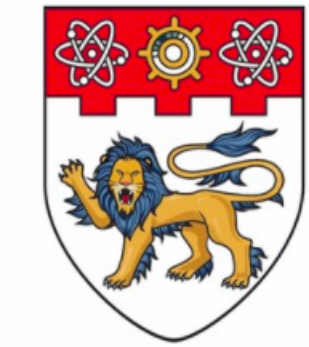

NANYANG TECHNOLOGICAL UNIVERSITY
SINGAPORE

ON ZERO-DIMENSIONAL POLYNOMIAL SYSTEMS AND THEIR APPLICATIONS TO THE ELLIPTIC CURVE DISCRETE LOGARITHM PROBLEM

YUN YANG

SCHOOL OF PHYSICAL & MATHEMATICAL SCIENCES

2017

ON ZERO-DIMENSIONAL POLYNOMIAL SYSTEMS AND THEIR APPLICATIONS TO THE ELLIPTIC CURVE DISCRETE LOGARITHM PROBLEM

YUN YANG

School of Physical & Mathematical Sciences

A thesis submitted to the Nanyang Technological University

in fulfillment of the requirement for the degree of

Doctor of Philosophy

2017

ACKNOWLEDGEMENTS

Firstly, I want to thank my supervisor Prof. Xing Chaoping for his guidance and support during my study at NTU. His personality and enthusiasm in research are worth learning. I am also grateful to my co-supervisor Dr. Yeo Szeling. We had a myriad of discussions in the past four years and I benefited a lot from them. In the period of my painful thesis writing, she gave me constant encouragement. She read my thesis so carefully and gave a lot of invaluable suggestions to revise it. It is impossible for me to finish this thesis without her help.

I am indebted to the collaborators in my first paper, Prof. Ming-Deh A. Huang, Dr. Michiel Kosters and Dr. Yeo Szeling for their help, support and encouragement. I learned a lot from this cooperation. I am also grateful to Dr. Christophe Petit and Li Liang for the discussions and meetings on Elliptic Curve Discrete Logarithm Problem.

I would like to thank Center for Strategic Infocomm Technologies(CSIT), which gave financial support to a project on Elliptic Curve Discrete Logarithm Problem for two years. I also thank Fan Junjie Bertrand, Tay Kian Boon and James Quah from CSIT and Bagus Santoso from A*STAR for the meetings and discussions in this project. I learned a lot about elliptic curve discrete logarithm from this project.

I would like to thank Prof. Mihai Putinar, Prof. Carles Padro, Prof. Wu Guohua, Dr.Ducoat Jerome and Dr. Markin Nadya for their lectures and seminars. Thank Mr. Soh Hwee Jin, Melvin from IT support office for helping me to deal with the problems of NTU server for computation.

Finally, I am thankful to my family members and friends for their support in all stages of my life.

CONTENTS

ABSTRACT

In this thesis, we first study the problem of solving zero-dimensional multivariate polynomial systems over finite fields and then study the elliptic curve discrete logarithm problem over binary fields.

First, we discuss a mostly theoretical framework for solving zero-dimensional polynomial systems. Complexity bounds are obtained for solving such systems using a new parameter, called the *last fall degree*, which does not depend on the choice of a monomial order. More generally, let k be a finite field with q^n elements and let k' be the subfield with q elements. Let $\mathcal{F} \subset k[X_0, \ldots, X_{m-1}]$ be a finite subset generating a zero-dimensional ideal. We give an upper bound of the last fall degree of the Weil descent system of $\mathcal{F}$ from k to k', which depends on q, m, the last fall degree of $\mathcal{F}$, the degree of $\mathcal{F}$ and the number of solutions of $\mathcal{F}$, but not on n.

Second, we introduce special vector spaces and use them in the index calculus method to solve ECDLP over binary fields. We provide heuristic complexity bounds for our approach and give conditions such that an efficient index calculus method will result. Finally, we provide some concrete examples of vector spaces with the nice properties.

1. INTRODUCTION

This thesis focuses on two major problems in algebraic geometry and cryptography. First, we study the problem of solving zero-dimensional polynomial systems. In particular, we propose a theoretical framework to help us study this problem in a more systematic manner. Second, we study some applications of solving zero-dimensional polynomials systems in cryptography, namely, cryptanalysis on the multi-HFE system as well as solving the elliptic curve discrete logarithm problem over finite fields.

1.1 Solving zero-dimensional multivariate polynomial system

Solving multivariate polynomial systems over finite fields is an important theoretical problem in Mathematics [45] and Computer Algebra [8]. It has found practical applications in various areas including error-correcting codes, robotics, cryptanalysis of ciphers, analysis of computer hardware and signal theory. The problem also forms the basis of multivariate cryptography such as in [37] that is considered as a strong candidate for post-quantum public key cryptography.

Let k be a field and let $\mathcal{F} \subset R = k[X_0, \ldots, X_{m-1}]$ be a finite subset which generates a zero-dimensional ideal I. By this we mean that $\dim_k(R/I) = e < \infty$. Suppose that we want to find the finitely many solutions of $\mathcal{F}$ in k^m (or in $\overline{k}^m$). We denote an algebraic closure of k by $\overline{k}$. In this thesis we will discuss a mostly theoretical framework for solving zero-dimensional polynomial systems.

One of the most common methods is the following. First fix a monomial order on R, such as the degree reverse lexicographic order, and then compute a Gröbner basis of the ideal

generated by $\mathcal{F}$ using for example F_4 or F_5 [15,16]. Then one computes a Gröbner basis for the lexicographic order using FGLM [18], and one uses this to find all the solutions. It is often very hard to estimate the complexity of such algorithms. The largest degree which one sees in such a computation of a Gröbner basis for the degree reverse lexicographic order is called the *degree of regularity*, and this degree essentially determines the complexity of such algorithms.

One approach to obtain heuristic complexity bounds on the degree of regularity is the use of the so-called *first fall degree assumption*. For $i \in \mathbb{Z}_{\geq 0}$, we let $V_{\mathcal{F},i}$ be the smallest k-vector subspace of $R_{\leq i}$ such that

(i) $\mathcal{F} \cap R_{\leq i} = \{f \in \mathcal{F} : \deg(f) \leq i\} \subseteq V_{\mathcal{F},i}$;

(ii) $hg \in V_{\mathcal{F},i}$, for all $g \in V_{\mathcal{F},i}$ and $h \in R$ with $\deg(hg) \leq i$.

The first fall degree is defined to be the first d such that $V_{\mathcal{F},d} \cap R_{\leq d-1} \neq V_{\mathcal{F},d-1}$ (and if it does not exist, it is defined to be 0; note that this definition of the first fall degree differs slightly from most definitions as in [39], but behaves a lot better). The heuristic claim is that the first fall degree is close to the degree of regularity for many systems (see for example [39]). A quote from [14] is "Our conclusions rely on no heuristic assumptions beyond the standard assumption that the Gröbner basis algorithms terminate at or shortly after the degree of regularity" (note that in [14] the definition of degree of regularity coincides with the first fall degree definition of [39]). It is quite often easy to give an upper bound on the first fall degree, just by counting arguments (see [14] for example). However, in [29], the authors raise doubt to the first fall degree heuristic.

In the first part of our method, section 5.1, we will try to rectify the situation. We will define the notion of *last fall degree*, which is the largest d such that $V_{\mathcal{F},d} \cap R_{\leq d-1} \neq V_{\mathcal{F},d-1}$. We denote the last fall degree of $\mathcal{F}$ by $d_{\mathcal{F}}$. We show how one can solve the system by computing $V_{\mathcal{F},\max\{d_{\mathcal{F}},e\}}$ and monovariate factoring algorithms (Proposition 5.1.12). We will also prove different properties of the last fall degree, for example, that the degree of regularity is bounded below by the last fall degree and above by the maximum of e and the last fall degree.

Furthermore, the last fall degree behaves well with respect to certain operations (such as linear change of variables and linear change of equations). It must be said that we do not know how to compute the last fall degree without having an upper bound, say coming from the degree of regularity. We will compare our approach with other approaches for solving systems, most notably with MutantXL and standard Gröbner basis algorithms (Subsection 5.1.4).

In the second part of our method, Section 5.2 and Section 5.3, we will give an application of our new framework around the last fall degree. Assume that k is a finite field of cardinality q^n with subfield k' of cardinality q. Let $\mathcal{F}'$ be the Weil descent system of $\mathcal{F}$ to k'. This is the system one obtains when one expresses all equations with the help of a basis of k/k'. This is a system in nm variables and hence seems to be much harder to solve than the original system. We give upper bounds on $d_{\mathcal{F}'}$ in terms of q, m, $d_{\mathcal{F}}$, the degree of $\mathcal{F}$ and the number of solutions of $\mathcal{F}$, but not on n. This generalizes practical and mathematical results, if $m = 1$ [5, 14, 20, 38]. This shows that some versions of multi-HFE (HFE stands for hidden field equations) are much easier to tackle than one would expect. Let us now give a precise formulation of the main theorem.

We denote by $Z(\mathcal{F})$ the set of zeros of $\mathcal{F}$ over $\overline{k}$. For $r \in \mathbb{R}_{\geq 0}$ and $c, t \in \mathbb{R}_{\geq 1}$ we set

$$\tau(r, c, t) = \lfloor 2t(c-1)\left(\log_c\left(\frac{r}{2t}+1\right)+1\right)\rfloor.$$

Note that this function increases when r increases.

Theorem 1.1.1. *Let k be a finite field of cardinality q^n. Let $\mathcal{F} \subset R = k[X_0, \ldots, X_{m-1}]$ be a finite subset. Let I be the ideal generated by $\mathcal{F}$. Assume that the following hold:*

- *I is zero-dimensional, say one has $|Z(\mathcal{F})| \leq s$;*
- *I is radical;*
- *there is a coordinate t such that the projection map $Z(\mathcal{F}) \to \overline{k}$ to coordinate t is injective;*

Let $\mathcal{F}'_f$ be the Weil descent system of $\mathcal{F}$ to the subfield k' of cardinality q using some basis of k/k', together with the field equations (Subsection 5.2.1). Then one has

$$d_{\mathcal{F}'_f} \leq \max\left(\tau(\max(d_{\mathcal{F}}, \deg(\mathcal{F}), (m+1)s, 1), q, m), m \cdot \tau(2s, q, 1), q\right).$$

When $m = 1$, we can obtain a slightly stronger version (Theorem 5.3.5).

In Section 5.4 we explain a version of the cryptographic protocol multi-HFE and we show how our results can be applied to show that this protocol is insecure.

In Section 5.5 we will explain why Theorem 1.1.1 is not useful to determine the complexity of solving systems coming from summation polynomials for the elliptic curve discrete logarithm problem, since such systems are not zero-dimensional.

1.2 Background of elliptic curve discrete logarithm problem

In around 1985, Miller and Koblitz suggested independently to use elliptic curves to replace finite fields to construct discrete logarithm problem. They had an intuition that the discrete logarithm problem on elliptic curves might be harder than that on finite fields. This gave rise to the birth of elliptic curve cryptography. Since then, the elliptic curve discrete logarithm problem has attracted wide concern of many experts and scholars. We abbreviate this problem to the initials ECDLP in this thesis. The hardness of ECDLP is the building block of the security of elliptic curve cryptography. Unlike the discrete logarithm problem on finite fields which can be solved by subexponential time methods, there is no known subexponential time algorithm to solve ECDLP for generic elliptic curves. There exists only exponential time algorithm to solve ECDLP for a random elliptic curve. It remains an open problem to propose a subexponential time algorithm to solve ECDLP for random elliptic curves. This problem has been a major research topic in computational number theory, cryptography and even in industry for the past few decades.

We briefly survey some results pertaining to the discrete logarithm problem on elliptic curves. A comprehensive exposition of this subject as well as other topics in elliptic curve cryptography can be found in [52]. As in any other generic group, general methods such as the Rho algorithm, baby-step, giant-step algorithm can be applied to solve the discrete logarithm problem on elliptic curves as well. Both the rho and baby-step, giant-step algorithms run on any curve with an exponential time. Special instances of elliptic curves exist in which solving the discrete logarithm problem on them become much simpler. The following elliptic curves are these special cases:

(a) Supersingular elliptic curves(for definition,see [44], Chapter V.) ECDLP on these elliptic curves can be solved through MOV algorithm [34]. This algorithm reduces ECDLP to discrete logarithm problem of finite fields and there exist subexponential time(for definition,see [27] Definition 3.2 of Chapter 2.) algorithm to solve discrete logarithm problem of finite fields.

(b) Anomalous elliptic curves(for definition,see [44],Proposition6.5 of Chapter XI.) ECDLP on these elliptic curves can be reduced to discrete logarithm problem of a group which is very simple. See [42], [48], [46].

(c) Elliptic curves such that the cardinality of the rational points only has small prime divisors. To be more precise, let $E/\mathbb{F}_q$ be an elliptic curve defined over a finite field $\mathbb{F}_q$, then $\#E(\mathbb{F}_q)$ only has small prime divisors. ECDLP on these elliptic curves can be solved by Pohlig-Hellman method [52].

The index calculus method is an approach which was introduced by Kraitchik [30] that can be optimized to a sub-exponential algorithm to solve the discrete logarithm problem for the multiplicative groups of finite fields [1]. For more details about this method, see section(2.3). It seems natural to attempt to adapt the index calculus approach to elliptic curves. The main issue here is to define a suitable factor base and with this factor base, we will require a sieving

algorithm to construct the linear relations from different multiples of the given elements as in the case of finite fields. A first step in this direction was taken by Semaev [43] who proposed the use of summation polynomials. He argued that upon a suitable choice of the factor base and a good algorithm to solve these polynomials, linear relations can be obtained which in turn leads to an efficient index calculus method for solving ECDLP. Unfortunately, the challenge to select a good factor base and to design such an efficient sieving method for elliptic curves over general finite fields remains wide open. Nonetheless, positive results were achieved independently by Gaudry [22] and Diem [11, 12] for elliptic curves over finite fields $\mathbb{F}_{q^n}$ for some classes of q and n. They essentially showed that the sieving process can be reformulated to that of solving a system of polynomial equations over $\mathbb{F}_q$. In particular, Diem showed in [12] that this led to sub-exponential time index calculus methods for some values of q and n.

Recently, in [21, 39], the authors concentrated on the case where $q = 2$ and n a prime. The resulting sieving process is then reduced to solving a set of multi-homogeneous boolean polynomials. It was suggested that these polynomials admit a certain structure (and are therefore not random). By employing the well-known Gröbner basis method to solve systems of polynomials, and making some assumptions (which remain to be proven), the authors showed that a sub-exponential index calculus algorithm for solving the discrete logarithm problem on elliptic curves over $\mathbb{F}_{2^n}$ is obtained.

Despite all these improvements in the ECDLP, the elliptic curves used in practice are still safe, all these methods or algorithms proposed in recent papers about ECDLP are not practical. There is still much work needed to suggest a practical method to beat Pollard's rho method.

In this thesis, we focus on the index calculus approach to solve ECDLP. First, using our framework on zero-dimensional polynomial systems, we argue that the first fall degree assumption used to estimate the complexity of the method via Weil descent may not be valid. We then provide a different method to solve the polynomial system arising from the relation search step of the index calculus approach. In particular, we investigate a sub-class of vector spaces

with nice characteristic polynomials. Using these vector spaces, we transform the polynomial system into one with smaller degrees. We provide complexity bounds for our approach and give conditions such that an efficient index calculus method will result. Finally, we provide some concrete examples of vector spaces with the nice properties.

1.3 Contributions of the thesis

The main contributions of this thesis are as follows:

Contribution 1. We discuss a mostly theoretical framework for solving zero-dimensional polynomial systems. Complexity bounds are obtained for solving such systems using a new parameter, called the *last fall degree*, which does not depend on the choice of a monomial order. We prove the main theorem(1.1.1). With this result, we prove that the multi-HFE cryptosystem is not secure.

Contribution 2. We identify some characteristics of a vector space that can serve as good factor bases for the index calculus approach to solve ECDLP. Using these vector spaces, we propose an approach to perform the relation search step in the index calculus algorithm. We provide complexity bounds for using our approach and give conditions for an efficient index calculus method. Finally, we present some concrete examples of the vector spaces that we seek.

1.4 Organization of the thesis

The rest of this thesis is organized as follows. In chapter 2, we review the basic definitions and the framework of index calculus method. In chapter 3, we summarized some known methods to solve zero-dimensional polynomial systems. In chapter 4, we summarized the recent developments of index calculus for solving ECDLP. Chapter 5 contains our method for solving zero-dimensional Weil descent system. In Section 5.1 we discuss the last fall degree. We will also discuss how one can solve zero-dimensional systems using the last fall degree and we will compare this method with other methods. In Section 5.2 we introduce Weil descent and an alternative version of

Weil descent. Section 5.3 is devoted to the proof of Theorem 1.1.1. In this section we first discuss the relation between the two Weil descent systems. Then we study the monovariate case and deduce the result for the multivariate case from the monovariate case using projection polynomials. Finally, we discuss how one can generalize the main theorem. In Section 5.4 we discuss the relation with multi-HFE. In Section 5.5 we discuss why the results in this article are not directly useful for studying systems coming from summation polynomials for the elliptic curve discrete logarithm problem. In chapter 6, we propose a sub-class of vector spaces with nice properties that can serve as good factor bases for the index calculus approach. We use these vector spaces and describe how the polynomial systems can be solved. We provide complexity bounds for our approach as well as the conditions for which the approach will be sub-exponential. Chapter 7 gives a summary of our results and some remaining problems for future research.

2. PRELIMINARIES

In this chapter, we will briefly review the basic properties of elliptic curves and formally define the elliptic curve discrete logarithm problem. We will then discuss some approaches to solve the elliptic curve discrete logarithm problem, notably the index calculus method. Finally, we will present a detailed exposition of an important class of polynomials called the summation polynomials associated with an elliptic curve.

2.1 Elliptic curves

In this section,we introduce the definition of an elliptic curve, its associated group law and some of its properties. For more about elliptic curves, see [44].

Definition 2.1.1. An elliptic curve E defined over a field $\mathbb{F}$, denoted by $E/\mathbb{F}$, is an algebraic curve given by the following Weierstrass equation:

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$$

where $a_1, a_2, a_3, a_4, a_6 \in \mathbb{F}$.

For the definition of an algebraic curve, see [44],Chapter II.

In this thesis, we will always consider elliptic curves defined over finite fields.

Remark 2.1.2. Let $\overline{\mathbb{F}}$ be an algebraic closure of $\mathbb{F}$ and let $\mathcal{O}$ denote the point of E at infinity.

We can regard an elliptic curve as the following set:

$$E = \{(x, y) \in \overline{\mathbb{F}} \times \overline{\mathbb{F}} \mid y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6\} \cup \{\mathcal{O}\}$$

Remark 2.1.3. For $char(\mathbb{F}) \neq 2, 3$, every elliptic curve defined over $\mathbb{F}$ can be given by a short Weierstrass equation:

$$E: \ y^2 = x^3 + a_4x + a_6.$$

for some $a_4, a_6 \in \mathbb{F}$.

Every elliptic curve can be equipped with an abelian group structure with $\mathcal{O}$ as the identity element. We give the explicit formula of the group law algebraically as follows:

Definition 2.1.4 (The Group Law)**.** Let $E/\mathbb{F}$ be an elliptic curve given by Weierstrass equation

$$E: \ y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6.$$

(a) (Identity): $P + \mathcal{O} = \mathcal{O} + P = P$, for all $P = (x, y) \in E$.

(b) (Inverse): Let $P_0 = (x_0, y_0)$. Then $-P_0 = (x_0, -y_0 - a_1x_0 - a_3)$.

(c) (Addition): Let $P_1 + P_2 = P_3$ with $P_i = (x_i, y_i) \in E$ for $i = 1, 2, 3$.

If $x_1 = x_2$ and $y_1 + y_2 + a_1x_2 + a_3 = 0$, then

$$P_1 + P_2 = \mathcal{O}.$$

Otherwise, define

$$\lambda = \begin{cases} \frac{y_2 - y_1}{x_2 - x_1}, & x_1 \neq x_2 \\ \frac{3x_1^2 + 2a_2x_1 + a_4 - a_1y_1}{2y_1 + a_1x_1 + a_3}, & x_1 = x_2. \end{cases}$$

and

$$\nu = \begin{cases} \frac{y_1x_2 - y_2x_1}{x_2 - x_1}, & x_1 \neq x_2 \\ \frac{-x_1^3 + a_4x_1 + 2a_6 - a_3y_1}{2y_1 + a_1x_1 + a_3}, & x_1 = x_2. \end{cases}$$

then $P_3 = P_1 + P_2$ has coordinates

$$x_3 = \lambda^2 + a_1\lambda - a_2 - x_1 - x_2, y_3 = -(\lambda + a_1)x_3 - \nu - a_3$$

Theorem 2.1.5. (a) *The above group law makes* $E = E(\overline{F}) = \{(x, y) \in \overline{\mathbb{F}} \times \overline{\mathbb{F}} \mid y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6\} \cup \{\mathcal{O}\}$ *into an abelian group with the identity* $\mathcal{O}$.

(b) *The rational points* $E(\mathbb{F}) = \{(x, y) \in \mathbb{F} \times \mathbb{F} \mid y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6\} \cup \{\mathcal{O}\}$ *form a subgroup of* E.

For a geometrical view of the group law and the proof of the above theorem, see [44],Chapter III.

For an elliptic curve defined over a finite field, we have the following theorem to estimate the number of the rational points of this elliptic curve.

Theorem 2.1.6 (Hasse)**.** *Let* $E/\mathbb{F}_q$ *be an elliptic curve defined over a finite field* $\mathbb{F}_q$. *Then the number of the rational points* $E(\mathbb{F}_q)$ *of* E *satisfies the following inequality:*

$$q + 1 - 2\sqrt{q} \leq \#E(\mathbb{F}) \leq q + 1 + 2\sqrt{q},$$

where $\#E(\mathbb{F}_q)$ *denotes the cardinality of* $E(\mathbb{F}_q)$.

Proof. See [44],Theorem1.1 of Chapter V. □

Remark 2.1.7. (a) The above theorem implies that the number of rational points on an elliptic curve over a finite field is bounded. In particular, $\#E(\mathbb{F}_q) = O(q)$.

(b) In Cryptography, we are mainly interested in those elliptic curves defined over finite fields whose rational points form a cyclic group or contain a large cyclic subgroup of the order $O(q)$.

2.2 The elliptic curve discrete logarithm problem

Many elliptic curve cryptosystems base their security on the difficulty for solving the elliptic curve discrete logarithm problem(we use ECDLP for short) efficiently for random elliptic curves. The formal definition of ECDLP is the following:

Definition 2.2.1 (The elliptic curve discrete logarithm problem)**.** For an elliptic curve $E/\mathbb{F}_q$ defined over a finite field $\mathbb{F}_q$, let $P, Q \in E(\overline{\mathbb{F}_q})$ be two points of E with $Q \in \langle P \rangle$. The elliptic curve discrete logarithm problem is to find an integer k such that $Q = kP$.

In practice, the order of P in the above definition is known and it is often a large prime number.

In general, for a randomly chosen elliptic curve, the fastest currently known algorithm to solve ECDLP over this elliptic curve is Pollard's ρ algorithm [40]. For ECDLP on random elliptic curves, there exist only generic algorithms such as baby-step giant-step algorithm, Pollard's ρ method and its variants [52]. These algorithms don't consider the structure of the group, so they apply to any discrete logarithm problem on any group. These algorithms have exponential time complexity and therefore attacking ECDLP is infeasible in practice at present.

2.3 Index calculus

In this section, we present an approach commonly known as the index calculus method. This method first appeared in the work of Kraitchik [30, 31] and later developed by Adleman [1] and many other mathematicians. It is a method originally dedicated to compute discrete logarithm problems on finite prime fields. Subsequently, it was adapted to compute discrete logarithm

problems on arbitary finite fields and hyperelliptic curves. The index calculus method is a probabilistic method which can be described as follows.

Let us first generalize the discrete logarithm problem to an arbitrary cyclic group $\mathcal{G}$ with order N. Let $+$ denote the operation in $\mathcal{G}$. Given a generator P of $\mathcal{G}$ and an element $Q \in \mathcal{G}$, the discrete logarithm problem seeks to find an integer x such that $Q = xP$, where xP denotes $P+P+\ldots+P$ (x times). The discrete logarithm of Q to the base P is denoted by $\log_P(Q)$. We give a brief outline of the index calculus method for solving this problem. The index calculus method involves the following three steps:

1. Factor Base definition.

   Choose a subset $\mathcal{F} = \{P_1, P_2, \ldots, P_t\} \subset \mathcal{G}$. This subset $\mathcal{F}$ is called a factor base. In general, we try to choose a factor base $\mathcal{F}$ with the following properties:

   - Randomly choose an element $h \in \mathcal{G}$. Then h can be expressed as a linear combination of the elements of the factor base with a high probability;
   - There exists an efficient algorithm to express h as a linear combination of the P_i's,if such an expression exists. If such an expression does not exist, then this algorithm also can detect this nonexistence efficiently.

2. Relation search(also known as sieving step).

   Randomly choose two integers a_i and b_i in $[1, N]$ and try to write $a_iP + b_iQ$ as a linear combination of the elements in the factor base, i.e. we want to get equations of the form:

   $$a_iP + b_iQ = c_{i1}P_1 + ... + c_{it}P_t$$

   where $c_{ij} \in \mathbb{Z}, j = 1, ..., t$.

   If the pair (a_i, b_i) does not produce such a linear combination, discard the pair and choose another pair. Choose sufficient pairs to obtain more than $\#\mathcal{F}$ such relations. Suppose we

collect $\#\mathcal{F}+1$ relations as follows:

$$a_iP + b_iQ = c_{i1}P_1 + ... + c_{it}P_t, i = 1, 2, ..., t+1.$$

3. Linear algebra.

The final step is to employ elementary linear algebra to deduce the discrete logarithm x through the relations collected in step 2 . Specifically, construct the matrix $C = (c_{ij})_{1\leq i\leq t+1, 1\leq j\leq t}$ and find a nontrivial solution of the following linear equations:

$$vC \equiv 0 \pmod{N}, v = (v_1, ..., v_{t+1}).$$

By nontrivial, we mean that

$$(\sum_{i=1}^{t+1} v_ia_i, \sum_{i=1}^{t+1} v_ib_i) \neq (0, 0) \pmod{N}.$$

This nontrivial solution exists and it can be solved by linear algebra if the $t+1$ relations collected are linearly independent.

With v denoting the nontrivial solution computed, we have the following:

$$\sum_{i=1}^{t+1} v_ia_iP + \sum_{i=1}^{t+1} v_ib_iQ = 0$$

If $\sum_{i=1}^{t+1} v_ib_i$ is invertible in $\mathbb{Z}/N\mathbb{Z}$, then we get the discrete logarithm

$$x = -\frac{\sum_{i=1}^{t+1} v_ia_i}{\sum_{i=1}^{t+1} v_ib_i} \pmod{N}.$$

On the other hand, if $\sum_{i=1}^{t+1} v_ib_i$ is not invertible in $\mathbb{Z}/N\mathbb{Z}$, then we need to try other nontrivial solutions. Specifically, we can collect additional relations to get a different

linear equation.

Remark 2.3.1. (i) Observe that steps 1 and 2 are the critical steps for the index calculus method as step 3 can be performed once the relations are formed. More precisely, the challenge to construct a good index calculus method is often to construct a good factor base $\mathcal{F}$ and an efficient sieving step (Step 2).

(ii) Time complexity of Step 3 is $T_3 = O(t^{\omega'})$, where ω' is the sparse linear algebra constant Since the matrix C is usually sparse.

(iii) Let p_0 denote the probability that an element $Q \in \mathcal{G}$ can be expressed as a linear combination of the elements in $\mathcal{F}$. Let T_0 be the time required to express Q as a linear combination of the P_i's. Then the time complexity for step 2 is $T_2 = O(tT_0/p_0)$.

We will demonstrate the power of the index calculus method with an explicit example in prime finite fields. We observe that it works well in these fields since there exists a good factor base, i.e., the set of prime numbers smaller than some given bound, as well as an efficient algorithm to factor integers with all prime factors less than a given bound.

Example 2.3.2. *Let $p = 16547$. We consider the group $\mathbb{F}_p^*$. $P = 11$ is a generator of the group. let $Q = 2392$. We want to find the discrete logarithm of Q to the base P, i.e. find k such that* $11^k \equiv 2392 \pmod{p}$.

First, we choose the factor base $\mathcal{F} = \{2, 3, 5, 7, 13\}$, so $t = \#\mathcal{F} = 5$. Next, we try to collect

enough relations. After some trials, we finally get the relations as follows:

$$\begin{aligned}
11^{137} &\equiv 8320 \equiv 2^7 \cdot 5 \cdot 13 \pmod{p} \\
11^{314} &\equiv 2700 \equiv 2^2 \cdot 3^3 \cdot 5^2 \pmod{p} \\
11^{499} &\equiv 504 \equiv 2^3 \cdot 3^2 \cdot 7 \pmod{p} \\
11^{518} &\equiv 13230 \equiv 2 \cdot 3^3 \cdot 5 \cdot 7^2 \pmod{p} \\
11^{949} &\equiv 12168 \equiv 2^3 \cdot 3^2 \cdot 13^2 \pmod{p} \\
2392 \cdot 11^{337} &\equiv 735 \equiv 3 \cdot 5 \cdot 7^2 \pmod{p}
\end{aligned}$$

Let $L_2 = \log_{11} 2, L_3 = \log_{11} 3, L_5 = \log_{11} 5, L_7 = \log_{11} 7, L_{13} = \log_{11} 13$ *and take logarithm to the base 11 for the above equations. We have the following linear equations:*

$$\begin{aligned}
7L_2 + L_5 + L_{13} &\equiv 137 \pmod{16546} \\
2L_2 + 3L_3 + 2L_5 &\equiv 314 \pmod{16546} \\
3L_2 + 2L_3 + L_7 &\equiv 499 \pmod{16546} \\
L_2 + 3L_3 + L_5 + 2L_7 &\equiv 518 \pmod{16546} \\
3L_2 + 2L_3 + 2L_{13} &\equiv 949 \pmod{16546} \\
L_3 + L_5 + 2L_7 &\equiv \log_{11} 2392 + 337 \pmod{16546}
\end{aligned}$$

Finally, in Step 3, we solve the linear system to yield the following solution:

$$\begin{aligned}
L_2 &\equiv 4741 \pmod{16546} \\
L_3 &\equiv 15350 \pmod{16546} \\
L_5 &\equiv 5483 \pmod{16546} \\
L_7 &\equiv 5214 \pmod{16546} \\
L_{13} &\equiv 11105 \pmod{16546}
\end{aligned}$$

and from the last relation, we obtain $k = \log_{11} 2392 = L_3 + L_5 + 2L_7 - 337 \equiv 15350 + 5483 + 2(5214) - 337 \equiv 14378 \pmod{16546}$. *Thus the desired discrete logarithm is* $k = 14378$.

Remark 2.3.3. The above example is a little different from the framework of the index calculus we presented. In fact, the original index calculus method for discrete logarithm problem for prime finite fields uses the same idea of the 3 steps of the above example. Then mathematicians adapted this idea and formed a variant of this method, i.e. the outline we stated above.

2.4 Summation polynomials

Adapting the index calculus method to solve the ECDLP will require us to define a factor base. Unfortunately, unlike the case of a finite field, there is no obvious factor base for the group of rational points of an elliptic curve defined over a finite field, and thus it is difficult to apply the index calculus method to elliptic curves. Indeed, the challenge is to construct a good factor base such that a point can be efficiently expressed as a linear combination of the elements in the factor base. A first step in this direction was taken by Igor Semaev. In 2004, Igor Semaev introduced the concept of summation polynomials in a preprint [43]. These polynomials build relationships between the x-coordinates of finite points on an elliptic curve which sum to the identity element.

Let $\mathbb{F}$ be a finite field and E be an elliptic curve defined over $\mathbb{F}$ by the following Weierstrass equation:

$$Y^2 + a_1XY + a_3Y = X^3 + a_2X^2 + a_4X + a_6 \tag{2.1}$$

And let $\overline{\mathbb{F}}$ denote an algebraic closure of $\mathbb{F}$. We have the following proposition.

Proposition 2.4.1. *Let E be an elliptic curve defined over $\mathbb{F}$,and $m \in \mathbb{N}$ with $m \geq 2$. Then there exists an unique (up to multiplication by a nonzero constant) irreducible polynomial $S_m \in \mathbb{F}[X_1, ..., X_m]$ such that $S_m(x(P_1), ..., x(P_m)) = 0$ if and only if there exist $\epsilon_1, ..., \epsilon_m \in \{1, -1\}$ with $\epsilon_1 P_1 + ... + \epsilon_m P_m = 0$, for any $P_1, ..., P_m \in E(\overline{\mathbb{F}}) \setminus \{0\}$, where $x(P_i)$ is the x-coordinate of P_i,$i = 1, ..., m$.*

Proof. See [11], Proposition 2.1. □

Definition 2.4.2. The polynomial S_m in the above proposition is called an m-th summation polynomial of E.

