Research Article

Kazuhiro Yokoyama*, Masaya Yasuda, Yasushi Takahashi, and Jun Kogure

Complexity bounds on Semaev's naive index calculus method for ECDLP

https://doi.org/10.1515/jmc-2019-0029
Received Jul 15, 2019; accepted Aug 05, 2020

Abstract: Since Semaev introduced summation polynomials in 2004, a number of studies have been devoted to improving the index calculus method for solving the elliptic curve discrete logarithm problem (ECDLP) with better complexity than generic methods such as Pollard's rho method and the baby-step and giant-step method (BSGS). In this paper, we provide a deep analysis of Gröbner basis computation for solving polynomial systems appearing in the point decomposition problem (PDP) in Semaev's *naive* index calculus method. Our analysis relies on linear algebra under simple statistical assumptions on summation polynomials. We show that the ideal derived from PDP has a special structure and Gröbner basis computation for the ideal is regarded as an extension of the extended Euclidean algorithm. This enables us to obtain a lower bound on the cost of Gröbner basis computation. With the lower bound, we prove that the naive index calculus method cannot be more efficient than generic methods.

Keywords: ECDLP, Summation polynomials, Index calculus methods, Gröbner basis computation, Fall degrees

2010 Mathematics Subject Classification: Primary 94A60, Secondary 14G50

1 Introduction

The RSA cryptosystem [30] and the elliptic curve cryptography (ECC) [20, 25] are the most widely used systems in modern information society. The security of RSA is based on the hardness of the integer factorization problem (IFP), whereas that of ECC is based on the hardness of the elliptic curve discrete logarithm problem (ECDLP). While there exist sub-exponential time algorithms for solving IFP, no such algorithm exists for solving ECDLP with cryptographic parameters. The best-known practical algorithm for solving ECDLP is Pollard's rho method [29], except in special cases such as supersingular [24] and anomalous curves [31, 32, 36]. Its complexity is the square root of the order of an elliptic curve over a finite field (see [3, 16] for details). The largest known records for solving ECDLP are currently 112-bit over prime fields [4], 117.35-bit over binary fields [2], and 113-bit over Koblitz curves [41], which were all solved by the parallelized rho method. (See also [42] for a hardness comparison of ECDLP and IFP.)

The index calculus method is useful for solving the discrete logarithm problem (DLP) over a general cyclic group; the basic concept behind this method is to reduce DLP to a linear algebra problem by collecting relations over a so-called factor base. After Semaev [33] introduced summation polynomials associated with an elliptic curve, one can solve ECDLP in the index calculus framework. In particular, finding a relation is reduced to a problem of solving a system including Semaev's summation polynomials in order to decompose a point of an elliptic curve into the sum of points of a factor base. Semaev's paper [33] triggered many studies aiming

***Corresponding Author: Kazuhiro Yokoyama:** Rikkyo University, Tokyo, Japan; Email: kazuhiro@rikkyo.ac.jp
Masaya Yasuda: Rikkyo University, Tokyo, Japan; Email: myasuda@rikkyo.ac.jp
Yasushi Takahashi: FUJITSU Laboratories LTD., Kawasaki, Japan; Email: t-yasushi@fujitsu.com
Jun Kogure: FUJITSU Laboratories LTD., Kawasaki, Japan; Email: kogure@jp.fujitsu.com

to develop index calculus algorithms with better complexity than the rho method. For instance, Gaudry [14] and Diem [6] fully developed Semaev's approach based on Weil descent for ECDLP over a finite field $\mathbb{F}_{q^n}$ with an extension degree n. In 2012, Faugère et al. [11] showed that a system arising in algorithms based on Weil descent has a special structure, making it easier to solve the system using Gröbner basis algorithms, which are based on Buchberger's criterion, that is, using S-polynomials such as F_4 and F_5 [8, 9]. Subsequently, Petit and Quisquater [28] revisited Faugère et al.'s work to claim the subexponentiality for ECDLP over any binary field under the heuristic assumption of the *first fall degree* (FFD) regarding the behavior of Gröbner basis algorithms. However, in 2015, Huang et al. [19] provided computational evidence that raised doubt on the validity of the FFD assumption and introduced another notion called the *last fall degree* (LFD) to develop complexity bounds for solving a polynomial system. As mentioned in a survey [13], there was no consensus (not at all definitive) whether there exists a sub-exponential time algorithm to solve ECDLP over binary fields, since the FFD assumption is too optimistic whereas the LFD approach looks more precise but more difficult to estimate. For prime fields $\mathbb{F}_p$, Petit et al. [27] in 2016 provided an approach of index calculus, which works when $p - 1$ has a smooth factor. In 2018, Amadori et al. [1] proposed a variant of index calculus for solving arbitrary ECDLP to decrease the required number of Gröbner basis computations. An experimental study by [22] showed that both methods of [1, 27] do not outperform generic algorithms such as the rho method for solving ECDLP over prime fields.

In this paper, we provide a *deep* analysis of Gröbner basis computations for a polynomial system using the naive index calculus method with Semaev's summation polynomials. Considering the special structure of the ideal associated with the index calculus method, we regard its Gröbner basis computation using S-polynomials as an extension of the *Euclidean algorithm* for the polynomial greatest common divisor (GCD). We also use the shape of its *coefficient polynomial* (see Equation (11) later in the text) to count the number of operations in Gröbner basis computation. In particular, we observe that *leading monomials* of coefficient polynomials can provide precise information on S-polynomials. Under simple statistical assumptions on the behaviors of the summation polynomials, the sequence of S-polynomials behaves similarly to *normal* polynomial reminder sequences in GCD computation in the context of the subresultant theory. (See [26] for this theory.) This enables us to obtain a *lower bound* on the complexity of Gröbner basis computation using S-polynomials (Sections 4.1 and 4.2), under such statistical assumptions. Compared to the assumptions on fall degrees in [19, 28], our assumptions are simple and expected to hold in practice. Through experiments over prime fields $\mathbb{F}_p$ with up to 25 bits p, we examined the validity of our assumptions (Section 4.3). Our analysis also provides *precise* results related to the notions of *degree bound* and *degree fall* (Section 5.2). With our complexity bounds, we prove that the naive index calculus method cannot be more efficient than the rho method over an arbitrary finite field. Moreover, based on our experimental observations, it cannot be more efficient than even the brute force method.

2 Index calculus method for ECDLP

In this section, we recall ECDLP and the index calculus method to solve it with Semaev's summation polynomials [33]. From now on, we assume that each elliptic curve is defined by a short Weierstrass form.

Definition 1 (ECDLP). *Let $\mathbb{F}_q$ be the finite field with q elements and let E be an elliptic curve defined over $\mathbb{F}_q$. Let $E(\mathbb{F}_q)$ denote the group of $\mathbb{F}_q$-rational points of E. Given two points P, Q in $E(\mathbb{F}_q)$ with $Q \in \langle P \rangle$, ECDLP asks to find an integer ℓ satisfying $Q = \ell P$. Such an integer ℓ is unique up to modulo n, where n denotes the order of P in $E(\mathbb{F}_q)$. (In cryptography, we assume that n is a huge prime.) In particular, such an integer ℓ with $0 \leq \ell < n$ is denoted by* $\log_P(Q)$.

2.1 Outline of index calculus method

Here, we provide an outline of index calculus methods for solving ECDLP. (Through this paper, we assume that $E(\mathbb{F}_q) = \langle P \rangle$ is a cyclic group of prime order n.) We note that $A \approx B$ implies that A is *almost the same* as B, that is, $A = B^{1+\varepsilon}$ for a number ε with $|\varepsilon| \ll 1$, and $A \gtrapprox B$ implies that $A \approx B$ or $A > B$.

Step 1: (Construction of factor base). Take a subset $\mathcal{B}$ of $E(\mathbb{F}_q)$, which is called a *factor base*. In the *naive* setting, we randomly take a subset V of $\mathbb{F}_q$ and set $\mathcal{B} = \{(x, y) \in E(\mathbb{F}_q) \mid x \in V\}$. Set $d = \#V$ and $t = \#\mathcal{B}$. Then, we have $d \approx t$ with high probability, since the ratio of (x, y) belonging to $E(\mathbb{F}_q)$ is almost $\frac{1}{2}$ for each $x \in V$.

Step 2: (Collection of relations). Generate randomly two integers a, b with $0 \le a, b < n$, and find a decomposition

$$aP + bQ = B_1 + \cdots + B_m \quad (B_i \in \mathcal{B}) \tag{1}$$

for some m. This problem is called the point decomposition problem (PDP) for the point $aP+bQ \in E$. When it succeeds in solving the problem, we obtain the linear relation

$$a + b\log_P(Q) \equiv \sum_{i=1}^{m} \log_P(B_i) \bmod n.$$

Step 3: (Linear algebra). After collecting $t + 1$ *linearly independent* relations, the solution $\log_P(Q)$ can be recovered by linear algebra.

2.2 Summation polynomials

Here, we revisit an algebraic approach for solving PDP by Semaev [33]. Specifically, PDP can be reduced to finding zeros of the system of algebraic equations derived from Semaev's summation polynomials.

Definition 2 (Summation polynomials). *For an integer $r \ge 2$, a polynomial $S_r(x_1, \ldots, x_r)$ over $\mathbb{F}_q$ in r variables $x_1, \ldots, x_r$ is called a* summation polynomial *of order r associated with an elliptic curve E over $\mathbb{F}_q$, if it satisfies the following property: For any zero $(\alpha_1, \ldots, \alpha_r)$ of S_r, there exist points $P_1, \ldots, P_r$ in $E(\overline{\mathbb{F}_q})$ such that $P_1 + \cdots + P_r = \mathcal{O}$ and $\alpha_i = x(P_i)$ for $1 \le i \le r$, where $\mathcal{O}$ denotes the infinity point of E and $x(P)$ the x-coordinate of a point P in E with $P \neq \mathcal{O}$. (We denote the algebraic closure of a field K by $\overline{K}$.)*

Semaev [33] described the construction of summation polynomials over any field by resultants in a recursive manner. The summation polynomial of order 3 is constructed from an elliptic curve, and this is explicitly given in [33]. The summation polynomial of order $r \ge 4$ can be computed from two summation polynomials of orders $r-1$ and 3 as $S_r = \mathrm{Res}_y(S_{r-1}(x_1, \ldots, x_{r-2}, y), S_3(x_{r-1}, x_r, y))$, where $\mathrm{Res}_y(f, g)$ denotes the resultant of two polynomials f, g with respect to y. This can be generalized as follows: $S_{r+k} = \mathrm{Res}_y(S_{r+1}(x_1, \ldots, x_r, y), S_{k+1}(x_{r+1}, \ldots, x_{r+k}, y))$. Semaev's summation polynomial S_r is symmetric in its variables for every $r \ge 3$. For each variable x_i, we have $\deg_{x_i}(S_r) = 2^{r-2}$ and $\deg(S_r) = (r-1)2^{r-2}$, where $\deg_{x_i}(f)$ is the degree of a multivariate polynomial f with respect to the variable x_i and by $\deg(f)$ its total degree. (See Section 4.1.1.)

Remark 1. *It is computationaly hard to compute S_r for $r > 7$ by resultant compuation. However, by using symmetry, a simplified modification (called symmetrized summation polynomials) was proposed in [10], by which a polynomial of order 8 was computed. The computational cost of S_r requires $O(2^{c(r-1)^2})$ arithmetic operations over $\mathbb{F}_q$ for a constant c.*

2.3 Point decomposition using summation polynomials

PDP in Step 2 can be reduced to finding zeros of a system of algebraic equations using the summation polynomial S_{m+1}. For $a, b \in \mathbb{F}_q$, finding m elements $B_1, \dots, B_m$ in the factor base $\mathcal{B}$ satisfying the relation (1) is reduced to finding a zero $(\alpha_1, \dots, \alpha_m) \in V^m$ of $S_{m+1}(x_1, \dots, x_m, x(aP + bQ)) = 0$, where each α_i corresponds to $x(B_i)$. We set a univariate polynomial as

$$F(x) = \prod_{\alpha \in V} (x - \alpha), \tag{2}$$

which is a factor of $x^q - x$ over $\mathbb{F}_q$ since $V \subseteq \mathbb{F}_q$. Then, we consider the following system of algebraic equations:

$$\begin{cases} S_{m+1}(x_1, \dots, x_m, x(aP + bQ)) & = & 0, \\ F(x_1) & = & 0, \\ \vdots & & \vdots \\ F(x_m) & = & 0. \end{cases} \tag{3}$$

Thus, solving PDP for a point $aP + bQ \in E$ is reduced to computing zeros of the system (3). In general, Gröbner basis computation is supposed to be the most efficient among algebraic methods for finding the zeros. Specifically, we perform Gröbner basis algorithms for the ideal associated with the system (3), that is,

$$I_m(a, b) = \langle S_{m+1}(x_1, \dots, x_m, x(aP + bQ)), F(x_1), \dots, F(x_m) \rangle, \tag{4}$$

where $\langle \mathcal{A} \rangle$ denotes the ideal generated by a set $\mathcal{A}$ of multivariate polynomials in the ring $\mathbb{F}_q[x_1, \dots, x_m]$. Since all zeros of the system (3) belong to $\mathbb{F}_q$ if they exist, the ideal $I_m(a, b)$ is trivial or 0-dimensional in commutative algebra.

2.4 General complexity estimation

Here, we provide a general but brief discussion on the complexity of the naive index calculus method. (The complexity depends on two parameters, m and $t = \#\mathcal{B}$.)

