Complexity Bound on Semaev's Naive Index Calculus Method for ECDLP

Kazuhiro Yokoyama[1], Masaya Yasuda[2], Yasushi Takahashi[3], and Jun Kogure[3]

[1] Rikkyo University, Tokyo, Japan
[2] Kyushu University, Fukuoka, Japan
[3] FUJITSU Laboratories LTD, Kawasaki, Japan

Abstract. Since Semaev introduced summation polynomials, a number of works have been devoted to improve and analyze index calculus algorithms for solving the elliptic curve discrete logarithm problem (ECDLP) with better complexity than generic algorithms such as the rho method. In this paper, we give a deep analysis on Gröbner basis computation for a polynomial system appearing in the point decomposition problem (PDP) in a *naive* index calculus method. Our analysis is based on linear algebra under simple statistical assumptions on Semaev's summation polynomials. Specifically, we show that the ideal derived from the PDP has a special structure, and regard Gröbner basis computation for the ideal as an extension of the extended Euclidean algorithm to estimate its cost. This approach allows us to obtain a precise analysis on a lower bound on the cost of Gröbner basis computation. We also prove that the naive index calculus cannot be more efficient than even the brute force method for the ECDLP over arbitrary finite field.

1 Introduction

In public-key cryptography, the RSA cryptosystem [31] and elliptic curve cryptography (ECC) [19, 26] have been most widely used in modern information society. The securities of RSA and ECC are based on the hardness of the integer factorization problem (IFP) and the ECDLP, respectively. While there exist sub-exponential time algorithms to solve the IFP, no such algorithm exists to solve the ECDLP for cryptographic parameters. The best algorithm for solving the ECDLP is Pollard's rho method [30] except special cases such as supersingular [25] and anomalous curves [32, 33, 37], and its complexity is the square-root of the order of an elliptic curve over a finite field. (See also textbooks [3, 16].) Currently known largest records for the ECDLP are 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. (As a special case of prime fields, a 114-bit ECDLP over a Barreto-Naehrig curve was solved in [22].)

The index calculus method (ICM) is useful to solve the DLP over a general cyclic group, and its idea is to reduce the DLP to linear algebra by collecting relations over so-called a factor base. After Semaev [34] introduced summation polynomials associated with an elliptic curve, one can solve the ECDLP in the

index calculus framework. In particular, finding a relation is reduced to a problem of solving a system including Semaev's polynomials, in order to decompose a point of an elliptic curve into a sum of points of a factor base. Semaev's paper [34] triggered many works aiming to develop index calculus algorithms with better complexity than the rho method. As typical works, Gaudry [14] and Diem [7] fully developed Semaev's approach based on Weil descent for the ECDLP over a finite field $\mathbb{F}_{q^n}$ with small extension degree n. In 2012, Faugère et al. [12] showed that a system arising in algorithms based on Weil descent has special structures, making it easier to solve the system by Gröbner basis algorithms such as F_4 and F_5 [9, 10]. Shortly after that, Petit and Quisquater [29] revisited Faugère et al.'s work to claim the subexponentiality for the ECDLP over any binary field, under the heuristic assumption of *first fall degree* (FFD) about the behavior of Gröbner basis algorithms. However, Huang et al. [18] in 2015 provided computational evidence that raises doubt on the validity of the FFD assumption, and also introduced another notion called the *last fall degree* (LFD) to develop complexity bounds on solving a polynomial system. At present, as mentioned in a survey [13], there is no consensus (not at all definitive) whether there exists a sub-exponential time algorithm to solve the ECDLP over binary fields, since the FFD assumption is too optimistic while the LFD approach looks more precise but much harder to estimate. On the other hand, for prime fields $\mathbb{F}_p$, Petit et al. [28] 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 of [21] showed that both methods of [1, 28] do not outperform generic algorithms such as the BSGS [36] and the rho methods for solving the ECDLP over prime fields.

In this paper, we give a *deep* analysis on Gröbner basis computation for a polynomial system appearing a naive index calculus using Semaev's polynomials, as the *first step* for investigating the complexity of Gröbner basis computation used in the ICM. Looking the special structure of the ideal associated to the ICM, we regard its Gröbner basis computation as an extension of the *Euclidean algorithm* for polynomial GCD, and use the shape of the *coefficient polynomial* (see Equation (5)) to count the number of operations in Gröbner basis computation. Especially, we observe that *leading monomials* of coefficient polynomials can bring precise information on S-polynomials. Under simple statistical assumptions on the behavior of Semaev's polynomials, the sequence of S-polynomials behaves like *regular* polynomial reminder sequences in GCD computation. This enables us to obtain a lower bound on the complexity of Gröbner basis computation (Sections 4.1 and 4.2), under such assumptions. (Compared to assumptions on fall degrees in [18, 29], ours are simple and reasonable to hold in practice.) Our analysis also gives *precise* results related to notions of *degree bound* and *degree fall* (Lemma 3 and Proposition 12). These results are applicable directly to other variants of index calculus like [1, 28].

Finally, with our complexity bounds, we prove that the naive index calculus method cannot be more efficient than even the brute force method for solving the

ECDLP over arbitrary finite field. This might be a known fact by seeing existing upper bounds on the complexity. (See [24].) However, it should be emphasized that our bound is a lower one, by which some hope for getting better actual complexity are vanished. We think that our approach can be applied to estimate some lower bound on advanced algebraic methods using Weil descent. Making good use of algebraic structures is very important to investigate precisely the complexity of Gröbner basis computation, and our trial shall be one of successful examples.

We remark that our approach is based on extended GCD and heavily related to resultant theory. Such GCD and resultant theory (or Bezout's theory) were already applied in [18, 27] to analyze the complexity of algebraic computations for solving ECDLP or Hidden Field Equation(HFE). But, it was used for obtaining some *upper bounds* on degrees of polynomials appearing in algebraic computations, such as Gröbner basis. On the other hand, to obtain a lower bound, it is more important not to estimate the largest degree, but to analyze monomials appearing in *coefficient polynomials*.

2 Index Calculus Method for ECDLP

Here we recall the ECDLP and the *index calculus method* (ICM) based on Semaev's summation polynomials.

Definition 1 (ECDLP) *Let $\mathbb{F}_q$ be the finite field of order q, E an elliptic curve defined over $\mathbb{F}_q$ and $E(\mathbb{F}_q)$ the group of rational points of E over $\mathbb{F}_q$. We call the following the elliptic curve discrete logarithm problem (ECDLP) on E:*

Given points P, Q in $E(\mathbb{F}_q)$, find a positive integer ℓ such that $Q = \ell P$.

Here we assume that Q belongs to the additive group $\langle P \rangle$ generated by P. Such a positive integer ℓ with $0 \leq \ell < \mathrm{ord}(P)$ is denoted by $\log_P(Q)$, where $\mathrm{ord}(P)$ denotes the order of P in the group $E(\mathbb{F}_q)$.

We give a general form of the ICM for solving the ECDLP. We note $A \approx B$ 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) Take a subset $\mathcal{B}$ of $E(\mathbb{F}_q)$, which is called a *factor base*. In our *naive setting*, we first take a subset V of $\mathbb{F}_q$ and $\mathcal{B} = \{(x, y) \in E(\mathbb{F}_q) \mid x \in V\}$. Then, $\#\mathcal{B} \leq 2\#V$. With high probability, $\#V \approx \#\mathcal{B}$, since, for each x in V, the ratio of (x, y) belonging to $E(\mathbb{F}_q)$ is almost $\frac{1}{2}$. We number elements of $\mathcal{B}$ as $\mathcal{B} = \{B_1, \ldots, B_t\}$. Here we denote the size $\#\mathcal{B}$ by t and we also denote by d that of V.

(Step 2) Generate randomly two integers a, b such that $0 \leq a, b < ord(P)$, and find the following decomposition;

$$aP + bQ = B_{s_1} + \cdots + B_{s_m}, \quad m \in \{1, \ldots, t\}, \tag{1}$$

where we fix m in advance. When it succeeds, one has the following linear congruence:

$$a + b \log_P(Q) \equiv \sum_{i=1}^{m} \log_P(B_{s_i}) \pmod{\mathrm{ord}(P)}. \tag{2}$$

We call this computational problem *the point decomposition problem* (PDP).

(Step 3) After collecting enough number of linear congruences, compute $\log_P(Q)$ by linear algebra. In fact, after collecting $t+1$ *linearly independent* congruences, $\log_p(Q)$ can be determined uniquely.

2.1 Semaev's Summation Polynomials

Here we revisit an approach for solving the PDP by Semaev [34], where the PDP can be reduced to finding zeros of the system of algebraic equations derived from so-called Semaev's summation polynomial.

From now on, we fix the finite field $\mathbb{F}_q$ and an elliptic curve E defined over $\mathbb{F}_q$. Moreover, let $\mathcal{B}$ be a factor base and $V = \{x(P) \in \mathbb{F}_q \mid P \in \mathcal{B}\}$, where $x(P)$ denotes the x-coordinate of a point P of E. We denote the algebraic closure of $\mathbb{F}_q$ by $\overline{\mathbb{F}}_q$, the total degree of a polynomial f by $\deg(f)$ and its degree with respect to a variable x_i by $\deg_{x_i}(f)$,

Definition 2 (Summation Polynomial) *Let r be a positive integer greater than 1. 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 *if it satisfies the following:*

For any zero $(b_1, \ldots, b_r)$ of S_r, that is, $S_r(b_1, \ldots, b_r) = 0$, there exist points $P_1, \ldots, P_r$ of $E(\overline{\mathbb{F}}_q)$ such that

$$P_1 + \cdots + P_r = \mathcal{O},$$

and $b_i = x(P_i)$ for $1 \leq i \leq r$, where $\mathcal{O}$ is the infinity point.

Semaev gave a method for constructing summation polynomials by resultant in a recursive manner, which we also call *Semaev's summation polynomials* (SSPs). The SSP of order r can be computed by the previously computed SSP of order $r-1$ and the SSP of order 3 as follows:

$$S_r = \mathrm{Res}_y(S_{r-1}(x_1, \ldots, x_{r-2}, y), S_3(x_{r-1}, x_r, y)),$$

where $\mathrm{Res}_y(f, g)$ denotes the resultant of two polynomials f, g with respect to a variable y. The correctness of this construction can be shown easily, by which the existence of SSPs is guaranteed. We can also generalize this construction by

$$S_{r+k} = \mathrm{Res}_y(S_{r+1}(x_1, \ldots, x_r, y), S_{k+1}(x_{r+1}, \ldots, x_{r+k}, y)).$$

We note that, over any field, SSPs can be defined and computed.