We can explicitly construct S_m recursively. First, $S_2 = X_1 - X_2$. For other summation polynomials we have the following two lemmas.

Lemma 2.4.3. *Let E be an elliptic curve given by the equation(2.1). Then the 3rd summation polynomial of E is:*

$$\begin{aligned} S_3 =&(X_1^2X_2^2 + X_1^2X_3^2 + X_2^2X_3^2) - 2 \cdot (X_1^2X_2X_3 + X_1X_2^2X_3 + X_1X_2X_3^2) \\ &- b_2 \cdot (X_1X_2X_3) - b_4 \cdot (X_1X_2 + X_1X_3 + X_2X_3) - b_6 \cdot (X_1 + X_2 + X_3) - b_8, \end{aligned}$$

where the b_i's are defined as:

$$\begin{aligned} b_2 =&a_1^2 + 4a_2, \\ b_4 =&a_1a_3 + 2a_4, \\ b_6 =&a_3^2 + 4a_6, \\ b_8 =&a_1^2a_6 - a_1a_3a_4 + a_2a_3^2 + 4a_2a_6 - a_4^2. \end{aligned}$$

Proof. Using the definition of group law for elliptic curves, one can do a lengthy calculation to verify the above. See [11], Lemma 3.4. □

Lemma 2.4.4. *Let E be an elliptic curve given by the equation(2.1). Let $p, q \in \mathbb{N}$ with $p, q \geq 2$. Then*

$$S_{p+q}(X_1, ..., X_{p+q}) = Res_X(S_{p+1}(X_1, ..., X_p, X), S_{q+1}(X_{p+1}, ..., X_{p+q}, X))$$

where Res_X denotes the Sylvester resultant(See [10],Definition7 of Chapter3,§5) with respect to X.

Proof. Consider any $p+q$ points $(x_1, y_1), ..., (x_{p+q}, y_{p+q}) \in E(\overline{\mathbb{F}})$ such that

$$(x_1, y_1) + ... + (x_{p+q}, y_{p+q}) = 0. \tag{2.2}$$

If $(x_1, y_1) + ... + (x_p, y_p) = (x, y)$ for some finite point $(x, y) \in E(\overline{\mathbb{F}})$, then $(x_{p+1}, y_{p+1}) + ... + (x_{p+q}, y_{p+q}) = -(x, y) = (x, -y - a_1x - a_3)$. It follows from proposition(2.4.1) that $S_{p+1}(x_1, ..., x_p, x) = 0$ and $S_{q+1}(x_{p+1}, ..., x_{p+q}, x) = 0$.

Thus by the property of resultants, we have

$$\begin{aligned} Res_X(S_{p+1}(x_1, ..., x_p, X), S_{q+1}(x_{p+1}, ..., x_{p+q}, X)) =& A(X) * S_{p+1}(x_1, ..., x_p, X) \\ &+ B(X) * S_{q+1}(x_{p+1}, ..., x_{p+q}, X) \end{aligned} \tag{2.3}$$

for some polynomials $A(X), B(X)$ which are integer polynomials in the coefficients of $S_{p+1}(x_1, ..., x_p, X)$ and $S_{q+1}(x_{p+1}, ..., x_{p+q}, X)$.

Now by letting $X = x$ in equation(2.3), we have

$$Res_X(S_{p+1}(x_1, ..., x_p, X), S_{q+1}(x_{p+1}, ..., x_{p+q}, X)) = 0$$

On the other hand, if $(x_1, y_1) + ... + (x_p, y_p) = 0$, then $(x_{p+1}, y_{p+1}) + ... + (x_{p+q}, y_{p+q}) = 0$. It follows from proposition(2.4.1) that $S_p(x_1, ..., x_p) = 0$ and $S_q(x_{p+1}, ..., x_{p+q}) = 0$.

Since the leading coefficients of $S_{p+1}(x_1, ..., x_p, X)$ and $S_{q+1}(x_{p+1}, ..., x_{p+q}, X)$ are $S_p(x_1, ..., x_p)^2$ and $S_q(x_{p+1}, ..., x_{p+q})^2$ respectively by proposition(2.4.5) which we will prove later, we have

$$Res_X(S_{p+1}(x_1, ..., x_p, X), S_{q+1}(x_{p+1}, ..., x_{p+q}, X)) = 0$$

Thus we have proven $(x_1, y_1) + ... + (x_{p+q}, y_{p+q}) = 0$ implies

$Res_X(S_{p+1}(x_1, ..., x_p, X), S_{q+1}(x_{p+1}, ..., x_{p+q}, X)) = 0$.

Next we prove the converse is true.

Suppose $Res_X(S_{p+1}(x_1, ..., x_p, X), S_{q+1}(x_{p+1}, ..., x_{p+q}, X)) = 0$.

If the leading coefficients of $S_{p+1}(x_1, ..., x_p, X)$ and $S_{q+1}(x_{p+1}, ..., x_{p+q}, X)$ are zeros, i.e. $S_p(x_1, ..., x_p) = 0$ and $S_q(x_{p+1}, ..., x_{p+q}) = 0$. Then by proposition(2.4.1), there exist $y_1, ..., y_{p+q} \in \overline{\mathbb{F}}$ such that $(x_1, y_1), ..., (x_{p+q}, y_{p+q}) \in E(\overline{\mathbb{F}})$,

$$(x_1, y_1) + ... + (x_p, y_p) = 0$$

and

$$(x_{p+1}, y_{p+1}) + ... + (x_{p+q}, y_{p+q}) = 0$$

Thus we have

$$(x_1, y_1) + ... + (x_{p+q}, y_{p+q}) = 0$$

as required.

If one of the leading coefficients of $S_{p+1}(x_1, ..., x_p, X)$ and $S_{q+1}(x_{p+1}, ..., x_{p+q}, X)$ is nonzero, then $S_{p+1}(x_1, ..., x_p, X)$ and $S_{q+1}(x_{p+1}, ..., x_{p+q}, X)$ have a common root $x \in \overline{\mathbb{F}}$. Again by proposition(2.4.1), we have

$$\pm(x_1, y_1) \pm ... \pm (x_p, y_p) = \pm(x, y)$$

and

$$\pm(x_{p+1}, y_{p+1}) \pm ... \pm (x_{p+q}, y_{p+q}) = \pm(x, y')$$

for some $y_1, ..., y_{p+q}, y, y' \in \overline{\mathbb{F}}$. Note that (x, y) and (x, y') have the same x-coordinate, thus $(x, y') = \pm(x, y)$.

Since $-(x, y) = (x, y - a_1x - a_3)$, change the value of y_i to $y_i - a_1x_i - a_3$ if necessary, we can always get the following:

$$(x_1, y_1) + ... + (x_{p+q}, y_{p+q}) = 0$$

for suitable choices of the $y_1, ..., y_{p+q} \in \overline{\mathbb{F}}$.

Thus we have proven

$$Res_X(S_{p+1}(x_1, ..., x_p, X), S_{q+1}(x_{p+1}, ..., x_{p+q}, X)) = 0$$

if and only if there exist $p + q$ points $(x_1, y_1), ..., (x_{p+q}, y_{p+q}) \in E(\overline{\mathbb{F}})$ such that

$$(x_1, y_1) + ... + (x_{p+q}, y_{p+q}) = 0$$

Since summation polynomial is unique up to a constant, it follows that

$$S_{p+q}(x_1, ..., x_{p+q}) = Res_X(S_{p+1}(x_1, ..., x_p, X), S_{q+1}(x_{p+1}, ..., x_{p+q}, X))$$

□

The above lemmas gives the explicit construction of all summation polynomials of an elliptic curve, namely, one can compute S_m for $m \geq 3$ from S_{m-1} and S_3 by applying lemma(2.4.4) with $p = m - 2$ and $q = 2$.

Summation polynomials have some nice properties which are given by the following proposition.

Proposition 2.4.5. *Let $m \in \mathbb{N}$ with $m \geq 3$. Then the m-th summation polynomial S_m of an*

elliptic curve E has the following properties:

(a) *S_m is symmetric.*

(b) *S_m has degree 2^{m-2} in each variable.*

(c) *S_m is absolutely irreducible.*

(d) $S_m(X_1, ..., X_{m-1}, X_m) = S_{m-1}^2(X_1, ..., X_{m-1})X_m^{2^{m-2}} + ...$

Proof. (a) By proposition(2.4.1) $S_m(x_1, ..., x_m) = 0$ iff there exist m points $(x_1, y_1), ..., (x_m, y_m) \in E(\overline{\mathbb{F}})$ such that

$$(x_1, y_1) + ... + (x_m, y_m) = 0$$

Since the group law on elliptic curve is abelian, by proposition(2.4.1) we again have

$$S_m(x_1, ...x_{i-1}, x_j, x_{i+1}, ..., x_{j-1}, x_i, x_{j+1}, ..., x_m) = 0$$

for any $1 \leq i < j \leq m$. Thus

$$S_m(x_1, ..., x_m) = 0$$

if and only if

$$S_m(x_1, ...x_{i-1}, x_j, x_{i+1}, ..., x_{j-1}, x_i, x_{j+1}, ..., x_m) = 0$$

Thus S_m is symmetric.

(b) By induction and the definition of resultant, one has $\deg_{X_m} S_m \leq 2^{m-2}$. In addition, one can always find $m-1$ points $(x_1, y_1), ..., (x_{m-1}, y_{m-1}) \in E(\overline{\mathbb{F}})$ such that the x-coordinates of

2^{m-2} points

$$(x_1, y_1) \pm \ldots \pm (x_{m-1}, y_{m-1})$$

are pairwise different. In fact, we can prove the following stronger claim.

Claim: There exist $m-1$ points $(x_1, y_1), ..., (x_{m-1}, y_{m-1}) \in E(\overline{\mathbb{F}})$ such that the following hold

1. $2*((x_{i_1}, y_{i_1})+...+(x_{i_k}, y_{i_k}))-2*((x_{j_1}, y_{j_1})+...+(x_{j_l}, y_{j_l})) \neq 0$, for any $1 \leq k \leq m-1$,$0 \leq l \leq m-1$ with $k+l \leq m-1$,where $1 \leq i_1, ..., i_k, j_1, ..., j_l \leq m-1$ are pairwise different.
2. $\pm(x_1, y_1) \pm \ldots \pm (x_{m-1}, y_{m-1}) \neq 0$.

We can find such $m-1$ points by induction on m. For $m=2$, since $\#E(\overline{\mathbb{F}}) = \infty$ and $\#E[2] = \{P \in E(\overline{\mathbb{F}}) : 2*P = 0\}$ is finite, we can find a point satisfying the above two conditions. Now suppose for $m=r$ we can find $r-1$ points $P_1 = (x_1, y_1), ..., P_{r-1} = (x_{r-1}, y_{r-1})$ satisfying the two results in the claim. For $m = r+1$, we just need to choose a point $P_r = (x_r, y_r)$ such that

- $2 * P_r \neq 2 * (P_{i_1} + ... + P_{i_k} - P_{j_1} - ... - P_{j_l})$, for any $1 \leq k \leq r-1, 0 \leq l \leq r-1$ with $k+l \leq r-1$,where $1 \leq i_1, ..., i_k, j_1, ..., j_l \leq r-1$ are pairwise different.
- $P_r \neq \pm P_1 \pm \ldots \pm P_{r-1}$

we can always find such a point P_r, since there are only finite many points such that one of the above two inequalities becomes an equality. Once we choose such a point P_r, the claims hold.

Since two points P and Q have the same x-coordinate if and only $P = Q$ or $P + Q = 0$, from the above claim, it is easy to see there exist $m-1$ points $(x_1, y_1), ..., (x_{m-1}, y_{m-1}) \in E(\overline{\mathbb{F}})$ such that the x-coordinates of 2^{m-2} points

$$(x_1, y_1) \pm \ldots \pm (x_{m-1}, y_{m-1})$$

are pairwise different.

Thus by proposition(2.4.1), the polynomial $S_m(x_1, ..., x_{m-1}, X_m)$ in the variable X_m has 2^{m-2} roots. So $\deg_{X_m} S_m = 2^{m-2}$. The same result holds for all other variables since S_m is symmetric by (a).

(c) See [11], Proposition 2.1.

(d) Let C_m denote the coefficient of S_m at $X_m^{2^{m-2}}$. Then $C_m * X_m^{2^{m-2}}$ is equal to the polynomial

$$Z_m^{2^{m-2}} S_m(X_1, ..., X_{m-1}, \frac{X_m}{Z_m})$$

evaluate at $Z_m = 0$.

By applying lemma(2.4.4) with $p = m - k - 1$ and $q = k + 1$,for $k \geq 1$, we have

$$\begin{aligned} Z_m^{2^{m-2}} S_m(X_1, ..., X_{m-1}, \frac{X_m}{Z_m}) =& Z_m^{2^{m-2}} Res_X(S_{m-k}(X_1, ..., X_{m-k-1}, X), S_{k+2}(X_{m-k}, ..., \frac{X_m}{Z_m}, X)) \\ =& Res_X(S_{m-k}(X_1, ..., X_{m-k-1}, X), Z_m^{2^k} S_{k+2}(X_{m-k}, ..., \frac{X_m}{Z_m}, X)). \end{aligned}$$

By induction,

$$\begin{aligned} & Res_X(S_{m-k}(X_1, ..., X_{m-k-1}, X), Z_m^{2^k} S_{k+2}(X_{m-k}, ..., \frac{X_m}{Z_m}, X))_{(Z_m=0)} = \\ & Res_X(S_{m-k}(X_1, ..., X_{m-k-1}, X), Z_m^{2^k} S_{k+2}(X_{m-k}, ..., \frac{X_m}{Z_m}, X)_{(Z_m=0)}) = \\ & Res_X(S_{m-k}(X_1, ..., X_{m-k-1}, X), X_m^{2^k} S_{k+1}^2(X_{m-k}, ..., X_{m-1}, X)) = \\ Res_X(S_{m-k}(X_1, ..., X_{m-k-1}, X), X_m^{2^k}) * & Res_X^2(S_{m-k}(X_1, ..., X_{m-k-1}, X), S_{k+1}(X_{m-k}, ..., X_{m-1}, X)) = \\ & X_m^{2^{m-2}} S_{m-1}^2(X_1, ..., X_{m-1}) \end{aligned}$$

Thus $C_m = S_{m-1}^2(X_1, ..., X_{m-1})$.

□

Notice that the number of terms of a summation polynomial grows very quickly with m. In

fact, the polynomial is rather dense and we can expect that S_m has $2^{O(m^2)}$ terms. By using the evaluation/interpolation method to construct resultants, it takes time $2^{O(m^2)}$ to construct the m-th summation polynomial S_m.

The following table shows the number of terms for S_m for a field with characteristic 2.

Tab. 2.1: Number of terms of summation polynomials

m	No. of terms	Degree
2	2	1
3	5	4
4	24	12
5	729	32
6	148300	80

In practice, efficiently computing summation polynomial is hard. To the best of our knowledge, the current record is to compute the 8-th summation polynomial, see [19]. As such, to find an efficient way to compute large summation polynomial is an interesting open problem.

For singular curves, we can define summation polynomials similarly. These summation polynomials are related to some famous problems. To the best of our knowledge, [29] is the first article considering summation polynomials of singular curves. In the remainder of this section, we briefly summarize the results of this article.

For any curve(singular or nonsingular)E defined over $\mathbb{F}$ by the following Weierstrass equation:

$$Y^2 + a_1XY + a_3Y = X^3 + a_2X^2 + a_4X + a_6,$$

we define summation polynomials of E as follows:

$$S_2 = X_1 - X_2 \in \mathbb{F}[X_1, X_2]$$

$$S_3 = (X_1^2X_2^2 + X_1^2X_3^2 + X_2^2X_3^2) - 2 \cdot (X_1^2X_2X_3 + X_1X_2^2X_3 + X_1X_2X_3^2)$$
$$- b_2 \cdot (X_1X_2X_3) - b_4 \cdot (X_1X_2 + X_1X_3 + X_2X_3) - b_6 \cdot (X_1 + X_2 + X_3) - b_8 \in \mathbb{F}[X_1, X_2, X_3],$$

where b_2, b_4, b_6, b_8 is the same as lemma(2.4.3).

For $m \geq 4$,

$$S_m = Res_X\left(S_{m-1}(X_1, \ldots, X_{m-2}, X), S_3(X_{m-1}, X_m, X)\right) \in \mathbb{F}[X_1, \ldots, X_m].$$

We have the following propositions for two singular curves. The first proposition gives the connection between discrete logarithm of field elements and evaluations of summation polynomials, while the second proposition provides a relationship between sums of field elements and the summation polynomial evaluations.

Proposition 2.4.6. *Let E be the singular curve defined over $\mathbb{F}$ by the Weierstrass equation:*

$$Y^2 + XY = X^3.$$

For $m \geq 3$, let $S_m \in \mathbb{F}[X_1, \ldots, X_m]$ denote the m-th summation polynomial of E. Let $x_1, x_2, ..., x_m \in \mathbb{F}^* \setminus \{1\}$. Then there are $n_i \in \{-1, 1\}$ $(i = 1, \ldots, m)$ such that $x_1^{n_1} \cdots x_m^{n_m} = 1$ if and only if

$$S_m\left(\frac{x_1}{(x_1 - 1)^2}, \ldots, \frac{x_m}{(x_m - 1)^2}\right) = 0.$$

Proposition 2.4.7. *Let E be the singular curve defined over $\mathbb{F}$ by the Weierstrass equation:*

$$Y^2 = X^3$$

For $m \geq 3$, let $S_m \in \mathbb{F}[X_1, \ldots, X_m]$ denote the m-th summation polynomial of E. Let $x_1, x_2, ..., x_m \in \mathbb{F} \setminus \{0\}$. Then there are $n_i \in \{-1, 1\}$ $(i = 1, \ldots, m)$ such that $n_1x_1 + \cdots + n_mx_m = 0$ if and

only if

$$S_m(\frac{1}{x_1^2},\ldots,\frac{1}{x_m^2})=0.$$

This proposition shows that there is a connection between summation polynomial and the famous subset sum problem. More concretely, let $\mathbb{F}_q$ be a finite field of cardinality $q=p^n$. The subset sum problem is the following. Given $x_1,\ldots,x_m,a\in\mathbb{F}_q$, determine whether or not there are $\epsilon_i\in\{0,1\}$ such that $\sum_{i=1}^m\epsilon_i x_i=a$. If $p\neq 2$, this problem is equivalent to decide if S_{m+1} evaluates to zero at certain points, where S_{m+1} is the summation polynomial of the singular curve E in proposition(2.4.7).

Assume that $x_1,\ldots,x_m,a\in\mathbb{F}_q$ are given as in the subset sum problem. Write $x_i=2y_i$,$i=1,2,...,m$. Then the subset sum problem has a solution if and only if $S_{m+1}(y_1,\ldots,y_m,a-y_1-\ldots-y_m)=0$. Indeed, if $S_{m+1}(y_1,\ldots,y_m,a-y_1-\ldots-y_m)=0$, then by proposition(2.4.7), we can write $\sum_{i=1}^m n_iy_i-(a-y_1-\ldots-y_m)=0$ ($n_i\in\{\pm 1\}$). Set $\epsilon_i=(n_i+1)/2\in\{0,1\}$. Then one has

$$a=\sum_i(n_i+1)y_i=\sum_i\epsilon_i x_i.$$

The proof of the other direction is similar.

Fix $p\geq 5$. The subset sum problem over $\mathbb{F}_{p^n}$ is NP-complete. Thus the above statement says to check if a summation polynomial evaluates at certain points to zero or not is difficult, namely, it is a NP-complete problem. From this, we again see that summation polynomial is a complicate polynomial.

3. KNOWN METHODS FOR SOLVING ZERO-DIMENSIONAL POLYNOMIAL SYSTEMS

In this chapter, we review some classical methods and time complexity for solving systems of multivariate polynomial equations over a finite field.

We first define the main problem.

Problem 3.0.8. *[Polynomial Systems Solving over Finite Fields] Let $\mathbb{F}_q$ be a finite field of cardinality q. Given m polynomials $f_1(X_1, ..., X_n), ..., f_m(X_1, ..., X_n) \in \mathbb{F}_q[X_1, ..., X_n]$,determine if there exists any one vector $(z_1, ..., z_n) \in \mathbb{F}_q^n$ such that $f_1(z_1, ..., z_n) = ... = f_m(z_1, ..., z_n) = 0$. Furthermore, output one such vector $(z_1, \ldots, z_n)$ if it exists.*

We remark that it is sufficient to find one solution of the polynomial system for cryptographic applications. We only consider zero-dimensional polynomial systems(see definition(3.0.9)) in this chapter.

Definition 3.0.9. An ideal $I \subseteq R = \mathbb{F}_q[X_1, ..., X_n]$ is called zero-dimensional if the dimension of the $\mathbb{F}_q$-vector space R/I is finite, i.e. $\dim_{\mathbb{F}_q}(R/I) < \infty$.

We have the following results about the relationship between the number of solutions of a zero-dimensional ideal and the dimension of the corresponding vector space determined by this ideal. The following proposition gives a bound on the number of solutions over algebraic closure of a zero-dimensional ideal and when this bound can be achieved.

Proposition 3.0.10. *For a zero-dimensional ideal $I = \langle f_1, ..., f_m \rangle \subseteq \mathbb{F}_q[X_1, ..., X_n]$ with*

$f_1(X_1, ..., X_n), ..., f_m(X_1, ..., X_n) \in \mathbb{F}_q[X_1, ..., X_n]$, *the following hold:*

(a) *the number of solutions of* I *in* $\overline{\mathbb{F}_q}$ *is bounded by* $\dim_{\mathbb{F}_q}(R/I)$, *i.e. the polynomial system*

$$f_1(X_1, ..., X_n) = ... = f_m(X_1, ..., X_n) = 0$$

has at most $\dim_{\mathbb{F}_q}(R/I)$ *solutions in* $\overline{\mathbb{F}_q}^n$.

(b) *if* I *is a radical ideal of* $\mathbb{F}_q[X_1, ..., X_n]$, *then the number of solutions of* I *in* $\overline{\mathbb{F}_q}$ *equal to* $\dim_{\mathbb{F}_q}(R/I)$.

Proof. (a) See [32], Proposition 3.7.5 of Chapter 3.

(b) See [32], Theorem 3.7.19 of Chapter 3. □

3.1 Gröbner basis method for solving polynomial systems

In this section, we first introduce some basic concepts and then review Gröbner basis method for solving polynomial systems. Gröbner basis remains one of the most effective approaches to solve polynomial equations.

3.1.1 Gröbner basis

The concept of Gröbner basis was introduced by Buchberger [6]. It is a special basis of an ideal and has some good properties. It is a useful tool to solve polynomial systems.

Definition 3.1.1 (Monomial ordering)**.** A monomial ordering $\leq$ on $R = \mathbb{F}_q[X_1, ..., X_n]$ is a total ordering on the set of monomials of R satisfying:

1. $1 \leq u$, for all monomials $u \in R$.
2. $\leq$ is compatible with multiplication, i.e. if $u \leq v$ then $uw \leq vw$, for all monomials $u, v, w \in R$.

3. $\leq$ is well-ordering, i.e. every nonempty subset of monomials of R has the smallest element with respect to $\leq$.

A monomial ordering $\leq$ on R is called graded if in addition to the above, one has:

$$\deg(u) < \deg(v) \implies u \leq v,$$

where deg is the total degree of the monomial.

The most commonly used monomial orderings are the lexicographical ordering (lex for short), the degree lexicographical ordering (dlex for short) and the degree reverse lexicographical ordering (drl for short). See [10] for the definition of the orderings. In particular, dlex and drl orderings are graded orderings.

To a polynomial $f = \sum_{\alpha \in \mathbb{Z}^n_{\geq 0}} c_\alpha X^\alpha \in R$ (where we use $X^\alpha = X_1^{\alpha_1} ... X_n^{\alpha_n}$ for $\alpha = (\alpha_1, ..., \alpha_n)$), define the leading term of f to be

$$\mathrm{LT}(f) = c_\beta X^\beta,$$

and the leading monomial of f to be

$$\mathrm{LM}(f) = X^\beta,$$

where X^β is the maximal monomial of f with respect to $\leq$ such that $c_\beta \neq 0$.

In addition, we set $\mathrm{LT}(0) = 0$.

Let F be a finite subset of R. Let $\mathrm{LT}(F) = \{\mathrm{LT}(f) : f \in F\}$.

Let $I \subseteq R$ be an ideal. We let

$$\mathrm{LT}(I) = \langle \mathrm{LT}(f) : f \in I \rangle \subseteq R$$

be the leading term ideal of I, that is, the ideal generated by the leading terms of all polynomials

in I.

Definition 3.1.2 (Gröbner basis)**.** Let $I \subseteq R = \mathbb{F}_q[X_1, ..., X_n]$ be an ideal and $\leq$ be a monomial ordering on R. A Gröbner basis of the ideal I w.r.t the monomial order $\leq$ is a finite set $G = \{g_1, ..., g_t\} \subseteq I$ satisfying:

$$\mathrm{LT}(I) = \langle \mathrm{LT}(g) : g \in G \rangle.$$

Definition 3.1.3 (Reduced Gröbner basis)**.** A reduced Gröbner basis of I is a Gröbner basis G of I such that:

1. The leading term of each $g \in G$ has coefficient 1.

2. No monomial of g lies in $\langle \mathrm{LT}(g') : g' \in G \setminus \{g\} \rangle$,for all $g \in G$.

Each ideal has a unique reduced Gröbner basis. For a proof of this fact, see [10], Proposition 6 of Chapter 2, §7.

In the following, we will review some algorithms for computing a Gröbner basis. Some basic concepts are necessary before introducing these algorithms.

Definition 3.1.4. Let K be a field. Let $R = K[X_1, ..., X_n]$, $f, r \in R$ with $f \neq 0$ and let $P = \{p_1, ..., p_s\} \subseteq R$ be a finite set. Fix a monomial order $\leq$ on R. Then we say that f reduces to r modulo P, written

$$f \longrightarrow_P r,$$

if there exist $a_1, ..., a_s \in R$ such that

$$f = \sum_{i=1}^{s} a_i p_i + r,$$

and

$$\mathrm{LT}(a_i p_i) \leq \mathrm{LT}(f),$$

whenever $a_i p_i \neq 0$, $i = 1, ..., s$.

See [10], Definition 1 of Chapter 2,§9, where the definition is for $r = 0$.

Definition 3.1.5 (Multivariate polynomial division)**.** Let K be a field, $R = K[X_1, ..., X_n]$, and fix a monomial order $\leq$ on R. Let $F = (f_1, ..., f_s)$ be an ordered finite subset of R. Then every $f \in R$ can be written as

$$f = \sum_{i=1}^{s} a_i f_i + r,$$

with some $a_i, r \in R, i = 1, ..., s$ satisfying one of the following:

1. $r = 0$; or

2. None of the terms that appears in r is divisible by $\mathrm{LT}(f_i), i = 1, ..., s$, and $\mathrm{LT}(a_i f_i) \leq \mathrm{LT}(f)$ if $a_i f_i \neq 0$.

We call r the remainder of f on division by F, written as $r = \overline{f}^F$.

Note that r usually depends on the order of the elements in F. Specifically, different orderings of the elements may result in different remainders. An ordering of the polynomials is called a reductor. Further, it is clear that for an ordered set $G = (g_1, ..., g_s) \subseteq R$ and $f, r \in R$, $\overline{f}^G = r$ implies that $f \longrightarrow_G r$. However, the converse is false.

Definition 3.1.6 (S-polynomial)**.** Let $f, g \in R = K[X_1, ..., X_n] \backslash \{0\}$. Let $t = \mathrm{lcm}(\mathrm{LM}(f), \mathrm{LM}(g))$, where lcm denotes the least common multiple. Then the S-polynomial of f and g is defined as

follows:

$$S(f,g) = \frac{t}{\mathrm{LT}(f)}f - \frac{t}{\mathrm{LT}(g)}g.$$

Using S-polynomials, we have the following criterion to check whether a given finite set of polynomials is a Gröbner basis of an ideal.

Theorem 3.1.7 (Buchberger's Criterion)**.** *Let $I \neq \{0\}$ be an ideal of $R = K[X_1, ..., X_n]$. Then $G = \{g_1, ..., g_t\} \subseteq I$ is a Gröbner basis of I for a monomial ordering on R if and only if $\overline{S(g_i, g_j)}^G = 0$, for $i \neq j \in \{1, ..., t\}$.*

Proof. See [10], Theorem 6 of Chapter 2, §6 □

We remark that in the above theorem, if we replace $\overline{S(g_i, g_j)}^G = 0$ by $S(g_i, g_j) \longrightarrow_G 0$, the theorem is still correct (see [10], Theorem 3 of Chapter 2, §9 for the proof).

Based on Buchberger's Criterion, we have the following Buchberger's algorithm to compute a Gröbner basis of a given ideal.

Algorithm 3.1.8 (Buchberger's algorithm)**.** *Input: $F = \{f_1, ..., f_s\} \subseteq R = K[X_1, ..., X_n] \setminus \{0\}$*

Output: a Gröbner basis of $I = \langle f_1, ..., f_s \rangle$

1 . $G := F$

2 . REPEAT

3 . $G' := G$

4 . FOR each pair $\{p, q\}$, $p \neq q$ in G' DO

5 . $S := \overline{S(p,q)}^{G'}$

6 . IF $S \neq 0$ THEN $G := G \cup \{S\}$

7 . UNTIL $G = G'$

This version of Buchberger's algorithm indeed computes a Gröbner basis of an ideal, but it is not efficient in practice. Let's explore this algorithm in greater detail. The algorithm begins with reducing S-polynomials of all pairs of the generators of the ideal. If the remainder is nonzero, then append this remainder to the generating set to form a new generating set and then repeat this process until all S-polynomials reduce to zero. Notice that the most time consuming part of the algorithm is the reduction of S-polynomials. In a concrete implementation of this algorithm, one has to make some choices at different stages: First, one has to decide on how to select a pair of the generators(some references call critical pair) to form an S-polynomial of this pair. Second, one needs to choose a reductor when performing the reduction of the S-polynomial to obtain a remainder, where a reductor means the order of the polynomials when carrying out the multivariate polynomial division. These two choices have a significant influence on the efficiency of the whole algorithm, since some choices of the order of the S-polynomials and reductors may result in drastically more numbers of reductions of S-polynomials and thus leading to a huge time complexity of the algorithm.

Many other algorithms have been proposed to improve Buchberger's algorithm. Essentially, these algorithms follow two main approaches. In practice, many pairs of polynomials may result in S-polynomials that reduce to 0. Thus, one method is to find a strategy to predict these S-polynomials that reduce to 0 without explicitly computing them. Another approach is to improve the time to perform reduction or the method of doing reduction. In the following, we will briefly introduce two algorithms F4 [15] and F5 [16] which are based on Buchberger's algorithm. Before introducing these two algorithms, we first describe the relation between linear algebra and polynomial division.

3.1.2 Gaussian elimination and polynomial division

We begin by describing how to construct a matrix from a set of polynomials. In other words, given a set of polynomials, we can construct a matrix corresponding to this set of polynomials.

Let $F = (f_1, ..., f_s)$ be an ordered finite subset of $R = K[X_1, ..., X_n]$ with K a field, and let $M_{\leq}(F)$ be the set of monomials appearing as terms of all f_i in F(For simplicity, in the remaining of this thesis when we say the set of monomials in a set of polynomials we mean the set of monomials appearing as terms of all polynomials in that set). Sort all the elements in $M_{\leq}(F)$ in descending order with respect to the monomial ordering $\leq$. Let $t = \#M_{\leq}(F)$ and m_i be the i-th element of $M_{\leq}(F)$. Write $f_i = \sum_{\lambda=1}^{t} a_{i\lambda} m_\lambda$, for all $1 \leq i \leq s$. The matrix M_F corresponding to F is defined to be:

$$M_F = (a_{ij})_{1 \leq i \leq s, 1 \leq j \leq t}$$

For example, let $F = (f_1, f_2, f_3)$,where $f_1 = x^3 - 2xy + 1, f_2 = x^2y + 2y^2 + y, f_3 = x^2 + y$ in $\mathbb{F}_5[x, y]$ with lexicographic ordering such that $x > y$. Then the matrix M_F corresponding to F is

$$M_F = \begin{array}{c} \begin{array}{ccccccc} x^3 & x^2y & x^2 & xy & y^2 & y & 1 \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & -2 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 2 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 \end{pmatrix} \end{array}$$

.