We let $\mathcal{P}$ denote the probability of successfully solving PDP for a random point $aP + bQ \in E$. It can be estimated approximately as

$$\frac{\#\{B_1 + \cdots + B_m \mid B_i \in \mathcal{B}\}}{\#E(\mathbb{F}_q)}.$$

More precisely, we need to consider how many m points in the factor base $\mathcal{B}$ can generate the same point; we denote the average by γ. (We expect $\gamma \geq m!$ since $B_{\sigma(1)} + \cdots + B_{\sigma(m)}$ gives the same point for any permutation σ on $\{1, \dots, m\}$.) Then, the probability $\mathcal{P}$ can be estimated approximately as $\frac{t^m}{\gamma q}$. We also let $\mathcal{C}$ denote the average cost of solving PDP for a random point $aP + bQ$. It includes the cost of constructing the summation polynomial S_{m+1} and the average cost of solving the system (3). We note that PDP has γ solutions on average. However, the number of *essential* solutions giving distinct points can be estimated roughly as $\frac{\gamma}{m!}$ due to the symmetry of the solutions. Since more than t (linearly independent) solutions are required, the total cost (on the number of arithmetic operations over $\mathbb{F}_q$) of solving ECDLP can be estimated approximately as

$$\frac{m! \cdot t \cdot \mathcal{C}}{\gamma \cdot \mathcal{P}} + \mathrm{Lin}(t). \tag{5}$$

Here, $\mathrm{Lin}(t)$ denotes the cost of solving a system of more than t linear equations in t variables, and it is estimated as $O(t^\omega)$ for the linear algebra constant $2 < \omega < 3$.

When we set t to be sufficiently large such that $\mathcal{P} \approx 1$ (this requires $t^m \gtrapprox \gamma q$), the total cost is estimated as

$$\frac{m! \cdot t \cdot \mathcal{C}}{\gamma} + \mathrm{Lin}(t) \gtrapprox m! \cdot q \cdot t^{1-m} \cdot \mathcal{C} + \mathrm{Lin}(t). \tag{6}$$

We next consider the case $\mathcal{P} < 1$, in which we may assume $t^m < m!q$. As long as $\frac{t^m}{m!q} = o(1)$, it holds $\gamma \approx m!$ and the probability $\mathcal{P}$ is almost equal to $\frac{t^m}{m!q}$ [34]. When we write $t = q^e$, the total cost can be estimated from (5) as

$$\begin{aligned}\frac{m! \cdot t \cdot \mathcal{C}}{\gamma\mathcal{P}} + \mathrm{Lin}(t) &\approx m! \cdot q \cdot t^{1-m} \cdot \mathcal{C} + \mathrm{Lin}(t) \\ &= m! \cdot q^{1-e(m-1)} \cdot \mathcal{C} + \mathrm{Lin}(q^e).\end{aligned} \tag{7}$$

This cost should be at most $q^{1/2}$ to make the index calculus method more efficient than generic algorithms such as Pollard's rho method. This implies that from the first term of (7), it should hold that $1 - e(m-1) < \frac{1}{2} \Leftrightarrow \frac{1}{2} < e(m-1)$, and the cost $m! \cdot \mathcal{C}$ should also be significantly smaller than $q^{1/2}$. (Note that only a small m is available for construction of the polynomial S_{m+1}, as mentioned in Remark 1.) In addition, it should hold that $e \leq \frac{1}{2\omega}$ from the last term $\mathrm{Lin}(q^e) = O(q^{e\omega})$.

3 Gröbner basis computation for special type ideals

Let $R = K[x_1, \ldots, x_m]$ be the multivariate polynomial ring over a field K. An ideal of R is said to be a *special type* if it is generated by m univariate polynomials $F(x_1), \ldots, F(x_m)$ and one multivariate polynomial $S(x_1, \ldots, x_m)$ for a square-free $F(x)$. The ideal $I_m(a, b)$ defined in (4) is a special type since it is generated by the set $\{F(x_1), F(x_2), \ldots, F(x_m), S_{m+1}(x_1, \ldots, x_m, x(aP + bQ))\}$ for a random factor $F(x)$ of $x^q - x$ over $\mathbb{F}_q$ with $\deg F = d$. (See (2) for $F(x)$, where V is a random subset of $\mathbb{F}_q$ with $\#V = d$.) We supposed that $d \geq 2$.

In this section, we analyze the cost of Gröbner basis computation for special type ideals. Below, we list several notations related to Gröbner basis computation. (We follow the notations in [5] for the terminology on monomials and ordering.)

- $\mathcal{M} = \{x_1^{e_1} \cdots x_m^{e_m} \mid e_i \in \mathbb{Z}_{\geq 0}\}$: the set of all monomials in the ring R.
- $\prec$: a graded monomial ordering (or a reverse lexicographic ordering on the set $\mathcal{M}$, since such a monomial ordering is the most efficient monomial ordering in both theory and practice).
- $\mathrm{supp}(f)$: the set of monomials appearing in a polynomial $f \in R$, that is, $f = \sum_{t \in \mathrm{supp}(f)} a_t t$ with $a_t \in K \setminus \{0\}$.
- $LC(f)$, $LM(f)$, $LT(f)$: the leading coefficient, monomial and term of a polynomial f with respect to $\prec$ (it satisfies $LT(f) = LC(f) \cdot LM(f)$).
- $\mathrm{Spol}(f, g)$: the S-polynomial of two polynomials f and g defined as

$$\frac{\mathrm{lcm}(LM(f), LM(g))}{LT(f)} f - \frac{\mathrm{lcm}(LM(f), LM(g))}{LT(g)} g,$$

  where $\mathrm{lcm}(u, v)$ denotes the least common multiple of $u, v \in \mathcal{M}$.
- $LM(U) = \{LM(f) \mid f \in U\}$: the set of leading monomials for a set U.
- $\langle \mathcal{S} \rangle$: the ideal of R generated by a subset $\mathcal{S}$ of R.
- $V(L)$: the set of all zeros of an ideal L of R over $\overline{K}$.

We recall that for an ideal I of R and a monomial ordering $\prec$, its subset $\mathcal{G}$ is a Gröbner basis of I with respect to $\prec$ if $LM(\mathcal{G})$ generates the (monomial) ideal generated by $LM(I)$. For our analysis, we count the number of *S-polynomials* during Gröbner basis computation where the notion of S-polynomial plays a key role in the following criterion. (See Theorem 6 of Chapter 2.6 in [5].)

Theorem 1 (Buchberger's Criterion). *For an ideal I of R and a monomial ordering $\prec$, its generating set $\mathcal{G}$ is a Gröbner basis with respect to $\prec$ if and only if the* remainder on division *of the S-polynomial* $\mathrm{Spol}(g_1, g_2)$ *by $\mathcal{G}$ is* 0 *for all pairs $\{g_1, g_2\}$ of $\mathcal{G}$. Here, the remainder on division means the final consequence of a number of (monomial) divisions with respect to leading monomials.*

Given a generating set $\mathcal{G}$, we can construct a Gröbner basis by *adding* the (non-zero) remainder on division of a computed S-polynomial to $\mathcal{G}$ incrementally. This is so-called *Buchberger's algorithm*. (See [5] and also Section 4.2.)

We introduce the additional notion of *signature* and consider the so called *signature-based algorithms* which use signatures and are the most efficient algorithms for computing the Gröbner basis of a given ideal of special type. Signature-based algorithms, such as F_5 or its variants, are designed to discard *unnecessary S-polynomials* as much as possible, by using the signature. By the famous F_5-criterion, for each signature s, once an S-polynomial with signature s is computed, other S-polynomials with the same signature are unnecessary. (See [7] and its simple usage for univariate case in Remark 2.) Since *division operations* are restricted to keep the signature, it is not proven rigidly that signature-based algorithms always produce fewer S-polynomials compared with other algorithms based on Buchberger's criterion. However, due to the special structure of the ideal, we can see that other algorithms using an efficient technique called *normal selection strategy* exhibit similar computational behavior. See Section 4.1 for the details.

3.1 Univariate case

We begin with the univariate case $m = 1$. In this case, any ideal of special type I is generated by two univariate polynomials $f(= S)$ and $g(= F)$. Furthermore, Gröbner basis computation for the ideal $I = \langle f, g\rangle$ just corresponds to the Euclidean algorithm for computing the GCD $\gcd(f, g)$.

Euclidean algorithm for GCD

Here, we briefly recall the Euclidean algorithm and its related notions. (For the fundamentals, refer to [26, 40].) For two polynomials f and g in R, there exist two polynomials A, B satisfying $\gcd(f, g) = Af + Bg$. We here call such A, B *coefficient polynomials*. By keeping track of the Euclidean algorithm, we obtain the *polynomial remainder sequence* (PRS) as

$$f_0 = f, f_1 = g, f_2, \cdots, f_r = \gcd(f, g). \tag{8}$$

Each f_{i+1} is the *remainder* of f_{i-1} on the division by f_i as

$$f_{i-1} = q_i f_i + f_{i+1}, \tag{9}$$

for some quotient polynomial q_i. Using the extended Euclidean algorithm, the coefficient polynomials A_i, B_i for each f_i $(i \geq 2)$ are computed as

$$f_i = A_i f + B_i g \tag{10}$$

with $\deg(A_i) < \deg(g) - \deg(f_i)$ and $\deg(B_i) < \deg(f) - \deg(f_i)$.

Correspondence to Gröbner basis computation

Buchberger's algorithm is the most basic algorithm for finding Gröbner bases, in which S-polynomials are computed for two distinct polynomials. In particular, the division (9) exactly corresponds to producing a new basis for the ideal $I = \langle f, g\rangle$ in Buchberger's algorithm; Letting $d_i = \deg(f_i)$ for $0 \leq i \leq r$, we have $LM(f_{i-1}) = x^{d_{i-1}}$, $LM(f_i) = x^{d_i}$ and $\mathrm{lcm}(LM(f_{i-1}), LM(f_i)) = x^{d_{i-1}}$ since $d_{i-1} > d_i$. By its definition, the S-polynomial $\mathrm{Spol}(f_{i-1}, f_i)$ is equal to (up to constant multiples)

$$f_{i-1} - \frac{LC(f_{i-1})}{LC(f_i)} x^{d_{i-1}-d_i} f_i.$$

This polynomial may be still divisible by f_i, and its remainder on division by f_i coincides with f_{i+1}. This reduction process is denoted by $\mathrm{Spol}(f_{i-1}, f_i) \xrightarrow{f_i} f_{i+1}$.

Remark 2. *In Buchberger's algorithm, we compute the S-polynomial for every distinct pair* (f_i, f_j)*. We may discard* unnecessary *S-polynomials using the notion of* signature. *(See [7] for a recent survey of signature-based algorithms.) In the above case, by setting the signature of* f_i *as* $LM(A_i)$*, S-polynomials are dealt with in ascending order of signatures, and at most one S-polynomial is computed for each signature. This implies that, for each* i*, we need only to compute* $\mathrm{Spol}(f_{i-1}, f_i)$ *and we can discard* $\mathrm{Spol}(f_j, f_i)$ *for* $j < i - 1$.

On the computation cost

The total degree of the polynomial Af is useful for obtaining an upper bound on the total cost of the Euclidean algorithm. (It exactly corresponds to the *regularity* of the ideal $I = \langle f, g\rangle$ and thus gives a degree bound. See Section 5.) In fact, the computation of $\gcd(f, g)$ can be reduced to solving a linear system by introducing indeterminate coefficients for possible A and B. We may extend this approach to estimate the complexity of Gröbner basis computation. (See [11] for example.) By contrast, monomials of A and B may be considered as *data* containing all histories on which *monomial multiplications* are used in the Euclidean algorithm. In particular, as $LM(A_i)$ can be the signature of f_i, they can tell us which S-polynomials appear in Gröbner basis computation.

Example 1. *We consider* $f = x^{d+1} + 1$ *and* $g = x^d$ *for* $d \geq 1$*. We see that* $\gcd(f, g) = 1$ *and the total degree of* A *is not greater than* d*. However, we have* $A = 1$ *and* $B = -x$*, and the Euclidean algorithm terminates within 1 step. This trivial example implies that the shape of* A *is important for estimating the cost.*

Example 2. *We suppose that the PRS* (8) *is* normal[1]*. Thus,* $\deg(f_{i-1}) = \deg(f_i) + 1$ *for* $i \geq 2$*. This is considered as the* worst case *for the Euclidean algorithm. For simplicity, we assume that* $\deg(f) = \deg(g) + 1$ *and let* $d = \deg(g)$ *and* $D = \deg(\gcd(f, g))$*. Then, we have* $\deg(f_i) = d + 1 - i$, $\deg(A_i) = i - 2$ *for each* $i \geq 2$*, and it requires* $d - D$ *polynomial divisions. The following equations illustrate the correspondence between PRS and S-polynomials (here* $q_i = a_i x + b_i$ *for each* i*):*

$$\begin{array}{ll}
f_2 = f_0 - q_1 f_1 & \mathrm{Spol}(f_0, f_1) = f_0 - a_1 x f_1 \xrightarrow{f_1} f_2 \\
f_3 = f_1 - q_2 f_2 & \mathrm{Spol}(f_1, f_2) = f_1 - a_2 x f_2 \xrightarrow{f_2} f_3 \\
\quad\vdots & \qquad\qquad\vdots \\
f_k = f_{k-2} - q_{k-1} f_{k-1} & \mathrm{Spol}(f_{k-2}, f_{k-1}) = f_{k-2} - a_{k-1} x f_{k-1} \xrightarrow{f_{k-1}} f_k \\
\quad\vdots & \qquad\qquad\vdots
\end{array}$$

3.2 Multivariate case

From this subsection, we consider the multivariate case $m \geq 2$. Let I be an ideal of the multivariate polynomial ring $R = K[x_1, \ldots, x_m]$ over a field K. Assume that I is a special type ideal, which is generated by the set

$$\mathcal{F}_S = \{S(x_1, \ldots, x_m)\} \cup \mathcal{F} \quad \text{with} \quad \mathcal{F} = \{F(x_1), \ldots, F(x_m)\}$$

for a square-free univariate polynomial F and a multivariate polynomial S. Set $d = \deg(F)$ and $d_S = \deg(S)$. We let J denote the ideal generated by the set $\mathcal{F}$. Then $I = J + \langle S\rangle$. (It is an auxiliary ideal to analyze the structure of the special type ideal I.)

