A Generalisation of the Conjugation Method for Polynomial Selection for the Extended Tower Number Field Sieve Algorithm

Palash Sarkar and Shashank Singh

Applied Statistics Unit
Indian Statistical Institute
palash@isical.ac.in, sha2nk.singh@gmail.com

Abstract. In a recent work, Kim and Barbulescu showed how to combine previous polynomial selection methods with the extended tower number field sieve algorithm to obtain improved complexity for the discrete logarithm problem on finite fields $\mathbb{F}_{p^n}$ for the medium prime case and where n is composite and not a prime-power. A follow up work by Sarkar and Singh presented a general polynomial selection method and showed how to lower the complexity in the medium prime case even when n is composite and a prime-power. This complexity, though, was higher than what was reported for the case of n composite and not a prime-power. By suitably combining the Conjugation method of polynomial selection proposed earlier by Barbulescu et al. with the extended tower number field sieve algorithm, Jeong and Kim showed that the same asymptotic complexity is achieved for any composite n. The present work generalises the polynomial selection method of Jeong and Kim for all composite n. Though the best complexity that can be achieved is not lowered, there is a significant range of finite fields for which the new algorithm achieves complexity which is lower than all previously proposed methods.

Keywords: finite fields, discrete logarithm, tower number field sieve.

1 Introduction

One of the important problems in cryptography is to compute discrete logarithms over the multiplicative group of a finite field. There are two known general approaches to this problem which lead to sub-exponential run-time. These are the function field sieve (FFS) [1, 2, 15, 17] and the number field sieve (NFS) [9, 16, 18] algorithms.

Let p be a prime, $n \geq 1$ be an integer and $Q = p^n$. 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 $a \leq 1/3$; medium characteristic, if $1/3 < a < 2/3$; boundary, if $a = 2/3$; and large characteristic, if $a > 2/3$.

There has been tremendous progress in the FFS algorithm leading to a quasi-polynomial time algorithm [5] for the small characteristic case. Using algorithms given in [14, 5], a record computation of discrete log in the binary extension field $\mathbb{F}_{2^{9234}}$ was reported by Granger et al [10]. The FFS algorithm also applies to the medium prime case and this has been reported in [17, 13, 23].

The NFS algorithm is generally considered to be the state-of-the-art for medium to large characteristic finite fields. The application of NFS to compute discrete logarithms over finite fields was first proposed by Gordon [9] for prime order fields, i.e., for $n = 1$. Application to composite order fields, i.e., for $n > 1$, was shown by Schirokauer [27]. Important improvements to the NFS for prime order fields were given by Joux and Lercier [16]. Joux, Lercier, Smart and Vercauteren [18] showed that the NFS algorithm is applicable to all finite fields. When the prime p is of a special form, Joux and Pierrot [19] showed the application of the special number field sieve algorithm to obtain improved complexity.

The basic structure of the NFS algorithm is to construct two polynomials $f(x)$ and $g(x)$ over the integers which have a common factor $\varphi(x)$ of degree n modulo p. The polynomial $\varphi(x)$ defines the field $\mathbb{F}_{p^n}$ while the polynomials $f(x)$ and $g(x)$ define two number fields. The efficiency of the NFS algorithm is crucially dependent on the properties of the polynomials $f(x)$ and $g(x)$ used to construct the number fields. Consequently, polynomial selection is an important step in the NFS algorithm and is an active area of research.

Sequence of recent works on NFS: Starting with the work of Barbulescu et al. [4], there have been several recent works which continuously improve polynomial selection algorithms.

1. Barbulescu et al. [4] introduced two new methods for polynomial selection, namely, the generalised Joux-Lercier (GJL) and the Conjugation method. For the boundary case, the best complexity obtained was $L_Q(1/3, (48/9)^{1/3})$. The best complexities for the medium and the large prime cases were respectively $L_Q(1/3, (96/9)^{1/3})$ and $L_Q(1/3, (64/9)^{1/3})$.
2. Pierrot [22] worked out 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.
3. Barbulescu et al. [6] provided a detailed analysis of the tower number field sieve (TNFS) variant which had earlier been proposed by Schirokauer [27].
4. Sarkar and Singh [25] provided a method (called Algorithm-$\mathcal{A}$) for polynomial selection which both generalised and subsumed the GJL and the Conjugation methods. Asymptotic complexity for NFS and MNFS were worked out. The best reported complexities in [4] and [22] are obtained for one particular value of c_p. While the new method of [25] could not lower these complexities, it was shown that there are significant ranges for the values of c_p, where in comparison to [4, 22], lower asymptotic complexities are obtained.
5. Gaudry et al. [8] discussed practical issues in relation collection and the consequences of polynomial selection to this phase.