Such matrices constructed from several polynomials is useful. In the following, we will illustrate how to do polynomial division by doing Gaussian elimination on a matrix constructed from several polynomials.

Let $f \in R = K[X_1, ..., X_n]$ and $\mathcal{F} = (f_1, ..., f_s)$ be an ordered finite subset of R. Suppose that we want to do polynomial division of f by F. Our aim is to construct an appropriate matrix and perform Gaussian elimination on this matrix and read off the remainder from the reduced row echelon form. Before introducing how to construct this matrix, we first illustrate the method with a simple example.

Example 3.1.9. *Divide $f = x^2 + xy^2 + 1$ by $f_1 = xy + 1$ and $f_2 = x + 1$ using the lexicographic ordering with $x > y$ in $\mathbb{Q}[x, y]$. The multivariate polynomial division does the following steps:*

1. $f - xf_2 = xy^2 - x + 1$,
2. $(xy^2 - x + 1) - yf_1 = -x - y + 1$,
3. $(-x - y + 1) + f_2 = -y + 2$.

So the remainder is $-y + 2$ and we have $f = yf_1 + (x - 1)f_2 + (-y + 2)$.

Now let $F = (f, xf_2, yf_1, f_2)$. Then $M_{\leq}(F) = \{xy^2, x^2, x, y, 1\}$. Thus the matrix M_F corresponding to F is

$$M_F = \begin{array}{c} \begin{matrix} x^2 & xy^2 & x & y & 1 \end{matrix} \\ \begin{pmatrix} 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \end{pmatrix} \end{array} \begin{matrix} \leftarrow f \\ \leftarrow xf_2 \\ \leftarrow yf_1 \\ \leftarrow f_2 \end{matrix}$$

The reduced row echelon form of M_F is:

$$\widetilde{M_F} = \begin{pmatrix} 1 & 0 & 0 & 0 & -1 \\ 0 & 1 & 0 & 0 & 2 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & -2 \end{pmatrix}$$

The last row of $\widetilde{M_F}$ corresponds to the polynomial $y - 2$, which is, up to multiplication by a nonzero constant, equal to the remainder. Thus doing Gaussian elimination on the matrix M_F does nearly the same as doing polynomial division, with the minor difference being that we can only read off the remainder up to multiplication by a nonzero constant from the reduced

row echelon form. However, this is enough, since we only need to know the remainder up to multiplication by a nonzero constant in the computation of Gröbner basis.

From the above example, we see that in order to obtain the remainder of dividing f by $\mathcal{F}$, we can construct the matrix corresponding to a set of polynomials F containing f, as well as polynomials of the form $m_i f_i$,where m_i is a monomial. It remains to determine these monomials. This problem can be solved through exploring the polynomial division algorithm [10]. In each step of the algorithm, the algorithm looks for the smallest i such that $\mathrm{LT}(f_i)$ divides $\mathrm{LT}(h)$, where h is the intermediate result. If such an f_i exists, one changes the intermediate result h to $h - q_i f_i$ with $q_i = \frac{\mathrm{LT}(h)}{\mathrm{LT}(f_i)}$. So it is natural to add the polynomial $\mathrm{LM}(q_i)f_i$ to F. However, we do not know which f_i will be chosen in each step apriori without doing the polynomial division algorithm. To predict this f_i chosen in each step, observe that the exact value of each intermediate result is not critical, since we only care about the monomials appearing in the intermediate result. As such, it is sufficient to assume that the intermediate result $h - q_i f_i$ contains all the monomials of h and $q_i f_i$ except for the leading term $\mathrm{LT}(h)$ as it is cancelled in the subtraction, where f_i and q_i are determined by the previous step. Using this assumption, we can predict all the monomials m_i needed to construct F possibly with some redundant monomials. In practice, the number of this redundancy is small. In general, we have the following algorithm to construct the matrix.

Algorithm 3.1.10. *Input: f, an ordered set $\mathcal{F} = (f_1, ..., f_s) \subseteq R = K[X_1, ..., X_n]$, a monomial ordering $\leq$ on R;*

Output: matrix M_F corresponding to a set of polynomials F such that the remainder $\overline{f}^F$ (up to multiplication by a nonzero constant) can be read off from the reduced row echelon form of M_F

1 . $F := \{f\}$, $D := M_{\leq}(F)$, where $M_{\leq}(F)$ denotes the ordered set of monomials in F sorted in descending order with respect to the monomial ordering $\leq$.

2 . REPEAT

3 . $D := D \setminus \{D[0]\}$, where $D[0]$ is the first element of the ordered set D.

4 . Look for the smallest i such that $\mathrm{LM}(f_i)$ divides $D[0]$. If such an i exists, let $q := \frac{D[0]}{\mathrm{LM}(f_i)}$.

5 . $F := F \cup \{qf_i\}$.

6 . $D := D \cup M(qf_i) \setminus \{D[0]\}$ sorted in descending order w.r.t $\leq$, where $M(qf_i)$ is the set of monomials of qf_i.

7 . UNTIL D is empty.

8 . Construct M_F according to F.

Doing Gaussian elimination on M_F in the above algorithm, we can read off the remainder $\overline{f}^F$(up to multiplication by a nonzero constant) from the reduced row echelon form $\widetilde{M_F}$. There are two cases, namely

1. $\overline{f}^F = 0$: this corresponds to a zero row of $\widetilde{M_F}$,
2. $\overline{f}^F \neq 0$: then $\mathrm{LT}(\overline{f}^F)$ cannot be divided by any $\mathrm{LT}(f_i)$,$i = 1, ..., s$, and thus this polynomial corresponds to a row of $\widetilde{M_F}$ having a pivot which is not a pivot in M_F. Finding the row in $\widetilde{M_F}$ with such a property gives the remainder $\overline{f}^F$(up to multiplication by a nonzero constant).

Note that the above algorithm can be easily modified to construct a matrix M such that performing Gaussian elimination on M gives us the remainders of several polynomials $g_1, g_2, ..., g_t$ divided by $\mathcal{F} = (f_1, ..., f_s)$ through the reduced row echelon form. In fact, the F4 algorithm uses this idea to reduce several S-polynomials simultaneously.

3.1.3 F4 and F5 algorithm

F4 algorithm is an enhanced version of Buchberger's algorithm with respect to efficiency. This algorithm introduces two main changes compared to Buchberger's algorithm: one of which is to select several critical pairs(see 3.1.11 for the definition) at each step instead of only one

pair in Buchberger's algorithm, while the other is to use Gaussian elimination on appropriate matrices to replace the reduction of S-polynomials through multivariate polynomial division. Before describing the F4 algorithm, we give some definitions needed in the algorithm.

Definition 3.1.11 (Critical pair)**.** Let $f, g \in R = K[X_1, ..., X_n] \backslash \{0\}$. Let $t = \text{lcm}(\text{LM}(f), \text{LM}(g))$, where lcm denotes the least common multiple. The critical pair of f and g is defined as

$$Pair(f, g) := (t, \frac{t}{\text{LM}(f)}, f, \frac{t}{\text{LM}(g)}, g).$$

Further, we define

$$Left(Pair(f, g)) := (\frac{t}{\text{LM}(f)}, f), Right(Pair(f, g)) := (\frac{t}{\text{LM}(g)}, g)$$

The following is a basic version of the F4 algorithm.

Algorithm 3.1.12 (Algorithm F4)**.** *Input:* $F = \{f_1, ..., f_s\} \subseteq R = K[X_1, ..., X_n] \setminus \{0\}$*;*

Output: G, a Gröbner basis of $I = \langle f_1, ..., f_s \rangle$

1 . $G := F$, $P := \{Pair(f_i, f_j) | 1 \leq i < j \leq s\}$, *d:=0.*

2 . REPEAT

3 . $d := d + 1$.

4 . $P_d := Select(P)$, *where* $Select(P)$ *is any function such that* $Select(P)$ *is not empty.*

5 . $P := P \setminus P_d$, $L_d := Left(P_d) \cup Right(P_d)$.

6 . $\widetilde{F_d}^+ := Reduction(L_d, G)$.

7 . For $h \in \widetilde{F_d}^+$ *do*

8 . $P := P \cup \{Pair(h, g) | g \in G\}$, $G := G \cup \{h\}$.

9 . UNTIL P is empty.

10 . Return G.

The Select function in step 4 is the selection strategy to compute S-polynomials. If the size of $Select(P)$ is always 1, then this algorithm is the Buchberger's algorithm. If the size of $Select(P)$ is larger than 1, then in step 6 the Reduction step reduces several S-polynomials at the same time. There are many different strategies available, see [15] for the details.

The Reduction in step 6 is the following subalgorithm

Algorithm 3.1.13 (Reduction)**.** *Input: L, a finite subset of $M \times R$, where M is the set of monomials in R,*

G, a finite subset of R

Output: a finite subset of R

1 . $F :=$Symbolic Preprocessing(L, G).

2 . Construct matrix M_F corresponding to F using the method in 3.1.2 and do Gaussian elimination on M_F, $\widetilde{M_F} :=$ Reduced row echelon form of M_F.

3 . $\widetilde{F} := \{f | f$ corresponds to nonzero row of $\widetilde{M_F}\}$.

4 . $\widetilde{F}^+ := \{f \in \widetilde{F} | \mathrm{LT}(f) \notin \mathrm{LT}(F)\}$.

5 . Return $\widetilde{F}^+$

In step 1 of subalgorithm Reduction, there is a Symbolic Preprocessing subalgorithm. The aim of Symbolic Preprocessing is to prepare to construct a matrix to read off remainders from the row echelon form of this matrix. This algorithm is the same as algorithm3.1.10. Specifically, it is as follows.

Algorithm 3.1.14 (Symbolic Preprocessing)**.** *Input: L, a finite subset of $M \times R$, where M is the set of monomials in R*

G, a finite subset of R

Output: a finite subset of R

1 . $F := \{tf | (t, f) \in L\}$, $D := \mathrm{LT}(F)$.

2 . REPEAT

3 . Choose an element $m \in M(F) \setminus D$*, where* $M(F)$ *is the set of all monomials in* F*,* $D := D \cup \{m\}$*.*

4 . If there exists $g \in G$ *such that* $\mathrm{LM}(g)$ *divides* m*, then* $m_1 := \frac{m}{\mathrm{LM}(g)}$*,* $F := F \cup \{m_1 * g\}$*.*

5 . UNTIL $M(F) = D$*.*

6 . Return F*.*

In this section, we have only described the basic version of F4 algorithm. For the improved version, see [15].

The F5 algorithm is also based on Buchberger's algorithm. This algorithm aims to reduce useless computations: namely to remove S-polynomials with zero reduction. It introduces a new powerful criterion to detect useless critical pairs and computes Gröbner basis incrementally. F5 algorithm is a very efficient algorithm in practice. For example , a Gröbner basis of cyclic 10 was first computed by the F5 algorithm. Also, in the case of regular sequence, it was proved that useless critical pairs are avoided in F5 algorithm, i.e. there is no useless critical pair generated in F5 algorithm. For details, see [16].

3.1.4 Finding solutions via Gröbner basis

In the previous subsections, we have described how one can compute Gröbner basis of a polynomial system. We now discuss how one can compute the solutions of the polynomial system from a Gröbner basis with respect to some appropriate monomial ordering.

Gröbner basis method for solving multivariate polynomial system is based on the following proposition.

Proposition 3.1.15 (The triangle form)**.** *For a zero-dimensional ideal* $I \subseteq R = \mathbb{F}_q[X_1, ..., X_n]$*, let* $G = \{g_1, ..., g_t\}$ *be the reduced Gröbner basis with respect to lex order with* $X_1 > X_2 > ... >$

X_n. Order the elements of G by $\mathrm{LT}(g_1) > \mathrm{LT}(g_2) > ... > \mathrm{LT}(g_t)$. Then for all $i \in \{1, ..., n\}$, there exists $j = j_i \in \{1, ..., t\}$ such that $\mathrm{LT}(g_j) = X_i^{d_i}$ for some integer $d_i > 0$.

Proof. See [10], Theorem 6 of Chapter 5,§3. □

This proposition shows that the Gröbner basis of a zero-dimensional ideal w.r.t lexicographic order can be used to solve problem(3.0.8) in the following way:

1. Compute the reduced Gröbner basis G of the ideal $I = \langle f_1, ..., f_m \rangle$ w.r.t. lex order with $X_1 > X_2 > ... > X_n$.

2. Take a univariate polynomial in X_n from G and solve it using Berlekamp's algorithm [51]. Note that the existence of such univariate polynomial is ensured by I is a zero-dimensional ideal, see proposition(3.1.15).

3. Use back substitution to get all the values of $X_1, ..., X_{n-1}$ through the triangle form of G.

We remark that the degrees of Gröbner basis w.r.t lexicographic order can be very large, and thus computing such a Gröbner basis can lead to large computational complexity. On the other hand, the degree reverse lexicographic order(denoted DRL) has been proven to be the fastest monomial order for computing Gröbner basis of an ideal [4]. Therefore, in practice, the method of Gröbner basis to solve problem(3.0.8) typically consists of two steps. First, compute the Gröbner basis of the ideal $I = \langle f_1, ..., f_m \rangle$ w.r.t. DRL using F4 or F5 algorithms [15] [16]. Second,use the FGLM algorithm [18] to change the Gröbner basis wrt DRL to a Gröbner basis wrt lex order and then using the triangle form of lex Gröbner basis to get the solutions of I.

Finally, we summarize the complexity of this approach. F4 and F5 are efficient algorithms to compute Gröbner basis. As previously discussed, these two algorithms primarily perform linear algebra on a Macaulay matrix.

Definition 3.1.16 (Macaulay matrix)**.** Let d be a positive integer and let $F = \{f_1, ..., f_l\} \subseteq R = \mathbb{F}_q[X_1, ..., X_n]$ with $\deg(f_i) \leq d$,for $i = 1, ..., l$. Let $T \subseteq R$ be all the monomials of degree

less than or equal to d sorted w.r.t a monomial ordering. The Macaulay matrix M_d is defined as follows. Consider all the polynomials $t_j f_i$ of $\deg(t_j f_i) \leq d$ with $t_j \in T$ and $f_i \in F$. The rows of M_d correspond to the coefficients of $t_j f_i$ written as a linear combination of elements in T and the columns correspond to the sorted monomials. Here d is called the degree of the Macaulay matrix M_d.

Essentially, the F4 and F5 algorithms successively construct a Macaulay matrix M_d for increasing d and do Gaussian elimination on this matrix until a Gröbner basis is found. Lazard [33] proved the following theorem:

Theorem 3.1.17. *Let $F = \{f_1, ..., f_l\} \subseteq R$. There exists a positive integer D such that performing Gaussian elimination on Macaulay matrices of F with degree $1, 2, ..., D$ will compute a Gröbner basis of the ideal I which is generated by F.*

Consequently, the complexity of these two algorithms is determined by the complexity of Gaussian elimination. In particular, the maximal degree D in the computation determines the complexity of the algorithms. Specifically, the maximal degree D in the computation using F4 or F5 algorithms under the degree reverse lexicographic order is called the degree of regularity and we denote it as d_{reg}. The complexity of these two algorithm is as follows.

Theorem 3.1.18. [**[17]**] *Let I be a zero-dimensional ideal with n variables and d_{reg} be the degree of regularity of I. Then the F4 and F5 Gröbner basis algorithms need*

$$O\left(\binom{n + d_{reg} - 1}{d_{reg}}^{\omega}\right)$$

field operations, where ω is the linear algebra constant.

The complexity of FGLM algorithm is as follows.

Theorem 3.1.19 ([18])**.** *Let I be a zero-dimensional ideal with n variables and G be a Gröbner basis for I with respect to DRL. Let Ω be the number of solutions of I in an algebraic closure.*

Then computing a Gröbner basis G_{lex} with respect to a lexicographic ordering using FGLM algorithm from G needs

$$O\left(n\Omega^3\right)$$

field operations.

Remark 3.1.20. It remains an open problem to determine the degree of regularity of a random zero-dimensional polynomial system. For a regular sequence $F = (f_1, ..., f_s)$(see [17] for definition), the degree of regularity of the polynomial system defined by F is bounded by its Macaulay bound:$\sum_{i=1}^{s}(\deg(f_i) - 1) + 1$([17]). And the last degree $d_{\mathcal{F}}$(see definition(5.1.5)) is also bounded by this Macaulay bound(see remark(5.1.10 for detail).

3.2 The XL algorithm

In this section, we review the method of using eXtended Linearization(XL) algorithm to solve multivariate polynomial systems. In [9], the authors introduce XL algorithm to solve polynomial equations. According to [9], the algorithm is described as follows:

Definition 3.2.1 (XL algorithm)**.** Let $F = \{f_1, ..., f_s\} \subseteq R = K[X_1, ..., X_n] \setminus \{0\}$. The XL algorithm is designed to solve the polynomial system defined by F. For a positive integer D, the procedure of XL algorithm at degree D is the following:

1. **Multiply:** Generate all polynomials of the form $m_i f_j$ with $\deg(m_i f_j) \leq D$, where m_i is a monomial.

2. **Linearize:** Consider each monomial of degree $\leq D$ as a new variable and do Gaussian elimination on the equations obtained in 1. The ordering on the monomials must be such that all the terms containing one specific variable (say X_1) are eliminated last.

3. **Solve:** Assume that step 2 yields at least one univariate equation in the powers of X_1. Solve this equation over the finite fields (e.g. with Berlekamp's algorithm).

4. **Repeat:** Simplify the equations and repeat the process to find the values of the other variables.

Observe that in order to perform Step 2 in the XL algorithm, one must have the number of equations at least as large as the number of monomials involved. As such, one must select a sufficiently big D such that this condition holds. The XL algorithm has many variants. For more details of the variants, see [53].

In [3] and [47], the authors showed that XL algorithm is in fact a less efficient version of the F4 algorithm. Indeed, notice that the XL algorithm seeks to construct a large enough Macaulay matrix that allows one to find the solutions by one Gaussian elimination operation. On the other hand, the main ideas of the F4 algorithm are to progressively construct Macaulay matrices of increasing sizes, performing Gaussian elimination at each stage, until one finds the solutions. This notion was similarly employed in [13] which proposed a variant of the XL algorithm known as the MutantXL algorithm. In the following, we briefly review the concept of mutants and the mutantXL algorithm.

Definition 3.2.2 (Mutant)**.** Let $I = \langle f_1, ..., f_m \rangle \subseteq R = K[X_1, ..., X_n]$ and $f \in I$. Write f as

$$f = \sum_{i=1}^{m} g_i f_i,$$

where $g_i \in R$,$i = 1, ..., m$. The level of this specific representation of f is defined as follows:

$$\max\{\deg(g_i f_i) : i = 1, ..., m\}.$$

Then, the level of f is defined to be the minimum level of all the different representations of f. The polynomial f is called a mutant with respect to $\{f_1, ..., f_m\}$ if $\deg(f)$ is less than the level

of f.

Intuitively, a mutant f is a polynomial that is formed by linear combinations of the generating polynomials $g_i f_i$ such that the leading terms of some of the $g_i f_i$ cancel out. Based on this concept, [13] introduced the mutantXL algorithm which improves the XL algorithm.

For a parameter D, the MutantXL algorithm carries out steps 1 and 2 of 3.2.1. Then mutantXL searches for univariate polynomials in the same way as step 3 of 3.2.1. If no univariate polynomial exists, then mutantXL searches for mutants which are new polynomials of degree less than D constructed from the generating polynomials. If mutants are found at this stage, these mutants are multiplied by monomials as in step 1 such that the new polynomials have degree bounded by D and added to the system. This is known as the mutant strategy. The algorithm then proceeds in the same way as in 3.2.1. This is the main idea underlying the mutantXL algorithm at degree D. Essentially, MutantXL algorithm seeks to enlarge the polynomial system without incrementing D through adopting the mutant strategy. Thus this algorithm is more efficient than the original XL algorithm. Following this first version, other variants of the MutantXL algorithm were subsequently proposed, namely the MXL2 and MXL3 algorithms (see [36] and [35] for details of these two algorithms). However, the article [2] shows mutant strategy is equivalent to some selection strategy used in Gröbner basis algorithm such as F4 and as a result, all the MXL family algorithms are redundant variants of the F4 algorithm.

4. THE INDEX CALCULUS METHOD FOR ECDLP – DEVELOPMENTS AND PROGRESS

In this chapter, we will survey the developments on the index calculus method for the elliptic curve discrete logarithm problem. Specifically, we will present the key ideas and results contained in the following research papers [11, 12, 21, 22, 39, 43].

4.1 Gaudry and Diem's Results

4.1.1 Gaudry's Result

In [22], Gaudry focused on the elliptic curve discrete logarithm problem defined over $\mathbb{F}_{q^n}$ with q prime or a prime power and n is small. In particular, he solved ECDLP defined over $\mathbb{F}_{q^3}$ in heuristic asymptotic running time $\tilde{O}(q^{4/3})$, where $\tilde{O}$ means that there exists an constant c such that $\tilde{O}(q^{4/3}) = O(q^{4/3}\log^c(q^{4/3}))$.

Gaudry's proposed factor base:

The factor base used in [22] is $\mathcal{F} = \{(x, y) \in E(\mathbb{F}_{q^n}) | x \in \mathbb{F}_q\}$. The size of $\mathcal{F}$ is roughly equal to q on a heuristic assumption.

In the sieving stage, we wish to compute $\#\mathcal{F} + 1$ relations, where a relation is expressing $aP + bQ$ with a sum of n elements in $\mathcal{F}$ for randomly chosen integers a and b. Gaudry used the summation polynomial S_{n+1} to find such a relation. Gaudry heuristically argued that the probability of finding one relation is $1/n!$.

Further, by using the "large primes" trick introduced in Thériault's algoritm [49] and its

variants [23], Gaudry obtained the following heuristic complexity result:

Heuristic result 4.1.1. *[22] Let $n \in \mathbb{Z}_{\geq 2}$ be fixed and q prime or a prime power that goes to infinity. Then the discrete logarithm problem on any elliptic curve defined over $\mathbb{F}_{q^n}$ can be solved by a randomized algorithm in heuristic asymptotic running time $\tilde{O}(q^{2-2/n})$, where the constant in $\tilde{O}()$ depends on n.*

Note that the above result applied only to small n, since the constant hidden in the $\tilde{O}(q^{2-2/n})$ is exponential in n which grows very fast with n.

4.1.2 Diem's Results

Extending the factor base

In [11], Claus Diem considered a similar factor base as in [22]. He showed that there exist a class of elliptic curves defined over finite fields such that index calculus method takes subexponential time to solve the discrete logarithm problem on these elliptic curves. We highlight the main results in [11] as follows.

The main result of Claus Diem's paper [11] is the following theorem.

Theorem 4.1.2. *The elliptic curve discrete logarithm problem over finite fields $\mathbb{F}_{q^n}$ can be solved in an expected time of*

$$e^{O(\max\{\log(q), n^2\})}.$$

Following this theorem, Diem derived the following results:

(i) Let sequences of prime powers $(q_i)_{i\in\mathbb{N}}$ and natural numbers $(n_i)_{i\in\mathbb{N}}$ with $n_i \to \infty$ and $\frac{n_i}{\log(q_i)} \to 0$ for $i \to \infty$ be given. Then the elliptic curve discrete logarithm problem over finite fields $\mathbb{F}_{q_i^{n_i}}$ can be solved in an expected time of $(q_i^{n_i})^{o(1)}$.

(ii) Let $\beta \in [\frac{1}{2}, 1)$ and $a, b > 0$ be fixed. Let $\alpha := \frac{1-\beta}{2\beta} \in (0, \frac{1}{2}]$, $\gamma := \frac{2\beta}{\beta+1} < 1$ and $n \in \mathbb{Z}_{\geq 0}$

such that

$$a \cdot \log(q)^{\alpha} \leq n \leq b \cdot \log(q)^{\beta}.$$

Then the elliptic curve discrete logarithm problem over finite fields $\mathbb{F}_{q^n}$ can be solved in an expected time of $e^{O((\log q^n)^{\gamma})}$.

(iii) Let positive real numbers $a < b$ be fixed and $n \in \mathbb{Z}_{\geq 0}$ such that

$$a \cdot \sqrt{\log(q)} \leq n \leq b \cdot \sqrt{\log(q)}.$$

Then the elliptic curve discrete logarithm problem over finite fields $\mathbb{F}_{q^n}$ can be solved in an expected time of $e^{O((\log q^n)^{2/3})}$.

In his subsequent work in [12], Claus Diem extended his work in [11] by considering a more generalized factor base. Let $1 < m < n' < n$ be positive integers such that $mn' \approx n$. Let V be a subspace of $\mathbb{F}_{q^n}$ over $\mathbb{F}_q$ of dimension n'.

In this case, Diem considered this factor base $\mathcal{F} = \{(x, y) \in E(\mathbb{F}_{q^n}) | x \in V\}$.

Observe that when $n' = 1$, we obtain the preceding factor base. It is heuristically assume that $|\mathcal{F}| \approx q^{n'}$. With this extended factor base and by employing results of algebraic geometry, Diem established the following stronger result.

Theorem 4.1.3. *The elliptic curve discrete logarithm problem over finite fields $\mathbb{F}_{q^n}$ can be solved in an expected time of*

$$e^{O\left(\max\{\log(q), n\cdot(\log q)^{1/2}, n^{3/2}\}\right)}.$$

Furthermore, if q is even, then the time complexity is

$$e^{O\left(\max\{\log q, n\cdot\log(q)^{1/2}, n\cdot(\log n)^{1/2}\}\right)}.$$

Theorem 4.1.3 leads to the following results.

(i) Let sequences of primes $(q_i)_{i\in\mathbb{N}}$ and natural numbers $(n_i)_{i\in\mathbb{N}}$ with $q_i \to \infty$ and $n_i \to \infty$ for $i \to \infty$ be given. Suppose we have the following additional conditions:

(a) $\frac{n_i}{\log(q_i)^2} \to 0$ for $i \to \infty$

or

(b) q_i is even for all i and $\frac{\log(n_i)}{\log(q_i)^2} \to 0$ for $i \to \infty$,

then the elliptic curve discrete logarithm problem over finite fields $\mathbb{F}_{q_i^{n_i}}$ can be solved in an expected time of $(q_i^{n_i})^{o(1)}$.

(ii) Let $\beta \geq \frac{1}{2}$ and $a, b > 0$ be fixed. Let q be even, $\alpha := \frac{1}{2\beta+1}$, $\gamma := 1 - \frac{1}{2}\frac{1}{\beta+1}$ and $n \in \mathbb{Z}_{\geq 0}$ such that

$$a \cdot \log(q)^{\alpha} \leq n \leq b \cdot \log(q)^{\beta} \tag{4.1}$$

Then the elliptic curve discrete logarithm problem over finite fields $\mathbb{F}_{q^n}$ can be solved in an expected time of $e^{O((\log q^n)^{\gamma})}$. Furthermore, if $\beta \leq 1$, then the same holds over all finite fields $\mathbb{F}_{q^n}$ with n satisfying (4.1).

A special case is the following: For $a, b > 0$ and $n \in \mathbb{Z}_{\geq 0}$ such that

$$a \cdot \log(q)^{1/3} \leq n \leq b \cdot \log(q).$$

Then the elliptic curve discrete logarithm problem over finite fields $\mathbb{F}_{q^n}$ can be solved in an expected time of $e^{O((\log q^n)^{3/4})}$.

(iii) Let $\beta \in [1, 2)$ and $a, b > 0$ be fixed. Let $\alpha := \frac{2-\beta}{3\beta}$, $\gamma := \frac{3}{2}\frac{\beta}{\beta+1}$ and $n \in \mathbb{Z}_{\geq 0}$ such that

$$a \cdot \log(q)^{\alpha} \leq n \leq b \cdot \log(q)^{\beta}.$$

Then the elliptic curve discrete logarithm problem over finite fields $\mathbb{F}_{q^n}$ can be solved in an expected time of $e^{O((\log q^n)^{\gamma})}$.

Remark 4.1.4. In this thesis, we have described a more simplified version of the factor bases proposed by both Gaudry and Diem [11, 12, 22]. This is to provide a more readable exposition of their work which contains the flavour of the general approach.

4.2 Solving the summation polynomials using Weil descent

So far, we have seen that in the case of $\mathbb{F}_{q^n}$, a possible factor base is $\mathcal{F} = \{(x, y) \in E(\mathbb{F}_{q^n}) |\ x \in V\}$ for some appropriate vector subspace V of $\mathbb{F}_{q^n}$. In the sieving step, our goal is to write a point as a sum of some of the points in $\mathcal{F}$. This can be done with the aid of summation polynomials as we proceed to show.

Suppose that we wish to write each point R as a sum of m points in the factor base so that we are dealing with the summation polynomial S_{m+1}. Let $\{\theta_1, \ldots, \theta_n\}$ be a basis of $\mathbb{F}_{q^n}$ over $\mathbb{F}_q$ and let $\{\nu_1, \ldots, \nu_{n'}\}$ be a basis of V over $\mathbb{F}_q$. Clearly, each $\nu_i, i = 1, \ldots, n'$ is a linear combination of the $\theta_1, \ldots, \theta_n$.

Define mn' new variables $x_{ij} \in \mathbb{F}_q$ such that $x_i = \sum_{j=1}^{n'} x_{ij}\nu_j, i = 1, \ldots, m, j = 1, \ldots, n'$. Let $R = (x, y) \in E(\mathbb{F}_{q^n})$. In this case, we will solve $S_{m+1}(x_1, \ldots, x_m, x) = 0$ or equivalently, $S_{m+1}(\sum_{j=1}^{n'} x_{1j}\nu_j, \ldots, \sum_{j=1}^{n'} x_{mj}\nu_j, x) = 0$. Here, the former equation has unknowns in $\mathbb{F}_{q^n}$ while the unknowns in the latter equation are in $\mathbb{F}_q$.

Expanding everything in terms of θ_i and equating the coefficients to 0, we now obtain:

$f_1(x_{11}, \ldots, x_{mn'}) = 0, \ldots, f_n(x_{11}, \ldots, x_{mn'}) = 0$ for some $f_1, \ldots, f_n \in \mathbb{F}_q[x_{11}, \ldots, x_{mn'}]$.

Let us illustrate with a concrete example.

Example 4.2.1. *Consider the field $\mathbb{F}_{2^4}$ and let α be its generator with $\alpha^4 = \alpha + 1$. Let E denote the elliptic curve over $\mathbb{F}_{2^4}$ be defined by $y^2 + xy = x^3 + \alpha x + 1 + \alpha$. Then $(1, 0) \in E$. Let V be the subspace with basis $\{1, \alpha\}$. Since $S_3(x_1, x_2, x_3) = x_1^2x_2^2 + x_1^2x_3^2 + x_2^2x_3^2 + x_1x_2x_3 + 1 + \alpha$, $S_3(x_1, x_2, 1) = x_1^2x_2^2 + x_1^2 + x_2^2 + x_1x_2 + 1 + \alpha$.*

Let $x_1 = x_{11} + x_{12}\alpha$ and $x_2 = x_{21} + x_{22}\alpha$. Substituting for x_1 and x_2 yields $(x_{11}^2 + x_{12}^2\alpha^2)(x_{21}^2 + x_{22}^2\alpha^2) + x_{11}^2 + x_{12}^2\alpha^2 + x_{21}^2 + x_{22}^2\alpha^2 + (x_{11} + x_{12}\alpha)(x_{21} + x_{22}\alpha) + 1 + \alpha = 0$.

Since $x_{ij} \in \mathbb{F}_2, i, j = 1, 2, x_{ij}^2 = x_{ij}$. *Thus* $x_{11}x_{21} + x_{11}x_{22}\alpha^2 + x_{12}x_{21}\alpha^2 + x_{12}x_{22}(1+\alpha) + x_{11} + x_{12}\alpha^2 + x_{21} + x_{22}\alpha^2 + x_{11}x_{21} + x_{11}x_{22}\alpha + x_{12}x_{21}\alpha + x_{12}x_{22}\alpha^2 + 1 + \alpha = 0$.

$x_{12}x_{22} + x_{11} + x_{21} + 1 + (x_{12}x_{22} + x_{12} + x_{22} + x_{11}x_{22} + x_{12}x_{21} + 1)\alpha + (x_{11}x_{22} + x_{12}x_{21} + x_{12}x_{22} + x_{11} + x_{22})\alpha^2 = 0$.

In particular, we have: $f_1 = x_{12}x_{22} + x_{11} + x_{21} + 1 = 0$, $f_2 = x_{12}x_{22} + x_{12} + x_{22} + x_{11}x_{22} + x_{12}x_{21} + 1 = 0$, $f_3 = x_{11}x_{22} + x_{12}x_{21} + x_{12}x_{22} + x_{11} + x_{22} = 0$.

