/(r+1)})$. If $Q = p^n$, then $\|\varphi\|_\infty = O(Q^{k/(n(r+1))})$.

Algorithm $\mathcal{D}$ describes the polynomial selection method for the extended TNFS. Apart from p, the other inputs are the factors η and κ of n, a divisor d of κ and an integer $r \geq \kappa/d$. The inputs d and r provide the flexibility whereby Algorithm $\mathcal{D}$ can be specialised to either the GJL or the Conjugation methods. We provide more details later.

The condition $\gcd(\eta, \kappa/d) = 1$ is required by Algorithm $\mathcal{D}$. The reason is the following. The polynomial $A_1(x)$ has integer entries and we wish to factorise $A_1(x)$ over $\mathbb{F}_p$ to obtain a factor $A_2(x)$ of degree k. This $A_2(x)$ is later used to define the polynomial $\varphi(x)$ which is required to be irreducible over $\mathbb{F}_{p^\eta}$. A necessary condition is that $A_2(x)$ must itself be irreducible over $\mathbb{F}_{p^\eta}$. Since $A_2(x)$ is a polynomial of

Algorithm: $\mathcal{D}$: Polynomial selection for TNFS.

Input: p, $n = \eta\kappa$, d (such that $d|\kappa$ and $\gcd(\eta, \kappa/d) = 1$) and $r \geq \kappa/d$.
Output: $h(z)$, $f(x)$, $g(x)$ and $\varphi(x)$.

Choose $h(z)$ to be a monic polynomial of degree η with small integer coefficients such that $h(z)$ is irreducible over $\mathbb{F}_p$;
let $k = \kappa/d$;
let $R = \mathbb{Z}[z]/(h(z))$;
let $\mathbb{F}_{p^\eta} = \mathbb{F}_p[z]/(h(z))$;
repeat
  randomly choose a monic polynomial $A_1(x) \in \mathbb{Z}[x]$ having the following properties:
    - $\deg A_1(x) = r + 1$;
    - $A_1(x)$ is irreducible over $\mathbb{Z}$;
    - $A_1(x)$ has coefficients of size $O(\ln(p))$;
    - over $\mathbb{F}_p$, $A_1(x)$ has a factor $A_2(x)$ of degree k such that $A_2(x)$ is irreducible over $\mathbb{F}_{p^\eta}$.

  randomly choose monic polynomials $C_0(x)$ and $C_1(x)$ in R such that $\|C_i\|_\infty$ is small for $i = 0, 1$; $\deg C_0(x) = d$ and $\deg C_1(x) < d$;
  define

$$f(x) = \mathrm{Res}_y\left(A_1(y), C_0(x) + y\, C_1(x)\right);$$
$$\varphi(x) = \mathrm{Res}_y\left(A_2(y), C_0(x) + y\, C_1(x)\right) \bmod p;$$
$$\psi(x) = \mathrm{LLL}(M_{A_2,r});$$
$$g(x) = \mathrm{Res}_y\left(\psi(y), C_0(x) + y\, C_1(x)\right).$$

until *$f(x)$ and $g(x)$ are irreducible over $\mathbb{Q}[z]/(h(z))$ (and hence over R) and $\varphi(x)$ is irreducible over $\mathbb{F}_{p^\eta} = \mathbb{F}_p[z]/(h(z))$.*

return $h(z)$, $f(x)$, $g(x)$ and $\varphi(x)$.

degree k with coefficients from $\mathbb{F}_p$ which is required to be irreducible over $\mathbb{F}_{p^\eta}$, it is necessary that $\gcd(\eta, k) = 1$. The condition $\gcd(\eta, \kappa/d) = 1$, however, does not restrict applicability. One can always choose $d = \kappa$ to obtain $k = 1$ and so $\gcd(\eta, \kappa/d) = 1$. Other values of d may also be appropriate, eg., if $\eta = 3$ and $\kappa = 4$, then one can choose $d = 2$.

Next we show how Algorithm $\mathcal{D}$ can be specialised to obtain the GJL and the Conjugation methods and their extensions.

1. If $\eta = 1$, $d = 1$, then we obtain the GJL method of [4], where different trade-offs are obtained by varying r; if $\eta = 1$, $d = \kappa = n$, then we obtain the Conjugation method of [4].
2. If $\eta = 1$, then we obtain Algorithm-$\mathcal{A}$ from [31].
3. If $d = \kappa$ and $r = 1$, then we obtain the method proposed by Jeong and Kim [25] which is essentially the Conjugation method in the exTNFS setting. Allowing $r > 1$ (or $d < \kappa$) provides a generalisation and leads to lower asymptotic complexity for certain ranges of finite fields.
4. If $\gcd(\eta, \kappa) = 1$, then choosing $d = 1$ and $r \geq \kappa$ provides the exTNFS-GJL algorithm. It is mentioned in [25] that combining their techniques with the GJL method gives rise to the exTNFS-GJL algorithm.

The following result states the basic properties of Algorithm $\mathcal{D}$. Bounds on the norms are obtained from the bounds on resultants given in [8] (see Section 2).

Proposition 1. *The outputs $f(x)$, $g(x)$ and $\varphi(x)$ of Algorithm $\mathcal{D}$ satisfy the following:*

1. $\deg(f) = d(r+1)$; $\deg(g) = rd$ *and* $\deg(\varphi) = \kappa$;
2. *over $\mathbb{F}_{p^n}$, both $f(x)$ and $g(x)$ have $\varphi(x)$ as a factor;*
3. $\|f\|_\infty = O(\ln(p))$ *and* $\|g\|_\infty = O(Q^{k/(n(r+1))})$.