Proposition 1 (Property of SSP) *For $r \geq 3$, S_r is symmetric in variables $x_1, \ldots, x_r$. For each variable x_i, $\deg_{x_i}(S_r) = 2^{r-2}$ and $\deg(S_r) = (r-1)2^{r-2}$.*

Remark 1 *The polynomial S_r has its total degree $(r-1)2^{r-2}$. Also, by our experiment, it is observed that $S(x_1, \ldots, x_{r-1}, a)$ is a dense polynomial for almost every a in $\mathbb{F}_q$. This implies certain difficulty on its computation by naive resultant computation. Actually, it is very hard to compute S_r for $r > 7$. On the other hand, by making good use of its symmetrical property in variables, its simplified modification (called symmetrized summation polynomial) is proposed in [11], where the polynomial of order 8 is computed. Also, the cost of computation of S_r requires $O(2^{(r-1)^2})$ arithmetic operations over $\mathbb{F}_q$.*

2.2 Solving PDP by SSP

The PDP in Step 2 can be reduced to a problem for finding zeros of a system of algebraic equations by using S_{m+1}. For randomly chosen a, b of $\mathbb{F}_q$, finding $B_1, \ldots, B_m$ in $\mathcal{B}$ such that $aP + bQ = B_1 + \cdots + B_m$ is reduced to finding a zero $(\alpha_1, \ldots, \alpha_m) \in V^m$ of $S_{m+1}(\alpha_1, \ldots, \alpha_m, x(aP + bQ)) = 0$, where each α_i corresponds to $x(B_i)$ and $V = \{x(P) \mid P \in \mathcal{B}\}$. Letting $F = \prod_{\alpha \in V}(x - \alpha)$, we have the following system of algebraic equations, which we denote by $\mathcal{S}_{a,b}$.

$$\mathcal{S}_{a,b}: \quad \begin{cases} S_{m+1}(x_1, \ldots, x_m, x(aP + bQ)) = 0, \\ F(x_1) = 0, \\ \vdots \quad\quad \vdots \\ F(x_m) = 0. \end{cases}$$

Since $\mathcal{B}$ is a subset of $E(\mathbb{F}_q)$, V is also a subset of $\mathbb{F}_q$ and $F(x)$ is a factor of $x^q - x$ over $\mathbb{F}_q$.

Thus, solving the PDP for $aP + bQ$ is reduced to computing zeros of the system $\mathcal{S}_{a,b}$. As we need exact zeros, we have to apply a symbolic and algebraic method for their computation. Among such algebraic methods, Gröbner basis is very suited and supposed to be the most efficient.

Now we let $I_m(a, b)$ be the ideal associated to the system of equations, that is,

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

where $\langle \mathcal{A} \rangle$ denotes the ideal generated by a subset $\mathcal{A}$ in the polynomial ring $\mathbb{F}_q[x_1, \ldots, x_m]$. Then, all zeros are rational, that is, they belong to $\mathbb{F}_q^m$, if exist. Thus, $I_m(a, b)$ is trivial or 0-dimensional. As the ideal has very special shape, here we give a special name to it, by which we can handle any *variant (modification)* of SSP.

Definition 3 *An ideal is said to be of* special type*, if it is generated by separable univariate polynomials $F(x_1), \ldots, F(x_m)$ and one multivariate polynomial.*

2.3 General Estimation on Complexity of Index Calculus Method

Here we give a general but brief discussion on the cost (complexity) of the ICM given in the previous subsection. We note that, although there are several improvements on ICM, we set the simplest one due to making our analysis more precise and also comprehensive.

The efficiency of the ICM heavily depends on the choice of m and $t = \#\mathcal{B}$. So, we have to optimize the values m and t (or equivalently d) for making the complexity smaller. We give rather simple estimation on the complexity for optimizing the parameters.

1. Let $\mathcal{P}rob_{\mathrm{PDP}}$ be the probability (or ratio) of *success* of solving PDP for randomly chosen a, b in $\mathbb{F}_q$, that is, $\mathcal{S}_{a,b}$ has a zero. Then $\mathcal{P}rob_{\mathrm{PDP}}$ can be estimated approximately as
$$\frac{\#\{B_{s_1} + \cdots + B_{s_m} \mid B_{s_i} \in \mathcal{B}\}}{\#E(\mathbb{F}_q)}.$$
More precisely, we consider how many m points in $\mathcal{B}$ can generate the same point, and let γ be its average. (We may expect $\gamma \geq m!$, since $B_{\sigma(s_1)} + \cdots + B_{\sigma(s_m)}$ gives the same point for any permutation σ on $\{s_1, \ldots, s_m\}$.) Then, $\mathcal{P}rob_{\mathrm{PDP}}$ can be estimated approximately as $\dfrac{t^m}{\gamma q}$.
2. Let $\mathcal{C}_{\mathrm{dec}}$ be the average cost of solving the PDP for randomly chosen $aP+bQ$ at Step 2 which includes the average cost, written by $\mathrm{cost}(\mathcal{S}_{a,b})$, of solving the system of algebraic equations and the cost of making $S_{m+1}(x_1, \ldots, x_m, x(aP+bQ))$. We note that the PDP has γ solutions in average. Among γ solutions, the number of *essential* ones which give distinct points can be estimated roughly as $\frac{\gamma}{m!}$ due to the symmetry on solutions. Since we need approximately t (linearly independent) solutions, the total cost (on the number of arithmetic operations over $\mathbb{F}_q$) for solving the ECDLP can be estimated as
$$O\left(\frac{m! \times t \times \mathcal{C}_{\mathrm{dec}}}{\gamma \times \mathcal{P}rob_{\mathrm{PDP}}} + \mathrm{Lin}(t)\right) \quad \text{or} \quad O\left(\frac{m! \times q \times \mathcal{C}_{\mathrm{dec}}}{t^{m-1}} + t^{\omega}\right),$$
where $\mathrm{Lin}(t)$ denotes the cost for solving a system of $O(t)$ linear equations in $O(t)$ variables and it can be estimated as $O(t^{\omega})$ for the linear algebra constant ω, where $2 < \omega < 3$.
3. If we set t large enough so that $\mathcal{P}rob_{\mathrm{PDP}} \approx 1$, the total complexity shall be
$$O\left(\frac{m! \times t \times \mathcal{C}_{\mathrm{dec}}}{\gamma} + t^{\omega}\right).$$
For such case, we have to set $t^m \geq \gamma q$.

Remark 2 *It can be shown that the probability (ratio) that $I_{m+1}(a, b)$ has a zero is almost equal to $\mathcal{P}rob_{\mathrm{PDP}}$. See the detail in [35]. We also remark that, although the estimations are given by* big O-notation*, they can be considered as the average estimation.*

Further Remarks on General Estimation Here we give more details on the estimation of the cost of the ICM for the ECDLP. (See [13].) We note that for computation of S_{m+1}, it costs $O(2^{m^2})$ arithmetic operations over $\mathbb{F}_q$.

Here we consider the case where $\mathcal{P}rob_{\mathrm{PDP}}$ is smaller than 1. (In this case, we may assume that $t^m < m!q$. Then, as $\frac{t^m}{m!q} = o(1)$, we have $\gamma \approx m!$ and $\mathcal{P}rob_{\mathrm{PDP}} \approx \frac{t^m}{m!q}$. See [7, 35].)

Let $t = q^{n'}$, that is, $n' = \log_q(t)$. Then, the total cost can be estimated as

$$O(q^{1-n'(m-1)}m!\mathcal{C}_{\mathrm{dec}} + q^{n'\omega}).$$

For making the ICM efficient, the cost function should be at most $q^{1/2}$, the cost of *generic algorithms* for DLP such as the BSGS and the Pollard's rho methods. Then, as the construction of $S_{m+1}(x_1, \ldots, x_m, x(aP + bQ))$ requires $O(2^{m^2})$, it follows that $m = O(\log(q)^c)$ for some constant $c < 1$. Also, from the first term $m!q^{1-n'(m-1)}\mathcal{C}_{\mathrm{dec}}$, it follows that $1 - n'(m-1) < \frac{1}{2}$ and $\mathcal{C}_{\mathrm{dec}}$ should be also much smaller than $q^{1/2}$. From the last term $q^{n'\omega}$, $n' \leq \frac{1}{2\omega}$.

Remark 3 *Basically, the setting $t = p^{n'}$ is designed for the extension field case, where $q = p^n$ and $\mathbb{F}_q/\mathbb{F}_p$ is the extension of degree n. In this case, V is set as a subfield of $\mathbb{F}_q$. When $\mathbb{F}_q$ is a prime field, that is, q is prime, it is quite natural to set V as a multiplicative subgroup of $\mathbb{F}_q^\times$. In this setting, we have $V = \{x \in \mathbb{F}_q \mid x^k - 1\}$ for $k = \#V$. This idea is discussed in [28]. However, we cannot apply the Weil descent technique. On the other hand, it is rather suited to analyze computational behaviors of Gröbner basis computation, since those behaviors might be the same as those for the extension field case.*

3 Gröbner Basis Computation for Ideal of Special Type

Here we present precise analysis on a lower bound of the computational cost of Gröbner basis computation of the ideal $I_m(a, b)$ generated by

$$\mathcal{F}_{a,b} = \{S_{m+1}(x_1, \ldots, x_m, x(aP + bQ)), F(x_1), F(x_2), \ldots, F(x_m)\}.$$

For the fundamentals on theory of Gröbner basis, see textbooks [5, 20]. We follow [5] for the terminology on monomials and ordering.

To simplify our notation and argument, we consider an ideal of *special type* whose generating set has a similar shape as that of $\mathcal{S}_{a,b}$. Replacing the summation polynomial $S_{m+1}(x_1, \ldots, x_m, x(aP + bQ))$ with a polynomial $S(x_1, \ldots, x_m)$, we let

$$\mathcal{G}_0 = \{F(x_1), F(x_2), \ldots, F(x_m)\} \text{ and } \mathcal{F} = \mathcal{G}_0 \cup \{S(x_1, \ldots, x_m)\},$$

where F is a square-free univariate polynomial, and consider the ideal I generated by $\mathcal{F}$ in the polynomial ring $R = K[x_1, \ldots, x_m]$ over a field K. (We mainly consider $K = \mathbb{F}_q$.) We also consider the ideal J generated by $\mathcal{G}_0$. By Buchberger's criterion (see [5]), since leading monomials of $F(x_1), \ldots, F(x_m)$ are co-prime to

each other, $\mathcal{G}_0$ is the reduced Gröbner basis of J with respect to any monomial ordering. Moreover we also assume that S is reduced with respect to $\mathcal{G}_0$ and $S \neq 0$. Then I is strictly larger than J. We denote the set of all zeros of the ideal J by $V(J)$ and that of I by $V(I)$. As $I \neq J$, $V(I)$ is a proper subset of $V(J)$. We note that, since each $F(x_i)$ is *square-free*, J and I are radical ideals. For simplicity, we write d for $\deg(F)$ and d_S for $\deg(S)$. Then $\#V(J) = d^m$.

Remark 4 *As an ideal of special type, we may replace $\mathcal{G}_0$ with a Gröbner basis of a 0-dimension radical ideal in R with respect to the specified ordering $\prec$. Even for this setting, many arguments below can be applied directly, and the same results might be obtained.*

From now on, we use the following as our setting and notations related to Gröbner basis theory. We denote the set of all monomials by $\mathcal{M}$ and fix a reverse lexicographical monomial ordering $\prec$ on $\mathcal{M}$ as it produces the most efficient computation in both theory and practice. For a polynomial f, we denote the set of monomials appearing in f by $\mathrm{supp}(f)$, that is, $f = \sum_{t \in \mathrm{supp}(f)} a_t t$, where each a_t is the coefficient of the monomial t and $a_t \neq 0$. We also denote the leading monomial of f by $LM(f)$ and the leading coefficient of f by $LC(f)$. For a subset U of R, we denote the set of leading monomials $\{LM(f) \mid f \in U\}$ by $LM(U)$.