Since leading monomials of $F(x_1), \ldots, F(x_m)$ are coprime, the set $\mathcal{F}$ is the reduced Gröbner basis of J with respect to any monomial ordering by Buchberger's criterion. (See Chapter 2 in [5].) For simplicity, we assume that $S \neq 0$ and S is reduced with respect to $\mathcal{F}$. Then, I is strictly larger than J. Since each $F(x_i)$ is square-free, it can be shown that two ideals I and J are radical. (See Proposition 4.5.1 in [15].) This is because $J \cap K[x_i]$ is

[1] Here, *normal* is used as in Chapter 11 in [40]. This is also called *regular* in Chapter 7.7 in [26].

generated by the square-free polynomial $F(x_i)$ for each i, and hence $I \cap K[x_i]$ is generated by its factor, which is also square-free. Thus, $\sqrt{I} = I \supsetneq J = \sqrt{J}$, and $V(I)$ is a proper subset of $V(J)$ by Hilbert's Nullstellensatz. (See Theorem 7 of Chapter 4 Section 2 in [5].)

3.2.1 Representation of Gröbner basis elements

Here, we regard Gröbner basis computation using S-polynomials for the ideal $I = \langle \mathcal{F}_S \rangle$ as an extension of the Euclidean algorithm. Specifically, we represent each $f \in I$ as

$$f = TS + A_1 F(x_1) + \cdots + A_m F(x_m) \tag{11}$$

for some $T, A_1, \ldots, A_m \in R$ (cf., the representation (10) in the univariate case). We focus on the "shape" of the "S-coefficient" T in our analysis. Note that since S is assumed to be reduced with respect to the set $\mathcal{F}$, every univariate polynomial $F(x_i)$ cannot reduce S, and thus it holds that $\deg_{x_i}(S) < d = \deg(F)$.

Definition 3 (Syzygy). *We call the ideal quotient of $J = \langle \mathcal{F} \rangle$ by the polynomial S the* syzygy ideal *with respect to S, and it is denoted as*

$$\mathrm{Syz} = (J : S) = \{f \in R \mid fS \in J\}.$$

Each element $h \in$ Syz gives a syzygy *among the polynomials in $\mathcal{F}_S$, that is, it satisfies $hS + B_1 F(x_1) + \cdots + B_m F(x_m) = 0$ for some $B_1, \ldots, B_m \in R$.*

With the syzygy ideal, we divide the set of monomials into two subsets as

$$\mathcal{M} = LM(\mathrm{Syz}) \sqcup NS(\mathrm{Syz})$$

with $NS(\mathrm{Syz}) = \mathcal{M} \setminus LM(\mathrm{Syz})$. When we take a Gröbner basis $\mathcal{H}$ of Syz with respect to the fixed ordering $\prec$, we can write

$$NS(\mathrm{Syz}) = \{t \in \mathcal{M} \mid LM(g) \nmid t \text{ for any } g \in \mathcal{H}\}$$

from the basic properties of Gröbner bases. We set $\mathcal{M}_{red} = \mathcal{M} \setminus LM(J)$. Then, it follows from the simple structure of the ideal $J = \langle \mathcal{F} \rangle$ that we represent

$$\mathcal{M}_{red} = \{x_1^{e_1} \cdots x_m^{e_m} \mid 0 \le e_i < d \text{ for } 1 \le i \le m\}.$$

Moreover, we obtain $NS(\mathrm{Syz}) \subseteq \mathcal{M}_{red}$ since $J \subseteq$ Syz and $LM(J) \subseteq LM(\mathrm{Syz})$.

Standard form and signature

Here, we consider a certain *minimal* representation of elements of I with respect to $\mathcal{F}_S$. The notion of *signatures* is useful for this purpose. (See [7, 9, 37, 38] for details on signatures.)

For each element $f \in I \setminus J$, there exist polynomials $T, A_1, \ldots, A_m$ satisfying (11). We call such $T, A_1, \ldots, A_m$ *coefficient polynomials*. Using the syzygy ideal, we can refine this representation. Let $\tilde{T}$ be the normal form (i.e., the remainder) of T with respect to the Gröbner basis $\mathcal{H}$ of Syz. Then, $\mathrm{supp}(\tilde{T}) \subseteq NS(\mathrm{Syz})$ and $T - \tilde{T} \in$ Syz. Since $(T - \tilde{T})S \in J$ by Definition 3, the polynomial $f - \tilde{T}S$ belongs to the ideal $J = \langle \mathcal{F} \rangle$. Moreover, by considering the *standard representation* of $f - \tilde{T}S$ with respect to the Gröbner basis $\mathcal{F}$ of J, there exist polynomials $\tilde{A}_1, \ldots, \tilde{A}_m$ such that $LM(\tilde{A}_i F(x_i)) \preceq LM(f - \tilde{T}S)$ for every i and

$$f = \tilde{T}S + \tilde{A}_1 F(x_1) + \cdots + \tilde{A}_m F(x_m). \tag{12}$$

(See Chapter 2 Section 9 in [5] for standard representation.) Here, we call the representation (12) the *standard form* of f. Note that $\tilde{T}$ is determined uniquely and does not depend on the construction. We have $LM(\tilde{T}S) \succeq LM(f)$ if f is reduced with respect to $\mathcal{F}$. This is because, if $LM(\tilde{T}S) \prec LM(f)$, then $LM(f - \tilde{T}S) = LM(f)$ cannot be reduced by $LM(\mathcal{F})$.

Definition 4 (Signatures). *We call the coefficient* $\tilde{T}$ *in* (12) the reduced S-coefficient *of* $f \in I \setminus J$ *and denote it by* $\mathrm{RSC}(f)$. *We also call its leading monomial the* signature *of* f *and denote it by* $\mathrm{sig}(f)$. *We set the signature of* $f \in J$ *as* 0.

For every monomial $t \in NS(\mathrm{Syz})$, we see that the polynomial tS has its signature t. Thus, $NS(\mathrm{Syz})$ is regarded as the set of *non-zero signatures*.

Remark 3. *As another definition, the signature of f is defined by keeping its computational record; this definition may differ from ours. Thus, the signature by our definition may be called the* minimal *signature. We can make both definitions coincide by handling S-polynomials as small as possible during Gröbner basis computaion. In the general setting, the signature is defined using a* module monomial ordering *and there are several module monomial orderings. Our definition corresponds to* position-over-terms (POT) orderings *and also to* Schreyer orderings. *(See Chapter 10 Section 4 in [5] and Chapter 2 Section 5 in [15].) Considering the special structure of the ideal, our definition is quite* natural *and* supposed *to make the computation the most efficient. Moreover, it is also suited for comparing the computational behavior with those of known efficient Gröbner basis algorithms. We give details in Remark 8.*

3.2.2 Applications of the set of non-zero signatures

As an easy application, we first show that the regularity of the ideal I can be estimated from the set of non-zero signatures $NS(\mathrm{Syz})$. We set

$$Reg = md + d_S - m \tag{13}$$

as an important number for our later analysis, which is related to a certain *regularity coming from Hilbert polynomials*. (We discuss this in Section 5.)

Lemma 2. *Let $\mathcal{G}$ denote the reduced Gröbner basis of the ideal $I = \langle \mathcal{F}_S \rangle$. For every element g in $\mathcal{G}$, it holds that* $\deg(g) \le Reg$.

Proof. If $g \in J = \langle \mathcal{F} \rangle$, then we have $g = F(x_i)$ for some i since it is reduced with respect to $\mathcal{F}$. Thus, $\deg(g) = d \le Reg$ since $m \ge 2$. Next, we consider $g \in I \setminus J$. Then, it holds that $\mathrm{supp}(\mathrm{RSC}(g)) \subseteq NS(\mathrm{Syz})$ from the construction. Since $NS(\mathrm{Syz}) \subseteq \mathcal{M}_{red}$, we have $\deg(\mathrm{RSC}(g)) \le md - m$. Moreover, since g is reduced with respect to $\mathcal{F}$, it holds that $LM(\mathrm{RSC}(g)S) \succeq LM(g)$. Thus, we have $\deg(g) \le \deg(\mathrm{RSC}(g)S) = \deg(\mathrm{RSC}(g)) + \deg(S) \le Reg$. □

Next, we estimate the number of non-zero signatures $\#NS(\mathrm{Syz})$, which plays an important role in our complexity analysis. Using the decomposition formula, we obtain $\sqrt{J} = \sqrt{J + \langle S \rangle} \cap \sqrt{J : S}$. (See chapters related to primary decomposition in [15, 39].) As J is radical and 0-dimensional, any ideal including J is also radical. (See Proposition 4.5.1 in [15].) Thus, $(J : S)$ is also radical, and we have $J = (J + \langle S \rangle) \cap (J : S) = I \cap (J : S)$. This induces the following exact sequence (see Exercise 5.3.3 in [15]):

$$0 \to R/J \to R/I \oplus R/(J : S) \to R/(I + (J : S)) \to 0.$$

We recall that for every 0-dimensional ideal L of R, the linear dimension of the residue class ring R/L coincides with the number of monomials in $\mathcal{M} \setminus LM(L)$. Furthermore, if L is radical, the linear dimension coincides with the number of its zeros over $\overline{K}$. (See Proposition 7 of Chapter 5 Section 3 in [5].) From this fact and the above exact sequence, we obtain

$$\#NS(\mathrm{Syz}) = \#V(J : S) = \#V(J) + \#V(I + (J : S)) - \#V(I).$$

We note that $I + (J : S)$ is also radical since it includes J.

Proposition 3. *We have* $\#NS(\mathrm{Syz}) = \#V(J) - \#V(I)$. *Thus, we estimate* $\#NS(\mathrm{Syz}) \approx d^m$ *if* $\#V(I)$ *is very small compared to* $\#V(J) = d^m$.

Proof. It suffices to show $I+(J:S) = R$, which implies that $\#V(I+(J:S)) = 0$. If $I = R$, then $I+(J:S) = R$. Thus, we consider the case $I \neq R$. Let f be an arbitrary element of R. By the *interpolation technique* (see Section 5.2 below), we can show that there exists a polynomial $h \in R$ such that $h(\alpha) = \frac{f(\alpha)}{S(\alpha)}$ for every $\alpha \in V(J) \setminus V(I)$. This implies that $(f - hS)(\alpha) = 0$ for every $\alpha \in V(J) \setminus V(I)$. As $S(\alpha) = 0$ for all $\alpha \in V(I)$, it follows that $(f - hS)S$ vanishes at all $\alpha \in V(J)$. By Hilbert's Nullstellensatz, the polynomial $(f - hS)S$ belongs to the radical ideal J. Thus, $f - hS$ belongs to the ideal quotient $(J : S)$. Furthermore, since the polynomial hS belongs to the ideal I, we have $f = hS + (f - hS) \in I + (J : S)$. This implies $I + (J : S) = R$, which completes the proof. □

3.2.3 Number of monomials in reduced S-coefficients

Here, we estimate the number of monomials in the reduced S-coefficient RSC(f) for each $f \in I \setminus J$. As those S-coefficients give a computational record for generating Gröbner basis elements, it is related to the total cost for computing a Gröbner basis of the ideal I. From Proposition 3, we set $V(J) \setminus V(I) = \{\alpha_1, \dots, \alpha_N\} \subseteq K^m$ and $NS(\mathrm{Syz}) = \{t_1, \dots, t_N\} \subseteq \mathcal{M}_{red}$ with $N = \#NS(\mathrm{Syz})$.

Consider the standard form (12) of f and represent RSC(f) $= \tilde{T} = \sum_{i=1}^N c_i t_i$ for some $c_i \in K$ since $\mathrm{supp}(\tilde{T}) \subseteq NS(\mathrm{Syz})$. Considering every c_i as an indeterminate, we obtain a system of linear equations derived from the N equations

$$\sum_{i=1}^{N} c_i t_i(\alpha_j) = \frac{f(\alpha_j)}{S(\alpha_j)} \text{ for } \alpha_j \in V(J) \setminus V(I).$$

Specifically, let M denote the $N \times N$ matrix whose i-th row is $(t_i(\alpha_1), \dots, t_i(\alpha_N))$ for $1 \le i \le N$. Then, we have the linear relation

$$\left(\frac{f(\alpha_1)}{S(\alpha_1)}, \dots, \frac{f(\alpha_N)}{S(\alpha_N)}\right) = (c_1, c_2, \dots, c_N)M. \tag{14}$$

Lemma 4. *The matrix M is invertible.*

Proof. We show that the system (14) has a unique solution, which proves that M is invertible. Let $\mathbf{c}' = (c'_1, \dots, c'_N)$ be an arbitrary solution to the system (14), and set $T_{\mathbf{c}'} = \sum_{i=1}^N c'_i t_i \in R$. Since $f(\alpha) = (T_{\mathbf{c}'}S)(\alpha) = 0$ for all $\alpha \in V(I)$, we have $f(\alpha) = (T_{\mathbf{c}'}S)(\alpha)$ for all $\alpha \in V(J)$. Thus, it follows by Hilbert's Nullstellensatz that $f - T_{\mathbf{c}'}S$ belongs to the radical ideal J. Hence, RSC(f)$S - T_{\mathbf{c}'}S$ also belongs to J. From Definition 3, we have RSC(f) $- T_{\mathbf{c}'} \in (J : S) = \mathrm{Syz}$. Since both RSC($f$) and $T_{\mathbf{c}'}$ are reduced with respect to the Gröbner basis $\mathcal{H}$ of Syz, we obtain RSC(f) $= T_{\mathbf{c}'}$ and the solution of the system (14) is unique. □

By Lemma 4, each c_i is expressed as the inner product of the vectors;

$$c_i = \left(\frac{f(\alpha_1)}{S(\alpha_1)}, \dots, \frac{f(\alpha_N)}{S(\alpha_N)}\right) \cdot {}^t\mathbf{m}_i, \tag{15}$$

where $\mathbf{m}_i$ denotes the i-th column of the inverse matrix of M.