Consequently, if ϕ is a sieving polynomial, then

$$|N(f,\phi)| \leq \mathfrak{C}(\eta,t,d(r+1),H) \times E^{(2d(r+1))/t} \times (\ln p)^{\eta(t-1)}; \tag{11}$$

$$|N(g,\phi)| \leq \mathfrak{C}(\eta,t,dr,H) \times E^{2dr/t} \times Q^{(k\eta(t-1))/(n(r+1))}$$

$$= \mathfrak{C}(\eta,t,dr,H) \times E^{2dr/t} \times Q^{(t-1)/(d(r+1))}. \tag{12}$$

Asymptotically, following Lemma 1 of [24], we have

$$N(f,\phi) = E^{2d(r+1)/t} \times L_Q(2/3, o(1));$$

$$N(g,\phi) = E^{2dr/t} \times Q^{\frac{t-1}{d(r+1)}} \times L_Q(2/3, o(1)); \tag{13}$$

$$N(f,\phi) \times N(g,\phi) = E^{(2d(2r+1))/t} \times Q^{\frac{t-1}{d(r+1)}} L_Q(2/3, o(1)). \tag{14}$$

Note. Instead of LLL reduction, it is possible to use HKZ or BKZ reductions in Algorithm $\mathcal{D}$. But the choice of lattice reduction algorithms does not matter in the asymptotic complexity analysis of TNFS, as the root Hermite factor is ignored asymptotically [9]. Practically, the lattice dimensions that arise in the application of Algorithm $\mathcal{D}$ are quite low. In such low dimensions, all the three reduction methods produce smallest vectors of similar norm. Our experiments confirm the same.

4. Asymptotic analysis

We consider the asymptotic analysis in two parts. The first part is an analysis of the asymptotic expression for the norm bound as given by (13) and the second part extends this to an asymptotic analysis of the run time of the algorithm.

4.1. Asymptotic analysis of the norm bounds.

The expression for norm bounds given by (13) hides the constant factors appearing in (11) and (12) in the $L_Q(2/3, o(1))$ factor. For an asymptotic analysis, we take $o(1)$ to be zero so that the factor $L_Q(2/3, o(1))$ can be taken to be 1. This gives the product of the norm bound to be $E^{(2d(2r+1))/t} \times Q^{\frac{t-1}{d(r+1)}}$. Similar asymptotic expressions for the product of norm bounds of polynomial selection algorithms appear in the literature [4, 31, 30].

In Table 1, we compare the expressions for the norm bounds for various polynomial selection algorithms. The last column in the row for exTNFS-$\mathcal{D}$ shows how NFS-GJL, NFS-Conj, NFS-$\mathcal{A}$, exTNFS-GJL and exTNFS-gConj are obtained as special cases of exTNFS-$\mathcal{D}$ by suitably instantiating the parameters of the algorithm. The norm bounds obtainable from exTNFS-$\mathcal{C}$ are either equalled or improved by norm bounds obtainable from exTNFS-$\mathcal{D}$.

The norm bounds obtained from exTNFS-JLSV1 cannot be derived as a special case of the norm bounds obtained from exTNFS-$\mathcal{D}$ and by implication cannot also be derived as a special case of the norm bounds obtained from either exTNFS-GJL or exTNFS-gConj. For all composite values of n from 4 to 24 and $t = 2$, we considered all the expressions for norm bounds that can be obtained from exTNFS-JLSV1, but not from exTNFS-$\mathcal{D}$. The work [4] lists values of Q-E pairs obtained from the

TABLE 1. Parameterised efficiency estimates for NFS obtained from the different polynomial selection methods.

Method	Norms Product	Conditions
NFS-JLSV1 [21]	$E^{\frac{4n}{t}}Q^{\frac{t-1}{n}}$	
NFS-GJL [4]	$E^{\frac{2(2r+1)}{t}}Q^{\frac{t-1}{r+1}}$	$r \geq n$
NFS-Conj [4]	$E^{\frac{6n}{t}}Q^{\frac{t-1}{2n}}$	
NFS-$\mathcal{A}$ [31]	$E^{\frac{2d(2r+1)}{t}}Q^{\frac{t-1}{d(r+1)}}$	$d\|n$, $r \geq n/d$
exTNFS-JLSV1 [24]	$E^{\frac{4\kappa}{t}}Q^{\frac{t-1}{\kappa}}$	$n = \eta\kappa$, $\gcd(\eta, \kappa) = 1$
exTNFS-GJL [24]	$E^{\frac{2(2r+1)}{t}}Q^{\frac{t-1}{r+1}}$	$n = \eta\kappa$, $\gcd(\eta, \kappa) = 1$, $r \geq \kappa$
exTNFS-$\mathcal{C}$ [30]	$E^{\frac{2d(2r+1)}{t}}Q^{\frac{(t-1)(r(\lambda-1)+k)}{\kappa(r\lambda+1)}}$	$n = \eta\kappa$, $k = \kappa/d$, $r \geq k$, $\lambda \in \{1, \eta\}$
exTNFS-gConj [25]	$E^{\frac{6\kappa}{t}}Q^{\frac{t-1}{2\kappa}}$	$n = \eta\kappa$
exTNFS-$\mathcal{D}$	$E^{\frac{2d(2r+1)}{t}}Q^{\frac{(t-1)}{d(r+1)}}$	$n = \eta\kappa$, $d\|\kappa$, $\gcd(\eta, \kappa/d) = 1$, $r \geq \kappa/d$
		NFS-GJL: $\eta = d = 1$
		NFS-Conj: $\eta = 1$, $d = \kappa = n$, $r = 1$
		NFS-$\mathcal{A}$: $\eta = 1$, $\kappa = n$, $d\|n$, $r \geq n/d$
		exTNFS-GJL: $d = 1$
		exTNFS-gConj: $d = \kappa$, $r = 1$

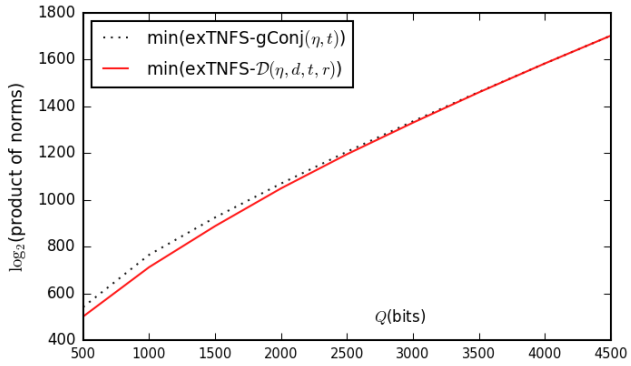
CADO-NFS software. For these Q-E pairs, none of these expressions obtained from exTNFS-JLSV1 achieve the minimum value of the norm, i.e., for each expression of norm bound obtained from exTNFS-JLSV1, there is another expression for norm bound obtained from exTNFS-$\mathcal{D}$ which evaluates to a lower value for all the Q-E pairs given in [4].