6. Kim and Barbulescu [21] (which is a merge of [20] and [3]) 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*. The complexity they achieved is $L_Q(1/3, (48/9)^{1/3})$ using classical NFS. An improvement of the second term of the sub-exponential expression was obtained using MNFS. The paper also reported improved complexities for special number field sieve algorithm.
7. Sarkar and Singh [26] provided an extension of Algorithm-$\mathcal{A}$ to provide a general method (called Algorithm-$\mathcal{B}$) for polynomial selection for the case considered in [21], i.e., for composite n which is not a prime power. It was shown that in this setting, the GJL and the Conjugation method are special cases of Algorithm-$\mathcal{B}$.
8. Sarkar and Singh [24] provided an extension of Algorithm-$\mathcal{B}$ called Algorithm-$\mathcal{C}$ which covered all composite n. When n is composite and a power of 2, the best complexity in the medium prime case using NFS was obtained to be $L_Q(1/3, (64/9)^{1/3})$. Progressively higher best complexities were reported for other prime-power values of n.
9. Guillevic et al. [11] reported a computation of discrete logarithm on an 170-bit MNT curve. They used the Conjugation method for selecting polynomials.
10. Jeong and Kim [12] showed how to combine the Conjugation method with exTNFS to cover all composite extension degrees. In particular, they showed that using classical NFS, the best complexity obtained for composite n (irrespective of whether it is a prime-power or not) in the medium prime case is $L_Q(1/3, (48/9)^{1/3})$.

The present paper provides a new general polynomial selection method, called Algorithm-$\mathcal{D}$. This algorithm works for all composite values of n and has the Conjugation method as a special case in the exTNFS setting. The best complexity achieved for the medium prime case is $L_Q(1/3, (48/9)^{1/3})$ for all composite n. This complexity is the same as that reported by Jeong and Kim [12]. On the other hand, for the medium prime case, there is a significant range of finite fields for which lower complexity is achieved. Suppose that $n = \eta\kappa$ and that η can be written as $\eta = c_\eta(\ln Q/\ln\ln Q)^{2/3-a}$ and let $c_\theta = c_p c_\eta$. For $c_\theta \in [3.39, 20.91]$, the complexity of exTNFS-$\mathcal{D}$ is lower than the complexities of all previous algorithms whether classical or MTNFS. 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-Conj 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.

We note that Algorithm-$\mathcal{D}$ does not subsume ether Algorithm-$\mathcal{B}$ or Algorithm-$\mathcal{C}$. Though the asymptotic complexity obtained by Algorithm-$\mathcal{D}$ is lower than that of Algorithm-$\mathcal{C}$, there are certain trade-offs in the norm bound that are achieved using Algorithm-$\mathcal{C}$ but, not using Algorithm-$\mathcal{D}$.

There is a mis-conception appearing in several works that the Sarkar-Singh method from [25] (i.e., Algorithm-$\mathcal{A}$) is applicable only when n is composite. We

have noted that such comments appear in [21, 11]. By extension, there may be a mis-conception that for composite $n = \eta\kappa$, Algorithms-$\mathcal{B}$, $\mathcal{C}$ (from [26] and [24]) and $\mathcal{D}$ appearing here are applicable only when κ is prime. We would like to clear this confusion.

1. Algorithm-$\mathcal{A}$ from [25] applies for all values of n. The algorithm has two parameters, namely a divisor d of n and a positive integer r such that $r \geq n/d$. For $d = 1$, Algorithm-$\mathcal{A}$ reduces to the GJL method. For $d = n$, Algorithm-$\mathcal{A}$ provides a *generalisation* of the Conugation method; the conjugation method is obtained by choosing $r = 1$; for certain ranges of c_p, it is possible to choose $r > 1$ to obtain asymptotic complexity which is lower than the Conjugation method. The cases of $d = 1$ and $d = n$ apply irrespective of whether n is prime or composite. If further, n is composite, then it is possible to choose $1 < d < n$ to obtain new trade-offs on the norm bounds. We note that the work [8] correctly describes this set-up.
2. For Algorithms-$\mathcal{B}$ and $\mathcal{C}$ the above statements apply with n replaced by κ. For Algorithm-$\mathcal{B}$, the condition $\gcd(\eta, \kappa) = 1$ is required while no such condition is required for Algorithm-$\mathcal{C}$.
3. Algorithm-$\mathcal{D}$ has the parameters r and d with the condition that $\gcd(\eta, \kappa/d) = 1$. By suitably choosing d (for example choosing $d = \kappa$), Algorithm-$\mathcal{D}$ can be made to work for all composite n. Choosing $d = \kappa$ and $r = 1$ provides the Conjugation method in the exTNFS setting. Choosing $r > 1$ (or $d < \kappa$, if appropriate) provides a generalisation of the Conjugation method.

2 The Set-Up of the Tower Number Field Sieve Algorithm

The target is to compute discrete logarithm in the field $\mathbb{F}_{p^n}$ where n is composite. Suppose that $n = \eta\kappa$ is a non-trivial factorisation of n.

Let $h(z)$ be a monic polynomial of degree η which is irreducible over both $\mathbb{Z}$ and $\mathbb{F}_p$. Let $R = \mathbb{Z}[z]/(h(z))$. Also, note that $\mathbb{F}_{p^\eta} = \mathbb{F}_p[z]/(h(z))$.

Let $f(x)$ and $g(x)$ be polynomials in $R[x]$ whose leading coefficients are from $\mathbb{Z}$. The other coefficients of f and g are polynomials in z of degrees at most $\eta - 1$. In particular, f and g can be viewed as bi-variate polynomials in x and z with coefficients in $\mathbb{Z}$. The following properties are required.

