Groebner Bases and Applications to the Security of Multivariate Public Key Cryptosystems

Daniel Esteban Escudero Ospina

Trabajo de Grado presentado como requisito para optar por el título de Matemático

Asesor
Daniel Cabarcas Jaramillo

UNIVERSIDAD NACIONAL DE COLOMBIA

Escuela de Matemáticas
Facultad de Ciencias
Universidad Nacional de Colombia, Sede Medellín
Colombia
Diciembre, 2016

Contents

List of Algorithms

Introduction

Motivation

A few words on Cryptography

In naive words, a cryptosystem is an algorithm or algorithms that allow two users to share secret information in the possible presence of a malicious third party, in such a way that they are the only ones capable of seeing and manipulating this information. The first idea that may come to our minds involve symmetric cryptosystems, where both parties need to have a common shared secret key and they use that key to both encrypt and decrypt information. This kind of cryptosystems impose a big problem, which is the process of according the common key. If the parties are able to establish a shared secret key securely, why do not they simply share the secret information in the same way? In historical contexts, this key was established in a secure channel like a personal meeting, or a secure line, and this key was used for some time. This may seem to work, but whenever they key must be replaced, the whole complicated process of establishing the key must be repeated. Moreover, communication today is performed between parties anywhere in the world, so a different approach is needed.

A new type of cryptosystems evade this issue. In asymmetric or public key cryptosystems, we don't have only one key but we have two keys per user, a *private key* which only the user knows and a *public key* which is accessible by everyone. Whenever user A wants to send a message to user B, he encrypts the message using B's public key and user B decrypts it using its private key. The well known RSA cryptosystem is a public key cryptosystem.

Post-Quantum Cryptography and MPKC

To introduce what post-quantum cryptography is, consider the cryptosystem RSA. It is widely accepted that computers today can not factor big integers into primes in an efficient manner. This is crucial to the security of RSA since, if one is able to factor large integers into primes, then one is able to find RSA private keys and therefore the cryptosystem is broken. However, quantum computers can perform this task in polynomial time so when these computers appear RSA will not be secure anymore. Moreover, the Diffie-Hellman key exchange protocol and many other cryptographic primitives widely used today will be useless once quantum computers appear [Sho99]. This means that, in order to mantain our communications secure, we need new cryptosystems whose security is based in problems that can not be solved neither by classical computers nor

quantum computers.

There are many problems that we can rely on to build quantum secure cryptosystems [BBD08]. The one of interest to us is that of finding the solutions of a quadratic multivariate polynomial system over a finite field, whose associated decisional problem is an NP-complete [GJ90], and public key cryptosystems whose security is based on the computational difficulty of solving this problem are within the field of **Multivariate Public Key Cryptography** (MPKC) [DGS06]. In these systems the public key is usually a tuple of multivariate quadratic polynomials and encryption is performed by evaluating those polynomials at the desired message, thus, being able to solve this system (set equal to some constants) gives us the ability to find secret messages.

Groebner Bases

Given an ideal I of a polynomial ring $\mathbb{F}[x_1, ..., x_n]$, where $\mathbb{F}$ is a field, a Groebner basis of I is a particular finite generating set of I that has some special and useful properties. In the context of multivariate polynomial rings, a *basis* for a polynomial ideal is a generating set of such. This fact, along the name of the thesis advisor of Bruno Buchberger, the developer of the theory [Buc65], gives the name to Groebner bases. Such basis can be used to solve many algebraic geometry and computational algebra problems, but the most important for MPKC is that it allows to find the zeros of polynomial systems quite efficiently.

A Groebner basis can be computed from any given finite basis and there has been a lot of work in developing more efficients algorithms to accomplish this. However, as we pointed out before, solving a system of polynomial equations over a finite field is known to be a hard computational problem so finding a Groebner basis is a hard computational problem by itself.

Recall that cryptosystems developed within the frame of multivariate public key cryptography (MPKC) can be broken if one is able to solve certain system of polynomial equations, therefore, finding a Groebner basis of the ideal generated by those polynomials is a critical step for breaking such cryptosystems. As mentioned before, finding a Groebner basis is not an easy task in the general case, however, the polynomial equations that arise from MPKC cryptosystems are far from being general because of the necessity of leaving a trapdoor for the legitimate user (the private key). Studying then the complexity of Groebner bases algorithms for the polynomial equations that arise from a particular cryptosystem has become critical for the security of such, and a better understanding of the factors affecting the computation has became imperative. A usual way to measure this complexity is to look at some intrinsic properties of the polynomial system known as the **degree of regularity** and the **falling degree**, which we will explain in detail.

Contribution

There are MPK signature schemes that are both, efficient and secure [DS05]. However, there is no such luck with MPK encryption schemes, and many efforts have been made for developing such. One of the leadings ideas for building these cryptosystems is that

used in HFE [Pat96], which consists of hiding certain easy to invert function over a large field with two affine transformations over a small field. Unfortunately, HFE (and some related schemes) got broken with two mainstream attacks: a min-rank attack by Faugère et al. [BFP13] (which improves Kipnis-Shamir attack [KS99]) and a Direct Algebraic Attack [FJ03] by Faugère and Joux over the binary field.

ZHFE is one of many MPK cryptosystems that arose from this idea. It was presented in 2014 by Porras et al. [PBD15] and it was very well received by the MPKC community for its new and creative idea. Unfortunately, it had efficiency issues in its very beginning. Almost one year after its release, an improvement on the efficiency of ZHFE and a security analysis based in the min-rank were published [BCE+16, PS16]. Although the former gave a hope on the future of ZHFE as a usable primitive, the latter showed a weakness on the cryptosystem that led to the necessity of reformulating it.

New ideas trying to avoid the MinRank attack and the Direct Algebraic Attack began to appear. In this work, we introduce new alternatives based on degree 3 polynomials, which has not been seen in the subject before. Surprisingly, we could develop a MinRank attack in the cubic case, which was not expected since we do not have a standard way of representing "cubic forms". Following this, we propose a variation that is vulnerable to a Direct Algebraic Attack. We study the reason why this attack works and based on this develop what we call "a second layer" that can increase the so-called *falling degree* up to a desired value. This concept, the Falling Degree, has served as measure of the level of security of MPK Cryptosystems. Nevertheless, we show that we must be careful when analyzing MPK Cryptosystems by means of this value by developing an "intelligent" algebraic attack on the latter construction.

As mentioned above, Groebner bases play a big role in this analysis as well as the degree of regularity and the falling degree of the systems generated in this New Alternative. These concepts are addressed in detail in this work.

Outline

This work is divided into two main parts. Part I is an attempt to summarize a lot of the theory known about Groebner Basis. It begins with chapter 1 which includes the basic theoretical definitions and properties of Groebner bases and then we discuss how to compute them in chapter 2. In chapter 3 we talk about the techniques used to estimate the arithmetical complexity of Groebner basis algorithms, including the degree of regularity and the falling degree.

Part II presents the contributions of this work. Chapter 4 gives necessary concepts from cryptography and discusses previous work on the subject. In chapter 5 we present the construction of some multivariate public key cryptosystems and related attacks.

Acknowledgments

I would like to thank Professor Daniel Cabarcas for his supervision throughout this work. Many thanks also to Professor John Bayron Baena, Professor Felipe Cabarcas,

PhD Jaiberth Porras and PhD student Javier Verbel, for the fruitful seminar meetings we had at the university.

This work was partially supported by “Fondo Nacional de Financiamiento para la Ciencia, la Tecnología y la Innovación Francisco José de Caldas”, Colciencias (Colombia), Project No. 111865842333 and Contract No. 049-2015.

I must also thank Cristina Ochoa, whose unconditional support has proven to be substantial for the culmination of this work and my studies in general.

Part I

Groebner Bases

Chapter 1

Groebner Bases: Definitions and Results

In this chapter, we state our main object of study: Groebner bases. We show its existence, some of its properties and its usefulness.

In few words, a Groebner basis of an ideal is a finite generating set of such that gives us information about it in a much deeper way than almost any other basis can do. Groebner bases allow us to obtain a lot of information about the ideal by only looking at them. For instance, all the elimination ideals can be known, and unique representatives for quotient rings can be obtained. A very interesting property of the ideal (perhaps, the most important) that we can read from a Groebner basis is the variety, and this application is basically why we care about Groebner bases in cryptography.