For the Q-E pairs given in [4] we have computed the values of the products of norms for exTNFS-gConj and exTNFS-$\mathcal{D}$. The plots are shown in Figure 3.

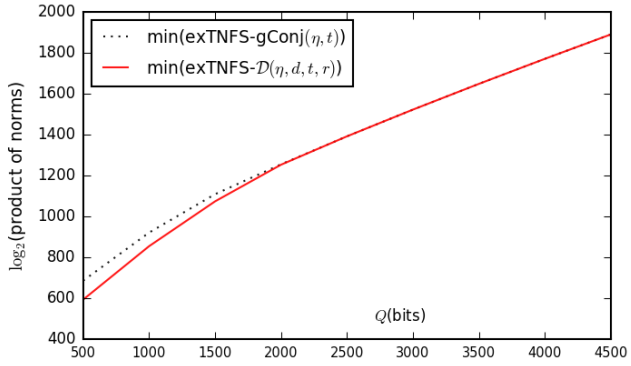
4.2. ASYMPTOTIC RUN TIMES FOR THE MEDIUM CHARACTERISTIC CASE. Following [24], the key observation for analysing exTNFS is that by suitably choosing η, the complexity analysis of the medium prime case can be transformed to the complexity analysis of the boundary case. Our analysis below follows [24, 30].

For the complexity analysis, as is customary, the $o(1)$ terms are ignored, i.e., they are taken to be 0. Also, as in all previous works, the complexity analysis is heuristic.

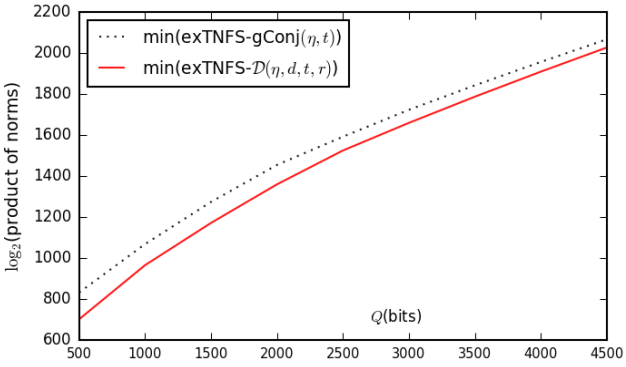
Two such important heuristic assumptions are the following.



(a) $\mathbb{F}_{p^{12}}$



(b) $\mathbb{F}_{p^{18}}$



(c) $\mathbb{F}_{p^{24}}$

FIGURE 3. Product of norms for various polynomial selection methods.

1. It is heuristically assumed that the B-smoothness behaviour of the norm of an ideal is same as that of a random integer of similar size.
2. It is heuristically assumed that the events of obtaining smoothness in the two number fields are independent so that the individual smoothness probabilities can be multiplied together to obtain the joint smoothness probability.

As before, let $Q = p^n$ and $n = \eta\kappa$ is a non-trivial factorisation of n. Suppose that for some a with $1/3 < a \leq 2/3$,

$$p = L_Q(a, c_p), \text{ where } c_p = \frac{1}{n}\left(\frac{\ln Q}{\ln\ln Q}\right)^{1-a} \text{ and so } n = \frac{1}{c_p}\left(\frac{\ln Q}{\ln\ln Q}\right)^{1-a}. \tag{15}$$

We write η in the following manner:

$$\eta = c_\eta \left(\frac{\ln Q}{\ln\ln Q}\right)^{2/3-a}. \tag{16}$$

Further, let

$$P = p^\eta. \tag{17}$$

Then it is easy to verify that

$$P = L_Q(2/3, c_\theta) \text{ and } \kappa = \frac{1}{c_\theta}\left(\frac{\ln Q}{\ln\ln Q}\right)^{1/3} \text{ where} \tag{18}$$

$$c_\theta = c_p c_\eta \text{ and so}$$

$$\kappa = c_\theta \left(\frac{\ln Q}{\ln\ln Q}\right)^{1/3}. \tag{19}$$

Even though P is not a prime, the magnitude of P corresponds to the boundary characteristic case analysis of NFS. The substitutions

$$p \leftarrow P,\ a \leftarrow 2/3,\ c_p \leftarrow c_\theta \text{ and } n \leftarrow \kappa$$

in (15) transform (15) into (18). So, the complexity analysis of the medium characteristic case reduces to the complexity analysis of the boundary characteristic case.

We recall the following.

1. The number of polynomials to be considered for sieving is E^2.
2. The factor base is of size B.

Let

$$B = L_Q(1/3, c_b). \tag{20}$$

As mentioned earlier, set $E = B$ so that asymptotically, the number of sieving polynomials is equal to the time for the linear algebra step.

Let $\pi = \Psi(\Gamma, B)$ be the probability that a random positive integer which is at most Γ is B-smooth. Let $\Gamma = L_Q(z, \zeta)$ and $B = L_Q(b, c_b)$. Using the L-notation version of the Canfield-Erdös-Pomerance theorem,

$$(\Psi(\Gamma, B))^{-1} = L_Q\left(z - b, (z-b)\frac{\zeta}{c_b}\right). \tag{21}$$

The bound on the product of the norms given by Proposition 1 is

$$\Gamma = E^{\frac{2}{t}d(2r+1)} \times Q^{\frac{t-1}{d(r+1)}}. \tag{22}$$

Note that in (22), $t-1$ is the degree of the sieving polynomial. Following the usual convention, we assume that the same smoothness probability π holds for the event that a random sieving polynomial $\phi(x)$ is smooth over the factor base.