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 $\mathbb{F}_{p^\eta}[x]/(\varphi(x)) = (R/pR)[x]/(\varphi(x))$.

Let K_f and K_g be the number fields associated with the polynomials f and g respectively. The above set-up provides two different decompositions of a homomorphism from $R[x]$ to $\mathbb{F}_{p^n}$. One of these goes through $R[x]/(f(x))$ and the other goes through $R[x]/(g(x))$.

With this set-up, it is possible to set up a factor base and perform the three main steps (relation collection, linear algebra and descent) of the NFS algorithm. For details we refer to [6, 21]. In this work, we will need only the following facts.

1. The factor base consists of B elements for some value B which determines the overall complexity of the algorithm.
2. A polynomial $\phi(x) \in R[x]$ generates a relation if both the norms $N(\phi, f)$ and $N(\phi, g)$ are B-smooth, where

$$N(\phi, f) := \mathrm{Res}_z(\mathrm{Res}_x(\phi(x), f(x)), h(z));$$
$$N(\phi, g) := \mathrm{Res}_z(\mathrm{Res}_x(\phi(x), g(x)), h(z)).$$

2.1 Bounds on Resultants

Let $f(z, x)$ be a bivariate polynomial with integer coefficients where $f_{i,j}$ is the coefficient of $x^i z^j$. Then

$$\|f\|_\infty = \max|f_{i,j}|.$$

Bounds on resultants of univariate and bivariate polynomials were given in [7]. In the following, we summarise these bounds.

Univariate polynomials: Let $a(u)$ and $b(u)$ be two polynomials with integer coefficients. From [7], we have

$$\begin{aligned}&|\mathrm{Res}_u(a(u), b(u))| \\ &\leq (\deg(a)+1)^{\deg(b)/2}(\deg(b)+1)^{\deg(a)/2}\|a\|_\infty^{\deg(b)} \times \|b\|_\infty^{\deg(a)}. \end{aligned} \quad (1)$$

Bivariate polynomials: Let $a(u, v)$ and $b(u, v)$ be two polynomials with integer coefficients. Let $c(u) = \mathrm{Res}_v(a(u, v), b(u, v))$. Then

$$\begin{aligned}&\|c\|_\infty \\ &\leq (\deg_v(a) + \deg_v(b))!\,(\max(\deg_u(a), \deg_u(b)) + 1)^{\deg_v(a)+\deg_v(b)+1} \\ &\quad \times \|a\|_\infty^{\deg_v(b)} \times \|b\|_\infty^{\deg_v(a)}. \end{aligned} \quad (2)$$

The bounds given by (1) and (2) combine to provide bounds on $N(\phi, f)$.

Let $\phi(x, z)$ and $f(x, z)$ be two polynomials and

$$\rho(z) = \mathrm{Res}_x(\phi(x, z), f(x, z)).$$

Further, suppose $\deg_x\phi \leq t-1$ and $\deg_z\phi \leq \eta - 1$. For $\|\phi\|_\infty = E^{2/(t\eta)}$, the number of possible $\phi(x, z)$'s is E^2. Assuming that $t, \eta, \deg_x f$ and $\deg_z f$ are small in comparison to E, using (2) we have

$$\|\rho\|_\infty = O\left(E^{2\deg_x(f)/(t\eta)} \cdot \|f\|_\infty^{t-1}\right).$$

Suppose $h(z)$ is a polynomial of degree η with $\|h\|_\infty = H$. Let

$$\Gamma = \mathrm{Res}_z\left(\mathrm{Res}_x(\phi(x), f(x)), h(z)\right).$$

Assuming that H is negligible in comparison to E, using (1) we have

$$\begin{aligned}|\Gamma| &= O\left(\left(\|\rho\|_\infty^\eta \cdot \|h\|_\infty^{\deg(\rho)}\right)\right) \\ &= O\left(E^{2\deg_x f/t} \cdot \|f\|_\infty^{\eta(t-1)}\right).\end{aligned}$$

Note that in the TNFS set-up described above $N(\phi, f) = \Gamma$.

Sieving polynomials: Sieving is done using polynomials $\phi(x) \in R[x]$ of degrees at most $t-1$ with $\|\phi\|_\infty = E^{2/\eta t}$. Then the number of sieving polynomials is E^2.

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 [16] and uses the LLL algorithm to select polynomials.

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

$$M_{\varphi,r} = \begin{bmatrix} p & & & & & \\ & \ddots & & & & \\ & & \ddots & & & \\ & & & p & & \\ \varphi_0 & \varphi_1 & \cdots & \varphi_{n-1} & 1 & \\ & \ddots & \ddots & & & \ddots \\ & & \varphi_0 & \varphi_1 & \cdots & \varphi_{n-1} & 1 \end{bmatrix} \tag{3}$$

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

$$\mathrm{LLL}(M_{\varphi,r}) = \psi(x) = \psi_0 + \psi_1 x + \cdots + \psi_r x^r \tag{4}$$

to denote the polynomial $\psi(x)$. The determinant of $M_{\varphi,r}$ is p^k and so by the LLL-reduced property, $\|\varphi\|_\infty = O(p^{k/(r+1)})$. 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. It has the condition $\gcd(\eta, k) = 1$, where $k = \kappa/d$. 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 degree k with coefficients from $\mathbb{F}_p$ which is required to be irreducible over $\mathbb{F}_{p^\eta}$, it is necessary that $\gcd(\eta, k) = 1$. This condition, however, does not restrict applicability. One can always choose $d = \kappa$ to obtain $k = 1$ and so $\gcd(\eta, k) = 1$. Other values of d may also be appropriate, eg., if $\eta = 3$ and $\kappa = 4$, then one can choose $d = 2$.