This is a very extensive subject that generalizes to other algebraic structures, however, in this work, we will be interested only in the theory of Groebner bases for polynomial rings over a field.

1.1 Basic Definitions

As a general notation throughout this work we let R denote the multivariate polynomial ring $\mathbb{F}[x_1, \ldots, x_n]$, where $\mathbb{F}$ is a field (not necessarily finite, nor algebraically closed). A **monomial** is any product of the form $x_1^{a_1} \cdots x_n^{a_n}$, with $(a_1, \ldots, a_n) \in \mathbb{N}^n$ (in this work, $\mathbb{N} = \{0, 1, 2, 3, \ldots\}$). We usually write

$$x^\alpha := x_1^{\alpha_1} \cdots x_n^{\alpha_n} \quad \text{and} \quad |\alpha| := \sum_{i=1}^{n} a_i,$$

where $\alpha = (a_1, \ldots, a_n) \in \mathbb{N}^n$. We denote by $\mathcal{M}$ the set of all monomials of R. Notice that

$$R = \bigoplus_{\alpha \in \mathcal{M}} \mathbb{F} x^\alpha$$

and hence each polynomial $f \in R$ can be written uniquely as

$$f = \sum_{x^\alpha \in \mathcal{M}} a_\alpha x^\alpha,$$

where almost all $a_\alpha \in \mathbb{F}$ are equal to zero.

Definition. A *monomial order* is any relation $\prec$ on $\mathcal{M}$ such that

(i) $\prec$ is a total order (every two distinct elements are comparable);

(ii) If $x^\alpha, x^\beta, x^\gamma \in \mathcal{M}$, then $x^\alpha \prec x^\beta$ implies $x^\alpha \cdot x^\gamma \prec x^\beta \cdot x^\gamma$;

(iii) $\prec$ is a well order.

Remark. Clearly, a monomial order can be uniquely identified with a total well-order over $\mathbb{N}^n$ that respects addition, and $\prec$ will denote both, indistinctly.

A Groebner basis will depend on the monomial order chosen, and the time needed for its computation has this dependence too. Some common monomial order are the following. Let $\alpha, \beta \in \mathbb{N}^n$,

- **Lexicographic order (lex):** $\alpha \prec_{\text{lex}} \beta$ if and only if the leftmost nonzero entry of $\beta - \alpha$ is positive. This has the advantage of being a particular case of an *elimination order* that we can use elimination theory methods with.

- **Reverse lexicographic order (rlex):** $\alpha \prec_{\text{rlex}} \beta$ if and only if the rightmost nonzero entry of $\beta - \alpha$ is negative. This order is useful for moving from affine spaces to projective spaces (polynomial homogenization).

- **Graded reverse lexicographic order (grevlex):** $\alpha \prec_{\text{grevlex}} \beta$ if and only if $|\alpha| < |\beta|$ or $|\alpha| = |\beta|$ and $\alpha \prec_{\text{rlex}} \beta$. This order bounds the total degree during the division algorithm. More generally, let $\prec$ be any monomial order, we define $\prec'$ as follows: $\alpha \prec' \beta$ if and only if $|\alpha| < |\beta|$ or $|\alpha| = |\beta|$ and $\alpha \prec \beta$. We say that $\prec'$ is the *graded order of* $\prec$, and it is a particular type of an order that *refines the degree*.

- Consider the ring $\mathbb{R}[X]$ of univariate polynomials in x with real coefficients. Let P be the subset of this ring formed by all polynomials whose leading coefficient is positive, and define for all $f, g \in \mathbb{R}[X] : f \lhd g$ if and only if $g - f \in P$. It can be easily seen that this is a linear order in $\mathbb{R}[X]$ that extends the natural order in $\mathbb{R}$ and satisfies $r \lhd X$ for all $r \in \mathbb{R}$. Let $g_1, \ldots, g_n \in \mathbb{R}[X]$ be $\mathbb{Q}-$linearly independent polynomials and define the following relation on $\mathbb{N}^n$:

$$(a_1, \ldots, a_n) \prec (b_1, \ldots, b_n) \Longleftrightarrow \sum_{i=1}^{n} a_i g_i \lhd \sum_{i=1}^{n} b_i g_i,$$

  one easily sees that this is a monomial order, that we call **monomial order induced by** $(g_1, \ldots, g_n)$. This may seem as a very particular type of monomial orders, but in fact, every monomial order can be obtained with this procedure. For instance, lexicographic order is induced by $(X^{n-1}, \ldots, X, 1)$ and reverse lexicographic is induced (quite unsurprisingly) by $(1, X, \ldots, X^{n-1})$.

From now on, unless otherwise stated, we fix a monomial order $\prec$.

Definition. Let $f = \sum_{x^\alpha \in \mathcal{M}} a_\alpha x^\alpha$ be a nonzero polynomial in R.

(i) The *support* of f is $\operatorname{supp}(f) := \{x^\alpha : a_\alpha \neq 0\}$;

(ii) The *exponent* of f is $\exp(f) := \max_\prec(\alpha : a_\alpha \neq 0)$;

(iii) The *leading monomial* of f is $\operatorname{LM}(f) := \max_\prec(\operatorname{supp}(f)) = x^{\exp(f)}$;

(iv) The *leading coefficient* of f is $\operatorname{LC}(f) := a_\alpha$ with $\alpha = \exp(f)$;

(v) The *leading term* of f is $\operatorname{LT}(f) := \operatorname{LC}(f) \operatorname{LM}(f)$;

(vi) The *degree* of f is $\deg(f) := \sum_{i=1}^n a_i$, where $(a_1, \ldots, a_n) = \exp(f)$.

By convention:

(i) $\exp(0) := -\infty$, which is smaller than any $\alpha \in \mathbb{N}^n$;

(ii) $\operatorname{LM}(f) = \operatorname{LT}(f) := 0$.

Note that the previous definition depends on the monomial order chosen. However, for the sake of simplicity, this is not being considered in the notation.

Example. Consider the case $n = 4$ and $\mathbb{F} = \mathbb{R}$, with the order being the lexicographical order. Let f be the polynomial

$$f = 5 \cdot x_1^2 x_2^3 x_3^1 x_4^1 + 3 \cdot x_2^4 x_3^2 x_4^3 - 5 \cdot x_2^2 x_3^1 x_4^5.$$

Note that this polynomial has been written in decreasing lexicographic order since

$$(2,3,1,1) \succ_{\text{lex}} (0,4,2,3) \succ_{\text{lex}} (0,2,1,5).$$

Also, we have

$$\operatorname{supp}(f) = \{x_1^2 x_2^3 x_3^1 x_4^1, x_2^4 x_3^2 x_4^3, x_2^2 x_3^1 x_4^5\}, \quad \exp(f) = (1,3,1,1),$$

$$\operatorname{LM}(f) = x_1^2 x_2^3 x_3^1 x_4^1, \quad \operatorname{LC}(f) = 5, \quad \operatorname{LT}(f) = 5 \cdot x_1^2 x_2^3 x_3^1 x_4^1.$$

1.2 A Division Algorithm in $\mathbb{F}[x_1, \ldots, x_n]$

In this section we will generalize the classical euclidean algorithm for $n = 1$, having a loss in the uniqueness of the quotient and remainder. We will see later that a Groebner basis is precisely a basis where uniqueness of the remainder is guaranteed.