Here we have 3 equations in 4 unknowns.

Remark 4.2.2. (1) In general, the problem reduces to solving n polynomials equations in $mn' \approx n$ variables over $\mathbb{F}_q$.

(2) The probability that a point R can be represented as a sum of m points in $\mathcal{F}$ is $1/m!$ by a heuristic assumption.

(3) This system of equations is typically solved using Gröbner basis algorithms. On the other hand, Diem derived his results by solving these polynomial systems using a geometric method introduced by Rojas.

4.3 ECDLP over binary fields

4.3.1 The Result of Faugère et al.

In [21], Faugère et al. considered ECDLP over binary fields $\mathbb{F}_{2^n}$. Inspired by the work of Gaudry and Diem, they proposed a factor base $\mathcal{F}_V := \{(x, y) \in E(\mathbb{F}_{2^n}) | x \in V\}$, where V is a vector subspace of $\mathbb{F}_{2^n}/\mathbb{F}_2$. In particular, they focused on the sieving step. Following the approach outlined in the preceding section, they first reduced the problem of finding a relation to a problem of solving a multivariate polynomial over $\mathbb{F}_{2^n}$. Second, in order to solve this multivariate polynomial, they employed the method of Weil descent to get an equivalent boolean polynomial system. They argued that this system has a special structure and they exploited this structure to devise a linearization algorithm to tackle this polynomial system.

Finally, they deduced that the complexity of solving ECDLP over $\mathbb{F}_{2^n}$ is $O(2^{\omega t})$, where $t \approx n/2$ under some heuristic assumptions.

We now concentrate on the case where $q = 2$ and n a prime. According to Diem's estimate, the time complexity of the algorithm is $\exp(O(n(\log n)^{1/2}))$, which is worse than exhaustive search.

More precisely, let $f \in \mathbb{F}_{2^n}[x_1, \ldots, x_m]$ be a multivariate polynomial in m variables and let V be a vector subspace of $\mathbb{F}_{2^n}/\mathbb{F}_2$, we want to solve $f(x_1, x_2, \ldots, x_m) = 0$ under the linear constraints $x_1, \ldots, x_m \in V$.

First, let us examine what happens to the degree of the variables when we apply the Weil Descent to a multivariate polynomial in $\mathbb{F}_{2^n}$. Let $x_1 = x_{11}\nu_1 + \ldots + x_{1n'}\nu_{n'}$. Let $e = \sum_{i=0}^{l} e_i 2^i$ be a positive integer, $e_i = 0, 1$. We have $x_1^e = (x_{11}\nu_1 + \ldots + x_{1n'}\nu_{n'})^e = \prod_{i=0}^{l}((x_{11}\nu_1)^{e_i 2^i} + \ldots + (x_{1n'}\nu_\prime)^{e_i 2^i}) = \prod_{i=0}^{l}(x_{11}^{e_i}\nu_1^{e_i 2^i} + \ldots + x_{1n'}^{e_i}\nu_{n'}^{e_i 2^i})$.

Hence we observe that the degree of each term is at most the Hamming weight of e.

We have thus proved the following lemma:

Lemma 4.3.1. *let $f \in \mathbb{F}_{2^n}[x_1, \ldots, x_m]$ be a multivariate polynomial in m variables such that the degree of each variable x_i is at most $2^{d_i} - 1$. Let $x_i = \sum_{j=1}^{n'} x_{ij}\nu_j, i = 1, \ldots, m$ for some $x_{ij} \in \mathbb{F}_2$. Then the resulting polynomial has degree at most $\sum_{i=1}^{m} d_i$.*

Let $\{\theta_i |\ i = 1, \ldots, n\}$ be a basis of $\mathbb{F}_{2^n}$ over $\mathbb{F}_2$ and let $\{\nu_i |\ i = 1, \ldots, n'\}$ be a basis of V over $\mathbb{F}_2$. We introduce mn' variables x_{ij} over $\mathbb{F}_2$ such that $x_i = \sum_{j=1}^{n'} x_{ij}\nu_j$. By replacing each x_i with the above equation and reducing by the field equations, we obtain

$$0 = f(x_1, x_2, \ldots, x_m) = f(\sum_{j=1}^{n'} x_{1j}\nu_j, \ldots, \sum_{j=1}^{n'} x_{mj}\nu_j) = f_1\theta_1 + f_2\theta_2 + \ldots + f_n\theta_n$$

for some $f_1, f_2, \ldots, f_n \in \mathbb{F}_2[x_{11}, \ldots, x_{mn'}]$ which depend on f and on the vector subspace V. Thus solving $f(x_1, x_2, \ldots, x_m) = 0$ under the linear constraints $x_1, \ldots, x_m \in V$ is equivalent to

solving the following boolean polynomial system:

$$f_1 = f_2 = \cdots = f_n = 0 \tag{4.2}$$

We return to our summation polynomial S_{m+1}. Recall that the degree of each variable x_i in S_{m+1} is 2^{m-1}. It follows that the degree with respect to each block of variables $(x_{i1}, x_{i2}, \ldots, x_{in'})$ is $m-1$. Consequently, our system in question consists of n boolean polynomials in $mn' \approx n$ variables with the degree of each polynomials at most $m(m-1)$.

Let d be a positive integer.

Consider the set $Mon(d)$ of all multi-linear monomials in $x_{11}, \ldots, x_{mn'}$ such that the degree with respect to each block of variables is at most d. The total number of such monomials can be estimated by $M(d) = \left(\sum_{d'=0}^{d} \binom{n'}{d'}\right)^m$.

Since $S_{m+1} = 0$, we have $gS_{m+1} = 0$ for any monomial g. We construct all polynomials of the form gS_{m+1} such that the degree with respect to each block of variables is at most d. The number of all such monomials g can be computed as $E(d) = \left(2^t(\sum_{d'=t}^{d} \binom{n'-t}{d'-t}\right)^m$, where $t = m-1$.

Let G be a matrix with the columns indexed by monomials in $Mon(d)$ such that G is the coefficient matrix constructed from all the polynomials obtained from gS_{m+1}'s above. This is commonly referred as the Macaulay matrix. It was shown in [21] that if $d \approx n'/2$,$nE(d) > M(d)$, and we can perform the Gaussian elimination on G to solve for the variables.

Assumption: Here, we assume that almost all the rows of G generated by this way are linearly independent with a high probability.

Experiments were performed to verify this assumption. Under such an assumption, with $d \approx n'/2$, the variables $x_{ij}, i = 1, \ldots, m, j = 1, \ldots, n'$, can be solved via linearization.

By choosing $m \approx n' \approx n^{1/2}$, this yields a complexity estimate of $2^{O(\omega n/2)}$ to solve the ECDLP.

4.3.2 Petit and Quisquater's Result

In [39], C. Petit and J. Quisquater used some heuristic assumptions on the degree of regularity and first fall degree of a polynomial system arising from a weil descent to achieve a subexponential complexity $O(2^{cn^{2/3}\log n})$.

In Gröbner basis computations, under some fixed monomial ordering, we essentially compute the s-polynomials of polynomials and recursively add the remainder to the list of polynomials. Let L be the set of all polynomials generated in the process of computing the Gröbner basis. The largest degree of the polynomials in L is known as the degree of regularity of the system.

It can be shown that in computing the Gröbner basis, we are in fact performing elimination on matrices constructed from the coefficients of the polynomials in L with respect to the monomials. Consequently, the degree of regularity d_{reg} gives us a good estimate of the complexity of the algorithm involved. Specifically, for n variables, the complexity can be estimated to be $O(n^{\omega d_{reg}})$, where ω is the linear algebra constant. It follows that a good estimate on the degree of regularity of the polynomial system will yield a good estimate on its complexity.

Definition 4.3.2. Let R be a multivariate polynomial ring over a field K. Let $h_1, \ldots, h_l$ be a set of polynomials in R. The first fall degree of $\{h_1, \ldots, h_l\}$, denoted by d_{ff}, is defined as the smallest degree $d \geq \deg(h_i)$ for all $i = 1, 2, \ldots, l$ such that there exist polynomials $g_1, \ldots, g_l$ in R with $\max_i\{\deg(g_i) + \deg(h_i)\} = d$ and $\deg(\sum_{i=1}^{l} g_i h_i) < d$, $\sum_{i=1}^{l} g_i h_i \neq 0$.

Let f be a polynomial over $\mathbb{F}_{2^n}$ in m variables $x_1, \ldots, x_m$ such that the degree of each variables is at most $2^t - 1$ for some integer t. As before, we do the Weil descent on f, i.e. replace each of the variables x_i by $x_{i1}\nu_1 + \ldots + x_{in'}\nu_{n'}, i = 1, \ldots, m$. Here, the degree with respect to each variable is bounded by t. We obtain a set of polynomials as in 4.2. By consider $x_1 f$ and performing weil descent again, we can deduce that the first fall degree of the system $\{f_1(x_{11}, \ldots, x_{mn'}), \ldots, f_n(x_{11}, \ldots, x_{mn'})\}$ is at most $mt + 1$.

In [39], Petit *et. al* proposed the following assumption:

Assumption 4.3.3. *Let E be an elliptic curve defined over $\mathbb{F}_{2^n}$. Let V be a random vector space of dimension n' over $\mathbb{F}_2$ and let R be a random point on the curve. Let $f := S_{m+1}(x_1, x_2, \ldots, x_m, x_R)$, where x_R denote the x-coordinate of R. For the system 4.2 from f, we have $d_{reg} = d_{ff} + o(1)$ with a high probability.*

Petit *et. al* verified the above assumption by experiments. The following is a brief complexity analysis of the index calculus method.

First, the complexity of computing the $(m+1)$th summation polynomial is 2^{t_1}, where $t_1 \approx m(m+1)$. By the above assumption, the degree of regularity $d_{red} \approx m^2 + 1$. Using dedicated block Gröbner basis algorithms, the complexity to solve this system is $O((n')^{\omega(m^2+1)})$, where ω is the linear algebra constant. The probability that a point $R_i := a_i P + b_i Q$ can be written as a sum of m points in the factor basis is about $1/m!$. We require around $2^{n'}$ such relations, and thus we need to solve about $2^{n'} m!$ summation polynomials. The total complexity for the sieving step of the index calculus method is 2^{t_2}, where $t_2 = \log m! + n' + (m^2+1)\omega \log n'$. Finally, the linear algebra step has the complexity of 2^{t_3}, where $t_3 \approx \log m + \log n + \omega' n'$ and ω' is the sparse linear algebra constant. Therefore, the total time is $T := 2^{t_1} + 2^{t_2} + 2^{t_3}$.

For $1/2 \leq \alpha \leq 1$, put $m = n^{1-\alpha}$ and $n' = n^{\alpha}$. We obtain $t_1 \approx n^{2(1-\alpha)}$, $t_2 \approx (1-\alpha)n^{1-\alpha} + n^{\alpha} + \omega\alpha(n^{2(1-\alpha)} + 1)\log n$, $t_3 \approx (2-\alpha)\log n + \omega' n^{\alpha}$. Taking $\alpha := 2/3$, which minimizes the total complexity, we eventually obtain a total complexity of $O(2^{cn^{2/3}\log n})$, where $c := 2\omega/3$ and ω is the linear algebra constant.

Remark 4.3.4. It seems that Assumption(4.3.3) is questionable. Recently, Ming-Deh A. Huang, Michiel Kosters and Sze Ling Yeo raised doubt on this assumption in [26, Section5.2]. They gave some experiment results to show their query on this assumption.

5. ON THE LAST FALL DEGREE OF ZERO-DIMENSIONAL WEIL DESCENT SYSTEMS

This chapter is a joint work with Ming-Deh A. Huang, Michiel Kosters and Sze Ling Yeo [25]. In this chapter, we give a method for solving zero-dimensional polynomial systems.

5.1 Last fall degree

In this section we define a new concept named *last fall degree*. It is related to a polynomial system. This notion is a parameter for the complexity of solving the polynomial system, and is independent of any monomial order. Later, we will use this notion to study the complexity of Weil descent systems.

Let k be a field and let $R = k[X_0, \ldots, X_{m-1}]$ be a polynomial ring. Note that the affine group $\mathrm{Aff}_m(k) = k^m \rtimes \mathrm{GL}_m(k)$ acts on R by affine change of variables, more precisely, let $A = (v, A_1) \in \mathrm{Aff}_m(k)$ with $v \in k^m$ and $A_1 \in \mathrm{GL}_m(k)$ and $f \in R$,then the action of A on f is defined as follows:

$$Af(X) := f(A_1 X + v)$$

where $X = (X_0, ..., X_{m-1})^t$ is the column vector for variables. This action preserves the total degree.

Here we fix some notations used throughout this chapter. The set of polynomials of degree $\leq i$ in R is denoted by $R_{\leq i}$.

Let $\mathcal{F}$ be a finite subset of R and let $I \subseteq R$ be the ideal generated by $\mathcal{F}$. We set $\deg(\mathcal{F}) = \max\{\deg(f) : f \in \mathcal{F}\}$. Furthermore, we set $\deg_{X_i}(\mathcal{F}) = \max\{\deg_{X_i}(f) : f \in \mathcal{F}\}$ for $i \in \{0, ..., m-1\}$.

5.1.1 Constructible polynomials

Definition 5.1.1. For $i \in \mathbb{Z}_{\geq 0}$, we let $V_{\mathcal{F},i}$ be the smallest k-vector subspace of R satisfying the following two conditions:

(i) $\mathcal{F} \cap R_{\leq i} = \{f \in \mathcal{F} : \deg(f) \leq i\} \subseteq V_{\mathcal{F},i}$;

(ii) $hg \in V_{\mathcal{F},i}$, for all $g \in V_{\mathcal{F},i}$ and $h \in R$ with $\deg(hg) \leq i$.

We set $V_{\mathcal{F},\infty} = I$ and $V_{\mathcal{F},-1} = \emptyset$.

If $\mathcal{F}$ is fixed, we just use V_i for abbreviation of $V_{\mathcal{F},i}$. Intuitively, V_i is the largest subset of I which can be constructed from $\mathcal{F}$ by doing operations of degree at most i. Note that V_i is a finite-dimensional k-vector space with dimension satisfying:

$$\dim_k(V_i) \leq \dim_k R_{\leq i} = \binom{m+i}{i} \leq (m+i)^i.$$

Notice that for any $f \in I$, there is an $i \in \mathbb{Z}_{\geq 0}$ such that $f \in V_i$. Phrased differently, we have $I = V_\infty = \bigcup_{i \in \mathbb{Z}_{\geq 0}} V_i$.

Definition 5.1.2. For $g, h \in R$ and $i \in \mathbb{Z}_{\geq 0} \sqcup \{\infty\}$. We write $g \equiv_{\mathcal{F},i} h$ if $g - h \in V_{\mathcal{F},i}$. If $\mathcal{F}$ is fixed, we often write $g \equiv_i h$. We write $g \equiv h$ if $g \equiv_\infty h$, which means $g - h \in I$.

Proposition 5.1.3. *Let $\mathcal{F}, \mathcal{G} \subset R$ be finite subsets, $i \in \mathbb{Z}_{\geq 0}$, $A \in \mathrm{Aff}_m(k)$ and k'/k a field extension. Then the following hold:*

(i) *$V_{\mathcal{F},i}$ can be constructed in a number of field operations which is polynomial in $(m+i)^i$ and in the cardinality of $\mathcal{F}$.*

(ii) *if* $\mathcal{F} \subseteq \mathcal{G}$, *then* $V_{\mathcal{F},i} \subseteq V_{\mathcal{G},i}$;

(iii) *if* $\mathrm{Span}_k(\mathcal{F}) = \mathrm{Span}_k(\mathcal{G})$ *and* $i \geq \deg(\mathcal{F})$, *then* $V_{\mathcal{F},i} = V_{\mathcal{G},i}$;

(iv) *one has* $AV_{\mathcal{F},i} = V_{A\mathcal{F},i}$, *where* $A\mathcal{F} = \{Af : f \in \mathcal{F}\}$;

(v) *one has* $V_{\mathcal{F},i} \otimes_k k' = V_{\{f \otimes_k 1:\ f \in \mathcal{F}\},i} \subset k'[X_0, \ldots, X_{m-1}]$.

Proof. i: One can construct the $V_{\mathcal{F},i}$ using linear algebra as follows. Fix a graded order on R. Construct a matrix for $\mathcal{F} \cap R_{\leq i}$ using the method in 3.1.2. Then put this matrix in reduced row echelon form and remove the 0 rows. Then we do the following step. For every row, multiply the corresponding polynomial g by all monomials t with $\deg(tg) \leq i$ and add a new row to the matrix corresponding to tg. After doing this, do Gaussian elimination on the new matrix to reduced row echelon form and remove 0 rows. If the number of rows increased in this step, then repeat the step. If not, the process is finished and one has computed a basis of $V_{\mathcal{F},i}$. Since the computations all occur in $R_{\leq i}$, a finite dimensional space of dimension bounded by $(m+i)^i$, one obtains the complexity result.

ii,v: Follows directly from the definitions.

iii: $\mathrm{Span}_k(\mathcal{F}) = \mathrm{Span}_k(\mathcal{G})$ implies that $\deg(\mathcal{F}) = \deg(\mathcal{G})$. For $i \geq \deg(\mathcal{F})$, we have $V_{\mathcal{G},i} \supseteq \mathcal{G} \cap R_{\leq i} = \mathcal{G}$, and hence

$$V_{\mathcal{G},i} \supseteq \mathrm{Span}_k(\mathcal{G}) = \mathrm{Span}_k(\mathcal{F}) \supseteq \mathcal{F}.$$

From the definitions of $V_{\mathcal{G},i}$ and $V_{\mathcal{F},i}$, we have

$$V_{\mathcal{G},i} \supseteq V_{\mathcal{F},i}.$$

Similarly, we have

$$V_{\mathcal{F},i} \supseteq V_{\mathcal{G},i}.$$

Thus

$$V_{\mathcal{F},i} = V_{\mathcal{G},i}.$$

iv: By the definition of $V_{\mathcal{F},i}$ and the action of $\mathrm{Aff}_m(k)$, we find that $AV_{\mathcal{F},i}$ is a k-vector subspace of R. For any element $g = Ag_1 \in AV_{\mathcal{F},i}$ with $g_1 \in V_{\mathcal{F},i}$ and $h \in R$ such that $\deg(hg) \leq i$, let $h_1 = A^{-1}h \in R$. Notice that the action of $\mathrm{Aff}_m(k)$ respects degrees, then we have

$$\deg(h_1 g_1) = \deg(h_1) + \deg(g_1) = \deg(h) + \deg(g) = \deg(hg) \leq i$$

hence $h_1 g_1 \in V_{\mathcal{F},i}$ and

$$hg = Ah_1 \cdot Ag_1 = A(h_1 g_1) \in AV_{\mathcal{F},i}$$

Also we have

$$AV_{\mathcal{F},i} \supseteq A(\mathcal{F} \cap R_{\leq i}) = A\mathcal{F} \cap R_{\leq i}.$$

So we have proven that $AV_{\mathcal{F},i}$ is a k-vector subspace of R satisfies the following two:

1. $A\mathcal{F} \cap R_{\leq i} \subseteq AV_{\mathcal{F},i}$;
2. if $g \in AV_{\mathcal{F},i}$ and if $h \in R$ with $\deg(hg) \leq i$, then $hg \in AV_{\mathcal{F},i}$.

By the definition of $V_{A\mathcal{F},i}$, we have

$$AV_{\mathcal{F},i} \supseteq V_{A\mathcal{F},i}.$$

Similarly, one has

$$A^{-1}V_{A\mathcal{F},i} \supseteq V_{\mathcal{F},i}.$$

Apply the action of A to both sides of the above, we have

$$V_{A\mathcal{F},i} \supseteq AV_{\mathcal{F},i}.$$

Thus finally we have $V_{A\mathcal{F},i} = AV_{\mathcal{F},i}$. □

Remark 5.1.4. Let $f_1, f_2, g_1, g_2 \in R$. Assume $f_1 \equiv_i f_2$, $g_1 \equiv_j g_2$. Assume that $\deg(f_1) \leq i$ and $\deg(g_2) \leq j$. Then one has

$$f_1g_1 - f_2g_2 = f_1(g_1 - g_2) + g_2(f_1 - f_2) \in V_{i+j}.$$

Hence we have $f_1g_1 \equiv_{i+j} f_2g_2$.

Let $\mathcal{F} \subseteq R$ be a finite subset generating a nonzero ideal I. Let $\leq$ be a graded order on R. We set

$$d_{\mathcal{F},\leq} = \min\{i : (\mathrm{LT}(v) : v \in V_{\mathcal{F},i}) = \mathrm{LT}(I)\}.$$

Equivalently, it is the minimal i such that $V_{\mathcal{F},i}$ contains a Gröbner basis for I with respect to $\leq$. Any algorithm which tries to compute a Gröbner basis has to do computations up to this degree. One particular order is the so-called degree reverse lexicographic order $\leq_{\mathrm{revlex}}$. One calls $d_{\mathcal{F},\leq_{\mathrm{revlex}}}$ the degree of regularity of $\mathcal{F}$ (in literature, there are many different definitions, but in this chapter we will choose this one). This definition of degree of regularity $\leq_{\mathrm{revlex}}$ is bounded up by another definition of degree of regularity d_{reg} which is introduced in section(3.1.4)(for detail see remark(5.1.10) below).

5.1.2 Last fall degree

We now define the last fall degree.

Definition 5.1.5. Let $\mathcal{F}$ be a finite subset of R and let I be the ideal generated by $\mathcal{F}$. We define the *last fall degree* of $\mathcal{F}$ to be the minimal $d \in \mathbb{Z}_{\geq 0} \sqcup \{\infty\}$ such that for all $f \in I$ we have $f \in V_{\max\{d,\deg(f)\}}$ and denote it by $d_{\mathcal{F}}$.

Note that the above definition implies that for $i \geq d_{\mathcal{F}}$, one has $V_{\mathcal{F},i} = I \cap R_{\leq i}$.

Proposition 5.1.6. *Let $\mathcal{F}, \mathcal{G} \subset R$ be finite subsets which generate ideals I respectively J. Let $A \in \mathrm{Aff}_m(k)$ and k'/k be a field extension. The following hold, where $\leq$ is any graded monomial order.*

(i) *One has: $d_{\mathcal{F}} \in \mathbb{Z}_{\geq 0}$.*

(ii) *One has $d_{\mathcal{F}} \leq d_{\mathcal{F},\leq}$.*

(iii) *One has: $d_{\mathcal{F}}$ is the largest $c \in \mathbb{Z}_{\geq 0}$ such that $V_c \cap R_{\leq c-1} \neq V_{c-1}$.*

(iv) *If $\mathrm{Span}_k(\mathcal{F}) = \mathrm{Span}_k(\mathcal{G})$, then one has $\max(d_{\mathcal{F}}, \deg(\mathcal{F})) = \max(d_{\mathcal{G}}, \deg(\mathcal{F}))$.*

(v) *One has: $d_{\mathcal{F}} = d_{A\mathcal{F}}$.*

(vi) *Consider the set $\{f \otimes 1 : \ f \in \mathcal{F}\} \subset k'[X_0, \ldots, X_{m-1}]$. One has: $d_{\{f \otimes 1: \ f \in \mathcal{F}\}} = d_{\mathcal{F}}$.*

(vii) *If $I = J$ and $\mathcal{F} \subseteq \mathcal{G}$, then one has $d_{\mathcal{G}} \leq d_{\mathcal{F}}$.*

(viii) *If $g \in V_{\mathcal{F},j}$, then one has $d_{\mathcal{F}} \leq \max(j, d_{\mathcal{F} \cup \{g\}})$.*

Proof. i, ii: i follows directly from ii. Assume that c is such that $V_{\mathcal{F},c}$ contains a Gröbner basis $\mathcal{B}$ with respect to $\leq$. We need to show $d_{\mathcal{F}} \leq c$. Take $f \in I$ and write $f = \sum_{b \in \mathcal{B}} a_b b$ with $\deg(a_b b) \leq \deg(f)$ for $b \in \mathcal{B}$. This is possible because $\mathcal{B}$ is a Gröbner basis for a graded order. Let $i = \max(c, \deg(f))$,note that we have the following:

$$\mathcal{B} \subseteq V_{\mathcal{F},c} \subseteq V_{\mathcal{F},i},$$

thus $\deg(a_b b) \in V_{\mathcal{F},i}$, and hence one finds $f \in V_{\max(c,\deg(f))}$, thus $d_{\mathcal{F}} \leq c$.

iii: Let c be as in the property, let $f \in V_c \cap R_{\leq c-1}$ and $f \notin V_{c-1}$. Then $\deg(f) \leq c-1$. Suppose $d_{\mathcal{F}} \leq c-1$, then $\max(d_{\mathcal{F}}, \deg(f)) \leq c-1$. By definition of last fall degree, we have

$$f \in V_{\max(d_{\mathcal{F}},\deg(f))} \subseteq V_{c-1}$$

this contradicts to $f \notin V_{c-1}$, thus we must have $d_{\mathcal{F}} \geq c$.

Again by definition of last fall degree, there exists $g \in I$ such that $g \notin V_{\max(d_{\mathcal{F}}-1,\deg(g))}$. Suppose $\deg(g) \geq d_{\mathcal{F}}$, then we have

$$g \in V_{\max(d_{\mathcal{F}},\deg(g))} = V_{\deg(g)} = V_{\max(d_{\mathcal{F}}-1,\deg(g))},$$

this contradicts to $g \notin V_{\max(d_{\mathcal{F}}-1,\deg(g))}$, thus we must have $\deg(g) \leq d_{\mathcal{F}} - 1$. Therefore we have

$$g \in I \cap R_{\leq d_{\mathcal{F}}-1}$$

and

$$g \notin V_{\max(d_{\mathcal{F}}-1,\deg(g))} = V_{d_{\mathcal{F}}-1}$$

So one finds

$$V_{d_{\mathcal{F}}} \cap R_{\leq d_{\mathcal{F}}-1} = I \cap R_{\leq d_{\mathcal{F}}-1} \neq V_{d_{\mathcal{F}}-1}.$$

iv: $\mathrm{Span}_k(\mathcal{F}) = \mathrm{Span}_k(\mathcal{G})$ implies that $\deg(\mathcal{F}) = \deg(\mathcal{G})$ and $\langle \mathcal{F} \rangle = \langle \mathcal{G} \rangle$. We have the following two cases.

1. If $d_{\mathcal{F}} > \deg(\mathcal{F})$, by (iii) we have $f \in V_{\mathcal{F},d_{\mathcal{F}}}$ and $f \notin V_{\mathcal{F},d_{\mathcal{F}}-1}$, so $\deg(f) \leq d_{\mathcal{F}} - 1$. From Proposition 5.1.3(iii), we have

$$V_{\mathcal{F},d_{\mathcal{F}}-1} = V_{\mathcal{G},d_{\mathcal{F}}-1} \tag{5.1}$$

By definition, we have

$$f \in V_{\mathcal{G},\max(d_{\mathcal{G}},\deg(f))} \tag{5.2}$$

Suppose $d_{\mathcal{G}} \leq d_{\mathcal{F}} - 1$,then $\max(d_{\mathcal{G}}, \deg(f)) \leq d_{\mathcal{F}} - 1$, thus by 5.2 and 5.1 we have

$$f \in V_{\mathcal{G},d_{\mathcal{F}}-1} = V_{\mathcal{F},d_{\mathcal{F}}-1},$$

this contradicts to $f \notin V_{\mathcal{F},d_{\mathcal{F}}-1}$. Thus we must have $d_{\mathcal{G}} \geq d_{\mathcal{F}} > \deg(\mathcal{F}) = \deg(\mathcal{G})$.

Now start from $d_{\mathcal{G}} > \deg(\mathcal{G})$ we get, we can prove that $d_{\mathcal{F}} \geq d_{\mathcal{G}}$ similarly as above, thus we obtain $d_{\mathcal{F}} = d_{\mathcal{G}}$ and $\max(d_{\mathcal{F}}, \deg(\mathcal{F})) = \max(d_{\mathcal{G}}, \deg(\mathcal{F}))$ as required.

2. If $d_{\mathcal{F}} \leq \deg(\mathcal{F})$, suppose $d_{\mathcal{G}} > \deg(\mathcal{F}) = \deg(\mathcal{G})$, from the above, we can deduce that $d_{\mathcal{F}} \geq d_{\mathcal{G}} > \deg(\mathcal{F})$, contradiction. Thus we obtain $d_{\mathcal{G}} \leq \deg(\mathcal{F})$. So $\max(d_{\mathcal{F}}, \deg(\mathcal{F})) = \max(d_{\mathcal{G}}, \deg(\mathcal{F}))$.

v: One easily finds $\langle A\mathcal{F}\rangle = A\langle\mathcal{F}\rangle$. Now for any $f \in \langle A\mathcal{F}\rangle$, we can write $f = Af_1$ for $f_1 \in \langle\mathcal{F}\rangle = I$. By definition, we have

$$f_1 \in V_{\mathcal{F},\max(d_{\mathcal{F}},\deg(f_1))} \tag{5.3}$$

Note that the action of $\mathrm{Aff}_m(k)$ respects degrees and from Proposition 5.1.3(iv) and 5.3, we have

$$f = Af_1 \in AV_{\mathcal{F},\max(d_{\mathcal{F}},\deg(f_1))} = V_{A\mathcal{F},\max(d_{\mathcal{F}},\deg(f_1))} = V_{A\mathcal{F},\max(d_{\mathcal{F}},\deg(f))}$$

Thus by definition of last fall degree, we have

$$d_{A\mathcal{F}} \leq d_{\mathcal{F}},$$

Similarly,

$$d_{A^{-1}(A\mathcal{F})} \leq d_{A\mathcal{F}},$$

thus $d_{A\mathcal{F}} = d_{\mathcal{F}}$.

vi: Follows directly from the definitions.

vii: For any $f \in J = I$, by definition we have

$$f \in V_{\mathcal{F},\max(d_{\mathcal{F}},\deg(f))}.$$

From $\mathcal{F} \subseteq \mathcal{G}$ and Proposition 5.1.3(ii), one has

$$f \in V_{\mathcal{G},\max(d_{\mathcal{F}},\deg(f))}.$$

Thus $d_{\mathcal{G}} \leq d_{\mathcal{F}}$ by definition.

viii: For $i \geq j$, since $g \in V_{\mathcal{F},j} \subseteq V_{\mathcal{F},i}$, we have

$$(\mathcal{F} \cup \{g\}) \cap R_{\leq i} \subseteq V_{\mathcal{F},i}.$$

Thus $V_{\mathcal{F}\cup\{g\},i} \subseteq V_{\mathcal{F},i}$. And we have $V_{\mathcal{F},i} \subseteq V_{\mathcal{F}\cup\{g\},i}$ by Proposition 5.1.3(ii). Therefore one has $V_{\mathcal{F}\cup\{g\},i} = V_{\mathcal{F},i}$.

It is easy to see that $\langle \mathcal{F} \rangle = \langle \mathcal{F} \cup \{g\} \rangle$ as $g \in V_{\mathcal{F},j}$. For any $f \in I = \langle \mathcal{F} \rangle = \langle \mathcal{F} \cup \{g\} \rangle$, we have

$$f \in V_{\mathcal{F}\cup\{g\},\max(d_{\mathcal{F}\cup\{g\}},\deg(f))} \subseteq V_{\mathcal{F}\cup\{g\},\max(\max(j,d_{\mathcal{F}\cup\{g\}}),\deg(f))} = V_{\mathcal{F},\max(\max(j,d_{\mathcal{F}\cup\{g\}}),\deg(f))} \tag{5.4}$$

note the equality in 5.4 we used $\max(\max(j, d_{\mathcal{F}\cup\{g\}}), \deg(f)) \geq j$ and $V_{\mathcal{F}\cup\{g\},i} = V_{\mathcal{F},i}$ for $i \geq j$. Thus it follows $d_{\mathcal{F}} \leq \max(j, d_{\mathcal{F}\cup\{g\}})$.

□

Property ii in combination with iii gives a method (using a monomial order and a Gröbner basis computation) to compute the last fall degree. It would be of great importance to find a

method which does not use a monomial order and which does not use a Gröbner basis computation.

Remark 5.1.7. Let $\mathcal{F}$ be a finite subset of R. It is in general not true that $V_{\mathcal{F},d_{\mathcal{F}}}$ generates the same ideal as $\mathcal{F}$. For example, if $m = 1$ and $\mathcal{F} = \{f\}$ with f not constant, then one has $d_{\mathcal{F}} = 0$, whereas $V_{\mathcal{F},0}$ does not generate (f).