If $d = \kappa$ and $r = 1$, then we obtain exactly the polynomial selection method proposed by Jeong and Kim [12] which is essentially the Conjugation method

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

Input: p, $n = \eta\kappa$, d (such that $d|\kappa$ and $\gcd(\eta, d/\kappa) = 1$) and $r \geq d/\kappa$.
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 $\mathbf{F}_p$;
let $k = \kappa/d$;
let $R = \mathbb{Z}[z]/(h(z))$;
let $\mathbb{F}_{p^n} = \mathbb{F}_p[z]/(h(z))$;
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 $\mathbb{F}_{p^n}$.

  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 $\mathbb{F}_{p^n} = F_p[z]/(h(z))$.*

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

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.

The following result states the basic properties of Algorithm $\mathcal{D}$. Bounds on the norms are obtained from the bounds on resultants given in [7] (see Section 2.1).

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

$$N(\phi, f) = E^{2d(r+1)/t} \times L_Q(2/3, o(1)); \tag{5}$$

$$\begin{aligned} N(\phi, g) &= E^{2dr/t} \times Q^{\frac{(t-1)k}{\kappa(r+1)}} \times L_Q(2/3, o(1)) \\ &= E^{2\kappa r/t} \times Q^{\frac{t-1}{d(r+1)}} \times L_Q(2/3, o(1)); \end{aligned} \tag{6}$$

$$N(\phi, f) \times N(\phi, g) = E^{(2d(2r+1))/t} \times Q^{\frac{t-1}{d(r+1)}} L_Q(2/3, o(1)). \tag{7}$$

We explain the main differences between Algorithm-$\mathcal{D}$ and Algorithms-$\mathcal{C}$ and $\mathcal{B}$ described in [26] and [24] respectively.

Differences in Algorithm-$\mathcal{D}$ and Algorithm-$\mathcal{B}$: Following [21], Algorithm-$\mathcal{B}$ requires the condition $\gcd(\eta, \kappa) = 1$. On the other hand, Algorithm-$\mathcal{D}$ has the condition $\gcd(\eta, k) = 1$. The two conditions are not equivalent and neither do one follow from the other. Further, in Algorithm-$\mathcal{B}$, the polynomials $C_0(x)$ and $C_1(x)$ are restricted to have integer coefficients, while in Algorithm-$\mathcal{D}$ these polynomials have coefficients from R.

Differences in Algorithm-$\mathcal{D}$ and Algorithm-$\mathcal{B}$: In Algorithm-$\mathcal{C}$, the polynomials $A_1(x)$ and $A_2(x)$ were allowed to have coefficients from $\mathbb{F}_{p^\eta}$ while the polynomials $C_0(x)$ and $C_1(x)$ were restricted to have coefficients in $\mathbb{Z}$. On the other hand, Algorithm-$\mathcal{D}$ restricts $A_1(x)$ and $A_2(x)$ to have coefficients in $\mathbb{Z}$ whereas polynomials $C_0(x)$ and $C_1(x)$ are allowed to have coefficients from $\mathbb{F}_{p^\eta}$.

4 Examples

We provide examples of polynomials obtained using Algorithm-$\mathcal{D}$.

Example 1. Let p be a 201-bit prime given below

$$p = 1606938044258990275541962092341162602522202993782792835301611 \tag{8}$$

and $n = 18$. Let $\eta = 3$, $\kappa = 6$. If we take $d = 3$, we have $k = \kappa/d = 2$ and so the condition $\gcd(\eta, k) = 1$ is satisfied. Taking $r = k$, we get the following polynomials.

$$h(z) = z^3 + 15z^2 + 10z + 11$$

$$\begin{aligned} f(x) = {} & x^9 + (12z^2 + 18z + 9)x^8 + (8355z^2 + 5364z + 6372)x^7 \\ & + (2081986z^2 + 1338976z + 1595132)x^6 \\ & + (7693226z^2 + 4947849z + 5894445)x^5 \\ & + (15696735z^2 + 10095449z + 12026805)x^4 \\ & + (19245585z^2 + 12378042z + 14746018)x^3 \\ & + (15661935z^2 + 10073185z + 12000266)x^2 \\ & + (7649910z^2 + 4920112z + 5861422)x + 2055692z^2 + 1322108z + 1575062 \end{aligned}$$