Then we will show that it requires, at least, (approximately) md^{m-1} arithmetic operations over K to compute the reduced Gröbner basis of I under *certain generic property* which is considered as an extension of the notion "regularity" of polynomial remainder sequence appearing in GCD computation. (See Assumption 2 in Section 3.4.) The idea comes from certain similarity among algorithms for Gröbner basis and the Euclidean algorithm for GCD of two univariate polynomials.

Remark 5 *Since the algebraic structure depends on the number of its zeros, our analysis on* Trivial Ideal Case*, where the ideal I has no zero, can give more precise estimation on the complexity as the easiest case. We also remark that existing works for estimating the complexity of the ICM concentrated on the case where I has some zeros. But, in Step 2 of our naive ICM, when the ratio of* success *is not exactly 1, the cost for handling* failure *cases becomes unignorable for estimating the total cost.*

Remark 6 *Basically, to compute zeros of a 0-dimensional ideal, its Gröbner basis with respect to a lexicographical ordering should be suited. However, in a computational view point, it is better to apply* change of basis *strategy, where we first compute the reduced Gröbner basis of I with respect to a reverse lexicographical ordering and then we convert it to one with respect to a lexicographical ordering. This conversion can be done very efficiently by the famous FGLM method. (See Chapter 10 in [5].)*

3.1 Euclidean Algorithm and Polynomial Remainder Sequence

We recall the Euclidean algorithm for GCD of two univariate polynomials f and g. As $\gcd(f,g)$ belongs to the ideal generated by f,g, it can be written as

$$\gcd(f,g) = Af + Bg \tag{3}$$

for some polynomials A, B. Here we call A, B the *coefficients polynomials* in the representation (3) of $\gcd(f,g)$. Tracing the Euclidean algorithm, we have its *polynomial remainder sequence* (PRS),

$$f_0 = f, f_1 = g, f_2, \ldots, f_r = \gcd(f,g),$$

where f_i is the *remainder* of the division of f_{i-2} by f_{i-1}, that is,

$$f_{i-2} = q_{i-1}f_{i-1} + f_i, \tag{4}$$

where q_{i-1} is the *quotient*. By the extended Euclidean algorithm, for each f_i, its coefficient polynomials A_i, B_i are computed;

$$f_i = A_i f + B_i g,$$

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

Remark 7 *We can translate the computation of GCD to that of Gröbner basis. The division* (4) *exactly corresponds to producing a new element in Buchberger's algorithm. The S-polynomial of g_{i-1} and g_i is $g_{i-1} - ax^{d_{i-1}-d_i}g_i$ for $d_i = \deg(f_i)$ and some $a \in K \setminus \{0\}$, and its remainder on division by $\{f_1, \ldots, f_i\}$ corresponds to f_{i+1}.*

Our computation of a Gröbner basis can be considered as certain extension of GCD and it may be quite natural to focus on the *representation*

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

for each element g of a computed Gröbner basis and use the "shape" of T for our complexity analysis.

Let us consider the costs of algorithms. The total degree $\deg(Af)$ is useful for giving an upper bound on the total cost of GCD computation. (This exactly corresponds to the *regularity* of the ideal and so it gives a degree bound. See Section 5.) Because, 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 the computation of Gröbner basis, by which we may estimate some upper bound of the complexity. See [12] for example.

On the other hand, monomials of A and B may be considered as *data* containing all history on which *monomial multiplications* are used in steps of the Euclidean algorithm. Also, they can tell which S-polynomials appear in the computation of Gröbner basis. Especially, leading monomials of A_i and B_i also give further data on the polynomial divisions appearing the whole procedure.

Example 1 (Simple Example) *We show a trivial example. Suppose that* $f = x^{n+1} + 1$ *and* $g = x^n$ *for some* $n \in \mathbb{N}$. *In this case,* $gcd(f, g) = 1$ *and the upper bound on the total degree of* A *is* n. *However, we have* $A = 1$ *and* $B = -x$ *and the GCD computation terminates within 1 step. Thus, it is important to analyze the shape of* A *for getting a precise estimation on the cost.*

Example 2 (Regular PRS) *We say that the generated PRS is* regular, *if* $\deg(f_{i-1}) = \deg(f_i) + 1$ *for* $i \geq 2$. *This case is considered as the* worst case *for the Euclidean algorithm. For simplicity, we assume that* $\deg(f) = \deg(g) + 1$ *and let* $d = \deg(f)$ *and* $D = \deg(\gcd(f, g))$. *Then, in the case,* $\deg(f_i) = d - i$ *and*

$$\deg(A_i) = d - (d - i) - 2 = i - 2,$$

and there are $d - D$ *polynomial divisions.*

Remark 8 *As mentioned in Section 1, the idea using GCD or resultant theory (or Bezout's theory) was already applied in [18, 27] to analyze the complexity of algebraic computations for solving ECDLP or HFE. It actually works for obtaining some* upper bounds *on degrees of polynomials appearing in algebraic computations, such as Gröbner basis.*

3.2 Representation of Elements of Gröbner Basis and Signature

Here we give some details on representation of elements of the computed Gröbner basis of I by $\mathcal{F}$. Eliminating unnecessary elements from it, we can extract the reduced Gröbner basis which we denote by $\mathcal{G}$. As S is assumed to be reduced with respect to $\mathcal{G}_0$, each $F(x_i)$ cannot reduce S and thus, we have $\deg_{x_i}(S) < d = \deg(F)$.

Now we consider certain *minimal* representation of elements of I with respect to $\mathcal{F}$, by which we can make precise analysis on the cost of computing Gröbner basis. For this purpose, the notion *signature* is very useful. (See [10, 8, 39] for details on signature.)

Let f be an arbitrary element in $I \setminus J$. By the definition of I, there are polynomials $T, A_1, \ldots, A_m$ such that

$$f = TS + A_1F(x_1) + \cdots + A_mF(x_m). \tag{5}$$

We call $T, A_1, \ldots, A_m$ the *coefficient polynomials* in the representation (5) of f.

Definition 4 (Syzygy) *We consider the ideal quotient of* J *by* S*;*

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

As each element h *in* $(J : S)$ *gives a* syzygy *among* $S, F(x_1), \ldots, F(x_m)$, *that is,* $hS + A_1F(x_1) + \cdots + A_mF(x_m) = 0$ *for some* $A_1, \ldots, A_m \in R$, *here we call the ideal the* syzygy ideal *with respect to* S, *and denote it by* Syz. *We also*

denote by $\tilde{\mathcal{G}}_0$ *its Gröbner basis with respect to the ordering* $\prec$*. Then, the set* $\mathcal{M}$ *of monomials is divided into two subsets;*

$$LM(\mathrm{Syz}) = \{LM(f) \mid f \in \mathrm{Syz}\} = \{t \in \mathcal{M} \mid LM(g) \mid t \text{ for some } g \in \tilde{\mathcal{G}}_0\},$$
$$NS(\mathrm{Syz}) = \mathcal{M} \setminus LM(\mathrm{Syz}) = \{t \in \mathcal{M} \mid LM(g) \not| t \text{ for any } g \in \tilde{\mathcal{G}}_0\}.$$

We also denote $\mathcal{M} \setminus LM(J)$ *by* $\mathcal{M}_{\mathrm{red}}$*. Then*

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

We note that $NS(\mathrm{Syz}) \subset \mathcal{M}_{\mathrm{red}}$*, as* $LM(J) \subset LM(\mathrm{Syz}) = LM(J : S)$*.*

Using the ideal Syz, we can *refine* the representation (5). Let $\tilde{T}$ be the normal form (remainder) of T with respect to $\tilde{\mathcal{G}}_0$. Then $T - \tilde{T}$ belongs to Syz and so $TS - \tilde{T}S$ belongs to J. Therefore, $f - \tilde{T}S$ also belongs to J. Considering the *standard representation* of $f - \tilde{T}S$ with respect to $\mathcal{G}_0$, there are polynomials $\tilde{A}_1, \ldots, \tilde{A}_m$ such that

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

from which we can show the following directly.

Lemma 2 *For each* $f \in I \setminus J$*, we consider its representation* (5)*. Then, there exist some* $\tilde{T}, A_i, \ldots, A_m \in R$ *such that* $\mathrm{supp}(\tilde{T}) \subset NS(\mathrm{Syz})$ *and*

$$f = \tilde{T}S + A_1 F(x_1) + \cdots + A_m F(x_m), \tag{6}$$

where $A_1, \ldots, A_m$ *can be taken from the standard representation of* $f - \tilde{T}S \in J$ *with respect to its Gröbner basis* $\mathcal{G}_0$*.* $\tilde{T}$ *is determined uniquely and* $\tilde{T}$ *is the* normal form *of* T *with respect to* $\tilde{\mathcal{G}}$*. Moreover, if* f *is reduced with respect to* $\mathcal{G}_0$*,* $LM(\tilde{T}S) \succeq LM(f)$*.*

Here we call the representation (6) the **standard form** of f.

Definition 5 (Signature) *For each* $f \in I \setminus J$*, we call the coefficient* $\tilde{T}$ *in* (6) the reduced S-coefficient *of* f *and denote it by* $\mathrm{RSC}(f)$*. Moreover, we call its leading monomial* $LM(\mathrm{RSC}(f))$ the signature *of* f *and denote it by* $\mathrm{sig}(f)$*. For each* $f \in J$*, we set its signature as* 0*.*

Remark 9 *As another definition, the signature of* f *is defined by keeping its computational record, and sometimes it differs from our definition. Thus, the signature by our definition may be called the* minimal *signature. However, by carefully handling S-polynomials as smallest as possible in the procedure, we keep that they coincide. Anyway, in order to minimize the number of S-polynomials, it is suited to use our definition.*

For each $f \in I \setminus J$, $\mathrm{supp}(\mathrm{RSC}(f)) \subset NS(\mathrm{Syz})$ by the definition of $\mathrm{RSC}(f)$. As $NS(\mathrm{Syz}) \subset \mathcal{M}_{red}$, we obtain

$$\deg(\mathrm{RSC}(f)) \le m(d-1) = md - m.$$

Therefore, for each $g \in \mathcal{G}$, as it is reduced with respect to $\mathcal{G}_0$, if $g \in J$, then $g = F(x_i)$ for some i, and otherwise, $LM(\mathrm{RSC}(g)S) \succeq LM(g)$ by Lemma 2. Thus, in this case, we have

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

Now we set

$$Reg = md + d_S - m$$

as an important number for our analysis which is related to certain *regularity coming from Hilbert polynomial*. We will discuss it in Section 5. As a direct consequence, we have the following which can be easily extended to the *homogenized* ideal. (See Section 5.2 and Proposition 12.)

Lemma 3 *For each element g of the reduced Gröbner basis $\mathcal{G}$ of I, its total degree does not exceed Reg.*

Now we estimate $\#NS(\mathrm{Syz})$. By the well-known decomposition formula, we have

$$\sqrt{J} = \sqrt{J + \langle S \rangle} \cap \sqrt{J : S}.$$

(See chapters related to primary decomposition in [15, 40].) As J is radical, we also have

$$J = (J + \langle S \rangle) \cap (J : S) = I \cap (J : S).$$

From this, the following exact sequence can be deduced:

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

(See Exercise 5.3.3 in [15].)