The expected number of polynomials to consider for obtaining one relation is π^{-1}. Since B relations are required, obtaining this number of relations requires trying $B\pi^{-1}$ trials. Balancing the cost of sieving and the linear algebra steps requires $B\pi^{-1} = B^2$ and so

$$\pi^{-1} = B. \tag{23}$$

Lemma 4.1. *Let $n = \eta\kappa$; d is a divisor of κ such that $k = \kappa/d$ is co-prime to η; $r \geq k$; $t \geq 2$; $p = L_Q(a, c_p)$ with $1/3 < a \leq 2/3$; and $\eta = c_\eta(\ln Q/\ln\ln Q)^{2/3-a}$. Then*

$$E^{\frac{2}{t}d(2r+1)} = L_Q\left(2/3, \frac{2c_b(2r+1)}{c_\theta kt}\right) \quad \text{and} \quad Q^{\frac{t-1}{d(r+1)}} = L_Q\left(2/3, \frac{kc_\theta(t-1)}{(r+1)}\right). \tag{24}$$

Proof. Noting that $E = B = L_Q(1/3, c_b)$, we have

$$\begin{aligned}
E^{\frac{2}{t}d(2r+1)} &= L_Q\left(1/3, c_b\frac{2}{t}d(2r+1)\right) \\
&= \exp\left(c_b\frac{2}{t}(2r+1)\frac{\kappa}{k}(\ln Q)^{1/3}(\ln\ln Q)^{2/3}\right) \\
&= \exp\left(c_b\frac{2}{c_\theta kt}(2r+1)\left(\frac{\ln Q}{\ln\ln Q}\right)^{1/3}(\ln Q)^{1/3}(\ln\ln Q)^{2/3}\right) \\
&= L_Q\left(2/3, \frac{2c_b(2r+1)}{c_\theta kt}\right).
\end{aligned}$$

The computation for the second relation is as follows:

$$\begin{aligned}
Q^{(t-1)/(d(r+1))} &= p^{(n(t-1))/(d(r+1))} \\
&= L_Q\left(a, c_p\frac{n(t-1)}{d(r+1)}\right) \\
&= \exp\left(c_p\frac{n(t-1)}{d(r+1)}(\ln Q)^a(\ln\ln Q)^{1-a}\right) \\
&= \exp\left(c_p\frac{k(t-1)}{(r+1)}c_\eta\left(\frac{\ln Q}{\ln\ln Q}\right)^{2/3-a}(\ln Q)^a(\ln\ln Q)^{1-a}\right) \\
&= \exp\left(\frac{c_\theta k(t-1)}{(r+1)}(\ln Q)^{2/3}(\ln\ln Q)^{1/3}\right) \\
&= L_Q\left(2/3, \frac{c_\theta k(t-1)}{(r+1)}\right).
\end{aligned}$$

□

This leads to the following result for the medium prime case which is the analogue of Theorem 1 in [31] for the boundary case.

Theorem 4.2. *Let $n = \eta\kappa$; d is a divisor of κ such that $k = \kappa/d$ is co-prime to η; $r \geq k$; $t \geq 2$; $p = L_Q(a, c_p)$ with $1/3 < a \leq 2/3$; and $\eta = c_\eta(\ln Q/\ln\ln Q)^{2/3-a}$. It is possible to ensure that the runtime of the exTNFS algorithm with polynomials chosen by Algorithm $\mathcal{D}$ is $L_Q(1/3, 2c_b)$ where*

$$c_b = \frac{2r+1}{3c_\theta kt} + \sqrt{\left(\frac{2r+1}{3c_\theta kt}\right)^2 + \frac{(t-1)kc_\theta}{3(r+1)}} \text{ and} \tag{25}$$

$$c_\theta = c_p c_\eta. \tag{26}$$

Proof. The product of the norms is

$$\Gamma = L_Q\left(2/3, \frac{2c_b(2r+1)}{c_\theta kt} + \frac{kc_\theta(t-1)}{(r+1)}\right). \tag{27}$$

Then π^{-1} given by (21) is

$$L_Q\left(1/3, \frac{1}{3}\left(\frac{2(2r+1)}{c_\theta kt} + \frac{kc_\theta(t-1)}{c_b(r+1)}\right)\right).$$

From the condition $\pi^{-1} = B$, we get

$$c_b = \frac{1}{3}\left(\frac{2(2r+1)}{c_\theta kt} + \frac{kc_\theta(t-1)}{c_b(r+1)}\right). \tag{28}$$

Solving the quadratic for c_b and choosing the positive root gives

$$c_b = \frac{2r+1}{3c_\theta kt} + \sqrt{\left(\frac{2r+1}{3c_\theta kt}\right)^2 + \frac{kc_\theta(t-1)}{3(r+1)}}.$$

□

Theorem 4.2 can be analysed in exactly the same manner as Theorem 1 of [31] with c_p in Theorem 1 of [31] being replaced by c_θ. Performing the analysis shows that the minimum complexity achieved by exTNFS-$\mathcal{D}$ is $L_Q(1/3, (48/9)^{1/3})$ and this complexity is achieved for $c_\theta = 12^{1/3}$, $d = \kappa$, $r = 1$ and $t = 2$. The choice $r = 1$ converts exTNFS-$\mathcal{D}$ to exTNFS-gConj. So exTNFS-gConj enjoys its superiority in term of the lowest asymptotic complexity but, it is achieved only for a fixed value of c_θ. The choices of $r = 1$ and $t = 2$ are not necessarily the best possible choices for other values of c_θ. This is highlighted in Figure 4 which compares exTNFS-gConj with exTNFS-$\mathcal{D}(t, k, r)$. Note that exTNFS-gConj corresponds to exTNFS-$\mathcal{D}(t, 1, 1)$ with $t \geq 2$. There are segments in Figure 4 where exTNFS-$\mathcal{D}(t, k, r)$ with $k > 1$ and/or $r > 1$ has lower complexity than exTNFS-$\mathcal{D}(t, 1, 1)$. In particular, we note that for $c_\theta > 3.39$, the complexity of exTNFS-$\mathcal{D}$ is lower than the complexities of all previous algorithms applicable to the medium characteristic finite fields.

5. Using multiple number fields

The multiple number field sieve algorithm uses several number fields to lower the asymptotic complexity. To be able to do this, it is required to generate several irreducible polynomials in $R[x]$ all of which have a common irreducible factor over $\mathbb{F}_{p^\eta}$. In this section, we describe how Algorithm $\mathcal{D}$ can be modified to generate such irreducible polynomials. This is an adaptation of a method described in [28].