Theorem 1.2.1. *Let $F = (f_1, \ldots, f_m)$ be an ordered $m-$tuple of nonzero polynomials in R, and for each i let $\alpha_i := \exp(f_i)$. Then, for every $f \in R$ there exist unique polynomials $q_1, \ldots, q_m, r \in R$ such that*

$$f = q_1 f_1 + \cdots + q_m f_m + r$$

where

$$q_i \in \bigoplus_{\alpha \in \Delta_i - \alpha_i} \mathbb{F}x^\alpha \quad \textit{with} \quad \Delta_i := (\alpha_i + \mathbb{N}^n) \setminus \bigcup_{j=1}^{i-1} (\alpha_j + \mathbb{N}^n)$$

and

$$r \in \bigoplus_{\alpha \in \Delta} \mathbb{F}x^\alpha \quad \textit{with} \quad \Delta := \mathbb{N}^n \setminus \bigcup_{i=1}^{m} (\alpha_i + \mathbb{N}^n).$$

Moreover, we have

$$\exp(f) \succeq \exp(q_i f_i), \exp(h).$$

Proof. We first show uniqueness. Given any two decompositions of this type, we can subtract them to obtain a decomposition for the polynomial 0, hence, it suffices to show that the only decomposition for the zero polynomial is the trivial one, given by $q_i, r = 0$. Notice that $\mathbb{N}^n = \Delta_1 \sqcup \cdots \sqcup \Delta_m \sqcup \Delta$ (disjoint union) and since $\exp(q_i f_i) \in \Delta_i, \exp(r) \in \Delta$, we can conclude that

$$\exp(f) = \max\left\{\exp(q_1 f_1), \ldots, \exp(q_m f_m), \exp(r)\right\}.$$

In particular, if $f = 0$, then $q_i = 0$ for all i and $h = 0$, which proves uniqueness. Actually, this also shows that $\exp(f) \succ \exp(q_i f_i)$ and $\exp(f) \succ \exp(h)$.

We now show existence for $f \neq 0$. Suppose that f has some term that is divisible by some $\mathrm{LM}(f_i)$ and take the greatest of all these terms, say $cx^\alpha \,\mathrm{LT}(f_i)$ with $c \in \mathbb{F}^*$ and $\alpha \in \mathbb{N}^n$. Assume that i is the smallest index with this property, then $\alpha + \alpha_i \in \Delta_i$ and therefore $\alpha \in \Delta_i - \alpha_i$. Now we apply the same procedure to $f' := f - cx^\alpha f_i$ instead of f. Notice that, due to the maximality of $cx^\alpha \,\mathrm{LT}(f_i)$, if $c' x^{\alpha'} \,\mathrm{LT}(f_{i'})$ is a term of f' then it is strictly smaller than $cx^\alpha \,\mathrm{LT}(f_i)$.

Since $\prec$ is a well order, this process must eventually stop. At this point, by grouping terms, we obtain polynomials $q_1, \ldots, q_m$ of the desired shape such that $r := f - q_1 f_1 - \cdots - q_m f_m$ has no terms divisible by some $\mathrm{LT}(f_i)$, hence $r \in \oplus_{\alpha \in \Delta} \mathbb{F}x^\alpha$. □

Remark. The process we just described for finding this decomposition is known as the *division algorithm*. Uniqueness of the outputs of this algorithm is guaranteed only if, on each iteration, we take the less index i such that some term of f is divisible by some $\mathrm{LM}(f_i)$. If we drop this restriction, uniqueness of the quotients q_i will be lost, but uniqueness of r can be kept if the set $G = \{f_1, \ldots, f_m\}$ is a Groebner basis, as we will see later on the chapter. Also, notice that this algorithm is precisely the classical euclidean division algorithm when $n = 1$ and $s = 1$, however, in the general case, we allow division by a set of polynomials (rather than only one polynomial) and division steps are performed using any monomial of the divisors, not only the leading monomials. Division algorithm is included as algorithm 1

We have to keep in mind that the outputs of this algorithm, the quotients q_i's and the remainder r, depend on the order of the f_i's. Different tuples of divisors can lead to different quotients and remainders, even if the tuples share the same polynomials. A more subtle fact is that these output depend on the monomial order chosen: different orders can lead to different quotients and remainders.

Algorithm 1: Polynomial division

Input: $f \in R$, $(f_1, \ldots, f_m) \in (R \setminus \{0\})^m$
Output: $(q_1, \ldots, q_m) \in R^m$ and $r \in R$ satisfying the conclusion of theorem 1.2.1

$q_1 := 0, \ldots, q_m := 0, r := 0$
$p := f$
while $p \neq 0$ **do**
    $i := 1$
    division_occurred := **false**
    while $i \leq m$ **and** division_occurred $=$ **false do**
        if $\mathrm{LT}(f_i)$ divides $\mathrm{LT}(p)$ **then**
            $q_i := q_i + \mathrm{LT}(f_i)/\mathrm{LT}(p)$
            $p := p - (\mathrm{LT}(f_i)/\mathrm{LT}(p))f_i$
            division_occurred := **true**
        else
            $i := i + 1$
        if division_occurred $=$ **false then**
            $r := r + \mathrm{LT}(p)$
            $p := p - \mathrm{LT}(p)$
return $(q_1, \ldots, q_m), r$

Let $f_1, \ldots, f_m \in R$, consider the ideal generated by these polynomials

$$I := \langle f_1, \ldots, f_m \rangle$$

and let $f \in R$. Let r be the remainder upon division by $(f_1, \ldots, f_m)$ using the division algorithm stated before. It is clear that if $r = 0$, then $f \in I$. However, this is not a necessary condition in general, i.e., remainder can be nonzero even if the dividend lies on the ideal generated by the divisors.

This shows that division algorithm does not determine whether a polynomial lies in a given finitely generated ideal or not, in contrast to the univariate case. Nevertheless, there are 'good' finite generating sets that possesses the following property: f lies in the ideal generated by this set if and only if remainder of division by this set is zero (no matter how the elements of this set are listed). These 'good' finite generating sets are what we will call Groebner bases.

1.3 Dickson's Lemma and Hilbert Basis Theorem

Hilbert basis theorem is a classical and important result in algebraic geometry, it states that all ideals of R are finitely generated. Although there are many ways to prove this result, we present a proof that uses an important lemma called *Dickson's lemma*, which will imply the existence of Groebner bases by its own. We need at first some definitions.

Definition. A *monomial ideal* is an ideal of R that is generated by monomials. More precisely, I is a monomial ideal if there exists $\mathcal{A} \subseteq \mathcal{M}$ such that $I = \langle \mathcal{A} \rangle$.

Remark. By convention, the ideal generated by the empty set is R, so it is in particular a monomial ideal.

Monomial ideals play a central role in the theory since the definition of Groebner basis we will give is involved with this type of ideals. We now state an important property that monomial ideals have, which is crucial in order to prove Dickson's lemma and has further applications by its own.

Proposition 1.3.1. *Let I be a monomial ideal of R. Then f lies in I if and only if* $\operatorname{supp}(f) \subseteq I$.

The proof of this proposition follows easily from the fact that the monomials of a monomial ideal are precisely the $R-$multiples of its generators. Notice that for an ideal I of R (not necessarily a monomial ideal) the condition stated by this proposition may not hold, i.e., there can be $f \in I$ with some terms outside I. In order to see this, take for example $\mathbb{F} = \mathbb{R}$, $n = 2$ and $I = \langle x + y \rangle$, then $x + y \in I$ but $x \notin I$. This condition is what makes monomial ideals important, since it implies that a monomial ideal is characterized by the monomials it contains, i.e., two monomial ideals are the same if and only if they contain the same monomials.

Definition. We define the partial order $\mid$ over $\mathbb{N}^n$ as follows. For $\alpha = (a_1, \ldots, a_n)$ and $\beta = (b_1, \ldots, b_n)$ in $\mathbb{N}^n$,

$$\alpha \mid \beta \iff \forall i : a_i \leq b_i.$$

The notation is consistent with the divisibility relation on the monomials, that is, $\alpha \mid \beta$ if and only if x^α divides x^β. The following is an easy to prove proposition.

Proposition 1.3.2. *Every nonempty subset $M \subset \mathbb{N}^n$ has a $\mid -$minimal element. In particular, for each $\beta \in M$ there exists a $\mid -$minimal element α of M such that $\alpha \mid \beta$.*

As a corollary, any nonempty subset $M \subset \mathbb{N}^n$ satisfying $M + \mathbb{N}^n = M$ is determined by its $\mid -$minimal elements:

$$M = \bigcup_{\alpha \mid -\min} (\alpha + \mathbb{N}^n).$$

We are now ready to state and prove a combinatorial version of Dickson's lemma.