Here we recall that for each 0-dimensional ideal L of R, the linear dimension of the residue class ring R/L coincides with the number of monomials in $\mathcal{M} \setminus LM(L)$. Also, if L is radical, the linear dimension coincides with the number of its zeros. Using this fact, we have

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

Proposition 4 *$\#NS(\mathrm{Syz}) = \#V(J) - \#V(I) = d^m - \#V(I)$. If $\#V(I)$ is very small compared to $\#V(J) = d^m$, then $\#NS(\mathrm{Syz}) \approx d^m$.*

Proof. It suffices to show $I + (J : S) = R$. Because $I + (J : S) = R$ implies $\#V(I + (J : S)) = 0$. When $I = R$, then $I + (J : S) = R$. Thus, we consider the case $I \neq R$.

Let f be an arbitrary element of R. By *interpolation technique*, see Section 5.2, we can show that there is an element h such that $h(\alpha) = \frac{f(\alpha)}{S(\alpha)}$ for $\alpha \in V(J) \setminus V(I)$. Then, $(f - hS)(\alpha) = 0$ for $\alpha \in V(J) \setminus V(I)$. As $S(\alpha) = 0$ for $\alpha \in V(I)$, it follows that $S(f - hS)$ vanishes on all α in $V(J)$. By Hilbert's Nullstellensatz, as J is radical, $S(f - hS)$ belongs to J and $f - hS$ belongs to $(J : S)$.

On the other hand, hS belongs to $I = J + \langle S \rangle$. Then f can be expressed as

$$f = (f - hS) + hS,$$

where $f - hS \in (J : S)$ and $hS \in I$. This implies that f belongs to $I + (J : S)$ and $R = I + (J : S)$. □

3.3 The Number of Monomials in #RSC(g) for g in $\mathcal{G}$

We discuss on the number of monomials in RSC(g) for each g in $\mathcal{G}$. We set $V(J) \setminus V(I) = \{\alpha_1, \ldots, \alpha_N\}$, where $N = \#(V(J) \setminus V(I)) = d^m - \#V(I)$. By Proposition 4, $\#NS(\mathrm{Syz}) = \#V(J) - \#V(I) = N$. Thus we set $NS(\mathrm{Syz}) = \{t_1, \ldots, t_N\}$. If I is trivial, then $NS(\mathrm{Syz}) = \mathcal{M}_{red}$, $\mathrm{Syz} = (J : S) = J$ and $\mathcal{G}_0 = \tilde{\mathcal{G}}_0$.

For each $g \in \mathcal{G} \setminus J$, we consider its standard form

$$g = \mathrm{RSC}(g)S + \sum_{i=1}^{m} A_i^{(g)} F(x_i),$$

where $\deg(A_i^{(g)} F(x_i)) \leq \deg(\mathrm{RSC}(g)S) \leq Reg$ and RSC(g) is reduced with respect to $\tilde{\mathcal{G}}_0$.

Let $\mathrm{RSC}(g) = \sum_{i=1}^{N} c_i t_i$. Considering c_i as an indeterminate for each t_i, we have a system of linear equations derived from the following:

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

Letting M be the $N \times N$ matrix whose i-th row is $(t_i(\alpha^{(1)}), \ldots, t_i(\alpha^{(N)}))$, we have

$$\left(\frac{g(\alpha^{(1)})}{S(\alpha^{(1)})}, \ldots, \frac{g(\alpha^{(N)})}{S(\alpha^{(N)})} \right) = (c_1, c_2, \ldots, c_N)M. \tag{7}$$

Lemma 5 *M is an invertible matrix.*

Proof. We show that the system (7) has a unique solution, form which M is proved to be invertible.

Let its arbitrary solution be $\mathbf{c}' = (c'_1, \ldots, c'_N)$, and set $T_{\mathbf{c}'} = \sum_{i=1}^{N} c'_i t_i$. Then, as $g(\alpha) = T_{\mathbf{c}'} S(\alpha) = 0$ for all $\alpha \in V(I)$, we have $g(\alpha) = T_{\mathbf{c}'} S(\alpha)$ for all $\alpha \in V(J)$. Thus, it follows that $g - T_{\mathbf{c}'} S$ belongs to J and so $\mathrm{RSC(g)}S - T_{\mathbf{c}'} S$ also belongs to J by Hilbert's Nullstellensatz. In the end, we have

$$\mathrm{RSC}(g) - T_{\mathbf{c}'} \in (J : S) = \mathrm{Syz},$$

Since both RSC(g) and $T_{\mathbf{c}'}$ are reduced with respect to the Gröbner basis $\tilde{\mathcal{G}}_0$ of $(J : S)$, we have $\mathrm{RSC}(g) = T_{\mathbf{c}'}$ and the system has a unique solution. □

By Lemma 5 we have

$$\left(\frac{g(\alpha^{(1)})}{S(\alpha^{(1)})},\ldots,\frac{g(\alpha^{(N)})}{S(\alpha^{(N)})}\right)M^{-1}=(c_1,c_2,\ldots,c_N), \tag{8}$$

and each c_i is expressed as the inner products of vectors;

$$c_i=\left(\frac{g(\alpha^{(1)})}{S(\alpha^{(1)})},\ldots,\frac{g(\alpha^{(N)})}{S(\alpha^{(N)})}\right)\cdot {}^t\mathbf{m}_i,$$

where $\mathbf{m}_i$ denotes the the i-th column of M^{-1}. Then, it is expected *with high probability* that the ratio of zero-products, that is, $c_i=0$, is approximately $\frac{1}{q}$. Thus, *roughly in average*, we may expect

$$\begin{aligned}\#\mathrm{supp}(\mathrm{RSC}(g)) &\approx \#NS(\mathrm{Syz})\left(1-\frac{1}{q}\right)=(d^m-\#V(I))\left(1-\frac{1}{q}\right)\\ &\approx (d^m-\#V(I)).\end{aligned} \tag{9}$$

When the property (9) holds for g we call it the **genericness of non-zero coefficients** of g.

Moreover, in a similar manner, we can extend our argument for any S-coefficient T of g not necessary reduced with respect to $\tilde{\mathcal{G}}_0$. In this case, we also expect the following for T:

$$\#\mathrm{supp}(T)=\#\{t\in\mathcal{M}\mid c_t\neq 0\}\gtrapprox (d^m-\#V(I))\left(\frac{q-1}{q}\right)\approx d^m-\#V(I). \tag{10}$$

We call this property (10) the **extended genericness of non-zero coefficients** of g.

Assumption on Genericness of Non-Zero Coefficients: Now we return to our original problem, where $S=S_{m+1}(x_1,\ldots,x_m,x(aP+bQ))$. So, the value $x(aP+bQ)$ ranges in almost a half of $\mathbb{F}_q$. Let

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

When $I_m(a,b)$ is trivial, the choice of g is unique, that is, $g=1$, and $NS(\mathrm{Syz})=\mathcal{M}_{red}$ and $N=d^m$. When $I_m(a,b)$ is non-trivial, $NS(\mathrm{Syz})$ differs corresponding to $V(I)$. So, in this case, we consider to embed $\mathbf{GS}_{m+1}(a,b)$ in a vector of size d^m whose components correspond to monomials in M_{red}.

Assumption 1. The distribution $\{\mathbf{GS}_{m+1}(a,b)\mid a,b\in\mathbb{F}_q\}$ *coincides with* that of randomly chosen vectors. And, for almost every a,b in $\mathbb{F}_q$, the reduced Gröbner basis $\mathcal{G}$ of the ideal $I_m(a,b)$ has some element g for which the genericness of non-zero coefficients holds.

When $I_m(a,b)$ is trivial, $\mathcal{G}_0 = \mathcal{G}_0 = \{F(x_1), \ldots, F(x_m)\}$. If $F(x)$ is binomial, then $\#\mathrm{supp}(T) \geq \#\mathrm{supp}(\mathrm{NF}_{\mathcal{G}_0}(T))$. This implies that under Assumption 1, the extended genericness of non-zero coefficients also holds. By our experiments, we expect that even for a non-binomial $F(x)$, and also, even for the non-trivial case, the extended genericness of none-zero coefficients holds.

Assumption 1a. For almost every a, b in $\mathbb{F}_q$, the reduced Gröbner basis $\mathcal{G}$ of the ideal $I_m(a,b)$ has some element g for which the extended genericness of non-zero coefficients holds.

3.4 The Number of S-polynomials

Here we extend the notion *regular PRS* in GCD computation to Gröbner basis computation in our case, by which we can estimate the number of S-polynomials computed during Gröbner basis computation. To do so, it is very useful to consider so-called *signature-based algorithms* (the F_5 algorithm and its variants) which can avoid unnecessary S-polynomials as many as possible. (See [10] for the original F_5 algorithm and also see the most recent survey [8].) Although it is not proven rigidly, such signature-based algorithms, along with the F_4 technique ([9]) for reduction step, are recognized as the fastest available algorithms nowadays.

Definition 6 ($\mathfrak{S}$-reduction) *Let $f, g, h \in I$. We say that f is $\mathfrak{S}$-reduced to g by h, if there are $t \in \mathcal{M}$ and $a \in K \setminus \{0\}$ such that $g = f - a \cdot t \cdot h$, $\mathrm{sig}(th) \prec \mathrm{sig}(f)$ and $LM(g) \prec LM(f)$. In this case, h (or $t \cdot h$) is called an* $\mathfrak{S}$-reducer *of f and signatures are stable through $\mathfrak{S}$-reduction as $\mathrm{sig}(f) = \mathrm{sig}(g)$. For $f \in I$ which has no $\mathfrak{S}$-reducer, we say that f is $\mathfrak{S}$-irreducible.*

Using $\mathfrak{S}$-reduction, we can show the following. (This is a translation of Proposition 2.13 in [39].)

Lemma 6 *For each $s \in NS(\mathrm{Syz})$, there is an element in I whose signature is s. Among such elements, there is an element, say f, which has the smallest leading monomial, say t. 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 set $\Phi(s) = \min_{\prec}\{\mathrm{LM}(f) \mid f \in I, \mathrm{sig}(f) = s\}$.

Next we give the definition of $\mathfrak{S}$-Gröbner basis for our case in a form suited for our discussion.

Definition 7 ($\mathfrak{S}$-Gröbner Basis) *A finite subset $\mathcal{H}$ is called an $\mathfrak{S}$-Gröbner basis of I if the following holds;*

(1) *$\mathcal{H}$ contains a Gröbner basis of J,*
(2) *for each $s \in NS(\mathrm{Syz})$, there exist $t \in \mathcal{M}$ and $g \in \mathcal{H}$ such that $t \times \mathrm{sig}(g) = s$ and tg is $\mathfrak{S}$-irreducible.*

Remark 10 (S-polynomial and F_5 criterion) *For each $s \in NS(\mathrm{Syz})$, if any pair (t, g) in $(\mathcal{M} \setminus \{1\}) \times \mathcal{H}$ does not satisfy the condition in (2) of Definition 7, an S-polynomial with signature s appears, from which an element of $\mathcal{H}$ can be computed. In more detail, there exist $g_1, g_2 \in \mathcal{H}, t_1, t_2 \in \mathcal{M}$ 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 $LM(t_1 g_1) = LM(t_2 g_2)$. Then, their S-polynomial is written as $\frac{1}{LC(g_1)} t_1 g_1 - \frac{1}{LC(g_2)} t_2 g_2$. By applying $\mathfrak{S}$-reduction to the S-polynomial, we obtain an $\mathfrak{S}$-irreducible element with signature s which is added to $\mathcal{H}$. Such a pair (g_1, g_2) is called a* normal pair*, and this procedure exactly corresponds to the F_5* criterion*.*

In actual computation of $\mathfrak{S}$-Gröbner basis, we deal with S-polynomials in ascending order of their signatures. In this way, polynomials which are $\mathfrak{S}$-irreducible are computed. By carefully dealing polynomials in ascending order of their signatures, we know their signatures correctly. (See [39, 38] for precise algorithms.)