Algorithm $\mathcal{D}$ produces two polynomials $f(x)$ and $g(x)$ of degrees $d(r+1)$ and dr respectively. The polynomial $g(x)$ is defined as $\mathrm{Res}_y(\psi(y), C_0(x) + yC_1(x))$ where $\psi(x) = \mathrm{LLL}(M_{A_2,r})$, i.e., $\psi(x)$ is defined from the first row of the matrix obtained after applying the LLL-algorithm to $M_{A_2,r}$. Let $\psi_1(x) = \psi(x)$ and $\psi_2(x)$ be the polynomial defined from the second row of the matrix $M_{A_2,r}$; let $g_1(x) = g(x)$ and $g_2(x) = \mathrm{Res}_y(\psi_2(y), C_0(x) + yC_1(x))$. For $i = 3, \ldots, V$, let $g_i(x) = s_i(x)g_1(x) + t_i(x)g_2(x)$ where $s_i(x)$ and $t_i(x)$ are polynomials of degrees less than η with $\|s_i\|_\infty, \|t_i\|_\infty \leq V^{1/(2\eta)}$.

From the manner in which $f(x)$, $g_1(x)$ and $g_2(x)$ have been constructed, it follows that over $\mathbb{F}_{p^\eta}$, $\phi(x)$ is a factor of all three of these polynomials. Then, $\phi(x)$ is also a factor of $g_i(x)$ for $i = 3, \ldots, V$. It is usually easy to ensure that the polynomials

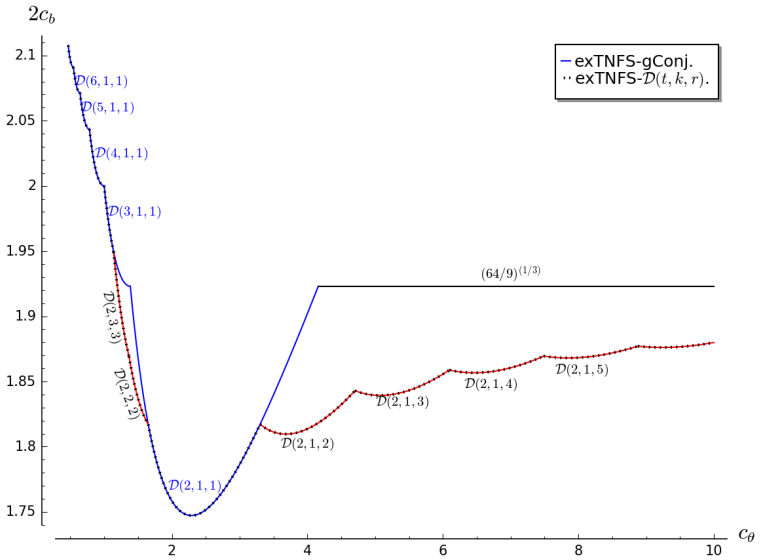


FIGURE 4. Complexity plot for medium characteristic finite fields

$f(x)$ and $g_i(x)$, $i = 1, \ldots, V$ are irreducible over the integers. Let $K_f = R[x]/(f(x))$ and $K_i = R[x]/(g_i(x))$ for $i = 1, \ldots, V$ be the number fields defined by $f(x)$ and $g_i(x)$, $i = 1, \ldots, V$. Further, let $\mathcal{O}_f$ be the ring of integers of $\mathcal{K}_f$ and for $i = 1, \ldots, V$ let $\mathcal{O}_i$ be the ring of integers of $\mathcal{K}_i$. In the usual way, there are homomorphisms from these number fields to the finite field $\mathbb{F}_{p^n} = \mathbb{F}_{p^\eta}/(\phi(x))$.

Clearly the g_i's have degree dr. Asymptotically, $\|g_1\|_\infty = \|\psi_1\|_\infty = \|\psi_2\|_\infty = \|g_2\|_\infty = Q^{k/(n(r+1))}$. As a result, for $i \geq 3$, $\|g_i\|_\infty = V^{1/(2\eta)} Q^{k/(n(r+1))}$.

The factor basis $\mathcal{F}$ is the disjoint union of $\mathcal{F}_f$ (the left side of the factor basis) and $\mathcal{F}_1, \ldots, \mathcal{F}_V$ (the right side of the factor basis), where $\mathcal{F}_f$ consists of ideals in $\mathcal{O}_f$ whose norms are bounded above by B and for $i = 1, \ldots, V$, $\mathcal{F}_i$ consists of ideals in $\mathcal{O}_i$ whose norms are bounded above by B'. The values of B and B' determine the complexity of the algorithm and are chosen so as to balance the cost of the relation collection and the linear algebra phases. The size of the entire factor basis is $B + VB'$. During relation collection, a relation is obtained if a sieving polynomial $\phi(x)$ is smooth over $\mathcal{F}_f$ (left side smoothness) and also over at least one of the factor bases $\mathcal{F}_i$ (right side smoothness).

The following condition is used to balance the left and right sides of the factor basis:

$$B = VB'. \tag{29}$$

As a consequence, the size of the factor basis is $B^{1+o(1)}$ and asymptotically the linear algebra step takes time B^2.

The sieving polynomials $\phi(x)$ are chosen from $R[x]$ and are of degrees at most $t - 1$. The coefficients of these polynomials are elements of $Z[z]/(h(z))$ and so are themselves polynomials in z. Let $\phi(x) = \phi_0(z) + \phi_1(z)x + \cdots + \phi_{t-1}x^{t-1}$ and the coefficients of $\phi_i(z)$ are chosen to have $E^{2/(\eta t)}$ distinct values. So, the number of sieving polynomials is E^2. To balance the cost of relation collection and linear algebra one sets $E = B$.

As before, let π be the probability that a random sieving polynomial $\phi(x)$ gives rise to a relation. Let π_1 be the probability that $\phi(x)$ is smooth over the factor basis for the first number field and π_2 be the probability that $\phi(x)$ is smooth over *at least* one of the other V factor bases. Further, let $\Gamma_1 = \mathrm{Res}_x(f(x), \phi(x))$ be the bound on the norm corresponding to the first number field and $\Gamma_2 = \mathrm{Res}_x(g_i(x), \phi(x))$ be the bound on the norm for any of the other number fields. Recall that Γ_2 is determined only by the degree and the L_∞-norm of $g_i(x)$. Heuristically, we have

$$\begin{aligned} \pi_1 &= \Psi(\Gamma_1, B); \\ \pi_2 &= V\Psi(\Gamma_2, B'); \\ \pi &= \pi_1 \times \pi_2. \end{aligned} \tag{30}$$

One relation is obtained in about π^{-1} trials and so total number of relations obtained after sieving would be $E^2\pi$ and this should be equal to B for linear algebra step to go through. Hence we have, as before, $B = E = \pi^{-1}$.