Proposition 5.1.8. *Let $I \subseteq R$ be a zero-dimensional ideal. Let $\leq$ be a graded monomial order. Let $\mathcal{B}$ be the reduced Gröbner basis of I with respect to $\leq$. Then one has* $\deg(\mathcal{B}) \leq \dim_k(R/I)$.

Proof. Let $\mathcal{B}$ be the reduced Gröbner basis of I. Let $X_1^{a_1}X_2^{a_2}\cdots X_m^{a_m}$ be the leading term of $b \in \mathcal{B}$. Note that the set

$$\{X_1^{b_1}X_2^{b_2}\cdots X_m^{b_m} : 0 \leq b_i \leq a_i, \text{ not all } b_i = a_i\}$$

is independent in R/I over k, because $\mathcal{B}$ is the reduced Gröbner basis. This set has cardinality $(a_1+1)\cdots(a_m+1)-1$. Hence we find

$$a_1 + a_2 + \ldots + a_m \leq (a_1+1)(a_2+1)\cdots(a_m+1) - 1 \leq \dim_k(R/I).$$

The result follows. □

Corollary 5.1.9. *Let $\mathcal{F} \subset R$ be finite subset which generates a zero-dimensional ideal I. Let $\leq$ be any graded monomial order on R. Then one has:*

$$d_{\mathcal{F}} \leq d_{\mathcal{F},\leq} \leq \max\{d_{\mathcal{F}}, \dim_k(R/I)\}.$$

Proof. One has $d_{\mathcal{F}} \leq d_{\mathcal{F},\leq}$ by Proposition 5.1.6. Let $\mathcal{B}$ be the reduced Gröbner basis of I with respect to $\leq$. By Proposition 5.1.8 one has $\mathcal{B} \subseteq V_{\max\{d_{\mathcal{F}},\dim_k(R/I)\}}$. We conclude $d_{\mathcal{F},\leq} \leq \max\{d_{\mathcal{F}}, \dim_k(R/I)\}$. □

Remark 5.1.10. For another definition of degree of regularity d_{reg} which is introduced in section(3.1.4), we have $d_{reg} \geq d_{\mathcal{F},\leq_{\text{revlex}}}$. By the definition of d_{reg}, all polynomials are contained in $V_{\mathcal{F},d_{reg}}$ in each loop of F4 or F5 algorithm and hence the Gröbner basis are also contained in $V_{\mathcal{F},d_{reg}}$. Thus we have $d_{reg} \geq d_{\mathcal{F},\leq_{\text{revlex}}}$ by the definition of $d_{\mathcal{F},\leq_{\text{revlex}}}$. By the above corollary, we have $d_{\mathcal{F}} \leq d_{reg}$, i.e., the last fall degree $d_{\mathcal{F}}$ is upper bounded by the degree of regularity d_{reg}.

Remark 5.1.11. We call an ideal $I \subseteq R$ radical if R/I has no nilpotent elements. Assume that I is a radical ideal. Then $\dim_k(R/I)$ is equal to the number of solutions of I over $\overline{k}$. We give a brief sketch. Note that R/I is reduced, since I is radical. Set $S = R/I \otimes_k \overline{k}$. Note that S is still reduced. Since S is a reduced Artinian ring and by the Nullstellensatz, it is isomorphic to $\overline{k}^e$ where e is the number of solutions of I over $\overline{k}$. One has $\dim_k(R/I) = \dim_{\overline{k}}(S) = e$.

5.1.3 Solving systems

We will now discuss how one can solve a multivariate zero-dimensional system once the last fall degree is known.

Proposition 5.1.12. *[26] Let k be a field. Assume that one can factor polynomials of degree at most t using a number of field equations which is polynomial in $g(t)$ where g is some function. Let $\mathcal{F} \subset R$ be a finite set. Assume that the ideal I generated by $\mathcal{F}$ is radical and that the system has at most e solutions over $\overline{k}$. Set $d = \max(d_{\mathcal{F}}, e)$. Then one can find all solutions of I in k in a number of field operations which is polynomial in the cardinality of $\mathcal{F}$, $g(d)$ and $(m+d)^d$.*

Proof. Compute V_d with a number of field operations polynomial in the input size of $\mathcal{F}$ and $(m+d)^d$ (Proposition 5.1.3i). We will work in V_d to find all the solutions.

Assume that all solutions over $\overline{k}$ of the system are

$$Z(\mathcal{F}) = \{(a_{0,0}, \ldots, a_{0,m-1}), \ldots, (a_{t,0}, \ldots, a_{t,m-1})\} \subset \overline{k}^m$$

with $t < e$. Since I is a radical ideal, by the Nullstellensatz and Galois theory, one has

$$h_0 = \prod_{a \in \{a_{i,0} : i=0,\ldots,t\}} (X_0 - a) \in I.$$

Using linear algebra, and the definition of the last fall degree, one can find h_0 as the nonzero polynomial of minimal degree d_0 in $V_d \cap \mathrm{Span}_k\{1, X_0, \ldots, X_0^e\}$. Factor h_0 with a number of operations polynomial in $g(t)$. Assume that a_0 is a root of h_0 in k. We will find all solutions over k with $X_0 = a_0$. Set $h_0' = h_0/(X_0 - a_0)$ of degree $d_0 - 1$. By the Nullstellensatz and Galois theory, one has

$$h_1 = h_0' \prod_{a \in \{a_{i,1} : i=0,\ldots,t, a_{i,0}=a_0\}} (X_1 - a) \in I.$$

Using linear algebra, one finds h_1 as the polynomial of minimal degree d_1 in

$V_d \cap \mathrm{Span}_k\{h_0', X_1 h_0', \ldots, X_1^{e-d_0+1} h_0'\}$. Factor h_1/h_0' over k. Pick a solution a_1 over k and find all solutions with $X_0 = a_0$, $X_1 = a_1$ using the similar recursive procedure. Hence one can find all solutions over k with the claimed number of field operations. □

If k is a finite field of cardinality q, one can factor a polynomial of degree bounded by t with operations polynomial in $\max(\log(q), t)$ in a probabilistic way and $\max(q, t)$ in a deterministic way [50].

Remark 5.1.13. Note that as input of Proposition 5.1.12 we need an upper bound on $\dim_k(R/I)$ and $d_{\mathcal{F}}$. One can bound $d_{\mathcal{F}}$ from computing a Gröbner basis first, but this defies the purpose of using this approach. If I is radical, then $\dim_k(R/I)$ is equal to the number of solutions of the system over $\overline{k}$ (Remark 5.1.11).

In practice one can often construct the $V_{\mathcal{F},i}$ until one finds monovariate polynomials and then eliminate variables. In that case, one does not need the bound on d, although one then does not know when the procedure terminates. The latter is the approach of MutantXL.

5.1.4 Comparison

In this subsection we will compare the above approach of solving a system $\mathcal{F}$ which generates a zero-dimensional ideal I. Set $e = \dim_k(R/I)$.

Let us first discuss the relation between our method for solving polynomial systems and algorithms such as XL and MutantXL. To emulate the MutantXL algorithm (see for example [7]) one computes V_0, V_1, V_2, ..., V_i until one produces a monovariate polynomial, which is bound to happen for V_i with $i = \max(d_{\mathcal{F}}, e)$. Once this monovariate polynomial has been found, one can factor it and solve various systems where the given variable is evaluated at a zero of the monovariate polynomial. Hence one reduces to solving similar systems with less variables. This MutantXL algorithm is an extension of the more classical XL algorithm, as in [9]. The XL algorithm is very similar, but one does not 'use relations which cause the degree to fall'. The method we have described has two advantages over the descriptions of MutantXL in literature. First of all, our formulation of MutantXL involving the V_i is much cleaner than what one finds in literature. Second, it shows that in many cases the substitution is not needed: one often finds monovariate polynomials in the same step and one can proceed as in Proposition 5.1.12.

Other algorithms for solving such a polynomial system apply the following strategy. One first computes a Gröbner basis for a monomial order $\leq$ of our choice. Then one uses the efficient FGLM algorithm [18] to compute a Gröbner basis for the lexicographic order. Once one has a Gröbner basis for a lexicographic order, one can easily solve the system. To achieve this efficiently, one needs to pick a good monomial order $\leq$ and a good algorithm for computing a Gröbner basis. We emulate the computation of a Gröbner basis as follows: one computes the V_0, V_1, V_2, ..., V_i until V_i contains a Gröbner basis with respect to $\leq$. Algorithms such as F_4 or F_5 [15, 16] very efficiently construct the various V_i , by creating for example sparse polynomials, and creating not too many new polynomials. Hence in practice one can solve much bigger systems than with naive implementations, but theoretically such algorithms should not be that much faster (they should have essentially the same complexity). The other question is

which monomial order $\leq$ one should pick. If there is a Gröbner basis $\mathcal{B}$ of degree d, then one has $\mathcal{B} \subseteq V_{\max\{d_{\mathcal{F}},d\}}$ by definition (and it seems unlikely to happen earlier). Hence we need to find an order such that there is a guaranteed Gröbner basis of a low degree. In practice one usually picks $\leq_{\text{revlex}}$, because a Gröbner basis of low degree exists, much smaller than the one for the lexicographic order (see [33]).

Which of the two methods is better? The complexity of the first method relies on $\max\{d_{\mathcal{F}}, \dim_k(R/I)\}$, whereas the complexity of the second method essentially relies on $d_{\mathcal{F},\leq}$ (or $d_{\mathcal{F},\leq_{\text{revlex}}}$ when the degree reverse lexicographic order is chosen). Note that by Proposition 5.1.9 one has $d_{\mathcal{F},\leq} \leq \max\{d_{\mathcal{F}}, \dim_k(R/I)\}$. Based on this, we conclude that it is general better to use a Gröbner basis algorithm than a MutantXL algorithm. If $\dim_k(R/I)$ is smaller than $d_{\mathcal{F}}$ (say if a system is radical and has a unique solution over an algebraic closure), it seems that both methods have essentially the same complexity. Hence in general it seems better to use a Gröbner basis algorithm in all situations. Similar results have also been obtained in for example [2], where it is shown that MutantXL algorithms are versions of F_4 and F_5 algorithms with redundancies.

So why would one be interested in looking at the last fall degree $d_{\mathcal{F}}$? Firstly, $d_{\mathcal{F}}$ does not rely on any monomial order. Picking a monomial introduces asymmetry and it makes it hard to prove certain theoretical complexity statements. In this chapter, we come across one such situation. We start with a polynomial system $\mathcal{F}$ and then apply Weil descent to this system to obtain a system $\mathcal{F}'$ in a different polynomial ring. It is not even clear how one should relate monomial orders between the two polynomial rings, hence it seems to be hard to compare the various $d_{\mathcal{F},\leq}$ and $d_{\mathcal{F}',\leq'}$. It seems more natural to compare their last fall degrees. We manage to compare both last fall degrees without the use of any heuristics (Theorem 1.1.1). One other advantage of the last fall degree is that it behaves well with respect to various operations (Proposition 5.1.6).

First fall degree

It is often hard to estimate the degree of regularity $d_{\mathcal{F},\leq_{\text{revlex}}}$ of a system. Hence one uses heuristical methods to find bounds. In heuristics, one often says that $d_{\mathcal{F},\leq_{\text{revlex}}}$ is close to the *first* c such that $V_c \cap R_{\leq c-1} \neq V_{c-1}$. This c is called the *first fall degree* of the system. Actually, most articles, such as [39], use a slightly different definition of the first fall degree. They say that the first fall degree $d_{\mathcal{F},f}$ is the first $d \geq \deg(\mathcal{F})$ such that there exists $g_f \in R$ for $f \in \mathcal{F}$ such that $d = \max_{f\in\mathcal{F}}(\deg(g_f f))$ and $\deg(\sum_{f\in\mathcal{F}} g_f f) < d$ and $\sum_{f\in\mathcal{F}} g_f f \neq 0$. By definition we have $d_{\mathcal{F},f} \leq d_{\mathcal{F}}$ if $d_{\mathcal{F}} \geq \deg(\mathcal{F})$ and $d_{\mathcal{F}} > 0$. The idea behind this heuristic is that once a degree fall occurs, many more must occur and the system will sort of 'collapse'. Proposition 5.1.6iv shows that the last fall degree is the *last* c such that $V_c \cap R_{\leq c-1} \neq V_{c-1}$ and as described above, this last fall degree captures the complexity of a polynomial system quite nicely. It is quite easy, with the help of combinatorics, to find an upper bound on the first fall degree. However, it seems to be much harder to directly bound the last fall degree. Quite often combinatorics gives a first fall degree which does not depend on the number of variables, which seem to be too optimistic for a degree of regularity in a non zero-dimensional system. See Section 5.5 for more discussions.

We hope that the framework with the last fall degree allows one to prove complexity statements of solving certain systems in a rigorous way.

5.2 Weil descent

Let q be a prime power. Let $n \in \mathbb{Z}_{\geq 1}$ and let k be a finite field of cardinality q^n. Let k' be the subfield of k of cardinality q. In this section, we introduce two Weil descent transforms for a finite subset of $R = k[X_0, \ldots, X_{m-1}]$.

Let $\mathcal{F} \subset R$ be a finite set of polynomials. Suppose we want to find the common zeros of

these polynomials in k. Let I be the ideal generated by

$$\mathcal{F}_f = \mathcal{F} \cup \{X_i^{q^n} - X_i : i = 0, \ldots, m-1\}.$$

We want to find the zeros of $\mathcal{F}_f$.

5.2.1 Weil descent

Let $\alpha_0, \ldots, \alpha_{n-1}$ be a basis of k/k'. Write $X_i = \sum_{j=0}^{n-1} \alpha_j X_{ij}$. For $f \in \mathcal{F}$ and $j = 0, \ldots, n-1$, we define $[f]_k \in R' = k'[X_{ij}, i = 0, \ldots, m-1, j = 0, \ldots, n-1]$ with $\deg_{X_{ij}}([f]_k) \leq q-1$ by

$$f\left(\sum_{j=0}^{n-1} \alpha_j X_{0j}, \ldots, \sum_{j=0}^{n-1} \alpha_j X_{m-1\ j}\right) \equiv \sum_{j=0}^{n-1} [f]_j \alpha_j \qquad (\text{mod } X_{ij}^q - X_{ij},\ i = 0, \ldots, m-1, j = 0, \ldots, n-1).$$

The system

$$\mathcal{F}' = \{[f]_j :\ f \in \mathcal{F},\ j = 0, \ldots, n-1\}$$

is called the *Weil descent system* of $\mathcal{F}$ with respect to $\alpha_0, \ldots, \alpha_{n-1}$. There is a bijection between the solutions over k (or $\overline{k}$) of $\mathcal{F}_f$ and the solutions over k' (or $\overline{k}$) of

$$\mathcal{F}'_f = \mathcal{F}' \cup \{X_{ij}^q - X_{ij} : i = 0, \ldots, m-1,\ j = 0, \ldots, n-1\}.$$

Note that the ideals generated by $\mathcal{F}_f$ and $\mathcal{F}'_f$ are radical ideals.

An interesting choice for the α_i is a normal basis, that is, a basis with $\alpha_i = \theta^{q^i}$ for some $\theta \in k$. Such a basis always exists.

Remark 5.2.1. A different choice of α_i merely results in a linear change of the variables X_{ij} and a linear change of the polynomials $[f]_i$ and the field equations $X_{ij}^q - X_{ij}$. Indeed, let

$\beta_0, \ldots, \beta_{n-1}$ be another basis. Let $[f]'_i$ be the corresponding Weil descent polynomials with respect to this new basis. We can write $\beta_i = \sum_{j=0}^{n-1} c_{ij}\alpha_j$ and $\alpha_i = \sum_{j=0}^{n-1} d_{ij}\beta_j$ with $c_{ij}, d_{ij} \in k$. Let $C = (c_{ij})_{i,j}$ be the corresponding matrix. One has:

$$\begin{aligned} f(\sum_{j=0}^{n-1}\beta_j X_{0j}, \ldots, \sum_{j=0}^{n-1}\beta_j X_{m-1\ j}) &= f(\sum_{k=0}^{n-1}\alpha_k \sum_{j=0}^{n-1} c_{jk}X_{0j}, \ldots, \sum_{k=0}^{n-1}\alpha_k \sum_{j=0}^{n-1} c_{jk}X_{m-1\ j}) \\ &\equiv \sum_{i=0}^{n-1} \mathrm{diag}(C, \ldots, C)[f]_i \alpha_i \\ &= \sum_{j=0}^{n-1} \left(\sum_{i=0}^{n-1} d_{ij}\mathrm{diag}(C, \ldots, C)[f]_i \right) \beta_j \\ &\equiv \sum_{j=0}^{n-1} [f]'_j \beta_j. \end{aligned}$$

In the first $\equiv$ we used that $\mathrm{diag}(C, \ldots, C)$ acts on $(X_{ij}^q - X_{ij} : i, j)$. Note that in the last step we might still need to reduce and we used that. We get a similar expression if we swith the roles of the two different bases. We first see that $\deg(\mathcal{F}')$ does not depend on the choice of basis.

If d is the last fall degree of $\mathcal{F}'_f$ with respect to the α_i, and d' with respect to the β_i, we conclude that $\deg(\mathcal{F}')$ does not depend on the choice of basis and that

$$\max\{d, \deg(\mathcal{F}')\} = \max\{d', \deg(\mathcal{F}')\}.$$

5.2.2 Another model for Weil descent

For practical reasons, we will often work with another model of Weil descent. This model is defined over k and not over the subfield k'.

Let $S = k[X_{ij} : i = 0, \ldots, m-1,\ j = 0, \ldots, n-1]$. Let $e_0, \ldots, e_{m-1} \in \mathbb{Z}_{\geq 0}$. Let $X_i^{e'_i}$ be the remainder of division of $X_i^{e_i}$ by $X_i^{q^n} - X_i$. Write $e'_i = \sum_{j=0}^{n-1} e'_{ij} q^j$ in base q with

$e'_{ij} \in \{0, 1, \ldots, q-1\}$. We set

$$\overline{\prod_{i=0}^{m-1} X_i^{e_i}} = \prod_{i=0}^{m-1} X_{i0}^{e'_{i0}} \cdots X_{i\ n-1}^{e'_{i\ n-1}} \in S.$$

We extend this definition k-linearly for all polynomials in R. This gives a map $\overline{\ }: R \to S$. We set

$$\overline{\mathcal{F}} = \{\overline{f} : f \in \mathcal{F}\}$$

and we set, where by convention $X_{in} = X_{i0}$,

$$\overline{\mathcal{F}}_f = \overline{\mathcal{F}} \cup \{X_{ij}^q - X_{i\ j+1} :\ i = 0, \ldots, m-1,\ j = 0, \ldots, n-1\}.$$

We let $\overline{I}$ be the ideal generated by $\overline{\mathcal{F}}_f$. Note that $\overline{I}$ is radical.

There is a bijection between the zero set of I (over k or $\overline{k}$) and that of $\overline{I}$ (over k or $\overline{k}$). If for example $X_i = a_i \in \overline{k}$ gives a zero of I, then $(X_{i0}, \ldots, X_{i\ n-1}) = (a_i, a_i^q, \ldots, a_i^{q^{n-1}})$ gives a zero of $\overline{I}$.

In the following, we prove several lemmas which will be used later. We define $\equiv$ and $\equiv_i$ with respect to $\overline{\mathcal{F}}_f$ unless stated otherwise.

Lemma 5.2.2. *Let $h_1, h_2 \in R$, $g \in S$.*

(i) $\overline{h_1 + h_2} \equiv_{\max(\deg(\overline{h_1}), \deg(\overline{h_2}))} \overline{h_1} + \overline{h_2}$;

(ii) $\overline{h_1} \cdot \overline{h_2} \equiv_{\deg(\overline{h_1}) + \deg(\overline{h_2})} \overline{h_1 h_2}$;

(iii) *There exists $h_3 \in R$ with $\deg_{X_i}(h_3) < q^n$ such that $g \equiv_{\deg(g)} \overline{h_3}$.*

Proof. i: By definition $\overline{\ }: R \to S$ is a k-linear map, thus we have $\overline{h_1 + h_2} = \overline{h_1} + \overline{h_2}$ which is a stronger result.

ii: One can first prove the cases that h_1 and h_2 are monomials(it is easy to check and thus we omit the proof) and then the general case follows by operation $\bar{}$ is k-linear.

iii: One can prove the case $g = X_{00}^{e_{00}} \cdots X_{0\ n-1}^{e_{0\ n-1}}$ first and use similar method to prove the statement is hold when g is a monomial and then the result follows by operation $\bar{}$ is k-linear. □

We define a k-algebras homomorphism $\varphi : S \to R$ which maps X_{ij} to $X_i^{q^j}$. This map has the following properties.

Lemma 5.2.3. *Let $h \in R$. The following statements hold:*

(i) $\varphi(\overline{h}) \equiv h \pmod{X_i^{q^n} - X_i,\ i = 0, \ldots, m-1}$;

(ii) $h \in I$ *if and only if* $\overline{h} \in \overline{I}$.

Proof. i: A simple computation shows the result holds when h is a monomial, then the general case holds by the two maps φ and $\bar{}$ are k-linear.

ii: Let $h \in I$. Recall I is the ideal generated by

$$\mathcal{F}_f = \mathcal{F} \cup \{X_i^{q^n} - X_i : i = 0, \ldots, m-1\}.$$

Write $h = \sum_{i=0}^{m-1} b_i(X_i^{q^n} - X_i) + \sum_{f \in \mathcal{F}} a_f f$, for some polynomials $b_i, a_f \in R$. Modulo $\overline{I}$ we find with Lemma 5.2.2:

$$\overline{h} = \overline{\sum_{i=0}^{m-1} b_i(X_i^{q^n} - X_i) + \sum_{f \in \mathcal{F}} a_f f} \equiv \sum_{i=0}^{m-1} \overline{b_i}(X_{i0} - X_{i0}) + \sum_{f \in \mathcal{F}} \overline{a_f}\,\overline{f} \equiv 0.$$

Thus $\overline{h} \in \overline{I}$ as required.

Conversely, let $\overline{h} \in \overline{I}$. Write $\overline{h} = \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} c_{ij}(X_{ij}^q - X_{i\ j+1}) + \sum_{f \in \mathcal{F}} b_f \overline{f}$, for some

polynomials $c_{ij}, b_f \in S$. By i, we have

$$\begin{aligned}\varphi(\overline{h}) &= \sum_{i=0}^{m-1}\sum_{j=0}^{n-1}\varphi(c_{ij})\varphi(X_{ij}^{q} - X_{i\ j+1}) + \sum_{f\in\mathcal{F}}\varphi(b_f)\varphi(\overline{f}) \\ &\equiv \sum_{i=0}^{m-1}\varphi(c_{i\ n-1})(X_i^{q^n} - X_i) + \sum_{f\in\mathcal{F}}\varphi(b_f)f \pmod{X_i^{q^n} - X_i,\ i=0,\ldots,n-1}.\end{aligned}$$

Thus $\varphi(\overline{h}) \in I$ and we conclude $h \in I$ by i and the condition $h \in R$. □

Degree bounds

Recall the definition of τ. For $r \in \mathbb{R}_{\geq 0}$ and $c, t \in \mathbb{R}_{\geq 1}$ we set

$$\tau(r, c, t) = \lfloor 2t(c-1)\left(\log_c\left(\frac{r}{2t}+1\right)+1\right)\rfloor.$$

The inequality of arithmetic and geometric means gives for $x_1, \ldots, x_t > 0$ the following:

$$\log(x_1 \cdot x_2 \cdot \ldots \cdot x_t) \leq t\log\left(\frac{x_1 + x_2 + \ldots + x_t}{t}\right).$$

Lemma 5.2.4. *Let $g \in R \setminus \{0\}$. Then one has*

$$\deg(\overline{g}) \leq \tau(\deg(g), q, m/2).$$

Proof. Let $g \in k[X] \setminus \{0\}$. Note that for any positive integer b, we can write

$$b = b_0 + b_1 q + \ldots + b_r q^r,$$

with $r = \lfloor \log_q(b) \rfloor$. Then one has

$$\deg(\overline{g}) \leq (q-1)\left(\log_q(\deg(g)+1)+1\right).$$

Let $g \in R\backslash\{0\}$. It is enough to prove the result for monomials. Assume that $g = X_0^{a_0} \cdots X_{m-1}^{a_{m-1}}$. Then by the first part and the inequality of arithmetic and geometric means, one has

$$\begin{aligned} \deg(\overline{g}) &\leq \sum_{i=0}^{m-1} (q-1)\left(\log_q(a_i+1)+1\right) = (q-1)\left(\log_q(\prod_{i=0}^{m-1}(a_i+1)) + m\right) \\ &\leq m(q-1)\left(\log_q(\frac{1}{m}\sum_{i=0}^{m-1}(a_i+1)) + 1\right) = m(q-1)\left(\log_q(\frac{\deg(g)}{m}+1)+1\right). \end{aligned}$$

□

Lemma 5.2.5. *Let $i \in \mathbb{Z}_{\geq 0}$. Set $s = \tau(i,q,m)$. Then one has*

$$\overline{V_{\mathcal{F}_f,i}} \subseteq V_{\overline{\mathcal{F}}_f,s}.$$

Proof. Assume $i > 0$. Let $f \in \mathcal{F}_f$ nonzero with $\deg(f) \leq i$. Then Lemma 5.2.4 gives $\overline{f} \in V_{\overline{\mathcal{F}}_f,s}$. Assume $g \in V_{\mathcal{F}_f,i}$, $h \in R$ both non constant such that $\deg(gh) \leq i$. Note that $\overline{gh} \equiv_{\overline{\mathcal{F}}_f,\deg(\overline{g})+\deg(\overline{h})} \overline{g}\overline{h}$ by Lemma 5.2.2ii. Then Lemma 5.2.4 gives, together with the inequality of arithmetic and geometric means,

$$\begin{aligned} \deg(\overline{g}\overline{h}) = \deg(\overline{g}) + \deg(\overline{h}) &\leq m(q-1)\left(\log_q(\frac{\deg(g)}{m}+1)+1\right) \\ &\quad + m(q-1)\left(\log_q(\frac{\deg(h)}{m}+1)+1\right) \\ &\leq 2m(q-1)\left(\log_q(\frac{i}{2m}+1)+1\right). \end{aligned}$$

The result then follows easily (use Lemma 5.2.2i for the additivity). □

5.3 Last fall degree and descent

5.3.1 Relating the types of Weil descent

Let k be a finite field of cardinality q^n and let k' be the subfield of k of cardinality q. Let $\mathcal{F} \subset R$ be a finite subset. We will now compare the systems $\overline{\mathcal{F}}_f$ and $\mathcal{F}'_f$. We imitate a proof from Granboulan et al. [24, Section 4.2].

Proposition 5.3.1. *One has:*

$$\max\{d_{\mathcal{F}'_f}, q, \deg(\mathcal{F}')\} \leq \max\{d_{\overline{\mathcal{F}}_f}, q, \deg(\mathcal{F}')\}.$$

Proof. By Remark 5.2.1 we may assume that the Weil descent is done with respect to a normal basis $\{\theta, \theta^q, \ldots, \theta^{q^{n-1}}\}$ of k/k'. Set

$$\mathcal{G} = \{\overline{f}, \overline{f^q}, ..., \overline{f^{q^{n-1}}} : f \in \mathcal{F}\} \cup \{X_{ij}^q - X_{i\ j+1} : i = 0, \ldots, m-1,\ j = 0, \ldots, n-1\}.$$

Note that we have $\overline{\mathcal{F}}_f \subseteq \mathcal{G}$. Note furthermore that both sets generate the same ideal since

$$\overline{f^{q^l}} \equiv_{\overline{\mathcal{F}}_f, \infty} \overline{f}^{q^l}$$

by Lemma 5.2.2ii. Hence we have $d_{\mathcal{G}} \leq d_{\overline{\mathcal{F}}_f}$ (Proposition 5.1.6vi, vii).

Since k/k' is a separable extension, the matrix $(\theta^{q^{i+j}})_{i,j=0}^{n-1}$ is invertible (independence of characters). Consider the linear change of variables defined by

$$Y_{ij} = \sum_{k=0}^{n-1} \theta^{q^{j+k}} X_{ik}.$$

By convention, we set $Y_{ij} = Y_{i\ j \pmod n}$. We first notice that the field equations of the two

systems are the same up to a linear change of equations:

$$\begin{aligned} Y_{ij}^q - Y_{i\ j+1} &= \sum_{k=0}^{n-1} \theta^{q^{j+k+1}} X_{ik}^q - \sum_{k'=0}^{n-1} \theta^{q^{j+1+k'}} X_{ik'} \\ &= \sum_{k=0}^{n-1} \theta^{q^{j+k+1}} (X_{ik}^q - X_{ik}). \end{aligned}$$

We claim:

$$\overline{f^{q^l}}(\ldots, Y_{ij}, \ldots) \equiv \sum_{k=0}^{n-1} \theta^{q^{k+l}} [f]_k \pmod{X_{ij}^q - X_{ij}, i = 0, \ldots, m-1,\ j = 0, \ldots, n-1}.$$

It is enough to prove the claim for $f = c \prod_{i=0}^{m-1} X_i^{e_i}$, since both Weil descent models are additive.

Let $X_i^{e_i'}$ be the remainder of division of $X_i^{e_i}$ by $X_i^{q^n} - X_i$ and $e_i' = \sum_{j=0}^{n-1} a_{ij} q^j$ with $a_{ij} \in \{0, 1, \ldots, q-1\}$.

This gives modulo $Y_{ij}^q - Y_{i\ j+1}$

$$\overline{f^{q^l}}(..., Y_{ij}, ...) = c^{q^l} \prod_{i=0}^{m-1} \prod_{j=0}^{n-1} Y_{i\ j+l}^{a_{ij}}.$$

Furthermore, modulo $X_{ij}^q - X_{ij}$, we have

$$\begin{aligned} f^{q^l}(\ldots, \sum_{k=0}^{n-1} \theta^{q^k} X_{ik}, \ldots) &= c^{q^l} \prod_{i=0}^{m-1} (\sum_{k=0}^{n-1} \theta^{q^k} X_{ik})^{q^l e_i} \equiv c^{q^l} \prod_{i=0}^{m-1} (\sum_{k=0}^{n-1} \theta^{q^k} X_{ik})^{q^l e_i'} \\ &= c^{q^l} \prod_{i=0}^{m-1} (\sum_{k=0}^{n-1} \theta^{q^k} X_{ik})^{q^l \sum_{j=0}^{n-1} a_{ij} q^j} \\ &\equiv c^{q^l} \prod_{i=0}^{m-1} \prod_{j=0}^{n-1} (\sum_{k=0}^{n-1} \theta^{q^{k+l+j}} X_{ik})^{a_{ij}}. \end{aligned}$$

Thus we get the following equation from the above two identities modulo $X_{ij}^q - X_{ij}$, since

$[f]_k^q \equiv [f]_k$:

$$\overline{f^{q^l}}(\ldots, Y_{ij}, \ldots) \equiv f^{q^l}(\ldots, \sum_{k=0}^{n-1} \theta^{q^k} X_{ik}, \ldots) \equiv \left(\sum_{k=0}^{n-1} \theta^{q^k} [f]_k\right)^{q^l} \equiv \sum_{k=0}^{n-1} \theta^{q^{k+l}} [f]_k.$$

In other words, there exist polynomials $h_{ij}^{(l)} \in S$, such that

$$\overline{f^{q^l}}(\ldots, Y_{ij}, \ldots) = \sum_{k=0}^{n-1} \theta^{q^{k+l}} [f]_k + \sum_{i,j} h_{ij}^{(l)} (X_{ij}^q - X_{ij}).$$

One has $\deg(\overline{f^{q^l}}) = \deg(\overline{f}) = \max_k(\deg([f]_k))$ by [28, Proposition 3.2]. Since $\{X_{ij}^q - X_{ij} : i = 0, \ldots, m-1, j = 0, \ldots, n-1\}$ forms a Gröbner basis for any graded order, we may assume that $\deg(h_{ij}^{(l)}(X_{ij}^q - X_{ij})) \leq \deg(\overline{f^{q^l}})$.