Now we consider a signature for which some S-polynomial is computed during $\mathfrak{S}$-Gröbner basis computation. Suppose that we have computed a $\mathfrak{S}$-Gröbner basis $\mathcal{H}_s$ up to a signature s. As S is the unique element of $\mathcal{H}_s$ with signature 1, $\mathrm{sig}(sS) = s \times \mathrm{sig}(S) = s$ and thus $\{ug \mid u \in T \setminus \{1\}, g \in \mathcal{H}_s, u\mathrm{sig}(g) = s\} \neq \emptyset$. Take an element, say $u_1 g_1$, from $\{ug \mid u \in T \setminus \{1\}, g \in \mathcal{H}_s, u\mathrm{sig}(g) = s\}$ which has the smallest leading monomial. If s is unnecessary, then $u_1 g_1$ should be $\mathfrak{S}$-irreducible and $\Phi(s) = LM(u_1 g_2) = u_1 \Phi(g_2)$.

But, for $s = u_1 \mathrm{sig}(g_1)$ with $u_1 \succ 1$, it is highly expected that

$$\Phi(\mathrm{sig}(g_1)) \succ \Phi(s) \text{ and } LM(u_1 g_1) = u_1 LM(g_1) = u_1 \Phi(\mathrm{sig}(g_1)) \succ \Phi(s),$$

which implies that s is necessary. Because,

$$\begin{aligned} \Phi(s) &= \min_{\prec}\{LM(f) \mid f \in I, \mathrm{sig}(f) = s\} \\ &= \min_{\prec}\{LM(f) \mid f = \left(\sum_{\substack{t \in NS(\mathrm{Syz}) \\ t \preceq s}} c_t t \right) S + A_1 F(x_1) + \cdots + A_m F(x_m)\} \end{aligned}$$

and therefore, it seems that the larger s becomes, the smaller $\Phi(s)$ becomes *in general*. We will discuss about this behavior in the next subsection.

Definition 8 *We say that I is* regular *with respect to the signature, if the following holds:*

$$(*) \quad \Phi(s) \not| \Phi(s') \text{ holds for any distinct pair } s, s' \text{ in } NS(\mathrm{Syz}) \text{ with } s \mid s'.$$

For a subset A of $NS(\mathrm{Syz})$, if the condition $(*)$ holds for almost every distinct pair $s, s' \in A$, we say that I is A-semi-regular*. As an extremal case, we say that I is* strongly regular *with respect to the signature, if $\Phi(s) \succ \Phi(s')$ holds for any distinct signature s, s' with $s \prec s'$.*

If I is regular, for every $s \in NS(\mathrm{Syz})$, some S-polynomial with signature s appears. Thus, its $\mathfrak{S}$-Gröbner basis $\mathcal{H}$ computed by a signature-based algorithm should contain the following set;

$$\{h_1, \ldots, h_N \mid \mathrm{sig}(h_i) = s_i, h_i \text{ is } \mathfrak{S}\text{-irreducible}\} \cup \{F(x_1), \ldots, F(x_m)\},$$

where $NS(\mathrm{Syz}) = \{s_1, \ldots, s_N\}$. Thus, at least, $(N-1)$ S-polynomials are computed.

To give more precise estimation for the case $d > \deg_{x_i}(S)$, we have to consider the effect of S whose signature is 1. Because, for smaller t, tS tends to break the condition $(*)$ in Definition 8. Actually, unnecessary signatures for S-polynomials are detected at the *beginning* stage of Gröbner basis computation. Here we assume that S is *symmetric* in variables, $d_S = m\delta$, where $\delta = \deg_{x_1}(S) = \cdots = \deg_{x_m}(S)$. This condition holds for almost every $S_{m+1}(x_1, \ldots, x_m, x(aP + bQ))$. Moreover we assume that S is not a *sparce* polynomial.

Lemma 7 *We assume that $d > \delta$. Each $t = x_1^{e_1} \cdots x_m^{e_m}$ $(0 \leq e_i < d - \delta)$ is a* useless *signature for $\mathfrak{S}$-Gröbner basis computation, that is, no S-polynomials with signature t appears.*

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.

Suppose that tS is not $\mathfrak{S}$-irreducible. Then, there is a polynomial T such that $LM(T) \prec t$ and the normal form of TS with respect to $\mathcal{G}_0$ has its leading monomial $LM(tS)$. Since $LM(TS) \prec LM(tS)$, it can be shown that $\deg_{x_i}(TS) < e_i + \delta$ for some i. But, in this case, any $F(x_i)$ cannot reduce TS and so $LM(TS)$ cannot coincides with $LM(tS)$. This is a contradiction and thus, tS is $\mathfrak{S}$-irreducible. □

Now we set

$$\overline{NS(\mathrm{Syz})} = NS(\mathrm{Syz}) \setminus \{x_1^{e_1} \cdots x_m^{e_m} \mid 0 \leq e_i < d - \delta\} \cup \{1\}.$$

Then $\#\overline{NS(\mathrm{Syz})} = d^m - \#V(I) - (d-\delta)^m + 1$. We note that it is very difficult to pick up all unnecessary signatures theoretically. Thus we define the set $\overline{NS(\mathrm{Syz})}$ as an easy approximation.

Assumption on Semi-Regularity: Now we return to our original problem. For $S = S_{m+1}(x_1, \ldots, x_m, x(aP + bQ))$, we may consider the value $x(aP + bQ)$ as a parameter z. Then, for distinct signatures s, s' such that $s = us'$ for $u \in \mathcal{M} \setminus \{1\}$, the condition $u\Phi(s') = \Phi(s)$ can be translated as some condition defined by a system of (semi) algebraic equations in z. (See Section 3.5.)

We note that we can consider the ideal generated by $S_{m+1}(x_1, \ldots, x_m, z)$ and $F(x_1), \ldots, F(x_m)$ over $\mathbb{Q}$. If the ideal is $\overline{NS(\mathrm{Syz})}$-semi-regular for some value z and modulo a prime p, then for almost every value of z in $\mathbb{Q}$ and a prime p, the ideal over $\mathbb{F}_p$ is $\overline{NS(\mathrm{Syz})}$-semi-regular.

Assumption 2: For almost every a, b in $\mathbb{F}_q$, $I_m(a, b)$ is $\overline{NS(\mathrm{Syz})}$-semi-regular.

Under Assumption 2, the number of elements of the computed $\mathfrak{S}$-Gröbner basis $\mathcal{H}$ has the same order as $\#\overline{NS(\mathrm{Syz})} \approx d^m - (d-\delta)^m + 1$. More precisely, there is some small constant $\varepsilon \ll 1$ such that

$$\#\mathcal{H} = \#\overline{NS(\mathrm{Syz})}^{1-\varepsilon} \approx (d^m - (d-\delta)^m + 1)^{1-\varepsilon}.$$

Since the number of S-polynomials is at least that of elements of the $\mathfrak{S}$-Gröbner basis, the number of S-polynomials is at least $(d^m - (d-\delta)^m + 1)^{1-\varepsilon}$.

We consider another algorithm for Gröbner basis computation not based on 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 ones available nowadays. This suggests that signature-based-algorithms can handle smaller number of S-polynomials compared with non-signature-based-algorithms.

Moreover, in our case, as our experimental results suggest, algorithms using two efficient techniques, *normal selection strategy* and *sugar degree*, compute S-polynomials in ascending order of total-degrees of their signature. This behavior is heavily related to the *semi-regularity* analyzed in Section 3.5. Because, in the normal strategy with sugar degree, at each step, a S-polynomial with smallest leading monomial is chosen. (See Page 116 in [5] for details.) Therefore, if Assumption 2 holds, this implies that such an S-polynomial has the smallest total degree of the signature. Thus, it is expected that the number of S-polynomials is at least $(d^m - (d-\delta)^m + 1)^{1-\varepsilon}$.

Assumption 2a: For almost every a, b in $\mathbb{F}_q$, any efficient algorithm for Gröbner basis of $I_m(a,b)$ computes at least $(d^m - (d-\delta)^m + 1)^{1-\varepsilon}$ S-polynomials.

Remark 11 *When d is larger than δ, the following inequality can be shown by induction argument on m;*

$$d^m - (d-\delta)^m + 1 \geq md^{m-1}. \tag{11}$$

In our experiment shown in Section 4.3, the number of S-polynomials which are not reduced to 0 is close to $2 \times md^{m-1}$. This might suggest $\#\mathcal{H} > c \times md^{m-1}$ for some constant $c > 1$.

3.5 Linear Algebra Related to Sub-Resultant Theory

Here we analyze the condition $(*)$ in Definition 8 by using linear algebraic methods related to subresultant theory. (See [6].) To make our argument simple and clear, we concentrate on Trivial Ideal Case, where $NS(\mathrm{Syz}) = \mathcal{M}_{red}$ and $\tilde{\mathcal{G}}_0 = \mathcal{G}_0$. We arrange all monomials in $\mathcal{M}_{red}$ in descending order. Thus, $\mathcal{M}_{red} = \{t_1, \ldots, t_N\}$, $N = \#\mathcal{M}_{red} = d^m$ and $t_i \succ t_j$ for $i < j$. (So, $t_N = 1$ and $t_1 = x_1^{d-1} \cdots x_m^{d-1}$.) As additional notations, for each polynomial f, we denote its normal form with respect to $\mathcal{G}_0$ by $\mathrm{NF}_{\mathcal{G}_0}(f)$. Also, for each polynomial f reduced with respect to $\mathcal{G}_0$, we define its corresponding vector as follows and denote it by $[f]$;

$$f = \sum_{i=1}^{N} a_i t_i \mapsto [f] = [a_1, \ldots, a_N].$$

Moreover, for a positive integer $k \leq N$, we write $[f]_k = [a_1, \ldots, a_k]$ for the vector consisting of the first k components.

For each signature s in $\mathcal{M}_{red}$, we consider an element f in I with signature s which is reduced with respect to $\mathcal{G}_0$, and its standard form;

$$f = TS + A_1F(x_1) + \cdots + A_mF(x_m), \tag{12}$$

where $\mathrm{supp}(T) \subset \mathcal{M}_{red}$ and $LM(T) = s$. Thus, for $t_L = s$, T is written as

$$T = \sum_{i=L}^{N} c_i t_i,$$

where c_i is the coefficient of t_i in T. As f is reduced with respect to $\mathcal{G}_0$, it follows that

$$\mathrm{NF}_{\mathcal{G}_0}(TS) = f.$$

Now we consider vectors $[\mathrm{NF}(t_LS)], \ldots, [\mathrm{NF}(t_NS)]$ and let M_s be an $(N - L+1) \times N$ matrix whose i-th row is $[\mathrm{NF}(t_{L+i-1}S)]$, and $(M_s)_k$ the matrix whose i-th row is $[\mathrm{NF}(t_{L+i-1}S)]_k$. Then, for the coefficient vector $\mathbf{c} = (c_L, \ldots, c_N)$, we have

$$\mathbf{c}M_s = [NF(TS)], \quad \mathbf{c}(M_s)_k = [NF(TS)]_k.$$

As a typical matrix, let $\hat{M}_s = (M_s)_{N-L+1}$ which is a square matrix.