The following choices of B and V are made:

$$\begin{aligned} E = B &= L_Q\left(\tfrac{1}{3}, c_b\right); \\ V &= L_Q\left(\tfrac{1}{3}, c_v\right); \text{ and so} \\ B' &= B/V = L_Q\left(\tfrac{1}{3}, c_b - c_v\right). \end{aligned} \tag{31}$$

For the case of multiple NFS we obtain the following result for the medium prime case which is the analogue of Theorem 4 in [31] for the boundary case.

Theorem 5.1. *Let $n = \eta\kappa$; d is a divisor of κ such that $k = \kappa/d$ is co-prime to η; $r \geq k$; $t \geq 2$; $p = L_Q(a, c_p)$ with $1/3 < a \leq 2/3$; and $\eta = c_\eta(\ln Q/\ln\ln Q)^{2/3-a}$. It is possible to ensure that the runtime of the exTNFS algorithm using multiple number fields with polynomials chosen by Algorithm $\mathcal{D}$ is $L_Q(1/3, 2c_b)$ where*

$$c_b = \frac{4r+2}{6kc_\theta t} + \sqrt{\frac{r(3r+2)}{(3kc_\theta t)^2} + \frac{(t-1)kc_\theta}{3(r+1)}} \text{ and} \tag{32}$$

$$c_\theta = c_p c_\eta. \tag{33}$$

Proof. Recall that $\|g_i\|_\infty = Q^{k/(n(r+1))}$ for $i = 1, 2$; and $\|g_i\|_\infty = V^{1/(2\eta)}Q^{k/(n(r+1))}$ for $i \geq 3$. In the computation below, we use $V^{1/(2\eta)}Q^{k/(n(r+1))}$ as the norm of g_i for all $i \geq 1$.

$$\begin{aligned} \Gamma_1 &= \|\phi\|_\infty^{\deg(f)} = E^{2\deg(f)/t} = E^{(2d(r+1))/t} = E^{(2\kappa(r+1))/(kt)} \\ &= L_Q\left(\frac{2}{3}, \frac{2(r+1)c_b}{ktc_\theta}\right); \\ \pi_1^{-1} &= L_Q\left(\frac{1}{3}, \frac{2(r+1)}{3ktc_\theta}\right); \\ \Gamma_2 &= \|\phi\|_\infty^{\deg(g)} \times \|g\|_\infty^{\deg(\phi)} = E^{2\deg(g)/t} \times Q^{(t-1)/(d(r+1))} \times V^{(t-1)/2} \\ &= E^{(2rd)/t} \times Q^{(t-1)/(d(r+1))} \times V^{(t-1)/2} \\ &= E^{(2rn)/(kt)} \times Q^{k(t-1)/(n(r+1))} \times V^{(t-1)/2} \\ &= L_Q\left(\frac{2}{3}, \frac{2rc_b}{c_\theta kt} + \frac{kc_\theta(t-1)}{r+1}\right) L_Q(1/3, (t-1)c_v/2); \end{aligned}$$

$$
\begin{aligned}
&= L_Q\left(\frac{2}{3}, \frac{2rc_b}{c_\theta kt} + \frac{kc_\theta(t-1)}{r+1}\right)\\
\pi_2^{-1} &= L_Q\left(\frac{1}{3}, -c_v + \frac{1}{3(c_b-c_v)}\left(\frac{2rc_b}{c_p kt} + \frac{kc_p(t-1)}{r+1}\right)\right);\\
\pi^{-1} &= L_Q\left(\frac{1}{3}, \frac{2(r+1)}{3ktc_p} - c_v + \frac{1}{3(c_b-c_v)}\left(\frac{2rc_b}{c_p kt} + \frac{kc_p(t-1)}{r+1}\right)\right);
\end{aligned}
$$

From the condition $\pi^{-1} = B$, we obtain the following equation:

$$
c_b = \frac{2(r+1)}{3ktc_\theta} - c_v + \frac{1}{3(c_b-c_v)}\left(\frac{2rc_b}{c_\theta kt} + \frac{kc_\theta(t-1)}{r+1}\right). \tag{34}
$$

We wish to find c_v such that c_b is minimised subject to the constraint (34). Using the method of Lagrange multipliers, the partial derivative of (34) with respect to c_v gives

$$
c_v = \frac{r+1}{3ktc_\theta}.
$$

Using this value of c_v in (34) provides the following quadratic in c_b:

$$
(3ktc_\theta)c_b^2 - (4r+2)c_b + \frac{(r+1)^2}{3ktc_\theta} - \frac{(c_\theta k)^2 t(t-1)}{r+1} = 0.
$$

Solving this and taking the positive square root, we obtain

$$
c_b = \frac{4r+2}{6ktc_\theta} + \sqrt{\frac{r(3r+2)}{(3ktc_\theta)^2} + \frac{c_\theta k(t-1)}{3(r+1)}}. \tag{35}
$$

Hence the overall complexity is $L_Q\left(\frac{1}{3}, 2c_b\right)$. □

The plot in Figure 5 shows that asymptotically, the variant using multiple number fields outperforms the variant using two number fields. From Theorem 5.1, the entire