Hence we have shown that the systems $\mathcal{G}$ and $\mathcal{F}'_f$ can be obtained from each other through an invertible linear change of variables and a change of polynomials. From Proposition 5.1.6iv,v we conclude

$$\max\{d_{\mathcal{F}'_f}, q, \deg(\mathcal{F}')\} = \max\{d_{\mathcal{G}}, q, \deg(\mathcal{F}')\} \leq \max\{d_{\overline{\mathcal{F}}_f}, q, \deg(\mathcal{F}')\}.$$

□

5.3.2 GCD computations

Let q be a prime power and let k be a finite field of cardinality q^n. Let $\mathcal{F} \subset k[X]$ be a finite set (hence we set $m = 1$). Consider the Weil descent system $\overline{\mathcal{F}}_f$ introduced in 5.2.2. Define $\equiv_j$ and V_j with respect to $\overline{\mathcal{F}}_f$. For $e \in \mathbb{Z}_{\geq 0}$, write $e = \sum_i a_i q^i$ in base q with $a_i \in \{0, 1, ..., q-1\}$, we set $w(e) = \sum_i a_i$. For $f = \sum_i b_i X^i \neq 0$, we set $w(f) = \max(w(i) : b_i \neq 0)$. Note that $w(f) \geq \deg(\overline{f})$, with equality if $\deg(f) < q^n$.

We start with a technical lemma, which is one of the main ingredients in the proof of the

main theorem.

Lemma 5.3.2. *[26] Let $h_2 \in k[X]$ nonzero of degree d. Set $u = \tau(2d, q, 1)$. Assume $\overline{h_2} \equiv_u 0$. Let $h_1 \in k[X]$. Let h_3 be the remainder of division of h_1 by h_2. Then one has $\overline{h_1} \equiv_{\max\{u, w(h_1)\}} \overline{h_3}$.*

Proof. If $d = 0$, the result follows by definition. Now we assume $d > 0$.

Write $h_2 = \sum_{i=0}^{d} b_i X^i$ where $b_d \neq 0$. It suffices to prove the result for $h_1 = X^e$ as taking remainders is additive. Let r_e denote the remainder of division of X^e by h_2. For $g \in k[X]$ with $\deg(g) \leq d$, one has $\deg(\overline{g}) \leq \tau(d, q, 1/2)$ by lemma 5.2.4. Furthermore, we have $\tau(d, q, 1/2) \leq u/2$ by a simple computation. In particular, we have $\deg(\overline{r_e}) \leq u/2$.

We will prove the following statements successively:

(i) for $e \in \{0, 1, \ldots, qd-1\}$, we have $\overline{X^e} \equiv_u \overline{r_e}$;

(ii) if e, e' satisfy $w(e) + w(e') \leq u$, $\overline{X^e} \equiv_u \overline{r_e}$ and $\overline{X^{e'}} \equiv_u \overline{r_{e'}}$, then $\overline{X^{e+e'}} \equiv_u \overline{r_{e+e'}}$;

(iii) for e with $w(e) \leq u$, we have $\overline{X^e} \equiv_u \overline{r_e}$;

(iv) one has $\overline{X^e} \equiv_{\max\{u, w(e)\}} \overline{r_e}$.

i: For $e = 0, \ldots, d-1$, the remainder is X^e itself and the result follows. One has $r_d = \frac{-1}{b_d} \sum_{i=0}^{d-1} b_i X^i$ and thus the condition $\overline{h_2} \equiv_u 0$ implies $\overline{X^d} \equiv_u \overline{r_d}$. We continue by induction. Assume the statement holds for cases smaller than e and that $e \leq qd - 1$. In the following we prove the statement for e. Write $r_{e-1} = \sum_{j=0}^{d-1} c_j X^j$. Note that r_e is the remainder of division of $X r_{e-1}$ by h_2, which gives $r_e = \sum_{j=0}^{d-1} c_j r_{j+1}$. Note that $e - 1 \leq qd - 2 = q^{\log_q(d)+1} - 2$. Write $e - 1 = a_0 + a_1 q + ... + a_s q^s$ with $a_i \in \{0, 1, \ldots, q-1\}$ and thus $s = \lfloor \log_q(e-1) \rfloor < 1 + \log_q(d)$. We have the following two cases:

1. if all $a_i = q - 1$, i.e $e - 1 = q^{s+1} - 1 \leq qd - 2$, this gives $s + 1 \leq \log_q(qd - 1) < 1 + \log_q(d)$ and thus $\deg(\overline{X^{e-1}}) = (q-1)(s+1) < (q-1)(1 + \log_q(d)) \leq (q-1)(2 + \log_q(d)) - 1$.

2. if some $a_i \neq q-1$, then $\deg(\overline{X^{e-1}}) \leq a_0+...+a_s \leq (q-1)(s+1)-1 < (q-1)(2+\log_q(d))-1$.

Thus we have proven $\deg(\overline{X^{e-1}}) \leq \lfloor (q-1)\left(\log_q(d)+2\right)-1 \rfloor$.

So we have

$$\deg(\overline{X}) + \deg(\overline{X^{e-1}}) \leq 1 + \lfloor (q-1)\left(\log_q(d)+2\right)-1 \rfloor = \lfloor (q-1)\left(\log_q(d)+2\right) \rfloor \leq u.$$

Using Lemma 5.2.2 and the induction hypothesis, we find

$$\overline{X^e} \equiv_u \overline{X} \cdot \overline{X^{e-1}} \equiv_u \overline{X} \cdot \overline{r_{e-1}} \equiv_u \overline{\sum_{j=0}^{d-1} c_j X^{j+1}} \equiv_u \overline{\sum_{j=0}^{d-1} c_j r_{j+1}},$$

and this gives the required remainder.

ii: Without loss of generality assume that $w(e') \leq u/2$. Then one has $u \geq \max(w(e) + w(e'), \deg(\overline{r_e}) + w(e'), \deg(\overline{r_e}) + \deg(\overline{r_{e'}}))$ and one has $\deg(r_e r_{e'}) \leq 2d-2 \leq qd-1$. Lemma 5.2.2 and i give

$$\overline{X^{e+e'}} \equiv_u \overline{X^e} \cdot \overline{X^{e'}} \equiv_u \overline{r_e} \cdot \overline{X^{e'}} \equiv_u \overline{r_e} \cdot \overline{r_{e'}} \equiv_u \overline{r_e r_{e'}} \equiv_u \overline{r_{e+e'}}.$$

Note that $r_{e+e'}$ is the remainder of $r_e r_{e'}$ divided by h_2 and thus the last $\equiv_u$ in the above equation follows by i.

iii: Using ii and induction, we easily reduce to the case where $e = q^i$. Note that $q^i = q \cdot q^{i-1}$ and that $u \geq q$. We can then apply ii and the proof follows by induction.

iv: We prove this statement by induction on $w(e) > u$. Write $e = e_1 + e_2$ with $u \leq w(e_1) < w(e)$, and $w(e_1) + w(e_2) = w(e)$. One has (Lemma 5.2.2 and iii)

$$\begin{aligned} \overline{X^e} &\equiv_{\max\{u,w(e)\}} \overline{X^{e_1}} \cdot \overline{X^{e_2}} \equiv_{\max\{u,w(e)\}} \overline{r_{e_1}} \cdot \overline{X^{e_2}} \\ &\equiv_{\max\{u,w(e)\}} \overline{r_{e_1}} \cdot \overline{r_{e_2}} \equiv_{\max\{u,w(e)\}} \overline{r_e}. \end{aligned}$$

□

The above lemma allows us to use the Euclidean algorithm to compute a gcd.

Proposition 5.3.3. *[26] Assume $\mathcal{F} = \{f\}$ with f nonzero. Set $u = \tau(2\deg(f), q, 1)$ and set $g = \gcd(f, X^{q^n} - X)$. We have: $\overline{g} \in V_u$.*

Proof. Let f_1 be the remainder of division of $X^{q^n} - X$ by f. One has $\overline{f} \equiv_u 0$. By Lemma 5.3.2, we have $\overline{f_1} \equiv_u 0$. Let f_2 be the remainder of division of f by f_1. Similarly, we find $\overline{f_2} \equiv_u 0$. Hence we can follow the Euclidean algorithm and we obtain $\overline{g} \in V_u$. □

5.3.3 Last fall degree of Weil descent systems

For a finite subset $\mathcal{F} \subset R$, we denote by $Z(\mathcal{F})$ the set of zeros of $\mathcal{F}$ over $\overline{k}$. Let k'' be a field extension of k. For $i = 0, \ldots, m-1$, we write

$$\pi_{i,\mathcal{F},k''} = \prod_{x \in \{x_i:\ \exists (x_0,\ldots,x_{m-1}) \in Z(\mathcal{F}) \cap k''^m\}} (X_i - x) \in k[X_i].$$

We write $\pi_{i,\mathcal{F}}$ for $\pi_{i,\mathcal{F},\overline{k}}$.

We are finally ready to prove the main theorem (Theorem 1.1.1).

Theorem 5.3.4. *Let k be a finite field of cardinality q^n. Let $\mathcal{F} \subset R$ be a finite subset. Let I be the ideal generated by $\mathcal{F}$. Assume that the following hold:*

- *I is zero-dimensional, say one has $|Z(\mathcal{F})| \leq s$;*
- *I is radical;*
- *there is a coordinate t such that the projection map $Z(\mathcal{F}) \to \overline{k}$ to coordinate t is injective;*

Let $\mathcal{F}'_f$ be the Weil descent system of $\mathcal{F}$ to the subfield k' of cardinality q using some basis of k/k', together with the field equations (Subsection 5.2.1). Then one has

$$d_{\mathcal{F}'_f} \leq \max\left(\tau(\max(d_{\mathcal{F}}, \deg(\mathcal{F}), (m+1)s, 1), q, m), m \cdot \tau(2s, q, 1), q\right).$$

Proof. We have $d_{\mathcal{F}'_f} \leq \max(d_{\overline{\mathcal{F}}_f}, q, \deg(\mathcal{F}'))$ by Proposition 5.3.1.

Without loss of generality, we may assume that $t = 0$. We can then write

$$Z(\mathcal{F}) = \{(a, \gamma_1(a), \ldots, \gamma_{m-1}(a)) : a \in \overline{k},\ \pi_{0,\mathcal{F}}(a) = 0\}$$

for some $\gamma_i \in k[X_0]$ of degree $< s$ by the Lagrange interpolation formula and by Galois theory. Indeed, we can just put

$$\gamma_i = \sum_{x=(x_0,\ldots,x_{m-1})\in Z(\mathcal{F})} x_i \prod_{(x'_0,\ldots,x'_{m-1})\in Z(\mathcal{F})\setminus\{x\}} \frac{X_0 - x'_0}{x_0 - x'_0}.$$

Note that $\gcd(\pi_{0,\mathcal{F}}, X_0^{q^n} - X_0) = \pi_{0,\mathcal{F},k}$ and one also has

$$Z(\mathcal{F}) \cap k^m = \{(a, \gamma_1(a), \ldots, \gamma_{m-1}(a)) : a \in \overline{k},\ \pi_{0,\mathcal{F},k}(a) = 0\}.$$

Set $r_0 = \max(d_{\mathcal{F}}, s, 1)$. By definition we have $\pi_{i,\mathcal{F}}, X_j - \gamma_j \in V_{\mathcal{F},r_0}$, since I is radical. Set $r_1 = \tau(r_0, q, m)$. By Lemma 5.2.5, we have $\overline{\pi_{i,\mathcal{F}}}, \overline{X_j - \gamma_j} \in V_{\overline{\mathcal{F}}_f,r_1}$. Set $r_2 = \max(r_1, \tau(2s, q, 1))$. We have $\overline{\pi_{0,\mathcal{F},k}}, \overline{\pi_{j,\mathcal{F}}}, \overline{X_j - \gamma_j} \in V_{\overline{\mathcal{F}}_f,r_2}$ (for $j = 1, \ldots, m-1$) by Proposition 5.3.3.

Now consider the system

$$\mathcal{G} = \{\pi_{0,\mathcal{F},k}, \pi_{1,\mathcal{F}}, \ldots, \pi_{m-1,\mathcal{F}}\} \cup \{X_1 - \gamma_1, \ldots, X_{m-1} - \gamma_{m-1}\}.$$

We have $\overline{\mathcal{G}} \subseteq V_{\overline{\mathcal{F}}_f,r_2}$. Let I' be the ideal generated by $\mathcal{F}_f$. Note that I' is the same as the ideal generated by $\mathcal{G}$, because both ideals are radical and have the same zero set. We first bound $d_{\mathcal{G}}$. Let $h \in I'$. One easily obtains

$$h \equiv_{\mathcal{G},\deg(h)} h'$$

for some $h' \in R$ with $\deg_{X_i}(h') < s$ using $\pi_{0,\mathcal{F},k}$ and $\pi_{i,\mathcal{F}}$ $(i = 1, \ldots, m-1)$. Then one can

replace X_i $(i > 0)$ with γ_i and do reductions with $\pi_{0,\mathcal{F},k}$ to make a polynomial in $k[X_0]$ and conclude

$$h \equiv_{\mathcal{G},\max(\deg(h),(m+1)s)} 0.$$

Hence we have $d_{\mathcal{G}} \leq (m+1)s$.

Let $h \in S$. We first claim that there is $h_1 \in R$ with $\deg_{X_i}(h_1) < s$ and

$$h \equiv_{\overline{\mathcal{F}}_f,\max\{\deg(h),m\cdot\tau(2s,q,1),r_2\}} \overline{h_1}.$$

We may assume that h is a monomial. By Lemma 5.2.2iii, there is a $h_3 \in R$ with $\deg_{X_i}(h_3) < q^n$ with $h \equiv_{\overline{\mathcal{F}}_f,\deg(h)} \overline{h_3}$. Note that h_3 can be chosen to be a monomial, say $h_3 = X_0^{a_0} \cdots X_{m-1}^{a_{m-1}}$. Set $w_i = \deg(\overline{X_i^{a_i}})$. Without loss of generality, we may assume $w_0 \geq w_1 \geq \ldots \geq w_{m-1}$. Let j be maximal such that $w_j > \tau(2s,q,1)$. Let g_i be the remainder of division of $X_i^{a_i}$ by $\pi_{i,\mathcal{F}}$ (and by $\pi_{0,\mathcal{F},k}$ if $i = 0$). By Lemma 5.3.2 for $i = 0,\ldots,j$ we have

$$\overline{X_i^{a_i}} \equiv_{\overline{\mathcal{G}}_f,w_i} \overline{g_i}$$

and for $i = j+1,\ldots,m-1$ we have

$$\overline{X_i^{a_i}} \equiv_{\overline{\mathcal{G}}_f,\tau(2s,q,1)} \overline{g_i}$$

We find (Remark 5.1.4)

$$\overline{X_0^{a_0}} \cdots \overline{X_j^{a_j}} \equiv_{\overline{\mathcal{G}}_f,w_0+\ldots+w_j} \overline{g_0} \cdots \overline{g_j}.$$

We obtain by Lemma 5.2.2ii and Remark 5.1.4:

$$\begin{aligned} h &\equiv_{\overline{\mathcal{F}}_f,\deg(h)} && \overline{h_3} \equiv_{\overline{\mathcal{F}}_f,\deg(h)} \overline{X_0^{a_0}} \cdots \overline{X_{m-1}^{a_{m-1}}} \\ &\equiv_{\overline{\mathcal{F}}_f,\max(\deg(h),m\cdot\tau(2s,q,1),r_2)} && \overline{g_0} \cdots \overline{g_j} \cdot \overline{X_{j+1}^{a_{j+1}}} \cdots \overline{X_{m-1}^{a_{m-1}}} \\ &\equiv_{\overline{\mathcal{F}}_f,\max(\deg(h),m\cdot\tau(2s,q,1),r_2)} && \overline{g_0} \cdots \overline{g_{m-1}} \\ &\equiv_{\overline{\mathcal{F}}_f,\max(\deg(h),m\cdot\tau(2s,q,1),r_2)} && \overline{g_0 \cdots g_{m-1}}. \end{aligned}$$

This finishes the proof of the claim.

Let $\overline{I}$ be the ideal generated by $\overline{\mathcal{F}}_f$. Assume $h \in \overline{I}$. By the above there is $h_1 \in R$ with $\deg_{X_i}(h_1) < s$ and

$$h \equiv_{\overline{\mathcal{F}}_f,\max(\deg(h),m\cdot\tau(2s,q,1),r_2)} \overline{h_1}.$$

From Lemma 5.2.3 it follows that $h_1 \in I'$. We have $h_1 \in V_{\mathcal{G},(m+1)s}$ by the above. From Lemma 5.2.5 we have $\overline{h_1} \in V_{\overline{\mathcal{G}}_f,\tau((m+1)s,q,m))}$. Hence we conclude:

$$h \in V_{\overline{\mathcal{F}}_f,\max(\deg(h),\tau((m+1)s,q,m),m\cdot\tau(2s,q,1),r_2)}$$

where $r_2 = \max(r_1, \tau(2s,q,1)) = \max(\tau(\max(d_{\mathcal{F}}, s, 1), q, m), \tau(2s,q,1))$. Summarizing, this gives

$$h \in V_{\overline{\mathcal{F}}_f,\max(\deg(h),\tau(\max((m+1)s,d_{\mathcal{F}},1),q,m),m\cdot\tau(2s,q,1))}.$$

The result then follows. □

5.3.4 Possible improvements of the main theorem

In this subsection, we will discuss how one can improve Theorem 5.3.4. Our main goal is to obtain a result for which the last fall degree of a Weil descent system does not depend on n.

If one reads the proof carefully, one notices that one can replace $(m+1)s$ by $m(s-1)-1+(s-1) = (m+1)(s-1)-1$ if $m > 1$. For $m = 1$, one can prove a much simpler theorem using mostly Proposition 5.3.3. The result is the following statement.

Theorem 5.3.5. *Let k be a finite field of cardinality q^n. Assume $m = 1$. Let $\mathcal{F} \subset R$ be a finite subset. Let $d \in \mathbb{Z}_{\geq 0}$ such that there $\exists f \in \mathcal{F}$ with $0 \leq \deg(f) \leq d$, and such that for all $g \in \mathcal{F}$ we have $\deg(\overline{g}) \leq \tau(2d, q, 1)$. Let $\mathcal{F}'_f$ be the Weil descent system of $\mathcal{F}$ to the subfield k' of cardinality q using some basis of k/k', together with the field equations (Subsection 5.2.1). Then one has*

$$d_{\mathcal{F}'_f} \leq \max(\tau(2d, q, 1), q).$$

Proof. (Sketch) As in the proof of Theorem 5.3.4, we work with the system $\overline{\mathcal{F}}_f$.

Set $u = \tau(2d, q, 1)$ and set $g = \gcd(\mathcal{F} \cup \{X^{q^n} - X\})$. Using Lemma 5.3.2 and Proposition 5.3.3, one can prove $\overline{g} \equiv_u 0$.

Let $h \in \overline{I}$. By Lemma 5.2.2iii, one has $h \equiv_{\deg(h)} \overline{h_2}$ for some $h_2 \in k[X]$. Since $\overline{h_2} \in \overline{I}$, it follows from Lemma 5.2.3ii that $h_2 \in I$. Hence h_2 has remainder 0 when divided by g. From Lemma 5.3.2, we conclude

$$h \equiv_{\max(\deg(h),u)} \overline{h_2} \equiv_{\max(\deg(h),u)} 0.$$

This finishes the proof. □

One can also study the Weil descent of a system $\mathcal{H}$ which consists of $\mathcal{F}$ and some polynomials in one of the variables of weight at most $\tau(2s, q, 1)$ (such as linear subspace constraints). One

can easily generalize as in Theorem 5.3.5 and exactly the same result should hold (the extra polynomials do not play a role). We did not use this formulation, because it looks a bit more complex.

We believe that the three conditions in the theorem, I is zero-dimensional, I is radical and a projection map is injective, can be replaced by the condition $\dim_k(R/I) \leq s$, but we do not know how to prove this. The following lemma says that if $|k| > \binom{s}{2}$, that after a linear change of variables the last condition automatically holds.

Lemma 5.3.6. *Let k be a field, $n \in \mathbb{Z}_{\geq 0}$ and let $v_1, \ldots, v_r \in k^n$ be distinct. Assume that $|k| > \binom{r}{2}$. Then there exists a matrix $A \in \mathrm{GL}_n(k)$ such that the first coordinates $Av_1, \ldots, Av_r$ are pairwise distinct.*

Proof. Assume that k is a finite field. Let $q = |k|$. Let $\langle\ ,\ \rangle$ be the standard inner product on k^n. It is equivalent to find $y \in k^n$ such that $\langle y, v_1\rangle, \ldots, \langle y, v_r\rangle$ are distinct, that is, such that for $i \neq j$ one has $\langle y, v_i - v_j\rangle \neq 0$. There q^{n-1} vectors y with $\langle y, v_i - v_j\rangle = 0$. There are at least $q^n - \binom{r}{2}q^{n-1}$ vectors which make none of the inner products zero. Hence if $q^n > \binom{r}{2}q^{n-1}$, the result follows. The proof for an infinite field follows in a similar way. □

With our techniques it seems impossible to remove the condition that the system is zero-dimensional (see also Section 5.5).

5.4 Multi-HFE

In this section we discuss the security of a version of the multi-HFE public key cryptosystem. Let us first describe the idea of this cryptosystem. The idea of multi-HFE is that it is easy to solve zero-dimensional systems with few variables, but it becomes harder when the number of variables increases (the complexity should be exponential in the number of variables). Using Weil descent, one can construct a system with a lot of variables from a system with only a

few variables. One can use this construction as a trap door. The idea of HFE (hidden field equations) was first introduced in [37]. We discuss a version of multi-HFE below.

We pick a public field k of cardinality q and public integers m, n (where m is not too big; the variant with $m = 1$ is called simply HFE). We let k_n be a field extension of k of degree n. The message space will be $k^{m\times n}$. The private key consists of a zero-dimensional system

$$\mathcal{F} \subset k_n[X_0, \ldots, X_{m-1}]$$

of at least $m + 2$ equations. The owner of the private key computes a Weil descent system $\{f'_0, \ldots, f'_{r-1}\} = \mathcal{F}' \subseteq k[X_{ij} : i = 0, \ldots, m-1, j = 0, \ldots, n-1]$ of $\mathcal{F}$ to k (with $r > mn$) and he stores the basis chosen for the Weil descent. Furthermore, also part of the private key are an $A \in \mathrm{Aff}_{nm}(k)$ and $B \in \mathrm{GL}_r(k)$. He then constructs a system $\mathcal{F}'' = \{f''_0, \ldots, f''_{r-1}\}$ defined by

$$[f''_0, f''_1, \ldots, f''_{r-1}]^T = B[Af'_0, Af'_1, \ldots, Af'_{r-1}]^T.$$

This system $\mathcal{F}''$ is made public and is a disguised Weil descent system (here A makes an affine change of variables, and B makes linear combinations of the equations themselves).

To encrypt a message $M \in k^{m\times n}$, one computes

$$M' = \mathrm{encr}(M) = (f''_0(M), f''_1(M), \ldots, f''_{r-1}(M)) \in k^r.$$

To decrypt $M' \in k^r$, one needs to find the usually unique M with $M' = \mathrm{encr}(M)$. One computes $[m''_0, \ldots, m''_{r-1}]^T = B^{-1}M'$. Consider the system $\mathcal{F}'_{M'} = \{f'_i - m''_i : i = 0, \ldots, r-1\}$. One easily finds $m_f \in k_n$ such that $\mathcal{F}'_{M'}$ is the Weil descent of $\mathcal{F}_{M'} = \{f - m_f : f \in \mathcal{F}\}$. The latter system is a system in a small number of variables with a usually unique solution. Hence one should be able to solve this system efficiently with the private key (with say a Gröbner basis algorithm) and find the solution M_0. One then finds $M = A^{-1}M_0$.

Without knowing the private keys, one might be tempted to solve

$$M' = (f_0''(M), f_1''(M), \ldots, f_{r-1}''(M))$$

directly, with the restriction that $M \in k^{m \times n}$. This is a system in $m \times n$ variables, which is a priori very hard to solve, especially when the security parameter n is chosen to be quite large. One expects that the degree of regularity or last fall degree of such a system, when the system $\mathcal{F} \subset k[X_0, \ldots, X_{m-1}]$ is fixed or if $d_{\mathcal{F}}$ is bounded, grows with n, resulting in a time which is exponential in n to solve the system.

Our results however, show that if $\mathcal{F} \subset k[X_0, \ldots, X_{m-1}]$ is fixed, and under some mild restrictions on the system $\mathcal{F}'_{M'}$ (which should almost always hold in practice), the last fall degree of such a system does not depend on n (Theorem 5.3.4). This shows that in most cases, one can solve such system using Proposition 5.1.12 in a way which depends only in a polynomial way on n, whereas one expects it to depend on n in say an exponential way. Similarly, with the help of Proposition 5.1.9, one sees that the degree of regularity of such a system often does not depend on n.

5.4.1 Comparison

There are various papers in literature regarding the hardness of solving HFE and multi-HFE.

One type of attack on HFE tries to solve the system $\mathcal{F}'_{M'}$, and this was first introduced in [20]. In this paper it was shown in a practical way that HFE can be cracked easily with the help of Gröbner basis computations. The maximal degree of the polynomials needed to solve the system does not seem to depend on n (see [20, Table 3]). This is very similar to our observation. The authors of [14] obtained heuristical arguments, based on the first fall degree assumption, which explain why the maximal needed degree does not grow with n. However, in [29], the authors raise doubt to the first fall degree heuristic. A first attempt to prove, without heuristics, that one can solve HFE efficiently using Gröbner basis techniques, is [38]. A first

complete proof of the complexity of solving HFE using the techniques from this chapter, can be found in [26]. See also Theorem 5.3.5. It is interesting to see that the upper bounds on say the last fall degree as in Theorem 5.3.5 we have obtained seem to be off by only a small constant factor (as in [38]).

In literature, one finds another (practical) method which has been used to attack Multi-HFE systems. See for example [5]. This strategy is different from the one we have described above, and seems unlikely to apply to our slightly more general setup. Often one restricts to a polynomial system $\mathcal{F}$ such that Weil descent only gives quadratic polynomials. That is, the only monomials appearing in $\mathcal{F}$ are of the form $X_i^{q^u} X_j^{q^v}$. This makes the Weil descent system easier and the key space much smaller. One can obtain a private key (multiple keys give equivalent systems) by solving certain systems, related to the MinRank problem, with Gröbner basis algorithms. With such a private key, one can easily solve the system. One can solve the required systems in polynomial time in n, since similar to our results, the degree of regularity of the systems does not depend on n. In [5] it is also discussed why HFE is safer than Multi-HFE and they say that choosing $m = 1$ seems to be optimal for security reasons.

As far as we are aware, our proofs are the first proofs which give complexity bounds for solving the systems $\mathcal{F}'_{M'}$ directly. Also, our systems $\mathcal{F}'_{M'}$ are more general than one usually finds in literature. We show that systems coming from multi-HFE are easier to solve than expected, that is, the last fall degree and the degree of regularity of $\mathcal{F}'_{M'}$ do not depend on n. However, we expect that our upper bounds are not as close to the true values as in the HFE case. We have not done any numerical computations to verify this.

5.5 Non zero-dimensional systems

Let k be a finite field of cardinality q and let k_n be an extension of k of degree q. Let $f \in R = k[X_0, \ldots, X_{m-1}]$ with $m \geq 2$. It has been suggested (see for example [39]) that the Weil descent system of $\{f\}$ (or in general a polynomial system which need not be zero-dimensional) from k_n

to k, the first fall degree is close to the degree of regularity, the largest degree reached during Gröbner basis computation. An example of the Weil descent of a single polynomial comes from one of the approaches to solve the elliptic curve discrete logarithm problem (ECDLP) using summation polynomials (see for example [11]). In this case the first fall degree does not depend on n and it is very tempting to adopt the first fall degree assumption as it leads to heuristically subexponential attack on ECDLP over finite fields of small characteristics. Such a subexponential algorithm would have a major impact on the security of many protocols. However more recent works (see for example [29] and [26]) have cast serious doubt on the first fall degree assumption: the degree of regularity does seem to depend on n.

What we have shown in this chapter is that to a large extent the *last* fall degree of the Weil descent system of a zero-dimensional polynomial system is independent of n (Theorem 5.3.4). This has enabled us to successfully solve HFE and multi-HFE systems with rigorously proven time complexity, as the underlying polynomial systems are zero-dimensional. Unfortunately, the system coming from a single multivariate polynomial, without field equations, is not zero-dimensional and our approach using projection polynomials does not work (Theorem 5.3.4). The system only becomes zero-dimensional when we add the field equations.

We do think that it is of great interest to study such systems coming from a single multivariate polynomial (or systems which are not zero-dimensional). We hope that the method of this chapter is a step in the right direction.

6. SPECIAL VECTOR SPACES AND APPLICATION TO BINARY ECDLP

In this chapter, we will consider the ECDLP over the field $F = \mathbb{F}_q$, see also definition 2.2.1, where $q = 2^n$ for some integer n. In Chapter 4, we have reviewed the recent works on solving ECDLP via the index calculus approach (Section 2.3). As mentioned, one of the main challenges is to construct a nice factor base that yields an efficient relation search step. Using summation polynomials, one promising approach in this direction is to consider vector spaces as factor bases and carry out the relation search by solving summation polynomials with suitable linearized constraints. Moreover, one common method to solve the latter problem is by means of Weil descent. However, some challenges remain with this line of approach:

- Find a rigorous way to estimate the complexity of the relation search;
- Is Weil descent the most efficient way to solve the polynomial system arising from the summation polynomial and the linearized constraints?
- Is the complexity for any vector space similar? In other words, are there vector spaces that result in more efficient relation search steps as compared to a random vector space?

The main focus of this chapter is to find suitable vector sub-spaces that gives rise to a more efficient relation search step for the index calculus approach. We recall the relation search step as: we seek to solve the following problem:

Problem 6.0.1. *Let V be a $\mathbb{F}_2$-vector subspace of $\mathbb{F}_q$ with dimension n'. Given a point $R \in E(\mathbb{F}_q)$, find, if any, m points $P_1, ..., P_m \in \mathcal{F}$, such that $R = P_1 + ... + P_m$.*

We investigate a sub-class of vector spaces with nice characteristic polynomials. Using these vector spaces, we transform the polynomial system into one with smaller degrees. We provide complexity bounds for our approach and give conditions such that an efficient index calculus method will result. Finally, we provide some concrete examples of vector spaces with the nice properties.

6.1 Solving a multivariate polynomial with vector space constraints

In this section, we describe a method to solve problem(6.0.1) when the $\mathbb{F}_2$-vector subspace of $\mathbb{F}_{2^n}$ with dimension n' has some nice properties.

6.1.1 Motivation

For an elliptic curve E defined over $\mathbb{F}_{2^n}$ and fixed integers m and n' with $mn' \approx n$, we typically choose $n' = \lceil \frac{n}{m} \rceil$ in our index calculus approach for optimal results. We consider the summation polynomial $S_{m+1}(x_1, \ldots, x_m, a)$, where $a = x(R)$ is the x-coordinate of some point $R \in E(\mathbb{F}_{2^n})$. Henceforth, we will simply write $f(x_1, \ldots, x_m)$ to denote $S_{m+1}(x_1, \ldots, x_m, a)$ throughout this chapter.

Let V be a vector subspace of $\mathbb{F}_q$ over $\mathbb{F}_2$ with dimension n'. Let $L(x)$ denote the *characteristic polynomial* of V:

$$L(x) := \prod_{v \in V} (x - v).$$

It is well known that $L(x)$ is a *linearized polynomial* over $\mathbb{F}_q$, that is,

$$L(x) = \sum_{i=0}^{n'} a_i x^{2^i}.$$

By proposition(2.4.1), to solve problem(6.0.1), it suffices to solve the following polynomial system:

$$\begin{aligned} f(x_1, \ldots, x_m) =& 0, \\ L(x_i) =& 0,\ i = 1, \ldots, m. \end{aligned} \tag{6.1}$$

In other words, we are solving a single multivariate polynomial $f(x_1, \ldots, x_m)$ with the constraints that $L(x_i) = 0$ for $i = 1, 2, \ldots, m$.

In order to motivate our method, let us review a special case. Concretely, we consider the case where n is a composite number and write $n = mn'$. In this special case, we choose the vector space as $V = \mathbb{F}_{2^{n'}}$. Recall that the usual method to solve problem6.0.1 is via Weil descent(see section5.2.1 or 4.2 for the details) together with Gröbner basis algorithms. Here, we briefly review the ideas involved. Choose a basis $\{1, w, ..., w^{n-1}\}$ of $\mathbb{F}_{2^n}$ over $\mathbb{F}_2$ and $\{v_1, ..., v_{n'}\}$ a basis of V over $\mathbb{F}_2$. Write

$$x_i = \sum_{j=1}^{n'} v_j x_{ij}$$

for new variables x_{ij},$i = 1, ..., m$,$j = 1, ..., n'$.