Let $t_Q = \Phi(s)$. Then, there is a vector $\hat{\mathbf{c}}$ such that the first $N-Q$ components of $\hat{\mathbf{c}}M_s$ is 0 and the $N-Q+1$ component, which corresponding to t_Q, is non-zero. This can be translated as

$$\hat{\mathbf{c}}_{N-Q}(M_s)_{N-Q} = (0, \ldots, 0), \quad \hat{\mathbf{c}}_{N-Q+1}(M_s)_{N-Q+1} = (0, \ldots, 1).$$

So, determining $\Phi(s)$ is reduced to solving some systems of linear equations.

In our original case, each M_s is a matrix with a parameter z, and $\det(\hat{M}_s)$ is a polynomial in z. So, if $\det(\hat{M}_s) \neq 0$ for some value $x(aP + bQ)$, $\deg(\hat{M}_s)$ is a non-trivial polynomial in z over K. We give two typical cases:

1. If $\hat{M}_s$ and $\hat{M}_{s'}$ are regular (invertible), where $s' = t_{L+1}$ (the previous element of s), it follows that any non-zero vector of size $N - L + 1$ whose last component is zero cannot make the first $N - L$ components zero but some vector with non-zero last component can make them zero Thus, in this case, we have $\Phi(s) = t_{N-L+1}$. Moreover, if $\hat{M}_s$ is regular for every $s \in NS(\mathrm{Syz})$, then I is strongly regular. This behavior was seen in the case $\delta = d - 1$ in our experiment.

2. For each D $(1 \leq D \leq m\delta)$, let t_D be the largest element among signatures with total degree D, that is, $t_D = \max_{\prec}\{t \in NS(\mathrm{Syz}) \mid \deg(t) = D\}$. If the matrix $\hat{M}_D$ is regular, we can show that for any signature $s \preceq t_D$, $\deg(\Phi(s)) \geq m\delta - D$. Thus, if $\hat{M}_D$ is regular for every D, it is expected with high possibility that $\deg(s) < \deg(s')$ implies $\deg(\Phi(s)) > \deg(\Phi(s'))$. This corresponds to Assumption 2.

4 Complexity of Gröbner Basis Computation and ICM

Here we give *lower bounds* on complexity of computation of Gröbner basis for an ideal of special type which satisfies special properties discussed in Section 3. Then, we give those for the naive ICM for the ECDLP. We consider an efficient algorithm for computing a Gröbner basis of I, and let $\mathcal{G}^*$ be the computed Gröbner basis from which the reduced one $\mathcal{G}$ is obtained.

4.1 Lower Bound Based on the Number of S-polynomials

We assume that the ideal I is $\overline{NS(\mathrm{Syz})}$-semi-regular. Then the number of S-polynomials which produce new elements added to $\mathcal{G}^*$ is at least $(d^m - (d-\delta)^m + 1)^{1-\varepsilon}$. (From the setting of our original ideal $I_m(a,b)$, we may assume that the degree d is much larger than δ, as S has more than δ^m monomials.) By Remark 11, we may use the bound (11);

$$d^m - (d - \delta^m) + 1 \geq md^{m-1}.$$

Thus, in this case, it requires at least $(md^{m-1})^{1-\varepsilon}$ S-polynomials and so the computational cost exceeds $(md^{m-1})^{1-\varepsilon}$ arithmetic operations.

Proposition 8 *Under Assumption 2a, for computing a Gröbner basis of $I_m(a,b)$, it requires at least $(md^{m-1})^{1-\varepsilon}$ field arithmetic operations for $\varepsilon \ll 1$.*

Case where Trivial Ideals mainly handled: In our setting in Section 2, we may assume $d \approx t$ and the cost $\mathcal{C}_{\mathrm{dec}}$ requires at least (approximately) mt^{m-1}. Then, as long as the ratio of *failure* is not ignorable, that is, larger than some constant, it follows that in the total estimation

$$O\left(\frac{m! \times q \times \mathcal{C}_{\mathrm{dec}}}{t^{m-1}} + t^{\omega}\right),$$

the first term exceeds $m!q$ under Assumption 2a. (More precisely, it exceeds $m!q^{1-\varepsilon}$.) We remark that, although the estimation is given by *big O-notation*, it can be considered as the average estimation. Hence, in this setting, we conclude that the naive ICM cannot be more efficient than the brute force method. Here we remark that by the brute force method, we mean not to search zeros of $S_{m+1}(x_1, \ldots, x_m, x(aP+bQ))$ but to search m points in $\mathcal{B}$ satisfying (1) directly.

Case where Non-Trivial Ideals mainly handled: In this case, we assume that $t \approx d$ and $t^m \gtrapprox \gamma q$. Under Assumption 2a, it appears a term $m!q$ in the total estimation

$$O\left(\frac{m! \times t \times \mathcal{C}_{\mathrm{dec}}}{\gamma} + t^{\omega}\right).$$

Hence, we conclude the same as Trivial-Ideal Case, that is, the ICM cannot be more efficient than the brute force method.

4.2 Another Lower Bound Based on #supp(T)

Here we discuss another approach for possible estimation which does not depend on any algorithm for Gröbner basis. We recall that during the Gröbner basis computation, at each step (in any algorithm based on Buchberger's criterion or its extension like F_5 criterion), a new polynomial is generated by multiplying some monomial to one already computed polynomial and by reducing it with respect to the set of already computed polynomials. Therefore, by accumulating (recording) this arithmetic procedures, such a computed polynomial, say f, is expressed as

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

where $T^{(f)}_{actual}, A^{(f)}_{1,actual}, \ldots, A^{(f)}_{m,actual}$ are polynomials and they depend on *the actual computation (algorithm)*. Here we call $T^{(f)}_{actual}$ the actual coefficient of T for computing f.

At the end, for each g in $\mathcal{G}$, we have

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

Then, each monomial of $T^{(g)}_{actual}$ can be constructed by *monomial multiplication* or *monomial reduction* at each step. Thus, the number of monomials can represent a *lower bound* on how many such arithmetic operations have occurred during the whole computation.

In more detail, at some step in the middle of computation, from the computed set $\mathcal{G}'$ for the Gröbner basis, a pair (g_1, g_2) is chosen and a new element g_3 is generated by the reduction of the S-polynomial $u_1 g_1 - u_2 g_2$ by $\mathcal{G}'$, where $u_i = \frac{lcm(LM(g_1), LM(g_2))}{LC(g_i) LM(g_i)}$. Then, we have

$$\begin{aligned} g_3 &= u_1 g_1 - u_2 g_2 - \sum_i \sum_t c_{t,i} t g_i \\ &= (u_1 T^{(g_1)}_{actual} - u_2 T^{(g_2)}_{actual} - \sum_i \sum_t c_{t,i} t T^{(g_i)}_{actual}) S + \cdots , \end{aligned} \tag{13}$$

where g_i belongs to $\mathcal{G}'$ and $c_{t,i}$'s are coefficients. This means that every monomial in the coefficient of T comes from some multiplication of a monomial and an already computed polynomial occurring this procedure. So, for each newly appearing monomial in the T-coefficient, there must occur one such multiplication. Thus, we may guess the following.

Conjecture: For producing g during the Gröbner basis computation for I, it requires at least $\#\mathrm{supp}(T^{(g)}_{actual})$ times of multiplications of one monomial and some polynomials appearing in the computation.

In our experiment, the number of such multiplication became much larger than $\#\mathrm{supp}(T^{(g)}_{actual})$. To give a theoretical proof for Conjecture should be our next work.

Case where Trivial Ideals mainly handled: In this case, the reduced Gröbner basis is $\{1\}$. Then, we consider the representation of 1;

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

Under Assumption 1a, we have

$$\#\mathrm{supp}(T^{(1)}_{actual}) \gtrapprox d^m,$$

and so $\mathcal{C}_{dec} \gtrapprox d^m$, which is larger than the estimation in Section 4.1. Hence, under Conjecture we conclude that the ICM here is worse than the brute force method.

Case where Non-Trivial Ideals mainly handled: Under Assumption 1a, there is an element g in $\mathcal{G}\backslash J$ such that the extended genericness of non-zero coefficients of $T^{(g)}_{actual}$ holds. Then, for such an element g, we have $\#\mathrm{supp}(T^{(g)}_{actual}) \gtrapprox d^m$ and thus,

$$\#\cup_{h\in\mathcal{G}}\,\mathrm{supp}(T(h)) \geq \#\mathrm{supp}(T^{(g)}_{actual}) \gtrapprox d^m.$$

Hence, we conclude that the ICM here is worse than the brute force method under Conjecture.

4.3 Experimental Data

Here we show experimental data which support our assumptions. For $m = 3$ or 4, we generated primes $p = 2^B + \alpha$ with very small α, and picked up integers k which divide $p - 1$. We also set $F(x) = x^k - 1$ for the binomial case, and $F(x) = x^{k-1} + \cdots + 1$ for the non-binomial case. (Thus, the degree d of $F(x)$ is either k or $k - 1$. and all zeors of $V(J)$ are rational over $\mathbb{F}_p$.) Next 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 reduced Gröbner bases of ideals $I_m(a, b)$ by a computer algebra system *Risa/Asir*. Its function `nd_gr` with *options* `gentrace=1` and `gensyz=1` records all history telling how elements of the computed Gröbner basis were constructed. As each of them comes from some S-polynomial, the number of them implies the number of *necessary* S-polynomails which are reduced to non-zero polynomials. As results, our assumptions (Assumption 1a, and Assumption 2a) seem to hold for all examples, even for cases d^m is much smaller than p. Moreover, from our exeprriment, it is also observed that the number of multiplications of monomials and polynomials occured in *reduction of S-polynomial* (see (13)) became very huge compared with $\#NS(\mathrm{Syz})$.

Remark 12 *We note the function* `nd_gr` *does use neither F_4 nor F_5, but it uses the* normal selection strategy *and* sugar degree*. Thus, although it may compute unnecessary S-polynomials, its computational behavior on selection of S-polynomials becomes close to signature-based algorithms.*

The Number of S-polynomials: Here we show our data for counting the number of S-polynomials. We denote by $\mathcal{G}^*$ the computed (non-reduced) Gröbner basis, from which the reduced Gröbner basis $\mathcal{G}$ is computed. All elements in $\mathcal{G}^*$ are computed from some *necessary* S-polynomials. Of course, there might be other *unnecessary* S-polynomials which are reduced to 0. But, to estimate some lower bound on the number of S-polynomials, we have to exclude those, and we count the number of such elements. In our experiment, we used $F(x) = x^d - 1$ and computed 5 examples for each parameter (B, d). In tables below, the symbol $[\#\mathcal{G}]$ means the average of $\#\mathcal{G} - \#\mathcal{G}_0$ and that *Ratio* means the ratio $\frac{[\#\mathcal{G}]}{\#NS(\mathrm{Syz})}$ or $\frac{[\#\mathcal{G}]}{[\#\overline{NS(\mathrm{Syz})}]}$. The symbol $[PA]$ means the average of the number of multiplications of monomials and polynomials occured in *reduction of S-polynomial*. In Non-Trivial Ideal Case, symbols $[\#NS(\mathrm{Syz})]$ and $[\#\overline{NS(\mathrm{Syz})}]$ mean their averages.