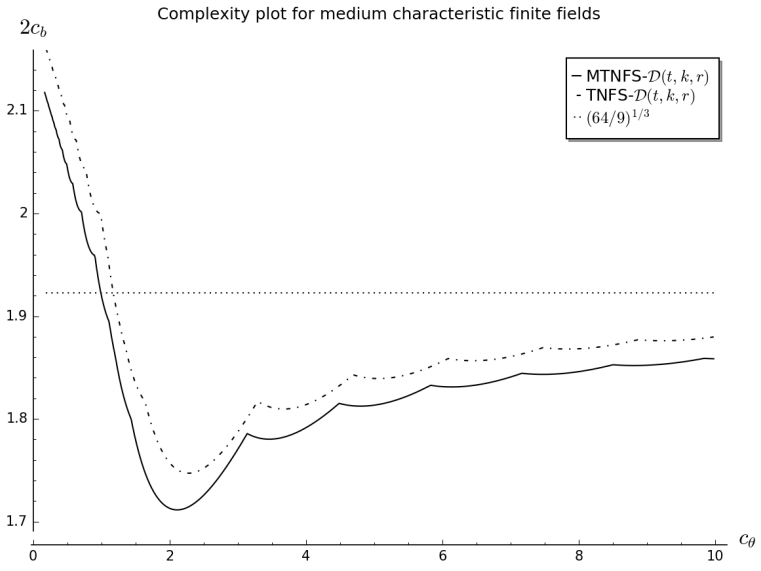


FIGURE 5. Complexity plot for medium characteristic finite fields

analysis carried out in Sections 8.1 and 8.2 of [31] apply with p replaced by P and c_p replaced by c_θ. This shows that the best complexity achieved by MexTNFS-$\mathcal{D}$ is $L_Q(1/3, 1.71)$ and this complexity is achieved for $c_\theta = 2.123$, $d = \kappa$, $r = 1$ and $t = 2$. Again, the choice $r = 1$ converts MexTNFS-$\mathcal{D}$ to MexTNFS-gConj. As in the case of exTNFS, the choices of $r = 1$ and $t = 2$ are not necessarily the best possible choices for other values of c_θ. In particular, we note that for $c_\theta \in (0, 1.12) \cup [1.45, 3.15]$, the complexity of MexTNFS-$\mathcal{D}$ is the same as that of MexTNFS-gConj and for $c_\theta \notin (0, 1.12) \cup [1.45, 3.15]$, the complexity of MexTNFS-$\mathcal{D}$ is lower than that of all previous methods.

6. Conclusion

In this paper, we have presented a new polynomial selection method for the exTNFS algorithm. This method provides a generalisation of the Conjugation method in the setting of exTNFS proposed by Jeong and Kim [25] For certain ranges of finite fields, the new method provides lower asymptotic complexity than the method of Jeong and Kim. Further, a concrete analysis of the different variants of TNFS algorithms shows that some of the new trade-offs obtained from Algorithm $\mathcal{D}$ provide better performance than the previous methods for some particular ranges of values of $\lg Q$.

Acknowledgment

We thank the reviewers for carefully reading the paper and providing comments which have helped in improving the work.

References

[1] L. M. Adleman, The function field sieve, In Leonard M. Adleman and Ming-Deh A. Huang, editors, *ANTS*, volume 877 of *Lecture Notes in Computer Science*, pages 108–121. Springer, 1994.

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

[3] R. Barbulescu and S. Duquesne, Updating key size estimations for pairings, *Journal of Cryptology*, 2018, 1–39. https://link.springer.com/article/10.1007/s00145-018-9280-5.

[4] R. Barbulescu, P. Gaudry, A. Guillevic and F. Morain, Improving NFS for the discrete logarithm problem in non-prime finite fields, In Elisabeth Oswald and Marc Fischlin, editors, *Advances in Cryptology – EUROCRYPT 2015*, volume 9056 of *Lecture Notes in Computer Science*, pages 129–155. Springer Berlin Heidelberg, 2015.

[5] R. Barbulescu, P. Gaudry, A. Joux and E. Thomé, A heuristic quasi-polynomial algorithm for discrete logarithm in finite fields of small characteristic, In Phong Q. Nguyen and Elisabeth Oswald, editors, *Advances in Cryptology - EUROCRYPT 2014 - 33rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Copenhagen, Denmark, May 11-15, 2014. Proceedings*, volume 8441 of *Lecture Notes in Computer Science*, pages 1–16. Springer, 2014.

[6] R. Barbulescu, P. Gaudry and T. Kleinjung, The tower number field sieve, In Tetsu Iwata and Jung Hee Cheon, editors, *Advances in Cryptology - ASIACRYPT 2015 - 21st International Conference on the Theory and Application of Cryptology and Information Security, Auckland, New Zealand, November 29 - December 3, 2015, Proceedings, Part II*, volume 9453 of *Lecture Notes in Computer Science*, pages 31–55. Springer, 2015.

[7] R. Barbulescu and C. Pierrot, The multiple number field sieve for medium and high characteristic finite fields, *LMS Journal of Computation and Mathematics*, **17** (2014), 230–246.

[8] Y. Bistritz and A. Lifshitz, Bounds for resultants of univariate and bivariate polynomials, *Linear Algebra and its Applications*, **432** (2010), 1995–2005. Special issue devoted to the 15th ILAS Conference at Cancun, Mexico, June 16-20, 2008.

[9] N. Gama and P. Q. Nguyen, Predicting lattice reduction, In Nigel Smart, editor, *Advances in Cryptology – EUROCRYPT 2008*, 31–51, Lecture Notes in Comput. Sci., 4965, Springer, Berlin, 2008.

[10] P. Gaudry, L. Grémy and M. Videau, Collecting relations for the number field sieve in GF(p^6), *LMS Journal of Computation and Mathematics*, **19** (2016), 332–350.

[11] D. M. Gordon, Discrete logarithms in GF(p) using the number field sieve, *SIAM J. Discrete Math.*, **6** (1993), 124–138.

[12] R. Granger, T. Kleinjung and J. Zumbrägel, Discrete logarithms in GF(2^{9234}), NMBRTHRY list, January 2014.

[13] A. Guillevic, Computing individual discrete logarithms faster in GF(p^n) with the NFS-DL algorithm, *Advances in cryptology–ASIACRYPT 2015. Part I,* 149–173, Lecture Notes in Comput. Sci., 9452, Springer, Heidelberg, 2015, `http://eprint.iacr.org/`.

[14] A. Guillevic, F. Morain and E. Thomé, Solving discrete logarithms on a 170-bit MNT curve by pairing reduction, *Selected Areas in Cryptography – SAC 2016*, 2017, 559–578. `http://eprint.iacr.org/`.