Lemma 1.3.3. (Combinatorial Dickson's lemma). *Let M be a nonempty subset of $\mathbb{N}^n$, then M has a finite number of $\mid -$minimal elements.*

Proof. We proceed by induction on n. For $n = 1$, the result is trivial. Assume that the result holds for $n-1$, we now show that it must hold for n. Let $\alpha = (a_1, \ldots, a_n) \in M$ be a $\mid -$minimal element of M, any other $\mid -$minimal element of M must lie in the set

$$M \setminus (\alpha + \mathbb{N}^n) = \bigcup_{i=1}^{n} \mathcal{A}_a^i$$

with

$$\mathcal{A}_a^i = \bigcup_{a=0}^{a_i} \{(c_1, \ldots, c_n) \in M : c_i = a\}.$$

Let $\beta = (b_1, \ldots, b_n)$ be another $\mid$ −minimal element of M, and let i be such that $b_i < a_i$, then β is a $\mid$ −minimal element of $\mathcal{A}^i_{b_i}$. Let π_i be the projection from $\mathbb{N}^n$ onto $\mathbb{N}^{n-1}$ that drops the $i-$th entry, we claim that $\pi_i(\beta)$ is a $\mid$ −minimal element of $\pi_i\left(\mathcal{A}^i_{b_i}\right)$. Indeed, if there exists $\gamma = (c_1, \ldots, c_n) \in \mathcal{A}^i_{b_i}$ with $\pi_i(\gamma) \mid \pi_i(\beta)$ but $\pi_i(\gamma) \neq \pi_i(\beta)$, given that $c_i = b_i$, we obtain that $\gamma \mid \beta$ but $\gamma \neq \beta$, a contradiction since β is a $\mid$ −minimal element of $\mathcal{A}^i_{b_i}$.

By the induction hypothesis, there is a finite number of $\mid$ −minimal elements of $\mathcal{A}^i_{b_i}$ for each i, hence, there is a finite number of $\mid$ −minimal elements of M. □

Corollary 1.3.4. *Let $\prec$ be a total order that respects addition on $\mathbb{N}^n$. The following are equivalent.*

(i) $\prec$ is a well order (every nonempty subset of $\mathbb{N}^n$ has a minimal element);

(ii) $0 \prec \alpha$ for all nonzero $\alpha \in \mathbb{N}^n$;

(iii) For all $\alpha, \beta \in \mathbb{N}^n$, if $\alpha \mid \beta$ and $\alpha \neq \beta$, then $\alpha \prec \beta$

Proof. Suppose that $0 \prec \alpha$ for all nonzero $\alpha \in \mathbb{N}^n$, if $\alpha, \beta \in \mathbb{N}^n$ are such that $\alpha \mid \beta$ and $\alpha \neq \beta$, then $\beta - \alpha$ is nonzero and lies in $\mathbb{N}^n$, hence $0 \prec \beta - \alpha$ and therefore, since $\prec$ respects addition, $\alpha \prec \beta$. Conversely, if $0 \prec \alpha$ and *(iii)* holds, since $0 \mid \alpha$ and $\alpha \neq 0$ we obtain that $0 \prec \alpha$. This shows that *(ii)* is equivalent to *(iii)*.

Suppose that there exists $\alpha \prec 0$, then $\cdots \prec 3\alpha \prec 2\alpha \prec \alpha$ is an infinite decreasing sequence, then $\prec$ can not be a well order. This shows that *(i)* implies *(ii)*.

Finally, we show that *(iii)* implies *(i)*. Suppose that *(i)* does not hold, that is, there exists an infinite chain $r_1 \succ r_2 \succ r_3 \succ \cdots$. If $r_i \mid r_{i+1}$ we would have by the hypothesis that $r_i \prec r_{i+1}$, which is absurd, hence $r_i \nmid r_{i+1}$. Consider the sets

$$T_i := \bigcup_{j=1}^{i} (r_i + \mathbb{N}^n),$$

we obtain a strictly increasing chain $T_1 \subsetneq T_2 \subsetneq T_3 \subsetneq \cdots$ of subsets of $\mathbb{N}^n$ that satisfy $T_i + \mathbb{N}^n = T_i$. By lemma 1.3.3, $T := \cup_{i=1}^{\infty} T_i$ has a finite number of $\mid$ −minimal elements, say $m_1, \ldots, m_\ell$. Let j be such that all of these elements lie in some T_j, since $T + \mathbb{N}^n = T$, it is characterized by its minimal elements:

$$T = \bigcup_{i=1}^{\ell} (m_i + \mathbb{N}^n) \subseteq T_j \subseteq T$$

and therefore $T_i = T_j$ for all $i \geq j$, which is absurd. □

This corollary shows that we can replace condition (iii) in the definition of a monomial order by any of the following:

- $1 \prec x^\alpha$ for all $x^\alpha \in \mathcal{M} \setminus \{1\}$;
- For all $x^\alpha, x^\beta \in \mathcal{M}$, if x^α divides properly x^β, then $x^\alpha \prec x^\beta$.

Lemma 1.3.5. (Dickson's lemma). *Let I be a monomial ideal, say $I = \langle \mathcal{A} \rangle$ with $\mathcal{A} \subseteq \mathcal{M}$, then there exists a finite subset $\mathcal{B} \subseteq \mathcal{A}$ such that $I = \langle \mathcal{B} \rangle$.*

Proof. If $\mathcal{A} = \emptyset$ the result is trivial, so assume this is not the case. Let

$$M = \exp\{\alpha \in \mathbb{N}^n : x^\alpha \in \mathcal{A}\} \neq \emptyset.$$

By lemma 1.3.3, M has a finite number of $\mid$ −minimal elements, say $m_1, \ldots, m_\ell$. We claim that $I = \langle \mathcal{B} \rangle$, where $\mathcal{B} = \{x^{m_1}, \ldots, x^{m_\ell}\} \subseteq \mathcal{A}$.

Indeed, by proposition 1.3.1, it suffices to see that every $x^\alpha \in \mathcal{A}$ lies in $\langle \mathcal{B} \rangle$, but this is trivial since in this case $\alpha \in M$ so $m_i \mid \alpha$ for some i, hence x^{m_i} divides x^α and therefore $x^\alpha \in \langle \mathcal{B} \rangle$. □

Dickson's lemma says that monomial ideals are finitely generated, which is in some sense a particular case of Hilbert basis theorem. However, the latter follows easily from the former, as we now show.

Theorem 1.3.6. (Hilbert basis theorem). *Let I be an ideal of R, then I is finitely generated.*

Proof. Assume $I \neq \{0\}$, for if this does not hold, the result is trivially true. Let $\mathrm{LM}(I)$ be the set of all leading monomials of the polynomials in I, then $\langle \mathrm{LM}(I) \rangle$ is a monomial ideal and therefore, by Dickson's lemma, there are $g_1, \ldots, g_m \in I$ such that

$$\langle \mathrm{LM}(I) \rangle = \langle \mathrm{LM}(g_1), \ldots, \mathrm{LM}(g_m) \rangle.$$

We claim that

$$I = \langle g_1, \ldots, g_m \rangle$$

and then, the theorem would be proved. Clearly $I \supseteq \langle g_1, \ldots, g_m \rangle$. Let $f \in I$ and apply the division algorithm to divide f by $(g_1, \ldots, g_m)$ to obtain $a_1, \ldots, a_m, r \in R$ such that

$$f = a_1 g_1 + \cdots + a_m g_m + r$$

and the other conclusions of theorem 1.2.1 hold. Notice that

$$r = f - a_1 g_1 - \cdots - a_m g_m \in I.$$

If $r \neq 0$, then $\mathrm{LM}(r) \in \langle \mathrm{LM}(I) \rangle = \langle \mathrm{LM}(g_1), \ldots, \mathrm{LM}(g_m) \rangle$ and therefore $\mathrm{LM}(r)$ is divisible by some $\mathrm{LM}(g_i)$, which contradicts the conclusions of theorem 1.2.1. Hence $r = 0$ and therefore

$$f = a_1 g_1 + \cdots + a_m g_m + 0 \in \langle g_1, \ldots, g_m \rangle,$$

this finishes the proof. □

Recall that a ring A is said to be **Noetherian** if one of the following equivalent conditions holds,

(i) Every ideal of A is finitely generated;

(ii) Every nonempty set of ideals of A has a maximal element;

(iii) Every increasing chain of ideals of A eventually stabilize.

The previous theorem shows that R satisfy condition (i), hence, it is Noetherian. In particular, every increasing chain of ideals of R eventually stabilize. This fact will be used in several consequent proofs.