$$
\begin{aligned}
g(x) = {} & 125348169769414251889064805141330425100 3x^6 \\
& + (100278535815531401511251844113064340080 24z^2 \\
& \qquad + 150417803723297102266877766169596510120 36z \\
& \qquad + 752089018616485511334388830847982550601 8)x^5 \\
& + (34972139365666576277049080634431188602983 70z^2 \\
& \qquad + 225125312905867996392760390033829443480 1388z \\
& \qquad + 266615557099544113768040840535609814188 3381)x^4 \\
& + (86465167506941950953076902586489727234186 94z^2 \\
& \qquad + 553537517701733336342110179504115157242 9248z \\
& \qquad + 660119320544740027161458572922805288714 7128)x^3 \\
& + (12351989508940569307175287612517657590593879z^2 \\
& \qquad + 792290236006956117968860357979422577027 9937z \\
& \qquad + 941973723264392418835362872977634624613 2532)x^2 \\
& + (86758862749936950664054499682080289237674 10z^2 \\
& \qquad + 558336920125468863119262984877677213638 7996z \\
& \qquad + 660943126641365666100079579850771811538 3806)x \\
& + 351619702189455869151510949810527592527373 2z^2 \\
& + 22784742753528374171240154042801167280134 28z \\
& + 268642092019404919884704663463643160827 4369
\end{aligned}
$$

$$
\begin{aligned}
& \phi(x) \\
& = x^6 + (8z^2 + 12z + 6)x^5 + (2790z^2 + 1796z + 2127)x^4 + (6898z^2 + 4416z \\
& \quad + 153911431113337476909206798363080245664714124206521967227 3746)x^3 \\
& \quad + (133564311175652824974238565749972201902195598691250018317 8940z^2 \\
& \qquad + 119999564550529723684259744007900172727183248347735385710 9144z \\
& \quad + 140346684488214375619227976621008216489701773863007334620 9732)x^2 \\
& \quad + (126781937863091274329249154878936187314689423519492702014 2876z^2 \\
& \qquad + 106434817925406622394280922265828143552170898004220753104 1015z \\
& \quad + 140346684488214375619227976621008216489701773863007334620 7490)x \\
& \quad + 133564311175652824974238565749972201902195598691250018317 1891z^2 \\
& \quad + 119999564550529723684259744007900172727183248347735385710 4641z \\
& \quad + 804500313598254784989104747199604209155950567849309816211 64
\end{aligned} \tag{9}
$$

Note that we have $\|g\|_\infty \approx 2^{143}$. If we take $d = 6$(i.e., $k = 1$ and so $\gcd(\eta, k) = 1$) and $r = k$, we get the following polynomials.

$$h(z) = z^3 + 14z^2 + 11z + 9$$

$$
\begin{aligned}
f(x) = x^{12} &+ (4z^2 + 16z + 14)x^{11} + (396z^2 + 354z + 279)x^{10} \\
&+ (2942z^2 + 2338z + 2018)x^9 + (9119z^2 + 7144z + 6221)x^8 \\
&+ (15438z^2 + 12050z + 10504)x^7 + (19707z^2 + 15345z + 13383)x^6 \\
&+ (27596z^2 + 21546z + 18809)x^5 + (32861z^2 + 25651z + 22383)x^4 \\
&+ (24795z^2 + 19352z + 16890)x^3 + (18601z^2 + 14519z + 12667)x^2 \\
&+ (16288z^2 + 12733z + 11085)x + 6394z^2 + 4978z + 4349
\end{aligned}
$$

$$
\begin{aligned}
g(x) = &-85664252522791494833070378 3790x^6 \\
&+ (-17132850504558298966614075675 80z^2 \\
&\qquad - 68531402018233195866456302703 20z \\
&\qquad - 59964976765954046383149264865 30)x^5 \\
&+ (-51398551513674896899842227027 40z^2 \\
&\qquad - 59964976765954046383149264865 30z \\
&\qquad - 59964976765954046383149264865 30)x^4 \\
&+ (-59964976765954046383149264865 30z^2 \\
&\qquad - 51398551513674896899842227027 40z \\
&\qquad - 51398551513674896899842227027 40)x^3 \\
&+ (-34265701009116597933228151351 60z^2 \\
&\qquad - 25699275756837448449921113513 70z \\
&\qquad - 17132850504558298966614075675 80)x^2 \\
&+ (-68531402018233195866456302703 20z^2 \\
&\qquad - 59964976765954046383149264865 30z \\
&\qquad - 25699275756837448449921113513 70)x \\
&- 51398551513674896899842227027 40z^2 \\
&- 17132850504558298966614075675 80z \\
&- 63003175632338133414557302387 01
\end{aligned}
$$

$$
\begin{aligned}
\phi(x) = x^6 &+ (2z^2 + 8z + 7)x^5 + (6z^2 + 7z + 7)x^4 + (7z^2 + 6z + 6)x^3 \\
&+ (4z^2 + 3z + 2)x^2 + (8z^2 + 7z + 3)x + 6z^2 + 2z \\
&+ 160925807199202358283751469410514542803176784794567058060684
\end{aligned}
$$

We note that $\|g\|_\infty \approx 2^{102}$. Taking $d = \kappa$ and $r > k$, we get the following polynomials which are not obtained by Conjugation method.