[15] K. Hayasaka, K. Aoki, T. Kobayashi and T. Takagi, A construction of 3-dimensional lattice sieve for number field sieve over $\mathbb{F}_{p^n}$, *JSIAM Lett.*, **6** (2014), 53–56.

[16] A. Joux, Faster index calculus for the medium prime case: Application to 1175-bit and 1425-bit finite fields, In Thomas Johansson and Phong Q. Nguyen, editors, *EUROCRYPT*, volume 7881 of *Lecture Notes in Computer Science*, pages 177–193. Springer, 2013.

[17] A. Joux, A new index calculus algorithm with complexity $L(1/4+o(1))$ in small characteristic, In Tanja Lange, Kristin E. Lauter, and Petr Lisonek, editors, *Selected Areas in Cryptography - SAC 2013 - 20th International Conference, Burnaby, BC, Canada, August 14-16, 2013, Revised Selected Papers*, volume 8282 of *Lecture Notes in Computer Science*, pages 355–379. Springer, 2014.

[18] A. Joux and R. Lercier, The function field sieve is quite special, In Claus Fieker and David R. Kohel, editors, *ANTS*, volume 2369 of *Lecture Notes in Computer Science*, pages 431–445. Springer, 2002.

[19] A. Joux and R. Lercier, Improvements to the general number field sieve for discrete logarithms in prime fields. A comparison with the Gaussian integer method, *Math. Comput.*, **72** (2003), 953–967.

[20] A. Joux and R. Lercier, The function field sieve in the medium prime case, In Serge Vaudenay, editor, *EUROCRYPT*, volume 4004 of *Lecture Notes in Computer Science*, pages 254–270. Springer, 2006.

[21] A. Joux, R. Lercier, N. P. Smart and F. Vercauteren, The number field sieve in the medium prime case, In Cynthia 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 Berlin Heidelberg, 2006.

[22] A. Joux and C. Pierrot, The special number field sieve in $\mathbb{F}_{p^n}$ - Application to pairing-friendly constructions, In Zhenfu Cao and Fangguo Zhang, editors, *Pairing-Based Cryptography - Pairing 2013 - 6th International Conference, Beijing, China, November 22-24, 2013, Revised Selected Papers*, volume 8365 of *Lecture Notes in Computer Science*, pages 45–61. Springer, 2013.

[23] A. Joux and C. Pierrot, Improving the polynomial time precomputation of Frobenius representation discrete logarithm algorithms - simplified setting for small characteristic finite fields, In Palash Sarkar and Tetsu Iwata, editors, *Advances in Cryptology - ASIACRYPT 2014 - 20th International Conference on the Theory and Application of Cryptology and Information Security, Kaoshiung, Taiwan, R.O.C., December 7-11, 2014. Proceedings, Part I*, volume 8873 of *Lecture Notes in Computer Science*, pages 378–397. Springer, 2014.

[24] T. Kim and R. Barbulescu, Extended tower number field sieve: A new complexity for the medium prime case, In Matthew Robshaw and Jonathan Katz, editors, *Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part I*, volume 9814 of *Lecture Notes in Computer Science*, pages 543–571. Springer, 2016.

[25] T. Kim and J. Jeong, Extended tower number field sieve with application to finite fields of arbitrary composite extension degree, In Serge Fehr, editor, *Public-Key Cryptography - PKC 2017 - 20th IACR International Conference on Practice and Theory in Public-Key*

Cryptography, Amsterdam, The Netherlands, March 28-31, 2017, Proceedings, Part I, volume 10174 of *Lecture Notes in Computer Science*, pages 388–408. Springer, 2017.

[26] A. K. Lenstra, H. W. Lenstra and L. Lovász, Factoring polynomials with rational coefficients, *Mathematische Annalen*, **261** (1982), 515–534.

[27] A. Menezes, P. Sarkar and S. Singh, Challenges with assessing the impact of NFS advances on the security of pairing-based cryptography, In *Mycrypt*, volume 10311 of *Lecture Notes in Computer Science*, pages 83–108. Springer, 2016.

[28] C. Pierrot, The multiple number field sieve with conjugation and generalized Joux-Lercier methods, In *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*, pages 156–170, Lecture Notes in Comput. Sci., 9056, Springer, Heidelberg, 2015.

[29] P. Sarkar and S. Singh, Fine tuning the function field sieve algorithm for the medium prime case, *IEEE Transactions on Information Theory*, **62** (2016), 2233–2253. http://ieeexplore.ieee.org/xpl/articleDetails.jsp?reload=true&arnumber=7405328.

[30] P. Sarkar and S. Singh, A general polynomial selection method and new asymptotic complexities for the tower number field sieve algorithm, In Jung Hee Cheon and Tsuyoshi Takagi, editors, *Advances in Cryptology - ASIACRYPT 2016 - 22nd International Conference on the Theory and Application of Cryptology and Information Security, Hanoi, Vietnam, December 4-8, 2016, Proceedings, Part I*, volume 10031 of *Lecture Notes in Computer Science*, pages 37–62, 2016.

[31] P. Sarkar and S. Singh, New complexity trade-offs for the (multiple) number field sieve algorithm in non-prime fields, In Marc Fischlin and Jean-Sébastien Coron, editors, *Advances in Cryptology - EUROCRYPT 2016 - 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Vienna, Austria, May 8-12, 2016, Proceedings, Part I*, volume 9665 of *Lecture Notes in Computer Science*, pages 429–458, Springer, 2016.

[32] O. Schirokauer, Discrete logarithms and local units, *Philosophical Transactions: Physical Sciences and Engineering*, **345** 91993), 409–423.

[33] O. Schirokauer, Using number fields to compute logarithms in finite fields, *Math. Comp.*, **69** (2000), 1267–1283.

[34] O. Schirokauer, Virtual logarithms, *J. Algorithms*, **57** (2005), 140–147.

[35] D. H. Wiedemann, Solving sparse linear equations over finite fields, *IEEE Trans. Information Theory*, **32** (1986), 54–62.

[36] P. Zajac, On the use of the lattice sieve in the 3d NFS, *Tatra Mountains Mathematical Publications*, **45** (2010), 161–172.

Received July 2018; revised January 2019.

E-mail address: palash@isical.ac.in
E-mail address: shashank@iiserb.ac.in