(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 huge memories.

In Trivial Ideal Case, from our experiment, it is observed that $\#NS(\mathrm{Syz}) = d^m$ coincides with $\#\mathcal{G}^*$ for $d \leq \delta = 8$ and $\#NS(\mathrm{Syz})$ is very close to $\#\mathcal{G}^*$ for $d \geq \delta = 8$. Thus, all ideals in Trivial Ideal Case are considered to be strongly regular or $NS(\mathrm{Syz})$-*semi-regular*.

In Non-Trivial Ideal Case, it is also observed that $\#V(I)$ is very small and $\#NS(\mathrm{Syz})(= d^m - \#V(I))$ is very close to $\#\mathcal{G}^*$. Thus, all ideals in Non-Trivial Case are considered to be $NS(\mathrm{Syz})$-semi-regular.

Trivial-Ideal Case

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
$[PA]$	122162	1175808	2902403	2902748	26852120.2	46097906.4

Non-Trivial-Ideal Case

B	12	13	15
d	5	6	7
d^m	625	1296	2401
$[\#NS]$	613.6	1287.8	2382.8
$[\#\overline{NS}]$	613.6	1287.8	2382.8
$[\#\mathcal{G}^*]$	613.6	1287.8	2382.8
Ratio	1	1	1
$[\#V(I)]$	11.4	8.2	18.2
$[PA]$	121820	408965	1173939

(II) For $m = 3$, we chose examples where d is very larger than $\delta = 4$. Here the symbol *Ration*1 means the ratio $\frac{[\#\mathcal{G}]}{dm^{m-1}}$. From our experiment, it is observed

that $\#\mathcal{G}^* \approx 5.5 \times \frac{N_1}{\log(N_1)}$, where $N_1 = \#\overline{NS(\mathrm{Syz})}$, and also $\#\mathcal{G}^* \approx 2.1 \times md^{m-1}$. This suggests that Assumption 2a holds for our examples.

Trivial-Ideal Case

d	12	20	30
$\#NS(=d^m)$	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
$[PA]$	147488	862113.6	3472845.4

Non-Trivial-Ideal Case

d	12	20	30
$[\#NS]$	1722.6	7992.8	26995.2
$[\#\overline{NS}]$	1211.6	3897.8	9420.8
md^{m-1}	432	1200	2700
$[\#\mathcal{G}^*]$	941.6	2591.8	5739.8
$Ratio$	0.78	0.66	0.61
$Ratio1$	2.17	2.16	2.13
$[\#V(I)]$	5.4	7.2	4.2
$[PA]$	147417.8	861979	3472685.4

The Number of Monomials: Our experiment shows that the extended genericness of non-zero coefficients holds for our examples. Also the distribution of $\mathbf{GS}_{m+1}(a, b)$ seems very *close* to that of randomly generated vectors. We made a preliminary experiment on the the distribution in a stastical view point for parameters $m = 3, 4$ and $B = 10, 15, 20, 25$ to examine that it is very close to the *uniform* distribution.

The details on our experiment is shown below, where for each parameters (m, b, d) we computed N samples.

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

$(*)$ For $m = 4$ and $B = 20$ in Trivial Case we used the total degree ordering (not the reverse lexicographical ordering) and computed up to $d = 7$.

Note on Trivial Ideal Case: For all examples, $\#\mathrm{supp}(T^{(1)}_{red})$ coincides with d^m. Also, $\#\mathrm{supp}(T^{(1)}_{actual})$ is close to $(m+1) \times d^m$ for Binomial Case.

Note on Non-Trivial Ideal Case: For every g in $\mathcal{G}$ except $g = F(x_i)$ 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. Also, for 53 examples among 60 ones, we have

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

and the largest gap between those values is 4, which appears at one example with $d = 11$ and $b = 10$. Moreover, in our examples, the largest gap between $\#\mathrm{supp}((T^{(g)}_{actual})_{red})$ and d^m is 18 which appears at one example with $d = 8$ and $b = 10$. In some examples, some $F(x_i)$ still remains in $\mathcal{G}$.

5 Further Discussion on Degree Bound and Degree Fall

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

5.1 Homogenization and Degree Fall

In general, it is hard to give a precise estimation on the complexity of Gröbner basis computation. However, for homogeneous ideals with a *graded* ordering such as a reverse lexicographical ordering, we can use properties of their graded structure such as Hilbert polynomials and its related regularities. (For definitions, see Chapter 2 and Chapter 9 in [5] or Chapter 5 in [20].) For analyzing the complexity of Gröbner basis computation for such ideals, the degree bound on elements of Gröbner basis is the most important. 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, its computational cost can be reduced to that of its *homogenized* ideal.

We begin by recalling some useful properties related to homogenization technique. (See [20] for details and proofs of propositions.) Then we define our *fall degree* which is much different from existing ones but does not depend on any algorithm for Gröbner basis computation. Our fall degree is defined for each element of the given ideal and, if the ideal has a smaller Gröbner basis, its *maximal* fall degree is expected to be close to the regularity.

Now we give a brief explanation for homogenization technique, where we introduce a new variable y and consider a ring $K[x_1, \ldots, x_n, y]$. For a polynomial $f(x_1, \ldots, x_n) \in K[x_1, \ldots, x_n]$, its homogenization, denoted by f^h, is defined as $f^h(x_1, \ldots, x_n, y) = y^{\deg(f)} f(x_1/y, \ldots, x_n/y)$.

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

Proposition 9 *Let $\tilde{\mathcal{H}}$ be a 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 9 means that, for a non-homogeneous ideal, its total cost of Gröbner basis computation can be reduced to that of its homogenized ideal.

Proposition 10 *Let L^h be the homogenization of L as an ideal. Then, there is a positive integer ℓ such that*

$$L^h = \langle \mathcal{H}^h \rangle : y^\infty = \langle \mathcal{H}^h \rangle : y^\ell. \tag{14}$$

Therefore, for each element f of L, there exists a non-negative integer $u \leq \ell$ such that $f^h y^u$ belongs to $\langle \mathcal{H}^h \rangle$. (For saturation, see Chapter 4 in [5].)

Now we give the definition of *fall degree* in our setting.

Definition 9 *For each f in L, the smallest positive integer u such that $f^h y^u$ belongs to $\langle \mathcal{H}^h \rangle$ is called the* fall degree *of f. If $u > 0$, we say that there occurs a degree-fall at f. Moreover, the smallest positive integer ℓ satisfying the formula* (14) *can be considered as the* maximal fall degree *of Gröbner basis computation, as there is an element in the reduced Gröbner basis with fall degree ℓ.*

We remark about efficient techniques, *normal selection strategy* and *sugar degree*, again. In Section 3.4, we mentioned that those techniques may make the computational behavior close to those of signature-based algorithms. Also, at the same time, these techniques make the computational behavior close to that of the homogenized ideal. For each f in the ideal, the fall-degree indicates the gap between the actual degree of f and the total degree of its image in the homogenized ideal. (See also Chapter 2 in [5].)

5.2 On Degree Bound

In Section 3, 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.

We begin by recalling the well-known upper bound based on *regularity*. By the word *regularity*, we may mean either the Hilbert regularity or the Castelnuovo-Mumford one. (See [17] for details and some extension of the following.)

Proposition 11 ([23]) *Consider an ideal L generated by a finite set $\{f_1, \ldots, f_r\}$ in $K[x_1, \ldots, x_n]$, where $\deg(f_1) \geq \deg(f_2) \geq \cdots \geq \deg(f_r)$ and $r \geq n$, and its reduced Gröbner basis $\mathcal{G}_L$ with respect to a revlex ordering. If the projective dimension of the homogeneous ideal $\tilde{L} = \langle f_1^h, \ldots, f_r^h \rangle$ 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 $\deg(f_{n+1}) = 1$ if $r = n$.

In our case, it can be easily seen that $Reg = md + d_S - m$ coincides with the bound in Proposition 11.

On the other hand, 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 $\mathcal{F}^h$, with high possibility under Assumption 1. Moreover, for the ideal $\tilde{I}$ in Non-Trivial Ideal Case, we can show that $Reg - 1$ gives a more tight bound by very simple arguments.

Trivial Ideal Case: Suppose that I has no zero, that is, its reduced Gröbner basis is $\{1\}$. Then the reduced Gröbner basis $\tilde{\mathcal{G}}$ of $\tilde{I}$ contains y^e for some positive integer e. We consider the standard form of 1;

$$1 = TS + A_1F(x_1) + \cdots + A_mF(x_m),$$

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

$$y^e = \tilde{T}S^h + \tilde{A}_1F(x_1) + \cdots + \tilde{A}_mF(x_m),$$

where $\tilde{T}$ is reduced with respect to $F^h(x_1), \ldots, F^h(x_m)$. Then it can be shown directly that $\tilde{T}|_{y=1} = T$. *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

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

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

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

Under Assumption 1, 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 13 *By our experiments shown in Section 4.3, we have also examined that the total degree of T_{red} coincides with Reg for every examples 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 $\mathcal{G} \setminus J$, we consider its standard form;

$$g = \mathrm{RSC}(g)S + \sum_{i=1}^m A_i^{(g)}F(x_i),$$

where $\deg(A_i^{(g)}F(x_i)) \leq \deg(\mathrm{RSC}(g)S) \leq R$. Since $(J : S) \supsetneq J$, $\dim_K R/(J : S) > \dim_K R/J$ and thus, there is some element h in $\tilde{\mathcal{G}}_0$ 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

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

Proposition 12 *For each element g of the reduced Gröbner basis $\mathcal{G}$ of I, if I has a zero, its total degree does not exceed $md + d_S - m - 1$. The same holds for the homogeneous ideal $\tilde{I}$ generated by $\mathcal{F}^h$.*

5.3 On Degree Fall

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 $S, F(x_1), \ldots, F(x_m)$ and our *degree falls* occur. Moreover the maximal fall degree is very close to the last fall degree defined in [18].

In our simple analysis under statistical assumptions, for Trivial Ideal Case, where I has no zero, 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 $\langle \mathcal{F}^h \rangle$, and with high possibility, e coincides with the bound Reg under our assumption. This implies that for such a case, at the final step, there occurs the largest degree fall, and Reg shall be the *last fall degree* of $\mathcal{F}$ defined in [18]. (See Definition 10 below.) Because, 1 belongs to $V_{\max(e,\deg(1))} = V_e$ and so, for any element f in I with $\deg(f) \leq e$, f also belongs to V_e. This exactly supports the last fall degree assumption[18].

Definition 10 *For a finite subset $\mathcal{S}$ and the ideal I of a polynomial ring R generated by $\mathcal{S}$, the last fall degree is defined as the smallest integer c such that for all f in I, f belongs to $V_{\max(c,\deg(f))}$, where V_i is the smallest K-vector space satisfying the following;*
(1) all $f \in \mathcal{S}$ with $\deg(f) \leq i$ are included in V_i,
(2) for $g \in V_i$ and $h \in R$, if $\deg(gh) \leq i$, then gh belongs to V_i.