$$
h(z) = z^3 + 19z^2 + 12z + 20
$$

$$
\begin{aligned}
f(x) = {} & x^{18} + (24z^2 + 15z + 6)x^{17} + (62625z^2 + 37131z + 68196)x^{16} \\
& + (54727718z^2 + 32447633z + 59463133)x^{15} \\
& + (37282287z^2 + 22104906z + 40509642)x^{14} \\
& + (149160537z^2 + 88436031z + 162067191)x^{13} \\
& + (166343264z^2 + 98624459z + 180739748)x^{12} \\
& + (276943888z^2 + 164198071z + 300908938)x^{11} \\
& + (290533998z^2 + 172256569z + 315676422)x^{10} \\
& + (335619487z^2 + 198986295z + 364663865)x^9 \\
& + (321699136z^2 + 190734514z + 349537809)x^8 \\
& + (283120090z^2 + 167859584z + 307621251)x^7 \\
& + (220526020z^2 + 130748936z + 239609783)x^6 \\
& + (152665575z^2 + 90514923z + 165877425)x^5 \\
& + (95525536z^2 + 56636128z + 103791992)x^4 \\
& + (49105732z^2 + 29114769z + 53355671)x^3 \\
& + (19530669z^2 + 11579565z + 21220837)x^2 \\
& + (5146409z^2 + 3051304z + 5591790)x + 611432z^2 + 362490z + 664458
\end{aligned}
$$

$$
\begin{aligned}
g(x) = {} & -404737481266104739 65x^{12} + (-647579970025767583440z^2 \\
& \qquad - 404737481266104739650z - 161894992506441895860)x^{11} \\
& + (-844970439639246864967305z^2 - 501145949303690888634630z \\
& \quad - 920373032399122177964100)x^{10} + (-383205447262747967500620z^2 \\
& \qquad - 225438777065220339985050z - 412751283395173613495070)x^9 \\
& + (-149182188219873545987 5935z^2 - 884351396566438856135250z \\
& \quad - 162413056482462509926 7520)x^8 + (-137707880625979476618 5160z^2 \\
& \qquad - 815303182262441387550960z - 149275277840564750077 7130)x^7 \\
& + (-191813187121623358214928 0z^2 - 113674568988398177178 0990z \\
& \quad - 208487410618787020737 4193)x^6 + (-151849817122971676307 9704z^2 \\
& \qquad - 897700170645904285463230z - 164847654657666754957 2676)x^5 \\
& + (-131995113303297033239 8771z^2 - 783968938410128853621580z \\
& \quad - 143358424766008798870 1134)x^4 + (-983242459565381772403466z^2 \\
& \qquad - 581257193171886544983086z - 106689965243531312165 8704)x^3 \\
& + (-568162913399041806440200z^2 - 336945997661804696189042z \\
& \qquad\qquad - 617045593825898731479033)x^2 \\
& + (-238220918670913222677520z^2 - 141103137787225140183294z \\
& \qquad - 258781582719231343451740)x - 426805896533969950113 01z^2 \\
& - 253290035249421296216 20z - 457349250532308408904 64
\end{aligned}
$$

$$
\begin{aligned}
\phi(x) = {} & x^6 + (8z^2 + 5z + 2)x^5 + (2z^2 + 5z + 8)x^4 + (7z^2 + 7z + 3)x^3 \\
& + (5z^2 + 4z + 6)x^2 + (5z^2 + 3z + 5)x + 2z^2 + 5z \\
& + 76786998417810400959689540320120624091302400319238236019324 3
\end{aligned}
$$

Note that $\|g\|_\infty \approx 2^{80}$.

Example 2. Consider $n = 4$ and p given in the equation (8). Taking $\eta = \kappa = d = 2$ and $r = 1$, we get the following set of polynomials.

$$h(z) = z^2 + 10z + 4$$

$$f(x) = x^4 + (12z + 2)x^3 + (-334z - 141)x^2 + (-814z - 334)x - 476z - 189$$

$$\begin{aligned} g(x) = {} & -715200327398308039994565525361x^2 \\ & + (-429120196438984823996739315 2166z \\ & \qquad - 715200327398308039994565525361)x \\ & - 500640229178815627996195867 7527z \\ & - 109573552768088175543806774 0325 \end{aligned}$$

$$\begin{aligned} \phi(x) = {} & x^2 + (6z + 1)x + 7z \\ & + 436504189383491355398355525988746242042734043988487290063285 \end{aligned}$$

Note that we have $\|g\|_\infty \approx 2^{101}$. Taking, in addition, $r = 2$, we get the following.

$$h(z) = z^2 + 9z + 12$$

$$\begin{aligned} f(x) = {} & x^6 + (6z + 18)x^5 + (-24z - 20)x^4 + (-104z - 168)x^3 \\ & + (656z + 1032)x^2 + (2640z + 4272)x + 2400z + 3895 \end{aligned}$$

$$\begin{aligned} g(x) = {} & -509498552799568921 74x^4 \\ & + (-203799421119827568696z - 611398263359482706088)x^3 \\ & + (20379942111982756869 6z - 118077227605609660905)x^2 \\ & + (34322351249456769147 18z + 540551926796116909 5450)x \\ & + 441887719645342300508 4z + 721266819494653653 7940 \end{aligned}$$

$$\begin{aligned} \phi(x) = {} & x^2 + (2z + 6)x + 4z \\ & + 13708736599510520364851080872883359648477543665871348424 6590 \end{aligned}$$

Note that $\|g\|_\infty \approx 2^{72}$.