1.4 Groebner Bases

We now turn our attention to Groebner bases, we show their existence and some useful properties they possess. At the end of the section, we show the existence of certain type of Groebner bases that happen to be unique.

Definition. Let I be an ideal of R. We say that $\{g_1, \dots, g_m\} \subseteq I$ is a *Groebner basis* of I if

$$\langle \mathrm{LM}(g_1), \dots, \mathrm{LM}(g_m) \rangle = \langle \mathrm{LM}(I) \rangle,$$

where $\mathrm{LM}(I)$ is the set of all leading monomials of the polynomials in I.

As expected, it turns out that Groebner bases are bases, i.e., they generate the ideal being considered. This fact, along the existence of Groebner bases, are shown in the following proposition.

Proposition 1.4.1. *Let I be a nonzero ideal of R, then I has a Groebner basis and such a set is indeed a basis for I.*

Proof. Actually, we have already proven it, we only need to take a look at the proof of Hilbert Basis theorem. As we saw there, $\langle \mathrm{LM}(I) \rangle$ is finitely generated by some $\mathrm{LM}(g_i)$'s, with each $g_i \in I$, so Groebner bases do exist. Also, using the division algorithm we saw that this condition implies that $I = \langle g_1, \dots, g_m \rangle$, so Groebner bases are indeed bases. □

Remark. Due to this proposition, we can call a Groebner basis G of I simply a Groebner basis, since it is a basis of $\langle G \rangle$

We mentioned before that Groebner bases are 'good' bases since they have the following property: f lies in the ideal generated by this set if and only if remainder of division by this set is zero. Although the definition we gave seems not to be related with this property, those are actually equivalent, as we now show.

Proposition 1.4.2. *Let $\{g_1, \dots, g_m\}$ be a subset of an ideal I of R. Then G is a Groebner basis if and only if for all $f \in I$, the remainder of division of f by $(g_1, \dots, g_m)$ is zero.*

Proof. ($\Rightarrow$). Let $f \in I$ and suppose $g \in I$ and $r \in R$ are such that $f = g + r$ and no monomial of r is divisible by any $\mathrm{LM}(g_i)$. Suppose $r \neq 0$. Since $r = f - g \in I$, $\mathrm{LM}(r) \in \langle \mathrm{LM}(I) \rangle = \langle \mathrm{LM}(g_1), \dots, \mathrm{LM}(g_m) \rangle$ so $\mathrm{LM}(r)$ is divisible by some $\mathrm{LM}(g_i)$, which contradicts the assumption. In particular, since division algorithm yields an expression of this form, we have that the remainder of division of f by $(g_1, \dots, g_m)$ is zero.

($\Leftarrow$). We show that $\langle \mathrm{LM}(I) \rangle \subseteq \langle \mathrm{LM}(g_1), \dots, \mathrm{LM}(g_m) \rangle$. Let $f \in I$, apply division algorithm, theorem 1.2.1, to divide f by G. By hypothesis, the remainder of this division is zero, we obtain then $q_1, \dots, q_m \in R$ satisfying the conclusion of theorem 1.2.1. As we saw in the proof of this theorem, this implies that

$$\exp(f) = \max \{\exp(q_1 f_1), \dots, \exp(q_m f_m)\}$$

and therefore $\mathrm{LM}(f) = \mathrm{LM}(q_i)\,\mathrm{LM}(f_i)$ for some i, thus $\mathrm{LM}(f) \in \langle \mathrm{LM}(g_1), \dots, \mathrm{LM}(g_m) \rangle$. □

Corollary 1.4.3. *Remainders of division algorithm with Groebner bases as divisors are unique, no matter how the elements are listed.*

Proof. Suppose that G is a Groebner basis and let $f \in R$. Suppose that $r \in R$ and $r' \in R$ are remainders of f upon division by G (with G possibly listed in different orders), then we can write

$$f = g + r, \quad f = g' + r'$$

where $g, g' \in I$ and no monomial of r and r' is divisible by any element of $\mathrm{LM}(G)$. This implies that

$$0 = (g' - g) + (r - r')$$

where $r - r'$ satisfy the same property. Since 0 and $g - g'$ lie in I, the proof of the first implication in the previous theorem implies that $r - r' = 0$, i.e., $r = r'$. □

The consequences of the previous corollary are very important. Let I be an ideal of R, and G be a Groebner basis of I, then for all $f, g \in R$ such that $f - g \in I$ we have that the remainder is zero. Since the remainder is linear on its inputs, the remainder of f and g must be the same. In other words, the remainder of every polynomial is well defined modulo I, so we have a unique way to represent the elements of the quotient ring R/I.

We now analyze uniqueness of Groebner bases. It can be easily seen that Groebner bases are not unique in general, for example, if G is a Groebner basis of an ideal I, it is clear that $G \cup \{f\}$ is a Groebner basis of I for all $f \in I$. Even more subtle redundancy can be found if there is $p \in G$ such that $\mathrm{LM}(p) \in \langle \mathrm{LM}(G - \{p\}) \rangle$, since in this case it can be easily shown that $G - \{p\}$ is also a Groebner basis of I. To avoid this kind of redundancy, we remove all $p \in G$ satisfying the previous condition and for simplicity we set all the polynomials in G to be monic. This procedure leads to the concept of minimal Groebner basis.

Definition. Given an ideal I of R and a Groebner basis G of I, we say that G is a *minimal Groebner basis* if every $g \in G$ is monic and for all $p \in G$, $\mathrm{LM}(p) \notin \langle \mathrm{LM}(G - \{p\}) \rangle$.

Let $G = \{g_1, \ldots, g_m\}$ be a minimal Groebner basis of I, and let $\alpha_i := \exp(g_i)$ for each i. It can be easily seen that the $\{\alpha_1, \ldots, \alpha_m\}$ is precisely the set of $|\ -$minimal elements of $\exp(I)$, so these exponents are intrinsic to I. In particular, any two minimal Groebner basis have the same cardinality. However, although the exponents in a minimal Groebner bases are unique, the basis per se is still not unique, this is easily seen since the definition only involves leading monomials, thus, anything can happen in the other terms of the polynomials. For instance, if $\mathbb{F} = \mathcal{R}$, $n = 2$, $f_1 = x^2 + axy$, $f_2 = xy$ and $f_3 = y^2 - (1/2)x$, one can verify that $\{f_1, f_2, f_3\}$ is a minimal Groebner basis of $I = \langle x^2, f_2, f_3 \rangle$, for each $a \in \mathcal{R}$.

We can tighten the previous definition to obtain a particular type of minimal Groebner bases, which happens to be unique.

Definition. Given an ideal I of R and a Groebner basis G of I, we say that G is a *reduced Groebner basis* if every $g \in G$ is monic and for all $p \in G$, no monomial of p lies in $\langle \mathrm{LM}(G - \{p\}) \rangle$.

It is clear that minimal Groebner bases exist, however, this is not so clear with reduced Groebner bases. This, along uniqueness, is shown in the following proposition.

Proposition 1.4.4. *Let $I \neq 0$ be a polynomial ideal, then I has a unique reduced Groebner basis.*

Proof. Let $G = \{g_1, \dots, g_m\}$ be a minimal Groebner basis of I, and let $\alpha_i := \exp(g_i)$ for each i, by changing the numeration, we can assume that $\alpha_1 \prec \alpha_2 \prec \cdots \prec \alpha_r$. Now we replace iteratively each g_i by its remainder when it is divided by $(g_1, \dots, g_{i-1})$, then we can see inductively that

$$g_i \in x^{\alpha_i} + \bigoplus_{\substack{\alpha \in \Delta \\ \alpha \prec \alpha_i}} \mathbb{F} x^{\alpha}$$

with $\Delta = \mathbb{N}^n \setminus \exp(I)$. From this it follows easily that the 'new' G is a reduced Groebner basis.

We now show uniqueness. Suppose that $\{f_1, \dots, f_m\}$ is another reduced Groebner basis of I, then for all i

$$f_i - g_i \in I \cap \left(\bigoplus_{\alpha \in \Delta} \mathbb{F} x^{\alpha} \right) = \{0\}$$

so $f_i = g_i$. □

1.5 Some Applications of Groebner Bases

In this chapter, we show some of the main applications of Groebner bases, making a particular emphasis on the problem of solving algebraic systems, since this is the main application of Groebner bases in cryptography.