Substituting x_i by the above representation, we obtain

$$f\left(\sum_{j=1}^{n'} v_j x_{1j}, \ldots, \sum_{j=1}^{n'} v_j x_{mj}\right) \equiv \sum_{j=0}^{n-1} f_j w^j \qquad (\text{mod } x_{ij}^2 - x_{ij},\ i = 1, \ldots, m, j = 1, \ldots, n')$$

for some $f_i \in \mathbb{F}_2[x_{11}, ..., x_{mn'}]$ with $\deg_{x_{ij}}(f_k) \leq 1$.

Then system6.1 is equivalent to the following:

$$\begin{aligned} f_i =& 0, i = 0, ..., n-1, \\ x_{ij}^2 - x_{ij} =& 0, \ i = 1, \ldots, m, j = 1, ..., n' \end{aligned} \tag{6.2}$$

Thus, we obtain a polynomial system over $\mathbb{F}_2$ which is typically solved via Gröbner basis to obtain a solution to problem6.0.1.

Instead of deriving a polynomial system over $\mathbb{F}_2$, we now construct a system over $\mathbb{F}_q$ as follows. Since our vector space is a subfield, $L(x) = x^{2^{n'}} - x$. Write $f = \sum \alpha_M M$, where M is a monomial in $x_1, \ldots, x_m$. Observe that for any $i = 0, \ldots, m-1$,

$$g_i \equiv f^{2^{in'}} = \sum \alpha_M^{2^{in'}} M^{2^{in'}} \equiv \sum \alpha_M^{2^{in'}} M \mod (L(x_1), \ldots, L(x_m)).$$

Note that here we have $L(x_i) = x_i^{2^{n'}} - x_i$.

Let $g_i = \sum \alpha_M^{2^{in'}} M$,$i = 0, ..., m-1$. We consider the following equivalent system of system(6.1)

$$\begin{aligned} g_i =& 0, i = 0, ..., m-1, \\ L(x_i) =& 0, \ i = 1, \ldots, m. \end{aligned} \tag{6.3}$$

Once again, we can use Gröbner basis algorithms to solve this system over $\mathbb{F}_q$. Note that $g_i, i = 1, ..., m-1$ have the same monomials as f, so we have added new polynomials to system(6.1) without increasing the number of monomials and thus we make the system(6.1) more over-defined through introducing these new polynomials. In practice, an over-defined system is widely believed to be solved more easily via Gröbner basis methods, so we expect

that system(6.3) may be easier to solve. We conducted some experiments to compare this method and the method of Weil descent. The experimental result shows that our method uses less computation time than the usual Weil descent method, i.e computing the Gröbner basis of system(6.3) uses less time than computing the Gröbner basis of system(6.2). The experimental result is recorded in table(6.1).

Tab. 6.1: The computation time of Weil descent method and our method

n	n'	m	$TimeW(s)$	$TimeO(s)$
12	4	3	0.31	0.19
15	5	3	2.87	0.59
18	6	3	28.91	0.6
21	7	3	129.95	21.97
24	8	3	1404.2	172.93

In the table, TimeW denotes the time for the Weil descent method and TimeO denotes the time for our method, both expressed in seconds.

Next, we analyze the time complexity of our approach.

For a symmetric polynomial $g \in \mathbb{F}_q[x_1, ..., x_m]$, let $\tilde{g}(s_1, \ldots, s_m)$ be the corresponding polynomial in the elementary symmetric variables $s_1, \ldots, s_m$, where

$$s_i = \sum_{1 \le j_1 < j_2 < \ldots < j_i \le m} x_{j_1} x_{j_2} \ldots x_{j_i}.$$

Now we consider the following system:

$$\tilde{g_i}(s_1, \ldots, s_m) = 0, i = 0, ..., m-1, s_1, ..., s_m \in \mathbb{F}_{2^{n'}}. \tag{6.4}$$

As each x_i is in a subfield, the elementary symmetric variables s_i are in the subfield as well. We have omitted the characteristic equations $L(s_i) = 0$ as they are field equations. Thus, we wish to solve the system over the field $\mathbb{F}_q$.

We follow the approach of Diem [12, Introduction] to solve the above system.

By lemma(6.1.3) below, we know that each polynomial in system(6.4) has total degree bounded by 2^{m-1}. Thus system(6.4) has at most $2^{m(m-1)}$ solutions over $\mathbb{F}_{2^{n'}}$. One can use an algorithm by M. Rojas([41]) to solve this system. The time complexity is polynomial in $2^{m(m-1)} \cdot \log(2^n)$. In particular, we suppose it is bounded by $(2^{m(m-1)} \cdot \log(2^n))^{C_1}$ for some constant C_1.

We invoke two standard heuristic assumptions in the literature below:

Assumption 6.1.1. *Assume the following:*

(a)

$$\#\mathcal{F} \approx 2^{n'};$$

(b) *the probability of a point $R \in E(\mathbb{F}_q)$ that can be splitted as a sum of m points in $\mathcal{F}$ is roughly $\frac{1}{m!}$.*

Note that once we get a solution of system(6.4), we can recover x_i (if they exist) by factoring a univariate polynomial with degree m over $\mathbb{F}_{2^{n'}}$. This can be done in time polynomial in $\max(\log(2^{n'}), m)$ in a probabilistic way [50]. We suppose it is bounded by $\max(\log(2^{n'}), m)^{C_2}$ for some constant C_2.

By the above analysis, the time complexity of the relation search step(collect roughly $2^{n'}$ relations) is bounded by:

$$2^{n'} \cdot m! \cdot \left((2^{m(m-1)} \cdot \log(2^n))^{C_1} + 2^{m(m-1)} \cdot (\max(\log(2^{n'}), m))^{C_2} \right).$$

A simple computation shows that the above number is bounded by:

2^t, where $t \approx n' + m \log m + C \cdot (\log n + m^2)$, where C is a constant.

Next, the linear algebra step needs time roughly $(2^{n'})^w$, where w is the linear algebra constant. The time to compute the summation polynomial is roughly 2^{t_1} with $t_1 \approx m(m+1)$ [39].

Thus the total time to solve ECDLP for this case via index calculus is:

$$2^t + 2^{t_1} + (2^{n'})^w.$$

Using the same method as Petit et al. [39, Section 5.3], let $n' := n^\alpha$ and $m := n^{1-\alpha}$, and take $\alpha = \frac{2}{3}$. It follows that the total time complexity is $O(2^{cn^{\frac{2}{3}}})$, which is subexponential.

Note that the above time complexity obtained is under the condition that $n = m \cdot n'$ with $n' = n^{\frac{2}{3}}$ and $m = n^{\frac{1}{3}}$ and some heuristic assumptions.

Finally, let us investigate the complexity of solving system(6.4) via Gröbner basis algorithms. Specifically, we conducted some experiments with the Magma Computational Algebra system to investigate the degree of regularity of m randomly generated polynomials over $\mathbb{F}_{2^n}$ in m variables, each with degree bounded d_i. Table(6.2) record our results.

From the table(6.2), we see that the degree of regularity of the system is of the form $\sum_i d_i + c(m, d_i)$, where $c(m, d_i)$ depends on m and d_i. According to the Macaulay bound, the degree of regularity of a regular system of m polynomials f_i in m variables is bounded by $B = 1 + \sum_i(\deg(f_i) - 1) = \sum_i md_i + 1 - m$. It seems reasonable to conjecture that the degree of regularity of our system is bounded by B. Consequently, the complexity of solving system(6.4) is $O((B+m)^{mw}) = O(2^{wm^2})$, where w is the linear algebra constant. This complexity is identical to that derived in our analysis above and thus, yields the same time complexity for the whole index calculus algorithm.

The above experimental results and time complexity analysis motivate us to ask:

When n is prime, can we modify the above method to obtain a nice time complexity for ECDLP?

Before we turn to that question, we observe that the complexity to solve system(6.4) plays an important part in deriving a sub-exponential complexity for our entire algorithm. In our system, the polynomials $\tilde{g}_i$ have relatively small degree. This is primarily because we have constructed m different g_i's with the same degree as f. In particular, the nice structure of the

Tab. 6.2: The degree of regularity of random polynomial system

n	m	d_i	D_{reg}
21	2	$(2,2)$	5
21	2	$(2,3)$	6
21	2	$(2,4)$	7
21	2	$(2,5)$	8
21	2	$(2,10)$	13
21	2	$(3,3)$	7
21	2	$(3,4)$	8
21	2	$(3,5)$	9
21	2	$(3,6)$	10
21	2	$(3,7)$	11
21	2	$(4,4)$	9
21	2	$(4,5)$	10
21	2	$(4,6)$	11
21	2	$(4,7)$	12
21	2	$(4,8)$	13
21	2	$(4,9)$	14
21	2	$(5,5)$	11
21	2	$(5,6)$	12
21	2	$(5,7)$	13
21	2	$(5,8)$	14
21	2	$(5,9)$	15
21	2	$(5,10)$	16
21	2	$(6,6)$	13
21	2	$(7,7)$	15
21	2	$(8,8)$	17
21	2	$(9,9)$	19
21	2	$(10,10)$	21
21	2	$(11,11)$	23
21	2	$(12,12)$	25
21	2	$(13,13)$	27
21	2	$(20,20)$	41
21	2	$(30,30)$	61
21	2	$(40,40)$	81

n	m	d_i	D_{reg}
21	3	$(2,2,2)$	7
21	3	$(2,3,4)$	11
21	3	$(2,3,5)$	12
21	3	$(2,3,6)$	13
21	3	$(2,3,7)$	14
21	3	$(3,3,3)$	11
21	3	$(3,4,5)$	14
21	3	$(3,4,6)$	15
21	3	$(3,4,7)$	16
21	3	$(3,5,6)$	16
21	3	$(3,5,7)$	17
21	3	$(3,6,7)$	18
21	3	$(4,4,4)$	14
21	3	$(4,5,6)$	18
21	3	$(4,5,7)$	19
21	3	$(4,6,7)$	20
21	3	$(5,5,5)$	18
21	3	$(5,6,7)$	21
21	3	$(6,6,6)$	21
21	3	$(7,7,7)$	24
21	3	$(8,8,8)$	28
21	3	$(9,9,9)$	31
21	3	$(10,10,10)$	35
21	3	$(11,11,11)$	38
21	3	$(12,12,12)$	42
21	3	$(13,13,13)$	45
21	3	$(14,14,14)$	49
21	3	$(15,15,15)$	52

n	m	d_i	D_{reg}
21	4	$(2,2,2,2)$	10
21	4	$(2,3,4,5)$	18
21	4	$(2,3,4,6)$	19
21	4	$(2,3,5,6)$	20
21	4	$(2,4,5,6)$	21
21	4	$(3,3,3,3)$	16
21	4	$(3,4,5,6)$	23
21	4	$(4,4,4,4)$	21
21	4	$(5,5,5,5)$	26

D_{reg} denotes the degree of regularity of the random polynomial system.

linearized polynomials $L(x_i)$ helps us construct sufficient polynomials of small degree.

6.1.2 Special vector subspaces

Let $F = \mathbb{F}_{2^n}$ and let V be a $\mathbb{F}_2$-vector subspace of F with dimension n'. Similar to the analysis in the above subsection, in this section, we consider V to be a vector subspace with a nice characteristic polynomial, namely, its characteristic polynomial $L(x)$ has the following form:

$$L(x) = x^{2^{n'}} + \sum_{i=0}^{n''} c_i x^{2^i}$$

with $n'' << n'$, where $c_i \in F$.

Remark 6.1.2. For a random n'-dimensional $\mathbb{F}_2$-vector subspace of $\mathbb{F}_{2^n}$, one will expect n'' to be around n'. Now, one can compute the number of vector subspaces of dimension n' to be

$$\prod_{i=0}^{n'-1} (2^n - 2^i) / \prod_{i=0}^{n'} (2^{n'} - 2^i) \approx 2^{n'(n-n')}.$$

On the other hand, let $L(x) = x^{2^{n'}} + \sum_{i=0}^{n'-1} a_i x^{2^i}$ represent an arbitrary linearized polynomial over $\mathbb{F}_{2^n}$. There are around $2^{n(n'-n'')}$ different ways to fix the coefficients a_i for $i = n'' + 1, \ldots, n' - 1$. Hence, if we let $n'' \approx n'^2/n$, we may find a linearized polynomial with $a_i = 0$ for $i = n'' + 1, \ldots, n' - 1$. However, it remains an open problem to construct such a linearized polynomial (if it exists) or to find other linearized polynomials with smaller n''.

In this section, we suppose the existence of special vector subspaces of $\mathbb{F}_{2^n}$ (that is, whose corresponding linearized polynomials have n'' sufficiently small relative to n') for various parameters n and n'. We will use this assumption to study the complexity to solve system(6.1).

Let V be a vector subspace of $F = \mathbb{F}_{2^n}$ with characteristic polynomial $L(x)$ of the following form:

$$L(x) = x^{2^{n'}} + \sum_{i=0}^{n''} c_i x^{2^i}.$$

For $0 \leq i \leq n$, let $L_i(x)$ denote the remainder of x^{2^i} divided by $L(x)$, i.e, $L_i(x) \equiv x^{2^i} \mod L(x)$. We have the following:

- $\deg(L_i(x)) = 2^{d_i}$, for $0 \leq d_i < n'$.
- $L_n(x) = x$.
- If $i < n' - 1$, then $d_{i+1} = d_i + 1$.

Let $I_0 = \{i | d_{i-1} \neq d_i - 1, 1 \leq i \leq n\}$. Let $i_1, \ldots, i_m$ be m indices in I_0 such that $d_{i_1} \leq \ldots \leq d_{i_m}$ and the sum $d = d_{i_1} + \ldots + d_{i_m}$ is the smallest.

For a positive integer j, let $\tilde{f}_j(s_1, ..., s_m)$ denote the symmetrized polynomial of

$$f(x_1, \ldots, x_m)^{2^j} \mod (L(x_1), \ldots, L(x_m)).$$

Consider the degree reverse lexicographic order on $\mathbb{F}_q[s_1, ..., s_m]$ with $s_m > s_{m-1} > ... > s_1$. Under this monomial order, we have the following lemma.

Lemma 6.1.3.
- *$f(x_1, \ldots, x_m)^{2^i} \mod (L(x_1,), \ldots, L(x_m))$ has degree bounded by 2^{d_i+m-1} in each variable x_j.*
- *Observe that each $f(x_1, \ldots, x_m)^{2^i} \mod (L(x_1), \ldots, L(x_m))$ is symmetric. After symmetrization, $\tilde{f}_i(s_1, \ldots, s_m)$ has degree bounded by 2^{d_i+m-1}.*

Proof.
- Since $L_i(x_j) \equiv x_j^{2^i} \mod L(x_j), j = 1, \ldots, m$, we have

  $$f(x_1, \ldots, x_m)^{2^i} \equiv f^{(i)}(x_1^{2^i}, \ldots, x_m^{2^i}) \equiv f^{(i)}(L_i(x_1), \ldots, L_i(x_m)) \mod (L(x_1,), \ldots, L(x_m)),$$

  where $f^{(i)}(x_1, ..., x_m)$ has the same monomials as $f(x_1, ..., x_m)$ but with the coefficients raise to power 2^i. Note that $f(x_1, \ldots, x_m)$ has degree bounded by 2^{m-1} in each variable x_j and $\deg(L_i(x_j)) = 2^{d_i}, j = 1, \ldots, m$.

It follows that $f^{(i)}(L_i(x_1), \ldots, L_i(x_m))$ has degree bounded by 2^{d_i+m-1} in each variable x_j.

- Since $f(x_1, \ldots, x_m)$ is a symmetric polynomial, then the first statement follows easily.

  Suppose the total degree of $\tilde{f}_i(s_1, \ldots, s_m)$ is larger than 2^{d_i+m-1}. Then there exists a term $\prod_{j=1}^{m} s_j^{a_j}$ such that $\sum_{j=1}^{m} a_j > 2^{d_i+m-1}$. Note that the degree of $\prod_{j=1}^{m} s_j^{a_j}$ with respect to variable x_1 is $\sum_{j=1}^{m} a_j$. This contradicts that each variable of $f(x_1, \ldots, x_m)^{2^i}$ mod $(L(x_1), \ldots, L(x_m))$ has degree bounded by 2^{d_i+m-1}.

□

To solve system (6.1), we consider the following alternative system:

$$\begin{aligned} \tilde{f}_{i_1} &= 0, \\ &\ldots \\ \tilde{f}_{i_m} &= 0. \end{aligned} \tag{6.5}$$

Note that the choice of $i_1, \ldots, i_m$ assures that for any $1 \leq i \neq j \leq m$, $\tilde{f}_{i_i}$ is not a power of $\tilde{f}_{i_j}$. Further, note that we have omitted the linearized constraints on x_i. In particular, this system has m equations in m variables. Experiments show that with high probability, this system is zero-dimensional. Thus, we will first find the solutions of this system, solve for the corresponding x_i's and then check if they satisfy the linearized constraints.

We use a similar method as in section(6.1.1) to estimate the complexity of our approach. In order to to solve system(6.1), we first solve system(6.5). Since each polynomial in system(6.5) has degree bounded by $2^{d_{i_j}+m-1}$, thus system(6.5) has at most $2^{m(m-1)+d}$ solutions over $\mathbb{F}_{2^n}$, where $d = \sum_{j=1}^{m} d_{i_j}$. One can use an algorithm by M. Rojas([41]) to solve this system. The time complexity is polynomial in $2^{d+m(m-1)} \cdot \log(2^n)$. Assume it is bounded by $(2^{m(m-1)+d} \cdot \log(2^n))^{C_1}$ for some constant C_1. Then one can recover x_i (if they exist) by factoring a univariate polynomial with degree m over $\mathbb{F}_{2^n}$, which can be done in time polynomial in $\max(\log(2^n), m)$

in a probabilistic way [50]. We suppose it is bounded by $\max(\log(2^n), m)^{C_2}$ for some constant C_2. Finally, we check if each x_i satisfies $L(x_i) = 0$, which is trivial.

By using the same heuristic assumptions as in section(6.1.1), the time complexity of the relation search step is bounded by:

$$2^t = 2^{n'} \cdot m! \cdot \left((2^{m(m-1)+d} \cdot \log(2^n))^{C_1} + 2^{m(m-1)+d} \cdot (\max(\log(2^n), m))^{C_2} \right).$$

The time for the linear algebra step is roughly $(2^{n'})^w$, where w is the linear algebra constant, and the time to compute the summation polynomial is roughly 2^{t_1} with $t_1 \approx m(m+1)$ [39].

Consequently, the total time complexity of the index calculus approach to solve ECDLP on $\mathbb{F}_{2^n}$ is:

$$2^t + 2^{t_1} + (2^{n'})^w.$$

Since $0 \leq d < n$, the above time complexity is totally decided by d. In particular, if $d = O(n^\omega)$ for some constant $0 < \omega < 1$, we obtain a sub-exponential algorithm.

Suppose that we have $n'' \approx n'^2/n = n'/m$. In this case, one checks that $d_{in'} = in''$ for $i = 1, 2, \ldots, m-1$. In particular, we have

$$d = \sum_{i=0}^{m-1} in'' = m(m-1)n''/2 = (m-1)n' \approx n.$$

So far, it is not clear if vector spaces with small values of d exist. In the next section, we generalize the ideas presented so far to try to further reduce the degrees of the polynomials in the system in order to obtain smaller values of d.

6.2 A transformation

In this section, we introduce a transformation on the variables of system(6.1). In addition, we restrict $L(x) \in \mathbb{F}_2[x]$. The other notations for this section are the same as before.

6.2.1 Polynomials $L(x)$ with coefficients in $\mathbb{F}_2$

Let $\mathcal{L}$ be the set of linearized polynomials with coefficients in $\mathbb{F}_2$ equipped with the symbolic multiplication $\otimes$. More precisely, for two linearized polynomials $L_1(x), L_2(x) \in \mathbb{F}_2[x]$, we define the symbolic multiplication $\otimes$ as follows:

$$L_1(x) \otimes L_2(x) := L_1(L_2(x)).$$

It is easy to verify that $L_1(x) \otimes L_2(x) = L_2(x) \otimes L_1(x)$. Under this symbolic multiplication $\otimes$, $\mathcal{L}$ forms an abelian group with x as the neutral element.

Consider the map $\varphi : \mathbb{F}_2[x] \to \mathcal{L}$ which maps x^i to x^{2^i}. Extend this map $\mathbb{F}_2$-linearly for all polynomials in $\mathbb{F}_2[x]$.Explicitly, under this map, a polynomial $l(x) = \sum_{i=0}^{t} a_i x^i \in \mathbb{F}_2[x]$ maps to $L(x) = \sum_{i=0}^{t} a_i x^{2^i} \in \mathcal{L}$. One can easily show that φ satisfies the following two properties:

1. $\varphi(l_1(x)l_2(x)) = \varphi(l_1(x)) \otimes \varphi(l_2(x))$,

2. $\varphi(l_1(x) + l_2(x)) = \varphi(l_1(x)) + \varphi(l_2(x))$,

where $l_1(x), l_2(x) \in \mathbb{F}_2[x]$.

Now for any $l(x) \in \mathbb{F}_2[x]$, we call $L(x) := \varphi(l(x))$ the associated linearized polynomial.

Lemma 6.2.1. *Let $l_1(x)$ and $l_2(x)$ be two polynomials in $\mathbb{F}_2[x]$ and let $L_1(x), L_2(x)$ be the associated linearized polynomials. We have $l_1(x)|l_2(x)$ if and only if $L_1(x)|L_2(x)$. In particular, let $l_2(x) = x^n - 1$. Then $l_1(x)|(x^n - 1)$ if and only if its linearized polynomial $L_1(x)$ has all its roots in an $\mathbb{F}_2$-vector subspace of $\mathbb{F}_{2^n}$.*

The proof of the above lemma is straightforward and we omit the details.

We fix two integers n, n' and an $\mathbb{F}_2$-vector subspace of $\mathbb{F}_q = \mathbb{F}_{2^n}$ with dimension n'. Let $L(x)$ be the characteristic polynomial of V. Suppose $L(x) \in \mathbb{F}_2[x]$.

For a linearized polynomial $L'(x) \in \mathbb{F}_2[x]$, it introduces an $\mathbb{F}_2$-linear map:

$$\begin{aligned} \mathbb{F}_q &\to \mathbb{F}_q \\ a &\mapsto L'(a). \end{aligned}$$

Since $L(L'(a)) = 0$, for $a \in V$, it follows that the above map introduces an $\mathbb{F}_2$-linear map on V when it restricts to V.

Let $l(x) \in \mathbb{F}_2[x]$ be the polynomial corresponding to $L(x)$ under the map φ introduced above. Similarly, let $l'(x) \in \mathbb{F}_2[x]$ corresponds to $L'(x)$. Now suppose $\gcd(l(x), l'(x)) = 1$, or equivalently, $\gcd(L(x), L'(x)) = x$. Under this condition, the following map is injective:

$$\begin{aligned} V &\to V \\ a &\mapsto L'(a). \end{aligned}$$

Thus this map is an isomorphism and its inverse exists.

Now we use the transformation $x_i = L'(y_i), i = 1, ..., m$ for system(6.1). This yields the following system in the variables y_i:

$$\begin{aligned} f(L'(y_1), \ldots, L'(y_m)) \mod (L(y_1), \ldots, L(y_m)) &= 0, \\ L(y_i) &= 0,\ i = 1, \ldots, m. \end{aligned} \tag{6.6}$$

Note that since $y_i \in V$, y_i satisfies the characteristic equation $L(y_i) = 0$ as well. To avoid using too many variables, we will simply use x_i in place of y_i in the above system, and thus we consider the following system which is equivalent to system(6.1):

$$\begin{aligned} f(L'(x_1), \ldots, L'(x_m)) \mod (L(x_1), \ldots, L(x_m)) &= 0, \\ L(x_i) &= 0,\ i = 1, \ldots, m. \end{aligned} \tag{6.7}$$

By treating $f(L'(x_1),\ldots,L'(x_m)) \mod (L(x_1),\ldots,L(x_m)$ as a multivariate polynomial, we can do the same thing as constructing system(6.5) from system(6.1). To be more precise, we consider the following problem:

Let V be a vector subspace of $\mathbb{F}_q$ over $\mathbb{F}_2$ with dimension n'. Let $L(x)$ be the characteristic polynomial of V. Assume that $L(x) \in \mathbb{F}_2[x]$. We look for a linearized polynomial $L'(x) \in \mathbb{F}_2[x]$ satisfying $\gcd(L'(x), L(x)) = x$. Similarly, we let $L_i(x) = (L'(x))^{2^i} \mod L(x)$ and $\deg(L_i(x)) = 2^{d_i}, i = 0, ..., n$. Let $I_0 = \{i | d_{i-1} \neq d_i - 1, 1 \leq i \leq n\}$. Let $i_1, \ldots, i_m$ be m indices in I_0 such that $d_{i_1} \leq \ldots \leq d_{i_m}$ and the sum $d = d_{i_1} + \ldots + d_{i_m}$ is the smallest.

To solve system (6.7), we consider the following system:

$$\begin{aligned} f_{i_1} =& 0, \\ & \ldots \\ f_{i_m} =& 0. \end{aligned} \tag{6.8}$$

where $f_{i_j} = (f(L'(x_1),\ldots,L'(x_m)))^{2^{i_j}} \mod (L(x_1), ..., L(x_m)), j = 1, ..., m$.

Since all polynomials in the above system are symmetric, we consider the corresponding system with every polynomial symmetrized:

$$\begin{aligned} \tilde{f}_{i_1} =& 0, \\ & \ldots \\ \tilde{f}_{i_m} =& 0. \end{aligned} \tag{6.9}$$

As in the previous case, the time complexity of solving this system primarily depends on the degrees $d'_{i_j}s$, namely, we like to have $d_{i_1} + \ldots + d_{i_m}$ as small as possible.

Next, we describe how we may find $L'(x)$ by exploiting the map φ. By 6.2.1, we have

$$\varphi(x^i * l'(x)) = \varphi(x^i) \otimes \varphi(l'(x)) = x^{2^i} \otimes L'(x) = L'(x^{2^i}) = (L'(x))^{2^i}.$$

Suppose that $x^i * l'(x) = q_i(x) * l(x) + l_i(x)$, where $l_i(x)$ is the remainder of $x^i * l'(x)$ divided by $l(x)$. By 6.2.1 and above, we have

$$(L'(x))^{2^i} = \varphi(q_i(x) * l(x) + l_i(x)) = \varphi(q_i(x)) \otimes \varphi(l(x)) + \varphi(l_i(x)) = \varphi(q_i)(L(x)) + \varphi(l_i(x)).$$

thus $\varphi(l_i(x))$ is the remainder of $(L'(x))^{2^i}$ divided by $L(x)$, i.e, $L_i(x) = \varphi(l_i(x))$.

From the above two equalities and 6.2.1, it is easy to see that $\gcd(L'(x), L(x)) = x$ if and only if $\gcd(l'(x), l(x)) = 1$.

Using the one-to-one correspondence between linearized polynomials with coefficients in $\mathbb{F}_2$ and polynomials in $\mathbb{F}_2[x]$, the above problem of finding a linearized polynomial $L'(x) \in \mathbb{F}_2[x]$ satisfying $\gcd(L'(x), L(x)) = x$ is equivalent to finding $l'(x) \in \mathbb{F}_2[x]$ with $\gcd(l'(x), l(x)) = 1$.

Furthermore, let $H_{l(x)}$ denote the group $\mathbb{F}_2[x]/l(x))^*$. Consider the cyclic subgroup $G = \langle \overline{x} \rangle$ of $H_{l(x)}$. Now, for each $l'(x) \in H_{l(x)}$, the coset $l'(x)G$ comprises the elements $\{l'(x), xl'(x), \ldots\}$. Thus, in finding for the m indices d_{i_j} that give a smallest sum d, one looks for such elements in the coset.

Note that the case of $L'(x) = x$ is the special case discussed in the preceding section. By allowing $L'(x)$ to vary, we have now greatly increased the search space to find m indices where the corresponding degrees are small. This is the main motivation for considering the transformation in this section.

It is clear that the complexity analysis when using transformations is identical to that carried out earlier. We have seen that a complexity bound on solving System (6.5) can be done via Rojas's results. We performed some experiments with Magma to investigate the degree of regularity for some vector spaces using our approach. Concretely, for each set of parameters

n, m, n', we select an $L(x)$ and $L'(x)$ and compute the respective d_i's. We then compute the Gröbner basis of both System (6.8) and System (6.9) using the 'GroebnerBasis" function in Magma to determine its regularity. The results are summarized in table(6.3) and table(6.4).

Tab. 6.3: The degree of regularity of System (6.8)

n	m	n'	d_i	$MacBound$	D_{reg}
7	2	3	$(0,1)$	11	8
7	3	3	$(0,1,2)$	43	25
15	2	7	$(0,3)$	35	32
17	2	8	$(3,3)$	63	40
21	2	10	$(4,4)$	127	88
23	2	11	$(3,5)$	159	128
31	2	15	$(2,5)$	143	128
35	2	17	$(7,7)$	1023	704

Tab. 6.4: The degree of regularity of System (6.9)

n	m	n'	d_i	$MacBound$	D_{reg}
7	2	3	$(0,1)$	5	4
7	3	3	$(0,1,2)$	13	12
15	2	7	$(0,3)$	17	16
17	2	8	$(3,3)$	31	32
21	2	10	$(4,4)$	63	64
23	2	11	$(3,5)$	79	80
31	2	15	$(2,5)$	71	72
35	2	17	$(7,7)$	511	512

In the above two tables, D_{reg} denotes the degree of regularity of the polynomial system and $MacBound$ denotes the Macaulay bound of this polynomial system.

From these results, one sees that the Macaulay bound seems to approximate the degree of regularity of the symmetrized system (System (6.9)) pretty well. Using the Macaulay bound as an approximate for the degree of regularity of the system, one again obtains the complexity bound derived in the previous section.

Finally, under the standard heuristic assumptions(6.1.1) and System (6.9) is zero-dimensional, we summarize the analysis and results of these two sections in the following heuristic result.

Heuristic result 6.2.2. *Using all the notations as above, suppose that there exists a vector subspace of* $\mathbb{F}_{2^n}$ *with dimension* n' *such that there exists a transformation* $L'(x)$ *and the corresponding set* $d_{i_1}, d_{i_2}, \ldots, d_{i_m}$ *with* $d = d_{i_1} + \ldots + d_{i_m} = O(n^\omega)$ *for some constant* $0 < \omega < 1$*, then we have a sub-exponential index calculus algorithm with* $\mathcal{F}_V = \{(x, y) \in E(\mathbb{F}_{2^n}) | x \in V\}$ *as the factor base.*

6.3 Examples

In the previous section, we have seen that for fixed n, m, n', we like to seek for a vector space of dimension n' with the following properties: Let $L(x)$ be its characteristic polynomial. Then, there exists some $L'(x)$ with $\gcd(L(x), L'(x)) = 1$ and there are m indices i_j for which the degree of $L'(x)^{2^{i_j}} \mod L(x)$ are as small as possible. In general, there does not seem to be a straightforward method to find such vector spaces. In this section, we present some examples for some choices of n, m and n'.

6.3.1 Subfield case

Suppose that $m|n$. We let $L(x) = x^{2^{n'}} - x$. With $i_1 = n', i_2 = 2n', \ldots, i_m = mn'$ yield $d_{i_1} = \ldots = d_{i_m} = 0$. In particular, we have $d = 0$.

6.3.2 More concrete examples

In the following, the examples do not apply the transformation introduced in section(6.2). In this case, we can always choose $i_m = n$ and thus $d_{i_m} = 0$. Therefore we omit $i_m = n$ and $d_{i_m} = 0$ in the examples.

Let k be a positive integer. Consider $l(x) = x^{2^{k-1}} + x^{2^{k-2}} + \ldots + x^2 + x + 1$. Squaring $l(x)$

produces:

$$\begin{aligned} l(x)^2 =& (x^{2^{k-1}} + \ldots + x^2 + x + 1)^2 \\ =& x^{2^k} + \ldots + x^{2^2} + x^2 + 1 \\ =& x^{2^k} - x + l(x). \end{aligned}$$

Since $l(x)$ and x are coprime, it follows that $l(x)|(x^{2^k-1} - 1)$. By Lemma 6.2.1, we conclude that $L(x) = x^{2^{2^{k-1}}} + x^{2^{2^{k-2}}} + \ldots + x^{2^2} + x^2 + x$ is a linearized polynomial with roots in $\mathbb{F}_{2^{2^k-1}}$.

This leads to the following example:

Let $m = 2, n = 2^k - 1, n' = 2^{k-1}$. $L(x) = x^{2^{2^{k-1}}} + x^{2^{2^{k-2}}} + \ldots + x^{2^2} + x^2 + x$. Hence, $d_{i_1} = 2^{2^{k-2}} \approx n/4$ and we have $d = n/4$.