Example 3. Consider $n = 12$ and p as above. Take $\eta = 3$ and $\kappa = 4$ and $d = 2$, $r = k = 2$.

$$h(z) = z^3 + 11z^2 + 15z + 3$$

$$\begin{aligned} f(x) = {} & x^6 + (18z^2 + 9z + 3)x^5 + (10341z^2 + 15912z + 3239)x^4 \\ & + (1770504z^2 + 2751672z + 562214)x^3 \\ & + (4945296z^2 + 7686900z + 1570620)x^2 \\ & + (4638000z^2 + 7209540z + 1473100)x + 1452552z^2 + 2257944z + 461359 \end{aligned}$$

$$
\begin{aligned}
g(x) = &-25901853699238517269631899865304470046 77x^4 \\
&+ (-31082224439086220723558279838365364056124z^2 \\
&\qquad - 155411122195431103617791399191826820280 62z \\
&\qquad - 518037073984770345392637997306089400935 4)x^3 \\
&+ (-89439100823470600132038950234896335071496 81z^2 \\
&\qquad - 13753884314295652670174538828476673594834870z \\
&\qquad - 27988495389355447626561153776604760933973 07)x^2 \\
&+ (-16808638345832811686298471405882039242411904z^2 \\
&\qquad - 26043481542097836103768559511213363723056322z \\
&\qquad - 53139195331316047895373728545815401349386 74)x \\
&- 79191221562541525593608033721095451223604 40z^2 \\
&- 12301715802165309681382520829319061454273924z \\
&- 25038154206255898395722035147006992207689 70
\end{aligned}
$$

$$
\begin{aligned}
&\phi(x) \\
&= x^4 + (12z^2 + 6z + 2)x^3 + (3453z^2 + 5310z \\
&\quad + 13153317136181854065289537242421374930975702769485783217444 14)x^2 \\
&\quad + (14642381046731513370058739760881745484966096865602985892600 46z^2 \\
&\qquad + 73211905233657566850293698804408727424830484328014929463683 3z \\
&\quad + 13153317136181854065289537242421374930975702769485783217453 85)x \\
&\quad + 14642381046731513370058739760881745484966096865602985892566 14z^2 \\
&\quad + 14642381046731513370058739760881745484966096865602985892583 06z \\
&\quad + 32222893200066469004691618582898863722025808633707935075847 5
\end{aligned}
$$

Note that we have $\|g\|_\infty \approx 2^{144}$. If we take $\eta = 2$, $\kappa = 6$ and $d = 2$. Taking $r = k$, we get the following polynomials.

$$h(z) = z^2 + 8z + 8$$

$$
\begin{aligned}
f(x) = x^8 &+ (4z + 12)x^7 + (-4z + 32)x^6 + (2z + 118)x^5 + (180z + 482)x^4 \\
&+ (196z + 672)x^3 + (-446z - 38)x^2 + (-853z - 683)x - 402z - 479
\end{aligned}
$$

$$
\begin{aligned}
g(x) = {} & 21466391773789776670008256922119248708347018 7x^6 \\
& + (643991753213693300100247707663577461250410561z \\
& \quad + 193197525964107990030074312299073238375123168 3)x^5 \\
& + 607350975571176840135562138552781845039042340 2x^4 \\
& + (340504035839159634206701463890050111189986519z \\
& \quad + 160479863491703225716258842371536244294128726 47)x^3 \\
& + (121470195114235368027112427710556369007808468 04z \\
& \quad + 497421081531868592425738917287385752419191265 21)x^2 \\
& + (181531745120800245290427510745274295100314086 68z \\
& \quad + 622318748863188771998232296871396849059226198 88)x \\
& + 110500393680173645241020694211871910664724602 24z \\
& + 528730449964899673120054076700143084546643667 84
\end{aligned}
$$

$$
\begin{aligned}
& \phi(x) \\
& = x^6 + (3z + 9)x^5 \\
& \quad + 133063783994622215345324551340069038202711820140877717368883 5x^4 \\
& \quad + (105433763563345403136452893446021816153203340903476151207600 4z \\
& \quad + 155607486264137181855162471103949188207389723332149170092647 1)x^3 \\
& \quad + (105433763563345403136452893446021816153203340903476151207605 9z \\
& \quad + 116695241386187743868313719006041839012002825541062319015466 4)x^2 \\
& \quad + (541504527394046900378336978445684610427996920133780742816963z \\
& \quad \quad + 315860038460594384818795094368482067510428549060075044498849)x \\
& \quad + 479545464544939099588903417708754059417518495058229028032985z \\
& \quad + 461243242085699950300079186153386681206871949929904822860464
\end{aligned}
$$

Note that $\|g\|_\infty \approx 2^{155}$.

5 Asymptotic Run Times for the Medium Prime Case

The norm bounds given by Proposition 1 are the same as those obtained in [25]. Consequently, the calculations for the asymptotic complexity follows almost verbatim from the calculations in [25] and leads to the following result for the classical NFS.