1.5.1 The Ideal Membership Problem

Given an ideal I or R, this problem is concerned in deciding whether a given polynomial $p \in R$ lies in I or not. In the univariate case, the problem is completely solved by the division algorithm, since in this case remainders on division algorithm are unique and any ideal is generated by any of its polynomials of smallest degree, so we only need to perform division on p by one of these polynomials and then p will lie in the ideal if and only if the remainder of this division is zero. However, as we noticed previously, when $n > 1$ the division algorithm is an imperfect generalization of its univariate version since remainders (and quotients) are not unique in general. Corollary 1.4.3 shows that this issue is not present when we have Groebner bases as divisors, and in this case, we can proceed as in the univariate case.

1.5.2 The Ideal Equality Problem

Given two polynomial ideals I, J of R, this problem aims to determine whether $I = J$. Keeping in mind the previous section, there is a direct approach to solve this problem, which consists on finding a Groebner basis for each ideal and then verifying if every

polynomial of each basis lies in the other ideal; this, by the previous application, is an efficient computation. However, proposition 1.4.3 shows that we can compute a reduced Groebner basis for each ideal and then those ideals will be the same if and only if these bases are so. Computing a reduced Groebner basis from any minimal Groebner basis is an efficient task that can be accomplished by means of algorithm 2.

Algorithm 2: Computation of reduced Groebner bases

Input: G a minimal Groebner basis of $\langle G \rangle$
Output: A reduced Groebner basis of $\langle G \rangle$
1: $F := G$
2: $p := f$
3: **for** $g \in F$ **do**
4: $\quad g' :=$ remainder on division of g by G
5: $\quad G := (G \setminus \{g\}) \cup \{g'\}$
6: **return** G

1.5.3 Elimination Theory

Definition. Let $I \subseteq R$ be a polynomial ideal. For any $0 \leq j < n$, we define the *j-th elimination ideal of I* as the ideal of $\mathbb{F}[x_{j+1}, \ldots, x_n]$ given by

$$I_{(j)} := I \cap \mathbb{F}[x_{j+1}, \ldots, x_n].$$

It is easy to see that $I_{(j)}$ is indeed an ideal of $\mathbb{F}[x_{j+1}, \ldots, x_n]$. Also, notice that $R_{(j)} = \mathbb{F}[x_{j+1}, \ldots, x_n]$ and $I_{(0)} = I$.

Definition. An *elimination order* of the variables $x_1, \ldots, x_n$ is a monomial order $\prec$ of $R[y_1, \ldots, y_m]$ such that, for all monomials $x^{\alpha_1}y^{\beta_1}$ and $x^{\alpha_2}y^{\beta_2}$, it holds that $x^{\alpha_1} \prec x^{\alpha_2}$ implies $x^{\alpha_1}y^{\beta_1} \prec x^{\alpha_2}y^{\beta_2}$. Such an order is also called a *block order* $y_1, \ldots, y_m \ll x_1, \ldots, x_n$.

Example. Given any two monomial orders $\prec_1$ and $\prec_2$ on $\mathbb{F}[x_1, \ldots, x_n]$ and $\mathbb{F}[y_1, \ldots, y_n]$ respectively, we can always construct a block order $y_1, \ldots, y_m \ll x_1, \ldots, x_n$ as follows: let $x^{\alpha_1}y^{\beta_1} \prec x^{\alpha_2}y^{\beta_2}$ if and only if $x^{\alpha_1} \prec_1 x^{\alpha_2}$ or $x^{\alpha_1} = x^{\alpha_2}$ and $y^{\beta_1} \prec_2 y^{\beta_2}$.

These monomial orders are of particular importance since they preserve Groebner bases when we project over the elimination ideals:

Theorem 1.5.1. ***(Elimination theorem).*** *[CLO07, Thm 2, §1, Chap 2] Let I be a polynomial ideal of $\mathbb{F}[x_1, \ldots, x_n, y_1, \ldots, y_m]$ and let G be a Groebner basis for I with respect to a block order $y_1, \ldots, y_m \ll x_1, \ldots, x_n$. Then $G_{(n)} = G \cap \mathbb{F}[y_1, \ldots, y_m]$ is a Groebner basis for $I_{(n)}$ as an ideal of $R_{(n)} = \mathbb{F}[y_1, \ldots, y_n]$.*

In particular, since any lex order with $x_1 \succ_{\text{lex}} x_2 \succ_{\text{lex}} \cdots \succ_{\text{lex}} x_n$ is a block order $x_{j+1}, \ldots, x_n \ll x_1, \ldots, x_j$ for all $1 \leq j < n$, we have the following corollary.

Corollary 1.5.2. *Let I be a polynomial ideal of R and let G be a Groebner basis for I with respect to lex order with $x_1 \succ_{lex} x_2 \succ_{lex} \cdots \succ_{lex} x_n$. Then $G_{(j)}$ is a Groebner bases for $I_{(j)}$ as an ideal of $R_{(j)}$.*

Thanks to this proposition, now we can compute $I_{(j)}$ for any ideal I, just by computing a Groebner basis of it and looking for polynomials in the variables $x_{j+1}, \ldots, x_n$.

Theorem 1.5.3. (Closure theorem). *Let I be an ideal in R and let $0 \leq j < n$. Let $\pi_j : \mathbb{F}^n \to \mathbb{F}^{n-j}$ be the projection given by $(a_1, \ldots, a_n) \mapsto (a_{j+1}, \ldots, a_n)$. Then $\mathbf{V}(I_{(j)})$ is the closure of $\pi_j(\mathbf{V}(I))$ in $\mathbb{F}^{n-j}$ with the Zariski topology.*

1.5.4 Solving Systems of Polynomial Equations

This problem deals with finding the common roots of a set of polynomials, if they exist. This is the most important application of Groebner bases to cryptography.

In linear algebra, the general situation for linear systems with equal number of equations and variables is that the number of solutions equals one. Moreover, when there are less equations than variables, we usually have an infinite number of solutions and when there are more equations than variables, no solutions are expected. Surprisingly, this behavior is also present with polynomial systems (not necessarily linear). More precisely, an overdetermined polynomial system is expected to possess no solutions, an undetermined polynomial system is expected to possess an infinite number of solutions, and those who have equal number of polynomials and variables are expected to have a finite number of solutions, bounded by Bézout bound (unique solution is not guaranteed here). This will be seen in more detail in chapter 3, when we study regular and semi-regular sequences.

This is not the only analogy of solving linear systems to solving nonlinear polynomial systems. When we have a linear system $Ax = b$, we perform some "legal" operations on the linear polynomials that guarantee two things: the solutions will not be changed, and these are much more easy to read from the obtained linear polynomials than it was from the original (triangular form, for instance); the key point here is that the solutions are shared among all the bases of the vector space spanned by the linear expressions, so finding new "structured" bases will help us get the solutions.

In polynomial system solving it is not enough to consider the vector space spanned by the polynomials, and this role is actually taken by the ideal generated by them. Moreover, structured bases for the vector space will be played by Groebner bases, and the Gaussian reduction that let us obtain new structured bases for the vector space will be replaced by Groebner basis algorithms.

More precisely, it can be easily seen that the solutions to the system $(f_1 = 0, \ldots, f_m = 0)$ are the same to $(g_1 = 0, \ldots, g_s = 0)$ whenever $\langle f_1, \ldots, f_m \rangle = \langle g_1, \ldots, g_s \rangle$, so we can compute these solutions using any basis of the ideal $\langle f_1, \ldots, f_m \rangle$.

Definition. A polynomial system $(f_1 = 0, \ldots, f_m = 0)$ is said to be *zero-dimensional* if it has finitely many solutions.

As we will show, computing the solutions of $(g_1 = 0, \ldots, g_s = 0)$ when $\{g_1, \ldots, g_s\}$ is a reduced lex Groebner basis and the system is zero-dimensional is a very efficient task,

basically because like Guassian reduction, a Groebner basis has a triangular form. Thus, in order to find the solutions to $(f_1 = 0, \ldots, f_m = 0)$ for given $f_1, \ldots, f_m \in R$, we simply compute a reduced Groebner basis $\{g_1, \ldots, g_s\}$ for $\langle f_1, \ldots, f_m \rangle$ and then compute the solutions to $(g_1 = 0, \ldots, g_m = 0)$.