Next, let t be a positive integer and let $n|(2^t - 1)$. Then $x^n - 1$ can be factored into a product of irreducible factors, each of degree dividing t. Let g be a product of some of these factors such that the total degree of g is n'. Consider the ring $\mathbb{F}_2[x]/\langle g \rangle$ and let G be its cyclic subgroup generated by x. We seek to find examples where G contains at least m low degree polynomials which are coprime to x.

Table(6.5) gives a list of polynomials $l(x)$ which are factors of $x^n - 1$ and their corresponding values of $d_{i_1}, d_{i_2}, \ldots, d_{i_{m-1}}$.

Tab. 6.5: Some parameters for d_i without using transformation

m	n	$l(x)$	d_i
3	31	$x^{10}+x^6+x^5+x^4+1$	$(5,6)$
3	31	$x^{11}+x^9+x^5+x^3+x+1$	$(5,8)$
3	43	$x^{14}+x^{12}+x^{10}+x^7+x^4+x^2+1$	$(7,10)$
3	43	$x^{15}+x^{14}+x^{13}+x^{12}+x^{11}+x^{10}+x^8+$ $x^7+x^5+x^4+x^3+x^2+x+1$	$(10,10)$
4	73	$x^{18}+x^{17}+x^{15}+x^{13}+x^{11}+x^{10}+x^9+$ $x^6+x^5+x^3+x^2+x+1$	$(10,11,14)$
4	73	$x^{19}+x^{17}+x^{14}+x^{11}+x^{10}+x^8+x^5+x^4+$ x^3+1	$(11,11,14)$
4	89	$x^{22}+x^{18}+x^{17}+x^{12}+x^{11}+x^6+x^2+x+1$	$(14,14,15)$
4	89	$x^{23}+x^{22}+x^{19}+x^{17}+x^{13}+x^{11}+x^7+$ x^6+x^3+1	$(14,15,19)$
3	109	$x^{36}+x^{34}+x^{32}+x^{30}+x^{28}+x^{27}+x^{26}+$ $x^{21}+x^{20}+x^{19}+x^{18}+x^{17}+x^{16}+x^{15}+$ $x^{10}+x^9+x^8+x^6+x^4+x^2+1$	$(29,29)$

We also perform some experiments to consider the transformation introduced in section(6.2). Table(6.6) gives a list of polynomials $l(x)$ with their corresponding values of $d_{i_1}, d_{i_2}, \ldots, d_{i_m}$ which are the smallest ones among all cosets. Note that in this case, i_m does not necessarily equal to n and d_{i_m} does not necessarily equal to 0.

From the above experimental results, we see that the degrees of list of d_i using transformation are smaller than the degrees of list of d_i without transformation. Thus after a transformation, d may become smaller and thus give an improved time complexity.

6.3.3 Examples with transformations

In this subsection, we will consider the transformation introduced in section(6.2).

First, let $m = 2$. In the following, we introduce a result needed. We only consider one variable polynomial ring $R = F[x] = \mathbb{F}_q[x]$ with q a prime power.

Definition 6.3.1. Let $f(x) = a_n x^n + ... + a_1 x + a_0 \in F[x]$ with $a_n \neq 0$. Then the *reciprocal*

Tab. 6.5: Some parameters for d_i without using transformation(continued)

m	n	$l(x)$	d_i
3	109	$x^{37}+x^{36}+x^{35}+x^{34}+x^{33}+x^{32}+x^{31}+$ $x^{30}+x^{29}+x^{26}+x^{22}+x^{15}+x^{11}+x^8+$ $x^7+x^6+x^5+x^4+x^3+x^2+x+1$	$(29,29)$
4	113	$x^{28}+x^{23}+x^{22}+x^{20}+x^{17}+x^{16}+x^{15}+$ $x^{14}+x^{13}+x^{12}+x^{11}+x^8+x^6+x^5+1$	$(19,23,23)$
4	113	$x^{29}+x^{27}+x^{26}+x^{22}+x^{21}+x^{18}+x^{16}+$ $x^{13}+x^{11}+x^8+x^7+x^3+x^2+1$	$(23,23,25)$
6	127	$x^{21}+x^{20}+x^{18}+x^{17}+x^{16}+x^{10}+x^9+$ $x^8+x^5+x^4+x^2+x+1$	$(5,10,10,13,15)$
6	127	$x^{22}+x^{20}+x^{19}+x^{16}+x^{11}+x^8+x^6+x^4+$ x^3+1	$(5,10,15,17,17)$
4	127	$x^{28}+x^{27}+x^{26}+x^{24}+x^{18}+x^{16}+1$	$(12,14,20)$
5	127	$x^{28}+x^{27}+x^{26}+x^{24}+x^{18}+x^{16}+1$	$(12,14,20,21)$
4	127	$x^{29}+x^{28}+x^{27}+x^{23}+x^{21}+x^{18}+x^{15}+$ $x^{13}+x^{10}+x^7+x^6+x^3+x+1$	$(17,19,20)$
5	127	$x^{29}+x^{28}+x^{27}+x^{26}+x^{23}+x^{22}+x^{19}+$ $x^{16}+x^{15}+x^{14}+x^{11}+x^5+x^4+1$	$(17,20,20,22)$
5	151	$x^{30}+x^{28}+x^{26}+x^{22}+x^{18}+x^{16}+x^{15}+$ $x^{14}+x^{12}+x^8+x^4+x^2+1$	$(15,19,19,22)$
5	151	$x^{31}+x^{30}+x^{28}+x^{26}+x^{24}+x^{23}+x^{22}+$ $x^{20}+x^{19}+x^{17}+x^{16}+x^{15}+x^{14}+x^{12}+$ $x^{11}+x^9+x^8+x^7+x^5+x^3+x+1$	$(19,19,25,25)$

polynomial f^* of f is defined by

$$f^*(x) = x^n f(\frac{1}{x}) = a_0x^n + a_1x^{n-1} + ... + a_n$$

Definition 6.3.2. Let $f(x) \in F[x]$ be a nonzero polynomial. If $f(0) \neq 0$, then the least positive integer e such that $f(x)$ divides x^e-1 is called the *order* of f and denoted by $ord(f) = ord(f(x))$. If $f(0) = 0$, then $f(x) = x^h g(x)$ for uniquely determined g with $g(0) \neq 0$; $ord(f)$ is then defined to be $ord(g)$.

Remark 6.3.3. Let f be a nonzero polynomial in $F[x]$ and f^* its reciprocal polynomial. Then $ord(f) = ord(f^*)$.

Tab. 6.6: Some parameters for d_i using transformation

m	n	$l(x)$	d_i
3	31	$x^{10}+x^8+x^7+x^5+x^3+x^2+1$	$(3,3,5)$
3	31	$x^{10}+x^6+x^5+x^4+1$	$(4,5,5)$
2	41	$x^{20}+x^{18}+x^{17}+x^{16}+x^{15}+x^{14}+x^{11}+$ $x^{10}+x^9+x^6+x^5+x^4+x^3+x^2+1$	$(8,8)$
3	43	$x^{14}+x^{11}+x^{10}+x^9+x^8+x^7+x^6+x^5+$ x^4+x^3+1	$(6,6,6)$
2	47	$x^{23}+x^{19}+x^{18}+x^{14}+x^{13}+x^{12}+x^{10}+$ $x^9+x^7+x^6+x^5+x^3+x^2+x+1$	$(7,11)$
2	71	$x^{35}+x^{33}+x^{28}+x^{27}+x^{26}+x^{25}+x^{24}+$ $x^{17}+x^{13}+x^8+x^7+x^5+x^4+x+1$	$(11,17)$
3	73	$x^{27}+x^{23}+x^{16}+x^{14}+x^{10}+x^8+x^6+x^3+1$	$(12,13,15)$
2	79	$x^{39}+x^{36}+x^{35}+x^{31}+x^{30}+x^{29}+x^{27}+$ $x^{26}+x^{25}+x^{24}+x^{21}+x^{20}+x^{19}+x^{18}+$ $x^{16}+x^{14}+x^{13}+x^{11}+x^5+x^4+x^2+x+1$	$(18,18)$

Proposition 6.3.4. *Let $l(x) \in F[x]$ with $l^*(x)$ as its reciprocal. Let $G = \langle \bar{x} \rangle$ be the subgroup of $(F[x]/l(x))^*$ and G^* be the corresponding subgroup in $(F[x]/l^*(x))^*$. Then $\{\deg(H) : H$ is a coset of $G\} = \{\deg(H^*) : H^*$ is a coset of $G^*\}$.*

Proof. Let $m = \deg(l)$. For any $f(x) \in F[x]$ with $\gcd(f,l) = 1$, f has a unique expression $f = x^h * g$ with $h \in \mathbb{N}$ and $g(0) \neq 0$. Then $fG = gG$. So from now on, when we say coset fG we assume $f(0) \neq 0$ and $\deg(f) < \deg(l) = m$. We have the following claim.

Claim: $\deg(fG) = \deg(f^*G^*)$.

For any element $x^a f \in fG$, it has a unique expression $x^a f \equiv x^b r \mod (l)$ with $b \in \mathbb{N}$, $r(0) \neq 0$ and $b + \deg(r) < m$.

Then we have $a \geq b$. Since if $a < b$, then $a + \deg(f) \geq m > b + \deg(r)$ and $f \equiv x^{b-a} r \mod (l)$, it follows that $b - a + \deg(r) < m$. Note that $\deg(f) < m$. Hence, we have $f = x^{b-a} r$, which contradicts $f(0) \neq 0$.

Now since $a \geq b$, we have $x^{a-b} f \equiv r \mod (l)$. If $a = b$,then $f = r$. Thus, we consider the case $a > b$. If $a - b + \deg(f) < m$,we must have $r = x^{a-b} f$ which is impossible since $r(0) \neq 0$. Thus $a - b + \deg(f) \geq m$. Suppose $x^{a-b} f - r = l * h$ for some $h \in F[x]$. It follows that $h(0) \neq 0$

and then $f^* - x^{a-b+\deg(f)-\deg(r)}r^* = l^*h^*$. Thus we have the following equation:

$$f^* \equiv x^{a-b+\deg(f)-\deg(r)}r^* \mod (l^*).$$

Note that $a-b+\deg(f) \geq m > \deg(r)$. It follows that $t := a-b+\deg(f)-\deg(r) > 0$ and we have

$$x^{(n-1)t+b}f^* \equiv x^{nt+b}r^* \equiv x^b r^* \mod (l^*),$$

where $n = ord(l)$.

So we prove that for every element $x^b r \in fG$, we can find an element $x^{(n-1)t+b}f^* \in f^*G^*$ with the same degree as $\deg(r) = \deg(r^*)$. By symmetry, for every element in f^*G^*, we can find an element in fG with the same degree. Thus $\deg(fG) = \deg(f^*G^*)$.

Using this claim, it is easy to see that the proposition follows. □

We will now give a rough bound for $\max(d_{i_1}, d_{i_2})$ based on some assumptions and proposition(6.3.4). First, let us see an concrete example to illustrate the basic idea.

Let $n = 31$ and $n' = 15$. Note that we have the following factorization of $x^{31}-1$ in $\mathbb{F}_2[x]$:

$x^{31}-1 = (x-1)*(x^5+x^2+1)*(x^5+x^3+1)*(x^5+x^3+x^2+x+1)*(x^5+x^4+x^2+x+1)*(x^5+x^4+x^3+x+1)*(x^5+x^4+x^3+x^2+1)$.

Let $l_i(x), i = 1, ..., 6$ denote the irreducible factors of $x^{31}-1$ with degree 5. We have $\binom{6}{3}$ different factors of $x^{31}-1$ with degree equal to n', i.e, there are $\binom{6}{3}$ different $l(x)$ to choose. Now we fix one $l(x)$ and let $L(x)$ be the image of $l(x)$ under the map φ introduced in section(6.2).

By section(6.2), we seek to find one coset of $(\mathbb{F}_2[x]/(l(x)))^\star$ with respect to $G = \langle \bar{x} \rangle$ such that this coset contains two polynomials with the least degrees among all possible cases and the quotient of these two polynomial is not equal to a power of x. Note that the cardinality of $(\mathbb{F}_2[x]/(l(x)))^\star$ is 31^3 and $\#G = 31$, and thus there are 31^2 cosets of $(\mathbb{F}_2[x]/(l(x)))^\star$ with respect to G.

We wish to bound $\max(d_{i_1}, d_{i_2})$ heuristically.

Let us suppose all elements of $(\mathbb{F}_2[x]/(l(x)))^\star$ are equally distributed in all these 31^2 cosets of G. Given a degree bound D, there are 2^D polynomials in $\mathbb{F}_2[x]$ with the constant term equal to 1, i.e., polynomials of the form $1+a_1x+a_2x^2+...+a_Dx^D$, $a_i \in \mathbb{F}_2$. Among these polynomials, at most $3 * 2^{D-5}$ of them are not coprime to $l(x)$. Hence, we may just assume that around 2^D polynomials of degree bounded by D and constant term 1 lie in the group $(\mathbb{F}_2[x]/l(x))^*$. Note that the quotient of any two of these polynomials is not equal to a power of x. The probability of at least one coset containing two of these elements can be computed similar to the *birthday problem*. In particular, among these 2^D elements, we ask the probability that at least two of them will lie in the same coset. In other words, referring to the birthday problem, the cosets represent the days of a year while the elements represent the birthdays of different people. This probability can be computed as:

$$P = 1 - \frac{(N-1) \times \ldots \times (N-B+1)}{N^B} = 1 - (1 - \frac{1}{N}) \times \ldots \times (1 - \frac{B-1}{N}),$$

where $N = 31^2$ is the number of cosets, $B \approx 2^D$ is the number of polynomials.

We have the following approximation:

$$(1 - \frac{i}{N}) \approx e^{-\frac{i}{N}}, i = 1, ..., B-1.$$

Thus $P \approx 1 - e^{\frac{-B^2}{2N}}$.

Since there are 20 different $l(x)$ and half of them have the same sets of degree by proposition(6.3.4), if $P > \frac{1}{10}$, then there exists some $l(x)$ such that one coset of $G = \langle \bar{x} \rangle$ in $(\mathbb{F}_2[x]/(l(x)))^\star$ contains at least two polynomials we are seeking. A simple computation shows that $B \approx 14$, so $D \approx 4$, i.e $\max(d_{i_1}, d_{i_2}) \leq 4$.

We perform experiments to run through all the $l(x)$ and $l'(x)$. In each case, we record the degrees of the two smallest elements in the coset. From our experiments, we find that we

can choose $l(x) = x^{15} + x^{11} + x^{10} + x^2 + 1$. In this case $D = 5$, and the two polynomials with the least degrees lie in the same coset are $x^2 + 1$ and $x^5 + x + 1$. Another choice is $l(x) = x^{15} + x^{13} + x^5 + x^4 + 1$, in which case $D = 5$, and the two polynomials with the least degrees lie in the same coset are $x^2 + 1$ and $x^5 + x^4 + 1$. The heuristic bound is close to the practical result.

Remark 6.3.5. Strictly speaking, each element in $(\mathbb{F}_2[x]/l(x))^*$ may not have an equal probability to be in each of the cosets. This is because not all the cosets have the same number of elements with constant term 1. However, for large enough parameters, the difference is negligible. Hence, the bound we obtained should be a good approximate for the bound we seek.

For some values of n, we can use the above example to give a heuristic bound for $\max(d_{i_1}, d_{i_2})$. We consider the following special form of n. Let $n = 2^t - 1$ be a prime number, and so t is also a prime. It is well known that $x^n - 1$ has the following factorization in $\mathbb{F}_2[x]$:

$$x^n - 1 = (x - 1) * f_1(x) * \ldots * f_{2r}(x),$$

where $f_i(x)$'s are irreducible factors of $x^n - 1$ with the same degree t, $r = \frac{2^{t-1}-1}{t}$.

We let $n' = 2^{t-1} - 1$. There are $\binom{2r}{r}$ different factors of $x^n - 1$ with degree n'. Fix one factor $l(x)$ of $x^n - 1$ with degree n'. The group $(\mathbb{F}_2[x]/(l(x)))^\star$ has $(2^t - 1)^r = n^r$ elements and $G = \langle \bar{x} \rangle$ has n elements. It follows that there are n^{r-1} cosets of G. Suppose D is the degree bound. Then the probability of at least one coset containing two elements we are seeking is:

$$P \approx 1 - e^{-\frac{B^2}{2N}},$$

where $N = n^{r-1}, B \approx 2^D$. Since there are $\binom{2r}{r}$ different $l(x)$, we let $1 - e^{-\frac{B^2}{2N}} \approx \frac{2}{\binom{2r}{r}}$. We give some values of t and $\max(d_{i_1}, d_{i_2})$ in table(6.7).

We see from the table that for small values of n, $\max(d_{i_1}, d_{i_2}) < n/4$ but it approaches to $n/4$ for larger n. According to the result in the previous subsection, for $n = 2^t - 1$, one may

Tab. 6.7: Some bound for $\max(d_{i_1}, d_{i_2})$

t	n	Bound for $\max(d_{i_1}, d_{i_2})$
7	127	22
11	2047	417
13	8191	1730
17	131071	28909
19	524287	117270

find $d = n/4$ when transformations are not exploited. It suggests that one may find cosets with smaller sums, especially when n is small.

Finally, with the help of Magma, we run through all possible $l(x)$ and $l'(x)$ for some parameters of n, m and n' to find the smallest sets of $d_{i_1}, \ldots, d_{i_m}$. Table(6.8) gives a list of polynomials $l(x)$ with their corresponding values of $d_{i_1}, d_{i_2}, \ldots, d_{i_m}$ which are the smallest ones among all cosets. Note that in this case, i_m does not necessarily equal to n and d_{i_m} does not necessarily equal to 0.

Tab. 6.8: Smallest d_i

m	n	$l(x)$	d_i
2	41	$x^{20}+x^{18}+x^{17}+x^{16}+x^{15}+x^{14}+x^{11}+$ $x^{10}+x^9+x^6+x^5+x^4+x^3+x^2+1$	$(8,8)$
3	43	$x^{14}+x^{11}+x^{10}+x^9+x^8+x^7+x^6+x^5+$ x^4+x^3+1	$(6,6,6)$
2	47	$x^{23}+x^{19}+x^{18}+x^{14}+x^{13}+x^{12}+x^{10}+$ $x^9+x^7+x^6+x^5+x^3+x^2+x+1$	$(7,11)$
2	71	$x^{35}+x^{33}+x^{28}+x^{27}+x^{26}+x^{25}+x^{24}+$ $x^{17}+x^{13}+x^8+x^7+x^5+x^4+x+1$	$(11,17)$
3	73	$x^{27}+x^{23}+x^{16}+x^{14}+x^{10}+x^8+x^6+x^3+1$	$(12,13,15)$
2	79	$x^{39}+x^{36}+x^{35}+x^{31}+x^{30}+x^{29}+x^{27}+$ $x^{26}+x^{25}+x^{24}+x^{21}+x^{20}+x^{19}+x^{18}+$ $x^{16}+x^{14}+x^{13}+x^{11}+x^5+x^4+x^2+x+1$	$(18,18)$

To summarize, in this chapter, we have identified a class of vector spaces with nice properties based on the characteristic polynomials that may help to speed up the relations search step in the index calculus approach to solve ECDLP. We have analyzed the time complexity to solve

ECDLP given such vector spaces. We have also provided some concrete examples of the vector spaces we seek for small parameters. It remains an open and interesting problem to construct such vector spaces for arbitrary parameters that leads to a more efficient ECDLP attack.

7. CONCLUSIONS

This thesis focuses on solving zero-dimensional polynomial systems and their applications to cryptography. Concretely, motivated by the many applications of polynomial systems in cryptography, such as in the design of multivariate cryptosystems and in the cryptanalytic attacks on elliptic curve cryptosystems, we investigated the problem of solving zero-dimensional polynomial systems in greater detail. We proposed a new notion called last fall degree and a corresponding framework which turned out to be useful to help us analyze the complexity of polynomial systems arising from Weil descent. These polynomial systems arise in both the design of multi-HFE cryptosystem as well as in the index calculus approach to solve the elliptic curve discrete logarithm problem. Using our proposed framework, we proved that the multi-HFE scheme is not secure. At the same time, we argued that our framework cannot be directly applied to the polynomial system from ECDLP applications. Instead, we proposed a different framework to solve such polynomial systems. Specifically, by using a class of vector subspaces with nice properties, we proposed a direct method to solve the polynomial system without using Weil descent. We provided complexity bounds on our approach based on some heuristic assumptions. More importantly, we derived some conditions for which our approach will become sub-exponential. Even though we did not break ECDLP with explicit constructions of the vector spaces we seek, we believe that our approach is interesting in its own right and provided some direction to find factor bases that may give rise to a sub-exponential index calculus attack on ECDLP.

7.1 Future work

Several possible research directions can be further explored. The first is from chapter5, where one can consider removing some conditions of theorem(5.3.4). For instance, it will be interesting to remove the condition that I is radical and/or the restriction that there is a coordinate t such that the projection map $Z(\mathcal{F}) \to \overline{k}$ to coordinate t is injective , and try to give a bound on the last fall degree of the Weil descent system.

As for our approach to solve ECDLP, it remains an open and interesting problem to determine if vector spaces with the conditions to result in a sub-exponential index calculus attack on ECDLP exists for prime n, and if so, construct such vector spaces explicitly.

BIBLIOGRAPHY

[1] L. M. Adleman. A subexponential algorithm for the discrete logarithm problem with applications to cryptography. In *20th Annual Symposium on Foundations of Computer Science, San Juan, Puerto Rico, 29-31 October 1979*, pages 55–60, 1979. [5, 12]

[2] M. R. Albrecht, C. Cid, J.-C. Faugere, and L. Perret. On the relation between the MXL family of algorithms and Gröbner basis algorithms. *Journal of Symbolic Computation*, 47(8):926–941, 2012. [47, 72]

[3] G. Ars, J.-C. Faugere, H. Imai, M. Kawazoe, and M. Sugita. Comparison between XL and Gröbner basis algorithms. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 338–353. Springer, 2004. [46]

[4] D. Bayer and M. Stillman. A theorem on refining division orders by the reverse lexicographic order. *Duke Mathematical Journal (C)*, 1987. [43]

[5] L. Bettale, J.-C. Faugere, and L. Perret. Cryptanalysis of HFE, multi-HFE and variants for odd and even characteristic. *Designs, Codes and Cryptography*, pages 1–52, 2013. [3, 93]

[6] B. Buchberger. *Ein Algorithmus zum Auffinden der Basiselemente des Restklassenrings nach einem nulldimensionalen Polynomideal.* PhD thesis, Universitat Innsbruck, Austria, 1965. [30]

[7] J. A. Buchmann, J. Ding, M. S. E. Mohamed, and W. S. A. E. Mohamed. MutantXL: Solv-

ing multivariate polynomial equations for cryptanalysis. In *Dagstuhl Seminar Proceedings*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2009. [71]

[8] J. S. Coron and B. de Weger. Hardness of the main computational problems used in cryptography. *IST-2002-507932,ECRYPT,European Network of Excellence in Cryptology*, 2007. [1]

[9] N. T. Courtois, A. Klimov, J. Patarin, and A. Shamir. Efficient algorithms for solving overdefined systems of multivariate polynomial equations. In *Advances in Cryptology—EUROCRYPT 2000 (Bruges)*, volume 1807 of *Lecture Notes in Computer Science*, pages 392–407. Springer, Berlin, 2000. [45, 71]

[10] D. Cox, J. Little, and D. O'Shea. *Ideals, varieties, and algorithms*. Undergraduate Texts in Mathematics. Springer-Verlag, Berlin, Heidelberg, New York, first edition, 1992. [19, 31, 32, 33, 34, 38, 43]

[11] C. Diem. On the discrete logarithm problem in elliptic curves. *Compositio Mathematica*, 147(1):75–104, 2011. [6, 18, 24, 49, 50, 51, 53, 94]

[12] C. Diem. On the discrete logarithm problem in elliptic curves II. *Algebra Number Theory*, 7(6):1281–1323, 2013. [6, 49, 51, 53, 99]

[13] J. Ding, J. Buchmann, M. S. E. Mohamed, W. S. A. E. Mohamed, and R.-P. Weinmann. MutantXL. *Proceedings of the 1st international conference on Symbolic Computation and Cryptography(SCC08)*, 2008. [46, 47]

[14] J. Ding and T. J. Hodges. Inverting HFE systems is quasi-polynomial for all fields. In *Advances in Cryptology – CRYPTO 2011: 31st Annual Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2011. Proceedings*, pages 724–742. Springer, 2011. [2, 3, 92]

[15] J. C. Faugère. A new efficient algorithm for computing Gröbner bases (F_4). *Journal of Pure and Applied Algebra*, 139(1-3):61–88, 1999. [2, 35, 41, 42, 43, 71]

[16] J. C. Faugère. A new efficient algorithm for computing Gröbner bases without reduction to zero (F_5). In *Proceedings of the 2002 International Symposium on Symbolic and Algebraic Computation*, pages 75–83. ACM, New York, 2002. [2, 35, 42, 43, 71]

[17] J.-C. Faugère, P. Gaudry, L. Huot, and G. Renault. Using symmetries in the index calculus for elliptic curves discrete logarithm. *Journal of Cryptology*, 27(4):595–635, 2014. [44, 45]

[18] J. C. Faugère, P. M. Gianni, D. Lazard, and T. Mora. Efficient computation of zero-dimensional Gröbner bases by change of ordering. *Journal of Symbolic Computation*, 16(4):329–344, 1993. [2, 43, 44, 71]

[19] J.-C. Faugère, L. Huot, A. Joux, G. Renault, and V. Vitse. *Symmetrized Summation Polynomials: Using Small Order Torsion Points to Speed Up Elliptic Curve Index Calculus*, pages 40–57. Springer Berlin Heidelberg, Berlin, Heidelberg, 2014. [25]

[20] J.-C. Faugère and A. Joux. Algebraic cryptanalysis of hidden field equation (HFE) cryptosystems using gröbner bases. In *Advances in Cryptology - CRYPTO 2003: 23rd Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 2003. Proceedings*, pages 44–60. Springer, 2003. [3, 92]

[21] J. C. Faugère, L. Perret, C. Petit, and G. Renault. Improving the complexity of index calculus algorithms in elliptic curves over binary fields. In *Advances in Cryptology—EUROCRYPT 2012*, volume 7237 of *Lecture Notes in Computer Science*, pages 27–44. Springer, Heidelberg, 2012. [6, 49, 54, 56]

[22] P. Gaudry. Index calculus for abelian varieties of small dimension and the elliptic curve discrete logarithm problem. *Journal of Symbolic Computation*, 44(12):1690–1702, 2009. [6, 49, 50, 53]

[23] P. Gaudry, E. Thomé, N. Thériault, and C. Diem. A double large prime variation for small genus hyperelliptic index calculus. *Mathematics of Computation*, 76(257):475–492, 2007. [50]

[24] L. Granboulan, A. Joux, and J. Stern. Inverting HFE is quasipolynomial. In *Advances in Cryptology—CRYPTO 2006*, volume 4117 of *Lecture Notes in Computer Science*, pages 345–356. Springer, Berlin, 2006. [80]

[25] M.-D. A. Huang, M. Kosters, Y. Yang, and S. L. Yeo. On the last fall degree of zero-dimensional Weil descent systems. Journal of Symbolic Computation, 2017. `https://doi.org/10.1016/j.jsc.2017.08.002`. [59]

[26] M.-D. A. Huang, M. Kosters, and S. L. Yeo. Last fall degree, HFE, and Weil descent attacks on ECDLP. In *Annual Cryptology Conference*, pages 581–600. Springer, 2015. [58, 69, 83, 85, 93, 94]

[27] N. Koblitz. *Algebraic aspects of cryptography*, volume 3. Springer Science & Business Media, 2012. [5]

[28] M. Kosters. Polynomial maps on vector spaces over a finite field. *Finite Fields and Their Applications*, 31:1–7, 2015. [82]

[29] M. Kosters and S. L. Yeo. Notes on summation polynomials. Cryptology ePrint Archive, 2015. `https://arxiv.org/abs/1503.08001.pdf`. [2, 25, 92, 94]

[30] M. Kraitchik. *Théorie des nombres*, volume 1. Gauthier-Villars, 1922. [5, 12]

[31] M. Kraitchik. *Recherches sur la théorie des nombres*, volume 1. Gauthier-Villars, 1924. [12]

[32] M. Kreuzer and L. Robbiano. *Computational commutative algebra 1*. Springer-Verlag Berlin Heidelberg, 2000. [30]

[33] D. Lazard. *Gröbner bases, Gaussian elimination and resolution of systems of algebraic equations*, pages 146–156. Springer Berlin Heidelberg, Berlin, Heidelberg, 1983. [44, 72]

[34] A. J. Menezes, T. Okamoto, and S. A. Vanstone. Reducing elliptic curve logarithms to logarithms in a finite field. *iEEE Transactions on information Theory*, 39(5):1639–1646, 1993. [5]

[35] M. S. E. Mohamed, D. Cabarcas, J. Ding, J. Buchmann, and S. Bulygin. MXL3: An efficient algorithm for computing Gröbner bases of zero-dimensional ideals. In *International Conference on Information Security and Cryptology*, pages 87–100. Springer, 2009. [47]

[36] M. S. E. Mohamed, W. S. A. E. Mohamed, J. Ding, and J. Buchmann. MXL2: Solving polynomial equations over GF (2) using an improved mutant strategy. In *International Workshop on Post-Quantum Cryptography*, pages 203–215. Springer, 2008. [47]

[37] J. Patarin. Hidden fields equations (HFE) and isomorphisms of polynomials (IP): Two new families of asymmetric algorithms. In *Advances in Cryptology—EUROCRYPT '96*, volume 1070 of *Lecture Notes in Computer Science*, pages 33–48. Springer, 1996. [1, 91]

[38] C. Petit. Bounding HFE with SRA. Cryptology ePrint Archive, 2014. `http://www0.cs.ucl.ac.uk/staff/c.petit/files/SRA_GB.pdf`. [3, 92, 93]

[39] C. Petit and J.-J. Quisquater. On polynomial systems arising from a Weil descent. In *Advances in Cryptology—ASIACRYPT 2012*, volume 7658 of *Lecture Notes in Computer Science*, pages 451–466. Springer, Heidelberg, 2012. [2, 6, 49, 57, 73, 93, 100, 101, 106]

[40] J. M. Pollard. Monte carlo methods for index computation (mod p). *Mathematics of computation*, 32(143):918–924, 1978. [12]

[41] J. M. Rojas. Solving degenerate sparse polynomial systems faster. *Journal of Symbolic Computation*, 28(1-2):155–186, 1999. [100, 105]

[42] I. Semaev. Evaluation of discrete logarithms in a group of p-torsion points of an elliptic curve in characteristic p. *Mathematics of Computation of the American Mathematical Society*, 67(221):353–356, 1998. [5]

[43] I. A. Semaev. Summation polynomials and the discrete logarithm problem on elliptic curves. Cryptology ePrint Archive, 2004. `http://eprint.iacr.org/2004/031.pdf`. [6, 17, 49]

[44] J. H. Silverman. *The arithmetic of elliptic curves*, volume 106 of *Graduate Texts in Mathematics*. Springer, Dordrecht, second edition, 2009. [5, 9, 11]

[45] S. Smale. Mathematical problems for the next century. *The Mathematical Intelligencer*, 20(2):7–15, 1998. [1]

[46] N. P. Smart. The discrete logarithm problem on elliptic curves of trace one. *Journal of cryptology*, 12(3):193–196, 1999. [5]

[47] M. Sugita, M. Kawazoe, and I. Hideki. Relation between the XL algorithm and Gröbner basis algorithms. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, 89(1):11–18, 2006. [46]

[48] S. Takakazu and A. Kiyomichi. Fermat quotients and the polynomial time discrete log algorithm for anomalous elliptic curves. *Commentarii mathematici Universitatis Sancti Pauli*, 47(1):81–92, 1998. [5]

[49] N. Thériault. Index calculus attack for hyperelliptic curves of small genus. In *Advances in Cryptology - ASIACRYPT 2003, volume 2894 of Lecture Notes in Comput. Sci.*, pages 75–92. Springer, 2003. [49]

[50] J. von zur Gathen and D. Panario. Factoring polynomials over finite fields: A survey. *Journal of Symbolic Computation*, 31(1):3 – 17, 2001. [70, 100, 106]

[51] J. vz Gathen and J. Gerhard. Modern computer algebra. *Cambridge*, 21, 2003. [43]

[52] L. C. Washington. *Elliptic curves: number theory and cryptography*. CRC press, 2008. [5, 12]

[53] B.-Y. Yang and J.-M. Chen. All in the XL family: Theory and practice. In *International Conference on Information Security and Cryptology*, pages 67–86. Springer, 2004. [46]