When $K = \mathbb{F}_q$ with a large q, we expect for a randomly chosen $f \in I$ that the probability that the ratio of zero-products, that is $c_i = 0$ in (15), is approximately $\frac{1}{q}$. Thus, we may expect that it satisfies roughly on average

$$\#\mathrm{supp}(\mathrm{RSC}(f)) \approx \#NS(\mathrm{Syz})\left(1 - \frac{1}{q}\right) \approx d^m - \#V(I).$$

We call this property the *genericness of non-zero coefficients* of $f \in I \setminus J$. In a similar manner, we can extend our above arguments for any S-coefficient T of f. In this case, we also expect the following property for $T = \sum_{t \in \mathcal{M}} c_t t$:

$$\#\mathrm{supp}(T) = \#\{t \in \mathcal{M} \mid c_t \neq 0\} \gtrsim (d^m - \#V(I))\left(\frac{q-1}{q}\right) \approx d^m - \#V(I).$$

We call this property the *extended genericness of non-zero coefficients* of f.

3.3 Signature-based algorithms and the number of S-polynomials

Here, we estimate the number of S-polynomials computed during Gröbner basis computation. It is useful to consider the so-called *signature-based algorithms* (the F_5 algorithm and its variants), which can avoid unnecessary S-polynomials as many as possible. (See [9] for the original F_5 algorithm as well as a recent survey [7].)

3.3.1 $\mathfrak{S}$-reduction and $\mathfrak{S}$-Gröbner bases

We recall important notions related to signature-based algorithms for special type ideals I. To deal with signatures, we must restrict the division operation.

Definition 5 ($\mathfrak{S}$-reduction). *Let $f, g, h \in I$. We say that f is* $\mathfrak{S}$-reduced *to g by h, if there are a monomial $t \in \mathcal{M}$ and an element $a \in K \setminus \{0\}$ such that*

$$g = f - a \cdot t \cdot h, \quad \operatorname{sig}(th) \prec \operatorname{sig}(f) \quad \textit{and} \quad LM(g) \prec LM(f).$$

In this case, the element h (or $t \cdot h$) is called an $\mathfrak{S}$-reducer *of f and signatures are stable through $\mathfrak{S}$-reduction as* $\operatorname{sig}(f) = \operatorname{sig}(g)$. *We also say that f is* $\mathfrak{S}$-irreducible *if it has no $\mathfrak{S}$-reducer.*

Using $\mathfrak{S}$-reduction, we can obtain the following result by which collecting $\mathfrak{S}$-irreducible elements of different signatures is sufficient for obtaining a Gröbner basis. (This is a translation of Proposition 2.13 in [38] for special type ideals.)

Lemma 5. *For every $s \in NS(\mathrm{Syz})$, there is an element in the special type ideal I whose signature is s. Among such elements, there is an element $f \in I$ which has the smallest leading monomial $t \in \mathcal{M}$. Then, f is $\mathfrak{S}$-irreducible and any $\mathfrak{S}$-irreducible element with signature s has the same leading monomial t.*

For each $s \in NS(\mathrm{Syz})$, we denote by $\Phi(s)$ the leading monomial of an $\mathfrak{S}$-irreducible polynomial f with signature s. By Lemma 5, it is uniquely determined from signature s as

$$\Phi(s) = \min\nolimits_{\prec}\{LM(f) \mid f \in I, \operatorname{sig}(f) = s\}. \tag{16}$$

Here, we provide the definition of $\mathfrak{S}$-Gröbner bases for our setting. It can be shown that any $\mathfrak{S}$-Gröbner basis is also a Gröbner basis.

Definition 6 ($\mathfrak{S}$-Gröbner Bases). *A finite subset $\mathcal{G}$ of the special type ideal I is called an* $\mathfrak{S}$-Gröbner basis *of I if it satisfies the following conditions:*

1. *The set $\mathcal{G}$ contains a Gröbner basis of the ideal $J = \langle \mathcal{F} \rangle$.*
2. *For every $s \in NS(\mathrm{Syz})$, there exist $t \in \mathcal{M}$ and $g \in \mathcal{G}$ such that $t \cdot \operatorname{sig}(g) = s$ and tg is $\mathfrak{S}$-irreducible. (Such g should be $\mathfrak{S}$-irreducible and $t \cdot LM(g) = t \cdot \Phi(\operatorname{sig}(g)) = \Phi(s)$.)*

For a signature s, if $\mathcal{G}$ satisfies both conditions for any signature $s' \prec s$, $\mathcal{G}$ is called an $\mathfrak{S}$-Gröbner basis up to s.

3.3.2 Basic frame of signature-based algorithms and the F_5 criterion

To compute an $\mathfrak{S}$-Gröbner basis, we consider (special) S-polynomials and add their non-zero remainders by $\mathfrak{S}$-reduction to $\mathcal{G}$ in ascending order of their signatures. The initial $\mathcal{G}$ is $\mathcal{F}_S$, where S is the unique $\mathfrak{S}$-irreducible element of signature 1. Condition (2) in Definition 6 gives an efficient criterion for detecting unnecessary S-polynomials.

Let s be a signature in $NS(\mathrm{Syz})$ and suppose that $\mathcal{G}$ is an $\mathfrak{S}$-Gröbner basis up to s. Then, $s \cdot S$ is an element of I of signature s, where $(s, S) \in \mathcal{M} \times \mathcal{G}$. If any pair (t, g) in $\mathcal{M} \times \mathcal{G}$ does not satisfy Condition (2) in Definition

6 for s, an S-polynomial with signature s exists and its remainder by $\mathfrak{S}$-reduction will be added to $\mathcal{G}$ as a necessary element for making $\mathcal{G}$ an $\mathfrak{S}$-Gröbner basis. In more detail, there exist $g_1, g_2 \in \mathcal{G}$, $t_1, t_2 \in \mathcal{M}$ such that $s = \mathrm{sig}(t_1 g_1) = t_1 \mathrm{sig}(g_1) \succ \mathrm{sig}(t_2 g_2) = t_2 \mathrm{sig}(g_2)$ and $LM(t_1 g_1) = LM(t_2 g_2)$. (As $t_1 g_1$ is not $\mathfrak{S}$-irreducible, there is a $\mathfrak{S}$ reducer $t_2 g_2$.) Then, their S-polynomial (up to constants) is of signature s. Applying $\mathfrak{S}$-reduction to the S-polynomial, we obtain an $\mathfrak{S}$-irreducible element with signature s, which is added to $\mathcal{G}$. Such a pair (g_1, g_2) is called a *normal pair*, and their S-polynomial is called a *special S-polynomial* here. This procedure exactly corresponds to the *F_5 criterion*. We note that $t_1 g_1$ is chosen so that $LM(t_1 g_1)$ is the smallest among all possible tg ($t \in \mathcal{M}, g \in \mathcal{G}$) with signature s, and if $t_1 g_1$ is $\mathfrak{S}$-irreducible, there does not exist such a normal pair for $t_1 g_1$ and we can avoid any special S-polynomial of signature s.

We note that, by carefully dealing with polynomials in ascending order of their signatures, we can accurately identify their signatures. (See [37, 38] for precise algorithms.)

Definition 7 (Necessary Signature). *For each s in $NS(\mathrm{Syz})$, we say that s is a necessary signature if any $\mathfrak{S}$-Gröbner basis contains an element of signature s.*

Remark 4. *By Definition 6, a signature s is necessary if there is no pair (t, g) in $\mathcal{M} \setminus \{1\} \times R$ such that g is $\mathfrak{S}$-irreducible, $t \cdot \mathrm{sig}(g) = s$, and $t \cdot LM(g) = t \cdot \Phi(\mathrm{sig}(g)) = \Phi(s)$. Of course, for a necessary signature s, some special S-polynomial of signature s is computed during $\mathfrak{S}$-Gröbner basis computation.*

3.3.3 Signatures for necessary S-polynomials

Here, we consider a signature for which some special S-polynomial is computed during $\mathfrak{S}$-Gröbner basis computation. Suppose that we have computed an $\mathfrak{S}$-Gröbner basis $\mathcal{G}_s$ up to a signature s, and its elements are $\mathfrak{S}$-irreducible. Since $S \in I \setminus J$ is the unique element of $\mathcal{G}_s$ with signature 1, it holds that $\mathrm{sig}(sS) = s \times \mathrm{sig}(S) = s$ and hence, $\{ug \mid u \in \mathcal{M} \setminus \{1\}, g \in \mathcal{G}_s, u \times \mathrm{sig}(g) = s\} \neq \emptyset$. From this set, we take an element $u_1 g_1$ with the smallest leading monomial.

If the signature s is unnecessary, then $u_1 g_1$ should be $\mathfrak{S}$-irreducible. Hence, it holds that $\Phi(s) = LM(u_1 g_1) = u_1 LM(g_1) = u_1 \Phi(g_1)$ and there is no special S-polynomial of signature s constructed by $u_1 g_1$. However, such a case seems to rarely happen. For $s = u_1 \times \mathrm{sig}(g_1)$ with $u_1 \succ 1$, we have $s \succ \mathrm{sig}(g_1)$, and it is expected that $\Phi(\mathrm{sig}(g_1)) \succ \Phi(s)$ and $LM(u_1 g_1) = u_1 LM(g_1) = u_1 \Phi(\mathrm{sig}(g_1)) \succ \Phi(s)$. This implies that the signature s is necessary. This is because,

$$\begin{aligned}\Phi(s) &= \min\nolimits_{\prec}\{LM(f) \mid f \in I, \mathrm{sig}(f) = s\} \\ &= \min_{\prec}\{LM(f) \mid f = \left(\sum_{t \in NS(\mathrm{Syz}), t \preceq s} c_t t\right) S + \sum_{i=1}^{m} A_i F(x_i),\ c_s \neq 0\}\end{aligned}$$

and it seems that $\Phi(s)$ decreases for sufficiently large s *in general*, since the set of non-zero signatures $NS(\mathrm{Syz})$ is finite. We will discuss this behavior in Subsection 3.4 below. Next, we present a generalization of the notion *normal* of PRS in our case.

Definition 8 (Normality). *We say that the special type ideal I is* normal *with respect to the signatures if the following condition holds:*

$$(\star)\ \ \Phi(s) \nmid \Phi(s') \textit{ holds for any distinct signatures } s, s' \textit{ in } NS(\mathrm{Syz}) \textit{ with } s \mid s'.$$

For a subset U of $NS(\mathrm{Syz})$, we say that I is U-semi-normal *if the condition $(\star)$ holds for distinct signatures s, s' with $s' \in U$ and $s \in NS(\mathrm{Syz})$. As an extremal case, we say that I is* strongly normal *if $\Phi(s) \succ \Phi(s')$ holds for any signatures s, s' with $s \prec s'$.*

By Remark 4, if the special type ideal I is normal, then every signature s is a necessary signature and some S-polynomial with signature s appears during $\mathfrak{S}$-Gröbner basis computation. Thus, letting $N = \#NS(\mathrm{Syz})$, at least $(N-1)$ S-polynomials are computed. In addition, if the special type ideal I is U-semi-normal for some $U \subset NS(\mathrm{Syz})$, then every signature in U is necessary.

Remark 5. *To give a more precise estimation for our case $d > \deg_{x_i}(S)$, we have to consider the effect of S whose signature is* 1. *This is because, for a smaller t, tS tends to break the condition* (*) *in Definition 8. Actually, unnecessary signatures for S-polynomials can be detected at the* beginning *stage of Gröbner basis computation.*

3.4 Linear algebra related to the subresultant theory

In this subsection, we analyze the condition (*) in Definition 8 using linear algebraic methods related to the subresultant theory. (See [26].) Then, based on such linear-algebraic analysis, we discuss the semi-normality of the ideal $I_m(a, b)$.

To simplify our arguments, we concentrate on a *trivial ideal case* where $I = R$, and thus $NS(\mathrm{Syz}) = \mathcal{M}_{red}$. (The same arguments are applicable to a non-trivial ideal case where $I \neq R$.) We arrange all monomials in $\mathcal{M}_{red}$ in descending order. Thus, $\mathcal{M}_{red} = \{t_1, \dots, t_N\}$, $N = \#\mathcal{M}_{red} = d^m$ and $t_i \succ t_j$ for $i < j$. (In particular, it satisfies $t_N = 1$ and $t_1 = x_1^{d-1} \cdots x_m^{d-1}$.) We denote by $\mathrm{NF}_{\mathcal{F}}(f)$ the normal form of a polynomial $f \in R$ with respect to $\mathcal{F}$. For a polynomial $g = \sum_{i=1}^{N} c_i t_i$ reduced with respect to $\mathcal{F}$, we define its *coefficient vector* as $[g] = (c_1, \dots, c_N)$. For an integer $1 \le k \le N$, we also write $[g]_k = (c_1, \dots, c_k)$ for the vector consisting of the first k components.