We dedicate what is left in the section to show how to compute the solutions in this case.

Proposition 1.5.4. *Let $(f_1 = 0, \ldots, f_m = 0)$ be a zero-dimensional system and $G = \{g_1, \ldots, g_\ell\}$ be a minimal Groebner basis of $I = \langle f_1, \ldots, f_m \rangle$ with respect to lexicographical order with $x_1 \succ_{lex} x_2 \succ_{lex} \cdots \succ_{lex} x_n$, such that $\mathrm{LM}(g_\ell) \prec_{lex} \cdots \prec_{lex} \mathrm{LM}(g_1)$. Then $g_\ell \in \mathbb{F}[x_n]$ and there exists a strictly increasing sequence $1 = i_1 < i_2 < \ldots < i_n = \ell$, such that for all $j \in \{1, \ldots, n-1\}$ and all $k \in \{i_j, \ldots, i_{j+1} - 1\}, g_k \in \mathbb{F}[x_j, \ldots, x_n]$ and $g_k \notin \mathbb{F}[x_{j+1}, \ldots, x_n]$. More graphically,*

$$G = \left\{ \begin{array}{c} g_1(x_1, \ldots, x_n) \\ \vdots \\ g_{i_2-1}(x_1, \ldots, x_n) \\ g_{i_2}(x_2, \ldots, x_n) \\ \vdots \\ g_{i_3}(x_3, \ldots, x_n) \\ \vdots \\ g_{\ell-1}(x_{n-1}, x_n) \\ g_\ell(x_n) \end{array} \right\}.$$

When the ideal is radical, this proposition can be strengthened as follows.

Lemma 1.5.5. ***(Shape lemma [GM89]).*** *Let I be a radical ideal. Then, after most linear changes of coordinates the lex reduced Groebner basis for I has the following simplified structure:*

$$\{x_1 - g_1(x_n), x_2 - g_2(x_n), \ldots, x_{n-1} - g_{n-1}(x_n), g_n(x_n)\},$$

where each g_i is a univariate polynomial.

The algorithm for solving the system is very straightforward, and, just like the linear case, it is based in backwards substitution. To solve the system we begin by finding the common roots of the last set of univariate polynomials, which by taking the greatest common divisor (using Euclid algorithm) of these can be reduced to the problem of finding the roots of one univariate polynomial; it must be noted that this problem is not always trivial, and several research has been devoted to finding roots of univariate polynomials, however, in finite fields, we have Berlekamp's algorithm and Cantor-Zanssenhaus algorithm that are decently efficient and make no bottleneck in polynomial system solving. After this, we substitute each solution obtained into the next set of bivariate equations making them univariate; the common roots of these polynomials are the roots of their greatest common divisor, which again can be computed efficiently using Euclid algorithm. Finally, we iterate this until all polynomials have been considered. This procedure is stated more formally below.

Proposition 1.5.6. *Let $G = \{g_1, \ldots, g_m\}$ be the reduced Groebner basis of $\langle G \rangle$ with respect to lexicographic order with $x_1 \succ_{lex} x_2 \succ_{lex} \cdots \succ_{lex} x_n$. Suppose that the system $(g_1 = 0, \ldots, g_m = 0)$ is zero-dimensional. Then, the following algorithm computes all the solutions of this system.*

Algorithm 3: Computation of $\mathbf{V}(g_1, \ldots, g_m)$

Input: *$G = \{g_1, \ldots, g_m\}$ a reduced Groebner basis of $\langle G \rangle$*
Output: *The set $\mathbf{V}(G)$*

1: ***if*** $1 \in G$ ***then***
2: ***return*** $\emptyset$
3: *Let g be the only polynomial in $G \cap \mathbb{F}[x_n]$*
4: $S_n := \{a \in \mathbb{F} : g(a) = 0\}$
5: ***for*** $j := n-1, n-2, \ldots, 1$ ***do***
6: ***for*** $(a_{j+1}, \ldots, a_n) \in S_{j+1}$ ***do***
7: $H := \{h(x_j, a_{j+1}, \ldots, a_n) : h \in (G \cap R_{j-1}) \setminus R_j\}$
8: $p := \gcd(H)$
9: $S_j := S_j \cup \{(a, a_{j+1}, \ldots, a_n) : p(a) = 0\}$
10: ***return*** S_1

Chapter 2

Computation of Groebner Bases

In this chapter we take a look at the root ideas of the algorithms used today for computing Groebner bases.

Up to this point, we do not have yet an algorithmic way to construct Groebner bases. Moreover, we do not even have an algorithmic way to check whether a given finite generating set is a Groebner basis or not, rather than going to the definition, i.e., verifying the equality $\langle \mathrm{LM}(g_1), \ldots, \mathrm{LM}(g_m) \rangle = \langle \mathrm{LM}(I) \rangle$.

We present in this chapter some algorithms that allow us to compute a Groebner Basis from any given basis, along with their theoretical foundations.

2.1 Buchberger's Algorithm

In this section, we present a characterization of Groebner bases that will lead to a Groebner basis test, and moreover, to an algorithm to produce such bases. By proposition 1.4.2, we know that $G = \{g_1, \ldots, g_m\}$ is a Groebner basis of I if the reminder on division of every polynomial in I by G is zero. The following proposition shows that we do not need to check this for every polynomial in I, but only for a finite number of them.

Proposition 2.1.1. *Let $g_1, \ldots, g_m \in R$ be such that for all $i = 2, \ldots, m$ and for each $|-$minimal element γ of*

$$\left[\bigcup_{j=1}^{i-1} (\alpha_j - \alpha_i + \mathbb{N}^n)\right] \bigcap \mathbb{N}^n,$$

the remainder of division of $x^\gamma g_i$ by $(g_1, \ldots, g_m)$ is zero Then $G = \{g_1, \ldots, g_m\}$ is a Groebner basis of $I = \langle g_1, \ldots, g_m \rangle$.

Proof. Using proposition 1.4.2, it suffices to show that the remainder on division by $(g_1, \ldots, g_m)$ is zero for each polynomial in I. Assume that this is not the case: there exists a nonzero polynomial $f \in I$ with a nonzero remainder when divided by $(g_1, \ldots, g_m)$. Given that $f \in I$, this polynomial can be written as a linear combination of polynomials of the form $x^\beta g_i$, and since the remainder of division algorithm is linear on the dividend, there must exist $x^\beta g_i$ such that its remainder is nonzero.

Recall the notation from theorem 1.2.1 with $\alpha_j := \deg g_j$ for each j. If $\beta \in \Delta_i - \alpha_i$, then the decomposition $x^\beta g_i = 0g_1 + \cdots + x^\beta g_i + \cdots 0g_m + 0$ would satisfy the conclusion of this theorem and therefore it would be the output of division algorithm, hence, $x^\beta g_i$ would have a zero remainder, which is absurd. It follows that

$$\beta \in \left[\bigcup_{j=1}^{i-1} (\alpha_j - \alpha_i + \mathbb{N}^n)\right] \bigcap \mathbb{N}^n,$$

we denote this set by B. By proposition 1.3.2, there exist a $\mid$ −minimal element γ of B such that $\gamma \mid \beta$, by the hypothesis we can write

$$x^\gamma g_i = \sum_{i=1}^{m} q_i g_i$$

where the q_i's satisfy the conclusion of division algorithm theorem, which in particular imply that

$$\gamma + \alpha_i = \exp(x^\gamma g_i) = \max\{\exp(q_j) + \alpha_j\},$$

and therefore $\exp(q_j) + \alpha_j \preceq \gamma + \alpha_i$ for all $j = 1, \ldots, m$. Moreover, this inequality is strict whenever $j \geq i$: for if $\exp(q_j) + \alpha_j = \gamma + \alpha_i$ with $j \geq i$, then $\exp(q_j) + \alpha_j \in (\alpha_i + \mathbb{N}^n)$, which contradicts theorem 1.2.1.