Theorem 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 with polynomials chosen by Algorithm $\mathcal{D}$ is $L_Q(1/3, 2c_b)$ where*

$$
c_b = \frac{2r+1}{3c_\theta kt} + \sqrt{\left(\frac{2r+1}{3c_\theta kt}\right)^2 + \frac{(t-1)kc_\theta}{3(r+1)}} \text{ and} \tag{10}
$$

$$
c_\theta = c_p c_\eta. \tag{11}
$$

The best complexity of $L_Q(1/3, (48/9)^{1/3})$ is achieved for $c_\theta = 12^{1/3}$, $r = k = 1$ and $t = 2$. The choice of $r = k = 1$ and $t = 2$ are not necessarily the best possible choices for other values of c_θ.

For the case of multiple NFS we obtain a similar result.

Theorem 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*

$$c_b = \frac{4r+2}{6kc_\theta t} + \sqrt{\frac{r(3r+2)}{(3kc_\theta t)^2} + \frac{(t-1)kc_\theta}{3(r+1)}} \text{ and} \tag{12}$$

$$c_\theta = c_p c_\eta. \tag{13}$$

From Theorem 2, the entire analysis carried out in Sections 8.1 and 8.2 of [25] apply with the constant c_p replaced by c_θ. This leads to the new asymptotic complexity results for the medium prime case that has been mentioned in the introduction.

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 [12]. For certain ranges of finite fields, the new method provides lower asymptotic complexity than the Conjugation method of Jeong and Kim.

References

1. Leonard 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. Leonard M. Adleman and Ming-Deh A. Huang. Function field sieve method for discrete logarithms over finite fields. *Inf. Comput.*, 151(1-2):5–16, 1999.
3. Razvan Barbulescu. An appendix for a recent paper of kim. *IACR Cryptology ePrint Archive*, 2015:1076, 2015.
4. Razvan Barbulescu, Pierrick Gaudry, Aurore Guillevic, and François 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. Razvan Barbulescu, Pierrick Gaudry, Antoine Joux, and Emmanuel 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. Razvan Barbulescu, Pierrick Gaudry, and Thorsten 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. Yuval Bistritz and Alexander Lifshitz. Bounds for resultants of univariate and bivariate polynomials. *Linear Algebra and its Applications*, 432(8):1995 – 2005, 2010. Special issue devoted to the 15th ILAS Conference at Cancun, Mexico, June 16-20, 2008.
8. Pierrick Gaudry, Laurent Grémy, and Marion Videau. Collecting relations for the number field sieve in $gf(p^6)$. Cryptology ePrint Archive, Report 2016/124, 2016. `http://eprint.iacr.org/`.
9. Daniel M. Gordon. Discrete logarithms in GF(p) using the number field sieve. *SIAM J. Discrete Math.*, 6(1):124–138, 1993.
10. Robert Granger, Thorsten Kleinjung, and Jens Zumbrägel. Discrete logarithms in GF(2^{9234}). *NMBRTHRY list*, January 2014.
11. Aurore Guillevic, Franois Morain, and Emmanuel Thomé. Solving discrete logarithms on a 170-bit mnt curve by pairing reduction. Cryptology ePrint Archive, Report 2016/507, 2016. `http://eprint.iacr.org/`.
12. Jinhyuck Jeong and Taechan Kim. Extended tower number field sieve with application to finite fields of arbitrary composite extension degree. Cryptology ePrint Archive, Report 2016/526, 2016. `http://eprint.iacr.org/`.
13. Antoine 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.
14. Antoine 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, 2013.
15. Antoine Joux and Reynald 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.
16. Antoine Joux and Reynald Lercier. Improvements to the general number field sieve for discrete logarithms in prime fields. A comparison with the gaussian integer method. *Math. Comput.*, 72(242):953–967, 2003.
17. Antoine Joux and Reynald 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.
18. Antoine Joux, Reynald Lercier, Nigel P. Smart, and Frederik 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.
19. Antoine Joux and Cécile 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.
20. Taechan Kim. Extended tower number field sieve: A new complexity for medium prime case. *IACR Cryptology ePrint Archive*, 2015:1027, 2015.
21. Taechan Kim and Razvan Barbulescu. Extended tower number field sieve: A new complexity for medium prime case. Cryptology ePrint Archive, Report 2015/1027, 2015. `http://eprint.iacr.org/`.
22. Cécile 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. Springer Berlin Heidelberg, 2015.
23. Palash Sarkar and Shashank Singh. Fine tuning the function field sieve algorithm for the medium prime case. *IEEE Transactions on Information Theory*, 62(4):2233–2253, April 2016. `http://ieeexplore.ieee.org/xpl/articleDetails.jsp?reload=true&arnumber=7405328`.
24. Palash Sarkar and Shashank Singh. A general polynomial selection method and new asymptotic complexities for the tower number field sieve algorithm. Cryptology ePrint Archive, Report 2016/485, 2016. `http://eprint.iacr.org/`.
25. Palash Sarkar and Shashank 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.
26. Palash Sarkar and Shashank Singh. Tower number field sieve variant of a recent polynomial selection method. Cryptology ePrint Archive, Report 2016/401, 2016. `http://eprint.iacr.org/`.
27. Oliver Schirokauer. Using number fields to compute logarithms in finite fields. *Math. Comp.*, 69(231):1267–1283, 2000.