For a signature $s \in \mathcal{M}_{red}$, let f be an element of I with signature s which is reduced with respect to the set $\mathcal{F}$. The standard form of f is represented as follows:

$$f = TS + A_1F(x_1) + \cdots + A_mF(x_m)$$

with $\mathrm{supp}(T) \subseteq \mathcal{M}_{red}$ and $LM(T) = s$. When $s = t_\ell$ (i.e., s is the ℓ-th monomial in $\mathcal{M}_{red}$), the reduced S-coefficient of f is written as $T = \sum_{i=\ell}^{N} c_i t_i$ with $c_\ell \neq 0$. Since f is reduced with respect to $\mathcal{F}$, it holds that $\mathrm{NF}_{\mathcal{F}}(TS) = f$. Now, we consider vectors $[\mathrm{NF}_{\mathcal{F}}(t_\ell S)], \dots, [\mathrm{NF}_{\mathcal{F}}(t_N S)]$. Let M_s be an $(N-\ell+1) \times N$ matrix whose i-th row is $[\mathrm{NF}_{\mathcal{F}}(t_{\ell+i-1}S)]$, and $(M_s)_k$ the $(N-\ell+1) \times k$ matrix whose i-th row is $[\mathrm{NF}_{\mathcal{F}}(t_{\ell+i-1}S)]_k$ for $1 \le k \le N$. Then, we clearly have

$$\mathbf{c}M_s = \sum_{i=\ell}^{N} c_i[\mathrm{NF}_{\mathcal{F}}(t_iS)] = [\mathrm{NF}_{\mathcal{F}}(TS)] = [f] \text{ and } \mathbf{c}(M_s)_k = [\mathrm{NF}_{\mathcal{F}}(TS)]_k = [f]_k$$

for the coefficient vector $\mathbf{c} = (c_\ell, \dots, c_N)$ of T. In particular, we set $\hat{M}_s = (M_s)_{N-\ell+1}$, which is a square matrix with size $N-\ell+1$. (This matrix shall be used later.)

Let $\Phi(s) = t_r$, the r-th monomial in $\mathcal{M}_{red}$. From (16) and the above argument, there exists a polynomial $\hat{T} = \sum_{i=\ell}^{N} \hat{c}_i t_i$ with $\hat{c}_\ell \neq 0$ so that the leading monomial of $\mathrm{NF}_{\mathcal{F}}(\hat{T}S)$ is equal to t_r. Therefore, the coefficient vector $\hat{\mathbf{c}} = (\hat{c}_\ell, \dots, \hat{c}_N)$ of $\hat{T}$ satisfies

$$\hat{\mathbf{c}}(M_s)_r = [\mathrm{NF}_{\mathcal{F}}(\hat{T}S)]_r = (0, \dots, 0, 1).$$

However, $\mathbf{c}(M_s)_r$ cannot be 0 for any non-zero vector $\mathbf{c} = (c_\ell, \dots, c_N)$ with $c_\ell \neq 0$. Thus, determining $\Phi(s)$ is reduced to determining the *solvability / non-solvability* of a number of linear equations systemes.

Example 3 (GCD and Subresultant). *We consider the univariate case dealt with in Example 2 to set $m = 1$, $x = x_1$, $f_0 = S$ and $f_1 = F$ and $\mathcal{M}_{red} = \{1, x, \dots, x^{d-1}\}$. We assume $f_2 = f_0$, as S is assumed to be reduced with respect to F. For each x^k, f_{k+2} is an $\mathfrak{S}$-irreducible element of signature x^k and f_{k+2} is the reminder of $A_{k+2}f_0$ by f_1, that is, the normal form of $A_{k+2}f_0$ with respect to $\{f_1\}$. Letting $A_{k+2} = \sum_{i=0}^{k} c_i x^i$, we have*

$$\mathrm{NF}_{\{f_1\}}\left(\sum_{i=0}^{k} c_i x^i \times f_0\right) = f_{k+2}.$$

Letting $f_{k+2} = \sum_{i=0}^{d-k-1} e_i x^i$*, we also have*

$$(c_k, \ldots, c_0) M_{x^k} = [f_{k+2}] = (0, \ldots, 0, e_{d-k-1}, \ldots, e_0),$$

where

$$M_{x^k} = \begin{pmatrix} [\mathrm{NF}_{\{f_1\}}(x^k f_0)] \\ \vdots \\ [\mathrm{NF}_{\{f_1\}}(f_0)] \end{pmatrix}.$$

Then, as the PRS $\{f_1, f_2, \ldots\}$ *is normal,* $e_{d-k-1} \neq 0$ *and* $(M_{x^k})_{k+1}$ *is invertible, and its determinant exactly corresponds to the* $(d-k)$*-th* principal subresultant coefficient (PSC) *defined by the determinant of the matrix obtained from the Sylvester matrix of* f_0 *and* f_1 *by deleting the first* $d-k$ *rows corresponding to* f_0*, the first* $d-k$ *rows corresponding to* f_1 *and the first and the last* $d-k$ *columns. (See Chapter 7 in [26] for details.)*

3.4.1 Discussion on normality of $I_m(a, b)$

In naive index calculus methods for solving ECDLP, we consider a number of ideals $I = I_m(a, b)$ defined in (4) for random a, b in $K = \mathbb{F}_q$. Then, by considering $x(aP + bQ)$ as *an additional parameter* z, our computation can be translated to solving a *parametric linear system*. As $F(x_i)$ does not involve the parameter z, $[\mathrm{NF}_{\mathcal{F}}(tS)]$ is a vector whose components are polynomials in z, and so $\hat{M}_s$ is a matrix whose components are polynomials in z. Thus, the linear systems for determining $\Phi(s)$ are given by such parametric matrices.

Remark 6 (Parametric solving)**.** *Since the determination of* $\Phi(s)$ *can be reduced to checking the* solvability/non-solvability *of linear systems, we can apply* quantifier elimination *technique based on* comprehensive Gröbner basis system *proposed by [12] as a general method. Using this method, the problem is reduced to the conjunction of* existential *problems in quantifier elimination theory. Then, we obtain the* conjunction *of systems of* semi-algebraic sets *given by polynomials in z.*

We provide a simpler discussion. In the case, $\det(\hat{M}_s)$ is a polynomial in z. Then, if $\det(\hat{M}_s) \neq 0$ for some value $x(aP + bQ)$, $\det(\hat{M}_s)$ is a non-trivial polynomial in z over K. Using this fact, we can discuss the two following typical cases:

- If $\hat{M}_s$ and $\hat{M}_{s'}$ are invertible, where $s' = t_{\ell+1}$ (the previous element of s), then any non-zero vector of size $N - \ell + 1$ whose last component is zero cannot make the first $N - \ell$ components zero; however, some vector with a non-zero last component can make them zero. Thus, in this case, we have $\Phi(s) = t_{N-\ell+1}$. Moreover, if $\hat{M}_s$ is invertible for every $s \in NS(\mathrm{Syz})$, then I is *strongly normal*. This behavior was observed in the case $\delta = d - 1$ in our experiment.
- For each D ($1 \leq D \leq m\delta$), let t_D be the largest element among signatures of total degree D, that is, $t_D = \max_{\prec}\{t \in NS(\mathrm{Syz}) \mid \deg(t) = D\}$. If the matrix $\hat{M}_{t_D}$ is invertible, then we can show that for any signature $s \preceq t_D$, $\deg(\Phi(s)) \geq m\delta - D$. Thus, if $\hat{M}_{t_D}$ is regular for every D, then it is expected with high probability that $\deg(s) < \deg(s')$ implies that $\deg(\Phi(s)) > \deg(\Phi(s'))$. This corresponds to Assumption 7 in the next section.

4 Complexity analysis of naive index calculus for ECDLP

In this section, we analyze the complexity of Gröbner basis computation for our target ideal $I_m(a, b)$ defined in (4), and present lower complexity bounds of the naive index calculus methods for solving ECDLP. Here, we suppose that $m \geq 2$. We can apply discussions given in the previous section to the ideal $I_m(a, b)$, since it is a special type ideal generated by m univariate polynomials $F(x_1), \ldots, F(x_m)$ and the summation polynomial $S_{m+1}(x_1, \ldots, x_m, x(aP + bQ))$ for a factor $F(x)$ of $x^q - x$ of degree d.

4.1 Analysis based on the number of S-polynomials

In this subsection, we present an analysis based on the number of S-polynomials in Gröbner basis computation for the ideal $I_m(a, b)$ with random $a, b \in \mathbb{F}_q$.

4.1.1 Modification of the set of non-zero signatures

Let $\delta = 2^{m-1}$. Then, for $S = S_{m+1}(x_1, \dots, x_m, x(aP + bQ))$, $LM(S) \preceq x_1^\delta \cdots x_m^\delta$ and $\deg_{x_i}(S) \leq \delta$ for each x_i, since $\deg_{x_i}(S_{m+1}(x_1, \dots, x_{m+1})) = \delta$ for each x_i. (Actually, $S_{m+1}(x_1, \dots, x_{m+1})$ has a term $x_1^\delta \cdots x_m^\delta$ and $\deg(S) = \deg(S_{m+1}(x_1, \dots, x_{m+1})) = m\delta$. This can be shown inductively by total degree estimation of S_{m+1} via the Sylvester matrix of S_3 and S_m, and the last statement of Theorem 1 in [33].) We note that, if $d > \delta$, S is reduced with respect to $\mathcal{F}$ and also $\mathfrak{S}$-irreducible. Otherwise, the normal form of S is $\mathfrak{S}$-irreducible. Then, we obtain the following result:

Lemma 6. *Assume $d > \delta$. Each $t = x_1^{e_1} \cdots x_m^{e_m}$ $(0 \leq e_i < d - \delta)$ with $t \neq 1$ is an unnecessary signature for $\mathfrak{S}$-Gröbner basis computation. In other words, no S-polynomials with signature t appear during $\mathfrak{S}$-Gröbner basis computation for the ideal $I_m(a, b)$.*

Proof. Let $t = x_1^{e_1} \cdots x_m^{e_m}$ $(0 \leq e_i < d - \delta)$. As any $F(x_i)$ cannot reduce tS, we can show that t does not belong to $LM(\mathrm{Syz})$ and tS has its signature t. Thus, it suffices to show that tS is $\mathfrak{S}$-irreducible, as $t \neq 1$ and S is an $\mathfrak{S}$-irreducible element of signature 1.

Suppose that tS is not $\mathfrak{S}$-irreducible. Then, there exists a polynomial T such that $\mathrm{supp}(T) \subseteq NS(\mathrm{Syz})$, $LM(T) \prec t$ and the normal form of TS with respect to $\mathcal{F}$ has the same leading monomial as $LM(tS)$. Since $LM(T) \prec t$, it can be shown that $\deg_{x_i}(LM(TS)) < \deg_{x_i}(LM(tS)) \leq e_i + \delta$ for some i. However, in this case, $F(x_i)$ cannot reduce TS and $LM(TS)$ cannot coincide with $LM(tS)$. This is a contradiction. □

Next, we modify the set of non-zero signatures as

$$\overline{NS(\mathrm{Syz})} = NS(\mathrm{Syz}) \setminus \{x_1^{e_1} \cdots x_m^{e_m} \mid 0 \leq e_i < d - \delta\}$$

Then, $\#\overline{NS(\mathrm{Syz})} = d^m - \#V(I) - (d - \delta)^m$. Since it is difficult to select all necessary signatures theoretically, we treat $\overline{NS(\mathrm{Syz})}$ as a *rough approximation.*

4.1.2 Assumptions on the semi-normality

As discussed in Remark 6, we can parameterize the value $x(aP + bQ)$ with a *parameter z* for the summation polynomial $S = S_{m+1}(x_1, \dots, x_m, x(aP + bQ))$ with random $a, b \in \mathbb{F}_q$. Then, for distinct signatures s, s' such that $s = us'$ for some $u \in \mathcal{M} \setminus \{1\}$, the condition $u\Phi(s') = \Phi(s)$ can be translated as a condition defined by a system of (semi) algebraic equations in z.

Remark 7 (Parametric property)**.** *By considering the summation polynomial S_{m+1} over $\mathbb{Q}$, we can consider the ideal generated by $S_{m+1}(x_1, \dots, x_m, z)$ and $F(x_1), \dots, F(x_m)$ over $\mathbb{Q}$. If the ideal is $\overline{NS(\mathrm{Syz})}$-semi-normal for some $z \in \mathbb{Q}$, then it holds for almost every $z \in \mathbb{Q}$, and the reduction ideal over $\mathbb{F}_p$ is $\overline{NS(\mathrm{Syz})}$-semi-normal for any prime p. (See Definition 8 for semi-normality.)*

From this remark, we assume the following:

Assumption 7. *For almost every a, b in $\mathbb{F}_q$, the ideal $I_m(a, b)$ is almost $\overline{NS(\mathrm{Syz})}$-semi-normal.*

Under Assumption 7, the number of necessary S-polynomials has almost the same order as $\overline{NS(\mathrm{Syz})} \approx d^m - (d - \delta)^m$, and is estimated as $(d^m - (d - \delta)^m)^{1-\varepsilon}$ for a small constant $0 < \varepsilon \ll 1$. Moreover, the number of elements of a computed $\mathfrak{S}$-Gröbner basis is at least $(d^m - (d-\delta)^m)^{1-\varepsilon}$, since S-polynomials of necessary signature form a part of the $\mathfrak{S}$-Gröbner basis.

Remark 8 (Comparison with other algorithms)**.** *We now consider another algorithm for Gröbner basis computation which uses S-polynomials, but not any signature. Although it is not proven rigidly, the F_5 algorithm or its variant, along with the F_4 technique for reduction step, is recognized as the fastest one. This is because it is highly supposed that, by defining the signature adequately, signature-based-algorithms can handle a smaller number of S-polynomials compared with non-signature-based-algorithms. In fact, from our experimental results, it is observed that algorithms using two efficient techniques,* normal selection strategy *and* sugar degree*, compute S-polynomials very likely in ascending order of total degree of their signature given in Definition 4. (See Chapter 2.10 in [5] for normal selection strategy and sugar degree.) This behavior should be heavily related to the* semi-normality *analyzed in Subsection 3.4, since, in the normal strategy with sugar degree, an S-polynomial with the smallest leading monomial is chosen at each step. (See page 116 in [5] for details.) Therefore, if Assumption 7 holds, such an S-polynomial may have the smallest total degree of the signature. Thus, even for another (efficient) algorithm using S-polynomials, it is expected that the number of S-polynomials is at least $(d^m - (d-\delta)^m)^{1-\varepsilon}$. Moreover, we also suppose that our definition of the signature is the most efficient among all possible definitions. (See Remark 3.)*

Lemma 8. *Under Assumption 7 and Remark 8, any efficient algorithm using S-polynomials for Gröbner basis of the ideal $I_m(a,b)$ requires the computation of at least $(d^m - (d-\delta)^m)^{1-\varepsilon}$ S-polynomials for almost every $a, b \in \mathbb{F}_q$.*

It can be shown by induction argument on m that the inequality $d^m - (d-\delta)^m \geq md^{m-1}$ holds when d is larger than $\delta \geq 4$. (In our experiments shown in Subsection 4.3, the number of *necessary S-polynomials*, which are not reduced to 0, is close to $2 \times md^{m-1}$.) Thus, it requires at least $(md^{m-1})^{1-\varepsilon}$ S-polynomials (or md^{m-1} S-polynomials by our experimental observation) and hence, the computational cost exceeds $(md^{m-1})^{1-\varepsilon}$ arithmetic operations. From a cryptographical point of view, the case in which the degree d is much larger than δ is important. We note that $\frac{d^m-(d-\delta)^m}{md^{m-1}} \approx \delta$ when $\frac{\delta}{d} \ll 1$.