Remark 14 *Shapes of ideals appearing in improved ICMs using the Weil descent technique are different from ours, and analysis on fall degrees and that on degree bound become complicated and difficult. By the Weil descent technique, Semaev's summation polynomials are divided into n distinct polynomials which essentially give the same solutions of the PDP. (Here, the PDP is defined over $\mathbb{F}_{p^n}$.) As the effect of such* division*, the upper bound on the degree in Proposition 11 becomes smaller than that of ours. Also, for smaller binary fields, an interesting behavior was observed, where the first fall degree coincided with the* actual

regularity*, and it is called the* first fall degree assumption*. Some authors tried to estimate the complexity based on the assumption. (See [29, 35].) But, counter examples were also reported in [18].*

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

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

where $S_{m+1,1}, \ldots, S_{m+1,n}$ *are polynomials derived from the Semaev's summation polynomial* $S_{m+1}(x_1, \ldots, x_m, x(aP + bQ))$*. Then the actual fall degree shall be estimated by the total degree of* $T_1^{(g)}, \ldots, T_n^{(g)}$ *and a lower bound on the cost of the Gröbner basis computation shall be estimated by* $\#NS(\mathrm{Syz})$*, where* Syz *is some ideal in the* R*-module defined by syzygies among* $S_{m+1,1}, \ldots, S_{m+1,n}$*. We believe that our simple arguments may investigate deep insights on the subject.*

6 Concluding Remarks

In this paper, we brought new simple arguments for giving a lower bound on the complexity of Gröbner basis computation of the ideal derived from the PDP, the dominant part of the ICM. As the first step of obtaining a meaningful bound, we considered an ICM of *very naive form* as our target. To do it, we extracted certain essential properties of the ideals appearing in solving the 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 computation. As a result, under simple statistical assumptions on Semaev's summation polynomials, we succeeded in getting a lower bound on Gröbner basis computation. By our experiments, the validity of assumptions was examined. By the obtained bound we concluded that the complexity of the naive ICM cannot be $O(q)$, where it is defined over $\mathbb{F}_q$, and thus, the naive ICM cannot be more efficient than the brute force method. We remark that when the ideal has no zero, that is, the PDP fails, computation of the Gröbner basis means producing the unique element 1 at the final stage. In this case, there occurs a huge degree fall at the final step of the computation, which exactly corresponds to the *last fall degree* assumption.

As our next work, we will be trying to make more experiments on larger examples in order to examine our statistical assumptions and make our arguments valid for any algorithm for Gröbner basis computation. Also we will be also applying our simple arguments to several improvements on ICM, as discussed in Section 5.3. If the system of algebraic equations still has the same special shape, our arguments can be applied directly. Therefore, our next target shall be other methods (e.g. [14, 7, 35]) based on the Weil descent for the ECDLP over binary extension fields. For such methods, the corresponding ideals have different shapes, that is, they are generated by polynomials defining the x-coordinate of points in the factor base and number of multivariate polynomials. By applying our arguments to those complicated ideals, we are thinking that certain new insights on the behavior of Gröbner basis computation of ideals can be extracted.

References

1. Amadori, A., Pintore, F., Sala, M.: On the discrete logarithm problem for prime-field elliptic curves. Finite Fields and Their Applications 51, 168–182 (2018)
2. Bernstein, D.J., Engels, S., Lange, T., Niederhagen, R., Paar, C., Schwabe, P., Zimmermann, R.: Faster elliptic-curve discrete logarithms on FPGAs. IACR Cryptology ePrint Archive 2016/382 (2016)
3. Blake, I.F., Seroussi, G., Smart, N.: Elliptic Curves in Cryptography, vol. 265. Cambridge University Press (1999)
4. Bos, J.W., Kaihara, M.E., Kleinjung, T., Lenstra, A.K., Montgomery, P.L.: Solving a 112-bit prime elliptic curve discrete logarithm problem on game consoles using sloppy reduction. International Journal of Applied Cryptography 2(3), 212–228 (2012)
5. Cox, D., Little, J., OShea, D.: Ideals, Varieties, and Algorithms. Undergraduate Text in Mathematics, Springer, fourth edn. (2015)
6. Cox, D.A., Little, J., O'shea, D.: Using algebraic geometry, Graduate Text in Mathematics, vol. 185. Springer (1998)
7. Diem, C.: On the discrete logarithm problem in elliptic curves. Compositio Mathematica 147(01), 75–104 (2011)
8. Eder, C., Faugère, J.C.: A survey on signature-based algorithms for computing gröbner bases. Journal of Symbolic Computation 80, 719–784 (2017)
9. Faugère, J.C.: A new efficient algorithm for computing Gröbner bases (F4). Journal of Pure and Applied Algebra 139(1-3), 61–88 (1999)
10. Faugère, J.C.: 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)
11. Faugère, J.C., Huot, L., Joux, A., Renault, G., Vitse, V.: Symmetrized summation polynomials: using small order torsion points to speed up elliptic curve index calculus. In: Advances in Cryptology–EUROCRYPT 2014. Lecture Notes in Computer Science, vol. 8441, pp. 40–57. Springer (2014)
12. Faugère, J.C., Perret, L., Petit, C., Renault, G.: Improving the complexity of index calculus algorithms in elliptic curves over binary fields. In: Advances in Cryptology–EUROCRYPT 2012. Lecture Notes in Computer Science, vol. 7237, pp. 27–44. Springer (2012)
13. Galbraith, S.D., Gaudry, P.: Recent progress on the elliptic curve discrete logarithm problem. Designs, Codes and Cryptography 78(1), 51–72 (2016)
14. Gaudry, P.: Index calculus for abelian varieties of small dimension and the elliptic curve discrete logarithm problem. Journal of Symbolic Computation 44(12), 1690–1702 (2009)
15. Greuel, G.M., Pfister, G.: A Singular Introduction to Commutative Algebra. Springer, second edn. (2008)
16. Hankerson, D., Menezes, A.J., Vanstone, S.: Guide to Elliptic Curve Cryptography. Springer Science & Business Media (2006)
17. Hashemi, A., Seiler, W.M.: Dimension-dependent upper bounds for gröbner bases. In: Proceedings of the 2017 ACM on International Symposium on Symbolic and Algebraic Computation–ISSAC 2017. pp. 189–196. ACM (2017)
18. Huang, M.D.A., Kosters, M., Yeo, S.L.: Last fall degree, HFE, and Weil descent attacks on ECDLP. In: Advances in Cryptology–CRYPTO 2015. Lecture Notes in Computer Science, vol. 9215, pp. 581–600. Springer (2015)

19. Koblitz, N.: Elliptic curve cryptosystems. Mathematics of Computation 48(177), 203–209 (1987)
20. Kreuzer, M., Robbiano, L.: Computational Commutative Algebra 2. Springer (2005)
21. Kudo, M., Yokota, Y., Takahashi, Y., Yasuda, M.: Acceleration of index calculus for solving ECDLP over prime fields and its limitation. In: Cryptology and Network Security–CANS 2018. Lecture Notes in Computer Science, vol. 11124, pp. 377–393. Springer (2018)
22. Kusaka, T., Joichi, S., Ikuta, K., Khandaker, M.A.A., Nogami, Y., Uehara, S., Yamai, N., Duquesne, S.: Solving 114-bit ECDLP for a Barreto-Naehrig curve. In: Information Security and Cryptology–ICISC 2017. Lecture Notes in Computer Science, vol. 10779. Springer (2017)
23. Lazard, D.: Gröbner bases, Gaussian elimination and resolution of systems. In: European Conference on Computer Algebra–EUROCAL 1983. Lecture Notes in Computer Science, vol. 162. Springer
24. McGuire, G., Mueller, D.: A new index calculus algorithm for the elliptic curve discrete logarithm problem and summation polynomial evaluation. IACR Cryptology ePrint Archive 2017/1261 (2017)
25. Menezes, A.J., Okamoto, T., Vanstone, S.A.: Reducing elliptic curve logarithms to logarithms in a finite field. IEEE Transactions on Information Theory 39(5), 1639–1646 (1993)
26. Miller, V.S.: Use of elliptic curves in cryptography. In: Advances in Cryptology–CRYPTO 1985. Lecture Notes in Computer Science, vol. 218, pp. 417–426. Springer (1985)
27. Petit, C.: Bounding HFE with SRA. Preprint, http://www0.cs.ucl.ac.uk/staff/c.petit/files/SRA_GB.pdf (2013)
28. Petit, C., Kosters, M., Messeng, A.: Algebraic approaches for the elliptic curve discrete logarithm problem over prime fields. In: IACR International Workshop on Public Key Cryptography–PKC 2016. Lecture Notes in Computer Science, vol. 9615, pp. 3–18. Springer (2016)
29. Petit, C., Quisquater, J.J.: On polynomial systems arising from a Weil descent. In: Advances in Cryptology–ASIACRYPT 2012. Lecure Notes in Computer Science, vol. 7658, pp. 451–466. Springer (2012)
30. Pollard, J.M.: Monte Carlo methods for index computation (mod p). Mathematics of Computation 32(143), 918–924 (1978)
31. Rivest, R.L., Shamir, A., Adleman, L.: A method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM 21(2), 120–126 (1978)
32. Satoh, T., Araki, K.: Fermat quotients and the polynomial time discrete log algorithm for anomalous elliptic curves. Commentarii Mathematici Universitatis Sancti Pauli 47(1), 81–92 (1998)
33. Semaev, I.A.: Evaluation of discrete logarithms in a group of p-torsion points of an elliptic curve in characteristic p. Mathematics of Computation 67(221), 353–356 (1998)
34. Semaev, I.A.: Summation polynomials and the discrete logarithm problem on elliptic curves. IACR Cryptology ePrint Archive 2004/031 (2004)
35. Semaev, I.A.: New algorithm for the discrete logarithm problem on elliptic curves. IACR Cryptology eprint Archive 2015/310 (2015)
36. Shanks, D.: Class number, a theory of factorization, and genera. In: Proc. of Symp. Math. Soc., 1971. vol. 20, pp. 41–440 (1971)
37. Smart, N.P.: The discrete logarithm problem on elliptic curves of trace one. Journal of Cryptology 12(3), 193–196 (1999)

38. Vaccon, T., Verron, T., Yokoyama, K.: On affine tropical F5 algorithm. In: Proceedings of the 2018 International Symposium on Symbolic and Algebraic Computation–ISSAC 2018. pp. 383–390. ACM (2018)
39. Vaccon, T., Yokoyama, K.: A tropical F5 algorithm. In: Proceedings of the 2017 ACM on International Symposium on Symbolic and Algebraic Computation–ISSAC 2017. pp. 429–436. ACM (2017)
40. Vasconcelos, W.V.: Computational Methods in Commutative Algebra and Algebraic Geometry, Algorithms and Computation in Mathematics, vol. 2. Springer (1998)
41. Wenger, E., Wolfger, P.: Solving the discrete logarithm of a 113-bit Koblitz curve with an FPGA cluster. In: International Conference on Selected Areas in Cryptography–SAC 2014. Lecture Notes in Computer Science, vol. 8781, pp. 363–379. Springer (2014)