Proposition 9. *Under Assumption 7 and Remark 8, it requires at least $(md^{m-1})^{1-\varepsilon}$ field arithmetic operations to compute a Gröbner basis of the ideal $I_m(a,b)$ for almost every $a, b \in \mathbb{F}_q$ by any algorithm using S-polynomials.*

4.1.3 Lower complexity bounds of naive index calculus

Based on Assumption 7 and Remark 8, we analyze the complexity of naive index calculus methods for solving ECDLP. For simplicity, we consider the case where the success probability $\mathcal{P}$ for solving the system (3) is almost equal to 1 for random $a, b \in \mathbb{F}_q$. (For other cases, the same estimation can be derived.) It follows from (6) that the total cost of the naive index calculus is at least

$$m! \cdot q \cdot t^{1-m} \cdot \mathcal{C} + \mathrm{Lin}(t) \gtrapprox m! \cdot m \cdot q + \mathrm{Lin}(d). \tag{17}$$

This is because $t = \#\mathcal{B} \approx d$ and it requires $t^m \gtrapprox \gamma q$ for the success probability $\mathcal{P} \approx 1$. Recall that $\mathcal{C}$ is the average cost of solving the system (3). From Proposition 9, the cost $\mathcal{C}$ requires at least md^{m-1}. (We can ignore the small constant ε, as our experiment suggests.) Then, the first term in (17) exceeds q. More precisely, by using $\mathcal{C} = (md^{m-1})^{1-\varepsilon}$, we can show that the total estimation is greater than $q^{\frac{m-1}{m}(1-\varepsilon)}$. This implies that naive index calculus methods with Gröbner basis computation by S-polynomials cannot be more efficient than Pollard's rho method. Moreover, based on our experimental observations, it cannot be more efficient than even the brute force method for solving ECDLP.

4.2 Another analysis based on the number of monomials

In this subsection, we discuss another approach for possible estimation which is not dependent on Gröbner basis algorithms. During Gröbner basis computation (based on Buchberger's criterion or its extension like the F_5 criterion), a new polynomial is generated through an S-polynomial by multiplying some monomial to a polynomial, and by reducing it with respect to the current set of polynomials for a Gröbner basis of I. Every such polynomial $f \in I$ is expressed as

$$f = T^{(f)}_{actual} S + A^{(f)}_{1,actual} F(x_1) + \cdots + A^{(f)}_{m,actual} F(x_m)$$

for some polynomials $T^{(f)}_{actual}, A^{(f)}_{1,actual}, \ldots, A^{(f)}_{m,actual}$. These coefficient polynomials depend on the *actual* computation (algorithm). For each element g of the reduced Gröbner basis, every monomial of the actual S-coefficient $T^{(g)}_{actual}$ is constructed by *monomial multiplication* or *monomial division* at each step. Thus, the number of monomials in $T^{(g)}_{actual}$ might represent a *lower bound* on the number of such arithmetic operations occurred to obtain the reduced Gröbner basis.

At some step during the computation, a pair (g_1, g_2) is chosen from the current set $\mathcal{G}'$ of polynomials for the reduced Gröbner basis, and a new element g is generated by the remainder on monomial divisions of the S-polynomial $\mathrm{Spol}(g_1, g_2) = u_1 g_1 - u_2 g_2$. Then,

$$\begin{aligned} g &= u_1 g_1 - u_2 g_2 - \sum_i \sum_t c_{t,i} t g_i \\ &= \underbrace{(u_1 T^{(g_1)}_{actual} - u_2 T^{(g_2)}_{actual} - \sum_i \sum_t c_{t,i} t T^{(g_i)}_{actual})}_{T^{(g)}_{actual}} S + \cdots \end{aligned} \tag{18}$$

for some polynomials $g_i \in \mathcal{G}'$ and some coefficients $c_{t,i}$. This shows that every monomial in the actual S-coefficient $T^{(g)}_{actual}$ comes from the multiplication of a monomial and an already computed polynomial which occurs during this procedure. In particular, there must occur one such multiplication for each newly appearing monomial in the actual coefficient. Thus, we guess the following result.

Conjecture 10. *To produce an element g appearing during Gröbner basis computation for the ideal I, it requires at least* $\#\mathrm{supp}(T^{(g)}_{actual})$ *times of multiplications between a monomial and a polynomial appearing during the computation.*

Remark 9. *In our experiments in Subsection 4.3, the number of such multiplications was much larger than* $\#\mathrm{supp}(T^{(g)}_{actual})$*. We leave a theoretical proof of the conjecture for our future work.*

4.2.1 Assumptions on the genericness of non-zero coefficients

We present our assumptions for estimating $\#\mathrm{supp}(T^{(g)}_{actual})$ for an element g of the reduced Gröbner basis $\mathcal{G}_{a,b}$ of $I = I_m(a, b)$ with random $a, b \in \mathbb{F}_q$. As in (15), for the summation polynomial $S = S_{m+1}(x_1, \ldots, x_m, x(aP + bQ)) \in I$, we let

$$\mathbf{GS}^{(g)}_{m+1}(a, b) = \left(\frac{g(\alpha_1)}{S_{m+1}(\alpha_1, x(aP + bQ))}, \ldots, \frac{g(\alpha_N)}{S_{m+1}(\alpha_N, x(aP + bQ))} \right),$$

where $V(J) \setminus V(I) = \{\alpha_1, \ldots, \alpha_N\} \subseteq \mathbb{F}_q^m$ as in Subsection 3.2. Then, we expect that the distribution

$$\{\mathbf{GS}^{(g)}_{m+1}(a, b) \in \mathbb{F}_q^N \mid a, b \in \mathbb{F}_q,\ g \in \mathcal{G}_{a,b}\}$$

would coincide almost with that of random vectors. Conversely, for fixed $a, b \in \mathbb{F}_q$, we expect that the vector $\mathbf{GS}^{(g)}_{m+1}(a, b)$ behaves like a random vector for almost every $g \in \mathcal{G}_{a,b}$, and thus, g would satisfy the genericness of non-zero coefficients. Moreover, from our experiments, we expect that the extended genericness of non-zero coefficients would hold for almost every $g \in \mathcal{G}_{a,b}$. Thus, we assume the following for our analysis:

Assumption 11. *For almost every a, b in $\mathbb{F}_q$, almost every element of the reduced Gröbner basis of the ideal $I_m(a, b)$ satisfies the extended genericness of non-zero coefficients.*

4.2.2 Complexity analysis of naive index calculus

We analyze the complexity of naive index calculus methods for solving ECDLP, based on Conjecture 10 and Assumption 11. Below, we consider two cases:

- **Case of trivial ideals.** The reduced Gröbner basis $\mathcal{G}_{a,b}$ is equal to $\{1\}$ in this case. Under Assumption 11, we have $\#\mathrm{supp}(T^{(1)}_{actual}) \gtrapprox d^m$ and $\mathcal{C} \gtrapprox d^m$ if Conjecture 10 holds. This cost is larger than the estimation md^{m-1} in Subsection 4.1. Hence, we obtain the same conclusion as in Subsection 4.1.
- **Case of non-trivial ideals.** In this case, there exists an element $g \in \mathcal{G}_{a,b} \setminus \mathcal{F}$ such that it satisfies the extended genericness of the non-zero coefficients of $T^{(g)}_{actual}$ under Assumption 11. Then, $\#\mathrm{supp}(T^{(g)}_{actual}) \gtrapprox d^m$, and thus
$$\# \cup_{h \in \mathcal{G}_{a,b}} \mathrm{supp}(T^{(h)}) \geq \#\mathrm{supp}(T^{(g)}_{actual}) \gtrapprox d^m.$$
Hence, we obtain the same conclusion as the case of trivial ideals.

4.3 Experimental data for our assumptions

Here, we present experimental data which support our assumptions. For $m = 3$ or 4, we generated primes $p = 2^B + \alpha$ with very small α. To count the number of S-polynomials, we randomly chose k elements from $\mathbb{F}_p$ and set $F(x)$ as the polynomial with these roots. Moreover, to count the number of monomials, we chose primes p with $k|p-1$ and set $F(x) = x^k - 1$ in Binomial case, and $F(x) = x^{k-1} + \cdots + 1$ in Non-binomial case. (Thus, the degree d of $F(x)$ is either k or $k-1$, and all zeros of $V(J)$ are rational over $\mathbb{F}_p$.) We generated elliptic curves with prime order over $\mathbb{F}_p$, their points P, Q and then generated randomly a, b in $\mathbb{F}_p$ for the point $aP + bQ$. Finally, we computed the reduced Gröbner bases of ideals $I_m(a, b)$ with respect to a graded reverse lexicographic order $\prec$ using a computer algebra system *Risa/Asir*. Its function `nd_gr` with *options* `gentrace=1` and `gensyz=1` records all history, illustrating how elements of the computed Gröbner basis were constructed. As results, Lemma 8 and Assumption 11 seem to hold for all examples. It is also observed that the number of multiplications of monomials and polynomials occurring in *reduction of S-polynomial* (see (18)) became very huge.

Remark 10. *The function* `nd_gr` *does not use any signature, but it uses the* normal selection strategy *and* sugar degree. *Thus, its computational behavior for the selection of S-polynomials becomes close to that of signature-based algorithms.*

4.3.1 Number of S-polynomials

We denote by $\mathcal{G}^\star$ the computed (non-reduced) Gröbner basis, that is, the set of all elements computed by S-polynomials through the computation, from which the reduced Gröbner basis $\mathcal{G}$ is computed. Thus, each element of $\mathcal{G}^\star$ corresponds to some computed S-polynomial, which was not reduced to 0. (Of course, there might be other *unnecessary* S-polynomials, which were reduced to 0.) For our experiments, we computed 5 examples for each parameter (B, d). In Tables 1,2,3,4, the symbol $[\#\mathcal{G}^\star]$ denotes the average of $\#(\mathcal{G}^\star \setminus \mathcal{F})$ and that $Ratio$ denotes the ratio $\frac{[\#\mathcal{G}^\star]}{\#\overline{NS(\mathrm{Syz})}}$. In Non-trivial ideal case, symbols $[\#NS(\mathrm{Syz})]$ and $[\#\overline{NS(\mathrm{Syz})}]$ denote their averages. The symbol $[PA]$ denotes the average of the number of multiplications of monomials and polynomials that occured in *division of S-polynomial*.

(I) For $m = 4$, we chose examples where d is close to δ. (In this case, $\delta = 8$.) We note that we could not deal with larger d in this case, since the computation requires large memory.

- In Trivial ideal case, it is observed that $\#NS(\mathrm{Syz}) = d^m$ coincides with $\#\mathcal{G}^*$ for $d \leq \delta = 8$ and $\#\overline{NS(\mathrm{Syz})}$ is very close to $\#\mathcal{G}^*$ for $d > \delta = 8$. Thus, all ideals in this case are considered to be strongly normal or almost $\overline{NS(\mathrm{Syz})}$-*semi-normal*. (See Table 1.)

Table 1: Trivial-ideal case ($m = 4$)

B	10	11	12	13	14	15
d	5	7	8	8	11	12
d^m	625	2401	4096	4096	14641	20736
$\#NS$	625	2401	4096	4096	14641	20736
$\#\overline{NS}$	625	2401	4096	4096	14560	20480
$[\#\mathcal{G}^*]$	625	2401	4096	4096	14301	19770
$Ratio$	1	1	1	1	0.98	0.97
$[PA]$	122170.8	1175402.6	2902360	2902738.4	26852012.8	46097921.6

Table 2: Non-trivial-ideal case ($m = 4$)

B	12	13	15
d	5	6	7
d^m	625	1296	2401
$[\#NS]$	605.2	1290	2390
$[\#\overline{NS}]$	605.2	1290	2390
$[\#\mathcal{G}^*]$	605.2	1290	2390
$Ratio$	1	1	1
$[\#V(I)]$	19.8	6	11
$[PA]$	120937.2	409084	1175366.4

Table 3: Trivial-ideal case ($m = 3$)

d	12	20	30
$\#NS$	1728	8000	27000
$\#\overline{NS}$	1216	3904	9424
md^{m-1}	432	1200	2700
$[\#\mathcal{G}^*]$	947	2599	5744
$Ratio$	0.78	0.67	0.61
$Ratio1$	2.19	2.17	2.13
$[PA]$	147487.4	862111.6	3472739

Table 4: Non-trivial-ideal case ($m = 3$)

d	12	20	30
$[\#NS]$	1723.2	7994.6	26994
$[\#\overline{NS}]$	1211.2	3898.6	9418
md^{m-1}	432	1200	2700
$[\#\mathcal{G}^*]$	942.2	2593.6	5738
$Ratio$	0.78	0.67	0.61
$Ratio1$	2.18	2.16	2.13
$[\#V(I)]$	4.8	5.4	6
$[PA]$	147427.2	862028.6	3472638.6

- In Non-trivial ideal case, it is observed that $\#V(I)$ is very small, and $\#NS(\mathrm{Syz})(= d^m - \#V(I))$ is very close to $\#\mathcal{G}^\star$. Thus, all ideals in this case are considered to be $NS(\mathrm{Syz})$-semi-normal. (See Table 2.)

(II) For $m = 3$, we chose examples where $B = 15$ and d is significantly larger than $\delta = 4$. Here, the symbol $Ratio1$ denotes the ratio $\frac{[\#\mathcal{G}^\star]}{dm^{m-1}}$. It is observed that $\#\mathcal{G}^\star \approx 5.5 \times \frac{N_1}{\log(N_1)}$, where $N_1 = \#\overline{NS}(\mathrm{Syz})$, and also $\#\mathcal{G}^\star \approx 2.1 \times md^{m-1}$. (See Tables 3 and 4.) This suggests that Lemma 8 holds for our examples.

4.3.2 Number of monomials

Our experiments show that the extended genericness of non-zero coefficients holds for our examples. This may suggest some similarities among the distribution of $\mathbf{GS}^{(g)}_{m+1}(a, b)$ and that of *random vectors*. We conducted a preliminary experiment on the the distribution from a statistical viewpoint for parameters $m = 3, 4$ and $B = 10, 15, 20, 25$. The details of our experiment are shown in Table 5, where for each parameter (m, B, d), we computed e samples.

Table 5: Number of monomials in coefficient polynomials
(*) For $m = 4$ and $B = 20$ in Trivial Case, we used a graded lexicographic order and computed up to $d = 7$.

Ideal	type of $F(x)$	m	B	d	e
Trivial	binomial	3	$15, 20, 25$	$5, \dots, 11$	20
		4	$15, 20^\star$	$5, \dots, 8^\star$	5
	non-binomial	3	$15, 20$	$4, \dots, 10$	20
Non-Trivial	binomial	3	$10, 15$	$6, \dots, 11$	5

- **Trivial ideal case.** For all examples, $\#\mathrm{supp}(T^{(1)}_{red})$ coincides with d^m. In addition, $\#\mathrm{supp}(T^{(1)}_{actual})$ is close to $(m + 1) \times d^m$ in Binomial Case.
- **Non-trivial ideal case.** For every g in $\mathcal{G}_{a,b}$, except for $g = F(x_i)$ for some i, $\#\mathrm{supp}(T^{(g)}_{actual})$ is much larger than d^m and $\#\mathrm{supp}((T^{(g)}_{actual})_{red})$ is almost equal to d^m. For 53 of the 60 examples, we have

$$\# \bigcup_{g \in \mathcal{G}_{a,b}} \mathrm{supp}((T^{(g)}_{actual})_{red}) = d^m$$

  and the largest gap between those values is 4, appearing at one example with $d = 11$ and $B = 10$. The largest gap between $\#\mathrm{supp}((T^{(g)}_{actual})_{red})$ and d^m, is 18, appearing at one example with $d = 8$ and $B = 10$ in our examples.

5 Further discussion on degree bound and fall degree

Here, we remark two important notions, *degree bound and fall degree*, which are considered as fundamental tools for estimating the complexity of Gröbner basis computations. In our setting, we can apply very simple arguments for these notions and achieve a more precise estimation.

5.1 Homogenization and fall degree

In general, it is difficult to precisely estimate the complexity of Gröbner basis computations. However, for homogeneous ideals with a *graded* monomial ordering such as a graded reverse lexicographic ordering, we can use properties of their graded structure such as Hilbert polynomials and their related regularities. (For definitions, see Chapters 2 and 9 in [5] or Chapter 5 in [21].) To analyze the complexity of Gröbner basis computations for such ideals, the degree bound on elements of Gröbner basis is crucial because by considering *Macaulay matrices*, an upper bound on the complexity can be easily calculated. Then, many of existing estimations on degree bounds were obtained by examining certain regularities. For a non-homogeneous ideal, the computational cost can be reduced to that of its *homogenization*.

We begin by recalling some useful properties related to homogenization technique. (See [21] for details and proofs of propositions.) Then, we define *local fall degree*, which is much different from last/first fall degree but does not depend on any algorithm for Gröbner basis computation. Our local fall degree is defined for each element of a given ideal and, if its Gröbner basis consists of elements of *smaller total degree*, its *maximal local fall degree* is expected to be close to the regularity.

Definition 9. *Let y be a new variable and consider $K[x_1, \dots, x_n, y]$.*

1. *For a polynomial $f(x_1, \dots, x_n)$ in $K[x_1, \dots, x_n]$, its* homogenization, *denoted by f^h, is defined as $f^h(x_1, \dots, x_n, y) = y^{\deg(f)} f(x_1/y, \dots, x_n/y)$. Moreover, for a subset $\mathcal{H}$ of $K[x_1, \dots, x_n]$, its* homogenization, *denoted by $\mathcal{H}^h$, is defined by $\mathcal{H}^h = \{f^h \mid f \in \mathcal{H}\}$.*
2. *For a homogeneous polynomial $\tilde{f}(x_1, \dots, x_n, y)$, its* dehomogenization *is defined as $\tilde{f}(x_1, \dots, x_n, 1)$ and denoted by $\tilde{f}|_{y=1}$. In the same manner, the* dehomogenization *of a set of homogeneous polynomials is defined.*
3. *For an ideal L of $K[x_1, \dots, x_n]$, its* homogenization L^h as an ideal *is defined as the ideal of $K[x_1, \dots, x_n, y]$ generated by $\{f^h \mid f \in L\}$.*
4. *For a graded ordering $\prec$ on the set of monomials in $\{x_1, \dots, x_n\}$, its* homogenization *$\prec_h$ is defined as follows: For monomials t_1, t_2 in $K[x_1, \dots, x_n, y]$, $t_1 \prec_h t_2$ if $\deg(t_1) < \deg(t_2)$ or $\deg(t_1) = \deg(t_2)$ and $t_1|_{y=1} \prec t_2|_{y=1}$.*

Now, we consider a subset $\mathcal{H}$ of $K[x_1, \dots, x_n]$, its homogenization $\mathcal{H}^h$, a graded monomial ordering $\prec$, its homogenization $\prec_h$, and the ideal L generated by $\mathcal{H}$.

Proposition 12 (Propositions 4.3.18, 4.3.21 in [21]). *Let $\tilde{\mathcal{H}}$ be the reduced Gröbner basis of $\langle \mathcal{H}^h \rangle$ with respect to $\prec_h$. Then, $\tilde{\mathcal{H}}$ consists of homogeneous polynomials and its dehomogenization $\tilde{\mathcal{H}}|_{y=1} = \{\tilde{f}|_{y=1} \mid \tilde{f} \in \tilde{\mathcal{H}}\}$ is a Gröbner basis of L with respect to $\prec$. If $\mathcal{G}_L$ is a Gröbner basis of L, then $\langle {\mathcal{G}_L}^h \rangle = L^h$ and ${\mathcal{G}_L}^h$ is a Gröbner basis of L^h with respect to $\prec_h$.*

Proposition 12 implies that, for a non-homogeneous ideal, the total cost of Gröbner basis computation can be reduced to that of its homogenized ideal.

Proposition 13 (Corollary 4.3.8 in [21]). *Let L^h be the homogenization of L as an ideal. Then, there is a positive integer ℓ such that*

$$L^h = (\langle \mathcal{H}^h \rangle : y^\ell). \tag{19}$$

Therefore, for each element f of L, there exists a non-negative integer $u \le \ell$ such that $f^h y^u$ belongs to $\langle \mathcal{H}^h \rangle$. The exponent ℓ can be determined as by saturation; *$(\langle \mathcal{H}^h \rangle : y^\infty) = (\langle \mathcal{H}^h \rangle : y^\ell)$. (See also Chapter 4 Section 4 in [5].)*

Now, we give the definition of *local fall degree*.

Definition 10. *For each f in L, the smallest non-negative integer u, such that $f^h y^u$ belongs to $\langle \mathcal{H}^h \rangle$, is called the* local fall degree *of f. If $u > 0$, we say that* a degree fall occurs locally *at f. Moreover, the smallest positive integer ℓ satisfying the formula* (19) *can be considered as the* maximal local fall degree *of Gröbner basis computation, as there is an element in the reduced Gröbner basis with local fall degree ℓ.*

Remark 11. *We remark on* normal selection strategy *and* sugar degree *again here. In Section 4.1.2, we mentioned that these techniques may make the computational behavior close to those of signature-based algorithms.*

Simultaneously, these techniques make the computational behavior close to that of the homogenization of the ideal. For each f in an ideal, its local fall degree indicates the gap between the actual total degree of f and that of its image in the homogenization.

5.2 On degree bound

By Lemma 2 in Section 3.2.2, the number Reg gives an upper bound on the total degree of elements of the reduced Gröbner basis. In fact, Reg just coincides with the well-known upper bound based on the regularity of the ideal I. By *regularity*, we may mean either the Hilbert or the Castelnuovo-Mumford regularity. (See [17] for details and some extensions of the following.)

Proposition 14 ([23]). *Consider an ideal L generated by a finite set $\{f_1, \dots, f_r\}$ in $K[x_1, \dots, x_n]$, where* $\deg(f_1) \geq \deg(f_2) \geq \cdots \geq \deg(f_r)$ *and $r \geq n$, and the reduced Gröbner basis $\mathcal{G}_L$ with respect to a graded reverse lexicographic ordering. If the projective dimension of the homogeneous ideal $\tilde{L} = \langle f_1^h, \dots, f_r^h \rangle$ is at most 0, then for any $g \in \mathcal{G}_L$,*

$$\deg(g) \leq \deg(f_1) + \cdots + \deg(f_{n+1}) - n + 1,$$

where we set $\deg(f_{n+1}) = 1$ *if $r = n$.*

In our case, it can be easily seen that $Reg = md + d_S - m$ coincides with the bound in Proposition 14. For Trivial ideal case, Reg coincides with e, where y^e is the unique element in $\mathbb{F}_q[y]$ of the reduced Gröbner basis of the homogeneous ideal $\tilde{I}$ generated by $\mathcal{F}_S^h$, with high probability under Assumption 11. Moreover, for the ideal $\tilde{I}$ in Non-trivial ideal case, we can show that $Reg - 1$ gives a tight-bound by simple arguments.

Trivial ideal case: Suppose that I has no zero, that is, its reduced Gröbner basis is $\{1\}$. Then, the reduced Gröbner basis of the ideal $\tilde{I}$ generated by $\mathcal{F}_S^h$ contains y^e for some positive integer e. We consider the standard form of 1;

$$1 = TS + A_1 F(x_1) + \cdots + A_m F(x_m),$$

where $T = \mathrm{RSC}(1)$ and $\deg(TS) \geq \deg(A_i F(x_i))$ for any i. Moreover, there are homogeneous polynomials $\tilde{T}, \tilde{A}_1, \dots, \tilde{A}_m$ such that

$$y^e = \tilde{T}S^h + \tilde{A}_1 F^h(x_1) + \cdots + \tilde{A}_m F^h(x_m),$$

where $\tilde{T}$ is reduced with respect to $F^h(x_1), \dots, F^h(x_m)$. Then, it can be shown directly that $\tilde{T}|_{y=1} = T$. Moreover, $e = \deg(\tilde{T}S^h) \leq Reg$. *In appearance*, the leading monomial of T is $x_1^{d-1} \cdots x_m^{d-1}$, whose coefficient, say C_S, can be calculated as

$$C_S = \sum_{\alpha=(\alpha_1,\dots,\alpha_m)\in V(J)} \frac{1}{S(\alpha)\prod_{i=1}^m F'(\alpha_i)},$$

where F' denotes the derivative of F, by the following interpolation;

$$T(x_1, \dots, x_m) = \sum_{\alpha=(\alpha_1,\dots,\alpha_m)\in V(J)} \left[\frac{1}{S(\alpha)} \prod_{i=1}^m \frac{F(x_i)}{(x_i - \alpha_i)F'(\alpha_i)} \right].$$

Under Assumption 11, we may expect that C_S does not vanish with high probability. In this case, $\deg(\tilde{T}) = \deg(T) = m(d-1)$ and hence, e coincides with $Reg = md + d_S - m$.

Remark 12. *In our experiments, as shown in Section 4.3, we have also examined that the total degree of T_{red} coincides with Reg for every example in Trivial ideal case. Thus, in this case, the signature of* 1 *is proved to be $x_1^{d-1} \cdots x_m^{d-1}$.*

Non-trivial ideal case: Next, we consider the case where I is non-trivial, that is, its reduced Gröbner basis $\mathcal{G}$ is not equal to $\{1\}$. For each element g of $\mathcal{G} \setminus \mathcal{F}$, we consider its standard form;

$$g = \mathrm{RSC}(g)S + A_1^{(g)} F(x_1) + \cdots + A_m^{(g)} F(x_m),$$

where $\deg(A_i^{(g)}F(x_i)) \le \deg(\mathrm{RSC}(g)S) \le Reg$. Since $\mathrm{Syz} = (J : S) \supsetneq J$, there is some element h in the Gröbner basis of Syz whose leading monomial belongs to $\mathcal{M}_{red}$. On the other hand, $x_1^{d-1}x_2^{d-1}\cdots x_m^{d-1}$ is the unique element in $\mathcal{M}_{red}$ whose total degree is $m(d-1)$, and any element in $\mathcal{M}_{red}$ divides $x_1^{d-1}x_2^{d-1}\cdots x_m^{d-1}$. Thus, it follows that the largest total degree of elements in $NS(\mathrm{Syz})$ is strictly less than $m(d-1)$ and $\deg(\mathrm{RCS}(g)) < md - m$. Thus,

$$\deg(g) \le \deg(\mathrm{RCS}(g)S) = \deg(\mathrm{RSC}(g)) + \deg(S) \le md + d_S - m - 1.$$

5.3 On fall degree

When I is trivial or has a few zeros, its reduced Gröbner basis is small, that is, its elements have smaller degrees. Such elements appear due to non-trivial syzygies among the generating polynomials $S, F(x_1), \dots, F(x_m)$, and our *local degree falls* occur. Moreover, the maximal local fall degree is very close to the last fall degree defined in [19]. As shown in Section 5.2, for Trivial ideal case, the generation of 1 as the unique element of the reduced Gröbner basis exactly corresponds to that of y^e as an element of the reduced Gröbner basis of $\langle \mathcal{F}_S^h \rangle$, and with high probability, e coincides with the bound Reg under our assumption. This implies that for such a case, at the final step, there occurs the largest local degree fall, and Reg shall be the *last fall degree* of $\mathcal{F}$ defined in [19]. (See Definition 11 below.) This occurs because 1 belongs to $V_{\max(e,\deg(1))} = V_e$ but does not belong to V_{e-1}. Moreover, as 1 belongs to V_e, for any element f in I with $\deg(f) \le e$, f also belongs to V_e. This exactly supports the last fall degree assumption.

Definition 11 (Last fall degree). *For a finite subset $\mathcal{H}$ and the ideal L of a polynomial ring R generated by $\mathcal{H}$, the* last fall degree *is defined as the smallest integer c, such that for all f in L, f belongs to $V_{\max(c,\deg(f))}$, where V_i is the smallest K-vector space satisfying the following:*
(1) all $f \in \mathcal{H}$ with $\deg(f) \le i$ *are included in V_i,*
(2) for $g \in V_i$ and $h \in R$, if $\deg(gh) \le i$*, then gh belongs to V_i.*

5.4 Index calculus with Weil descent technique for extension fields

Shapes of ideals appearing in improved index calculus methods using Weil descent technique are different from ours, and analyzing local fall degrees and degree bounds becomes complicated and difficult. Using Weil descent technique, Semaev's summation polynomials are divided into n distinct polynomials, which yield the same solutions of PDP. (Here, PDP is defined over $\mathbb{F}_{p^n}$.) As an effect of such a *division*, the upper bound on the degree in Proposition 14 becomes smaller. Meanwhile, for smaller binary fields, an interesting behavior, called the *first fall degree assumption*, was observed; the *first fall degree* coincided with the *actual regularity*. Some authors tried to estimate the complexity based on this assumption. (See [28, 34].) However, counter examples were reported in [19]. We note that, in a special case where parameters of Weil descent technique are set in *very particular way*, it was shown that some improved index calculus methods can outperform Pollard's rho method. (See [6, 18].)

Using our approach, we may focus on *coefficient* polynomials $T_1^{(g)}, \dots, T_n^{(g)}$ for each element g appearing in the computation of Gröbner basis;

$$g = T_1^{(g)}S_{m+1,1} + \cdots + T_n^{(g)}S_{m+1,n} + (\text{ other terms }),$$

where $S_{m+1,1}, \dots, S_{m+1,n}$ are polynomials derived from the Semaev's summation polynomial $S_{m+1}(x_1, \dots, x_m, x(aP + bQ))$. Then, the actual local fall degree will be estimated by the total degree of $T_1^{(g)}, \dots, T_n^{(g)}$ and a lower bound on the cost of Gröbner basis computation may be estimated by the quotient module of R^m by the submodule consisting of syzygies among $S_{m+1,1}, \dots, S_{m+1,n}$. We expect that our simple arguments may investigate deep insights on the subject.

6 Concluding remarks

In this paper, we presented simple arguments for giving a lower bound on the complexity of Gröbner basis computation (using S-polynomials) of the ideal derived from PDP, a dominant part of the index calculus method. As the first step in obtaining a meaningful bound, we considered an index calculus method of *very naive form* as our target. To do this, we extracted certain essential properties of the ideals appearing in solving PDP. As such ideals have very special shape (named *of special type* here), we applied rather simple and easy arguments for analyzing behaviors of Gröbner basis computations.

Conclusion

As a result, under simple statistical assumptions on Semaev's summation polynomials, we succeeded in obtaining a lower bound on Gröbner basis computation. In our experiments, the validity of assumptions was examined. (Since our computation is related to *parametric linear systems solving*, once our assumption holds for some examples, it may hold for almost every examples.) By the obtained bound, we concluded that the naive index calculus method cannot be more efficient than Pollard's rho method and even the brute force method under our experimental observations. We remark that when the ideal has no zero, that is, PDP fails, computation of the Gröbner basis means producing the unique element 1 at the final stage. In this case, a large local degree fall is obsearved at the final step of the computation, which exactly corresponds to the *last fall degree* assumption.

Future work

In our next work, we will attempt to conduct more experiments on larger examples for examining our statistical assumptions and to prove that our definition of the signature is the most efficient and, with this signature, signature-based algorithms can always produce fewer number of S-polynomials in our setting. In addition, we will apply our simple arguments to several improvements in the index calculus method, as discussed in Section 5.4. Thus, our next target shall be other methods (e.g. [6, 14, 34]) using Weil descent for ECDLP over binary extension fields, where the corresponding ideals have complicated shapes. Applying our arguments to these complicated ideals, new insights on the behavior of Gröbner basis computations of ideals might be extracted. As a byproduct, we also expect to find some easily computable parameters for ECDLP.

Acknowledgement: The authors would like to thank Masayuki Noro for his support on *Risa/Asir* system for analyzing the actual behavior of Gröbner basis computation. They are grateful to anonymous reviewers for their helpful comments.

References

[1] Alessandro Amadori, Federico Pintore and Massimiliano Sala, On the discrete logarithm problem for prime-field elliptic curves, *Finite Fields and Their Applications* **51** (2018), 168–182.

[2] Daniel J. Bernstein, Susanne Engels, Tanja Lange, Ruben Niederhagen, Christof Paar, Peter Schwabe and Ralf Zimmermann, Faster elliptic-curve discrete logarithms on FPGAs, *IACR Cryptology ePrint Archive 2016/382* (2016).

[3] Ian F Blake, Gadiel Seroussi and Nigel Smart, *Elliptic curves in cryptography*, 265, Cambridge university press, 1999.

[4] Joppe W Bos, Marcelo E Kaihara, Thorsten Kleinjung, Arjen K Lenstra and Peter L Montgomery, Solving a 112-bit prime elliptic curve discrete logarithm problem on game consoles using sloppy reduction, *International Journal of Applied Cryptography* **2** (2012), 212–228.

[5] David Cox, John Little and Donal O'Shea, *Ideals, varieties, and algorithms*, fourth ed, Undergraduate Text in Mathematics, Springer, 2015.

[6] Claus Diem, On the discrete logarithm problem in elliptic curves, *Compositio Mathematica* **147** (2011), 75–104.

[7] Christian Eder and Jean-Charles Faugère, A survey on signature-based algorithms for computing Gröbner bases, *Journal of Symbolic Computation* **80** (2017), 719–784.

[8] Jean-Charles Faugère, A new efficient algorithm for computing Gröbner bases (F4), *Journal of pure and applied algebra* **139** (1999), 61–88.

[9] Jean-Charles Faugère, A new efficient algorithm for computing Gröbner bases without reduction to zero (F5), in: *International Symposium on Symbolic and Algebraic Computation–ISSAC 2002*, pp. 75–83, ACM, 2002.

[10] Jean-Charles Faugère, Louise Huot, Antoine Joux, Guénaël Renault and Vanessa Vitse, Symmetrized summation polynomials: Using small order torsion points to speed up elliptic curve index calculus, in: *Advances in Cryptology–EUROCRYPT 2014*, LNCS 8441, Springer, pp. 40–57, 2014.

[11] Jean-Charles Faugère, Ludovic Perret, Christophe Petit and Guénaël Renault, Improving the complexity of index calculus algorithms in elliptic curves over binary fields, in: *Advances in Cryptology–EUROCRYPT 2012*, LNCS 7237, Springer, pp. 27–44, 2012.

[12] Ryoya Fukasaku, Shutaro Inoue and Yosuke Sato, On QE algorithms over an algebraically closed field based on comprehensive Gröbner systems, *Mathematics in Computer Science* **9** (2015), 267–281.

[13] Steven D Galbraith and Pierrick Gaudry, Recent progress on the elliptic curve discrete logarithm problem, *Designs, Codes and Cryptography* **78** (2016), 51–72.

[14] Pierrick Gaudry, Index calculus for abelian varieties of small dimension and the elliptic curve discrete logarithm problem, *Journal of Symbolic Computation* **44** (2009), 1690–1702.

[15] Gert-Martin Greuel and Gerhard Pfister, *A Singular Introduction to Commutative Algebra*, second ed, Springer, 2008.

[16] Darrel Hankerson, Alfred J Menezes and Scott Vanstone, *Guide to elliptic curve cryptography*, Springer Science & Business Media, 2006.

[17] Amir Hashemi and Werner M Seiler, Dimension-dependent upper bounds for Gröbner Bases, in: *Proceedings of the 2017 ACM on International Symposium on Symbolic and Algebraic Computation–ISSAC 2017*, ACM, pp. 189–196, 2017.

[18] Ming-Deh A Huang, Michiel Kosters, Christophe Petit, Sze Ling Yeo and Yang Yun, Quasi-subfield polynomials and the elliptic curve discrete logarithm problem, *Journal of Mathematical Cryptology* (to appear).

[19] Ming-Deh A Huang, Michiel Kosters and Sze Ling Yeo, Last fall degree, HFE, and Weil descent attacks on ECDLP, in: *Advances in Cryptology–CRYPTO 2015*, LNCS 9215, Springer, pp. 581–600, 2015.

[20] Neal Koblitz, Elliptic curve cryptosystems, *Mathematics of Computation* **48** (1987), 203–209.

[21] Martin Kreuzer and Lorenzo Robbiano, *Computational Commutative Algebra 2*, Springer, 2005.

[22] Momonari Kudo, Yuki Yokota, Yasushi Takahashi and Masaya Yasuda, Acceleration of index calculus for solving ECDLP over prime fields and Its limitation, in: *Cryptology and Network Security–CANS 2018*, LNCS 11124, Springer, pp. 377–393, 2018.

[23] Daniel Lazard, Gröbner bases, Gaussian elimination and resolution of systems, in: *European Conference on Computer Algebra–EUROCAL 1983*, LNCS 162, 1983.

[24] Alfred J Menezes, Tatsuaki Okamoto and Scott A Vanstone, Reducing elliptic curve logarithms to logarithms in a finite field, *IEEE Transactions on Information Theory* **39** (1993), 1639–1646.

[25] Victor S Miller, Use of elliptic curves in cryptography, in: *Advances in Cryptology–CRYPTO 1985*, LNCS 218, pp. 417–426, Springer, 1985.

[26] Bhubaneswar Mishra, *Algorithmic Algebra*, Text and Monographs in Computer Science, Springer, 1992.

[27] Christophe Petit, Michiel Kosters and Ange Messeng, Algebraic approaches for the elliptic curve discrete logarithm problem over prime fields, in: *IACR International Workshop on Public Key Cryptography–PKC 2016*, LNCS 9615, pp. 3–18, Springer, 2016.

[28] Christophe Petit and Jean-Jacques Quisquater, On polynomial systems arising from a Weil descent, in: *Advances in Cryptology–ASIACRYPT 2012*, LNCS 7658, Springer, pp. 451–466, 2012.

[29] John M Pollard, Monte Carlo methods for index computation (mod p), *Mathematics of Computation* **32** (1978), 918–924.

[30] Ronald L. Rivest, Adi Shamir and Leonard Adleman, A method for obtaining digital signatures and public-key cryptosystems, *Communications of the ACM* **21** (1978), 120–126.

[31] Takakazu Satoh and Kiyomichi Araki, Fermat quotients and the polynomial time discrete log algorithm for anomalous elliptic curves, *Commentarii Mathematici Universitatis Sancti Pauli* **47** (1998), 81–92.

[32] Igor A Semaev, Evaluation of discrete logarithms in a group of p-torsion points of an elliptic curve in characteristic p, *Mathematics of Computation* **67** (1998), 353–356.

[33] Igor A Semaev, Summation polynomials and the discrete logarithm problem on elliptic curves, *IACR Cryptology ePrint Archive 2004/031* (2004).

[34] Igor A Semaev, New algorithm for the discrete logarithm problem on elliptic curves, *IACR Cryptology eprint Archive 2015/310* (2015).

[35] Daniel Shanks, Class number, a theory of factorization, and genera, in: *Proc. of Symp. Math. Soc., 1971*, 20, pp. 41–440, 1971.

[36] Nigel P Smart, The discrete logarithm problem on elliptic curves of trace one, *Journal of Cryptology* **12** (1999), 193–196.

[37] Tristan Vaccon, Thibaut Verron and Yokoyama Kazuhiro, On affine tropical F5 algorithm, in: *Proceedings of the 2018 International Symposium on Symbolic and Algebraic Computation–ISSAC 2018*, ACM, pp. 383–390, 2018.

[38] Tristan Vaccon and Kazuhiro Yokoyama, A tropical F5 algorithm, in: *Proceedings of the 2017 ACM on International Symposium on Symbolic and Algebraic Computation–ISSAC 2017*, ACM, pp. 429–436, 2017.

[39] Wolmer V Vasconcelos, *Computational Methods in Commutative Algebra and Algebraic Geometry*, Algorithms and Computation in Mathematics 2, Springer, 1998.

[40] Joahim von zur Gathen and Jürgen Gerhard, *Modern Computer Algebra*, Cambridge University Press, 1999.

[41] Erich Wenger and Paul Wolfger, Solving the discrete logarithm of a 113-bit Koblitz curve with an FPGA cluster, in: *International Conference on Selected Areas in Cryptography–SAC 2014*, LNCS 8781, pp. 363–379, Springer, 2014.

[42] Masaya Yasuda, Takeshi Shimoyama, Jun Kogure and Tetsuya Izu, Computational hardness of IFP and ECDLP, *Applicable Algebra in Engineering, Communication and Computing* **27** (2016), 493–521.