Since $x^\beta g_i$ has a nonzero remainder and

$$x^\beta g_i = \sum_{j=1}^{m} x^{\beta-\gamma} q_j g_j,$$

there exists $i_1 \in \{1, \ldots, m\}$ such that $x^{\beta-\gamma} q_{i_1} g_{i_1}$ has a nonzero remainder hand hence, in the expansion of $x^{\beta-\gamma} q_{i_1}$ there is a monomial x^{β_1} with $\beta_1 \preceq \beta - \gamma + \exp(q_{i_1})$ such that $x^{\beta_1} g_{i_1}$ has nonzero remainder. We have that

$$\beta_1 + \alpha_{i_1} \preceq \beta - \gamma + \exp(f_{i_1}) + \alpha_{i_1} \preceq \beta + \alpha_i,$$

where the latter inequality is strict whenever $i_1 \geq i$.

Iterating this procedure, we obtain a sequence $(\beta_j, \alpha_j) \in \mathbb{N}^n \times \{1, \ldots, m\}$ such that

$$\beta_{j+1} + \alpha_{i_{j+1}} \preceq \beta_j + \alpha_{i_j}$$

with strict inequality if $i_j \leq i_{j+1}$. Since $i_j \in \{1, \ldots, m\}$, there exists a constant subsequence of $\{i_j\}$, and without loss of generality we can assume that such sequence is the sequence itself. In this case, the α's cancel out in the previous inequality and the inequality becomes strict, thus

$$\beta_{j+1} \prec \beta_j$$

for all $j \in \mathbb{N}$, which is absurd since $\prec$ is a well order. □

A $\mid$-minimal element of

$$\left[\bigcup_{j=1}^{i-1} (\alpha_j - \alpha_i + \mathbb{N}^n)\right] \bigcap \mathbb{N}^n,$$

is a $|$-minimal element of the finite set $\{\alpha_{ji} : j = 1, \ldots, i-1\}$ where $\alpha_{ji} := \alpha_j - \alpha_i$ but making the substitution of the negative entries by zeros, that is,

$$x^{\alpha_{ji}} = \frac{\operatorname{lcm}(x^{\alpha_i}, x^{\alpha_j})}{x^{\alpha_i}} = \frac{x^{\alpha_j}}{\gcd(x^{\alpha_i}, x^{\alpha_j})}.$$

Definition. Given two polynomials $f, g \in R$, we define the *$S-$polynomial of f and g* as

$$S(f, g) := \frac{x^\gamma}{\operatorname{LT}(f)} \cdot f - \frac{x^\gamma}{\operatorname{LT}(g)} \cdot g,$$

where $\gamma = \max\{\exp(f), \exp(g)\}$ so that $x^\gamma = \operatorname{lcm}(\operatorname{LM}(f), \operatorname{LM}(g))$.

Continuing with our previous discussion, in the first step of the computation of the remainder of $x^{\alpha_{ji}} g_i$, there appear (up to a constant factor) the polynomials $S(g_i, g_j)$, so it is sufficient for their remainders to be zero to obtain a Groebner basis, according to the previous proposition. Moreover, since these $S-$polynomials lie in the ideal generated by the g_i's, this becomes a necessary condition. This important property is stated in the following proposition.

Proposition 2.1.2. (Buchberger's criterion). *Let I be a polynomial ideal and $G = \{g_1, \ldots, g_m\}$ a basis for I. Then I is a Groebner basis if and only if for each $1 \leq i, j \leq m$, the remainder on division of $S(g_i, g_j)$ by G is zero.*

Buchberger's criterion gives an efficient algorithmic way to test whether a given finite generating set is a Groebner basis or not. We can easily use this criterion to produce Groebner bases from finite generating sets by using algorithm 4, which is known as the *Buchberger's algorithm*. Basically, the algorithm begins with a finite basis for the ideal,

Algorithm 4: Buchberger's algorithm

Input: $f := (f_1, \ldots, f_s) \in R^s$
Output: A Groebner basis G for $\langle f_1, \ldots, f_s \rangle$ with $\{f_1, \ldots, f_s\} \subseteq G$

```
1: G := F
2: repeat
3:     for each pair {p, q}, p ≠ q in G' do
4:         S := remainder on division of S(p, q) by G'
5:         if S ≠ 0 then
6:             G := G ∪ {S}
7: until G = G'
8: return G
```

and then adds every $S-$polynomial that did not go to zero on division by the actual basis. If the algorithm terminates, then every $S-$polynomial leaves zero remainder when divided by the actual basis, so Buchberger's criterion assures that this basis is actually a Groebner basis.

The following proposition shows that this algorithm always terminates, and it is basically a consequence of the fact that R is a Noetherian ring, so it satisfies the ascending chain condition.

Proposition 2.1.3. *Algorithm 4 terminates in a finite number of steps.*

Proof. At every loop of the algorithm, we have a basis G and we enlarge it with an element $S(f,g)$ for some $f,g \in G$ with the property that the remainder r on division of $S(f,g)$ by G is nonzero. Set $G' := G \cup \{r\}$, we show that $\langle \mathrm{LM}(G)\rangle \subsetneq \langle \mathrm{LM}(G')\rangle$. Indeed, if these monomial ideals were equal, they would have the same monomials and therefore $\mathrm{LM}(r) \in \langle \mathrm{LM}(G)\rangle$. In this case, we would have that $\mathrm{LM}(r)$ is divisible by some element of $\mathrm{LM}(G)$, which contradicts the fact that r is a remainder upon division by G.

The latter shows that every time the algorithm executes a loop, we enlarge an ideal of R. Since R is a Noetherian ring, this enlargement eventually stops and as we have shown, this happens precisely when no element is added to the actual basis, i.e., when the algorithm stops. □

2.2 Lazard's Algorithm

In order to present this algorithm, we need at first some definitions. These will be useful later in chapter 3, when we analyze the complexity of this algorithm.

R is naturally a **graded algebra**, that is, it can be written as a direct sum of $\mathbb{F}-$vector spaces

$$R = \bigoplus_{d\geq 0} R_d$$

such that $R_s R_t \subseteq R_{s+t}$. The natural gradation is that given by

$$R_d := \{f \in R : \text{ for all } x^\alpha \in \mathrm{supp}(f),\ \deg(x^\alpha) = d\} \cup \{0\},$$

but there are many other gradations. However, this is the only gradation we will use in this work.

Definition. Let I be an ideal of R. We define $I_d := R_d \cap I$, and we say that I is an *homogeneous ideal* if

$$I = \bigoplus_{d\geq 0} I_d.$$

We also denote by $I_{\leq d}$ the vector space of polynomials in I of degree at most d. Notice that if I is homogeneous, then

$$I_{\leq d} = \bigoplus_{\ell=0}^{d} I_\ell.$$

If $f \in R_d$, we say that f is a **homogeneous polynomial of degree** d. Using Hilbert Basis Theorem (theorem 1.3.6), it can be easily proven that an ideal is homogeneous if and only if it is generated by finitely many homogeneous polynomials.

Notice that I_d, being the intersection of two vector spaces, is a vector space by itself. Moreover, since the dimension of R_d equals

$$\binom{n+d-1}{d},$$

(which is the number of monomials of total degree equal to d), we conclude that I_d is finite dimensional.

2.2.1 Groebner Bases and Linear Algebra

The idea of using linear algebra to compute Groebner bases dates back to [GM86] and [Laz83], where the key observation is that an ideal I of R is a $\mathbb{F}-$vector space, as well as its components I_d (in the homogeneous case) and $I_{\leq d}$ (in the affine case).

Definition. Let V be a $\mathbb{F}-$linear subspace of R. A subset B of $V \setminus \{0\}$ is called a *staggered linear basis* of V, if it generates V and B is staggered, i.e., for all $f, g \in B$, $\mathrm{LM}(f) = \mathrm{LM}(g)$ implies $f = g$.

Remark. Notice that a staggered linear basis of V is in particular a Hamel basis of V.

Getting a hold of a staggered linear basis of an ideal of R it leads to a Groebner basis of it, as the following theorem shows.

Proposition 2.2.1. *Let B be a staggered linear basis for an ideal I of R. Then, the set*

$$\mathcal{G} := \{f \in B : \textit{ for all } f \neq g \in B, \mathrm{LM}(g) \textit{ does not divide } \mathrm{LM}(f)\}$$

is a minimal Groebner basis for I. Conversely, if $G := \{g_1, \ldots, g_m\}$ is an (ordered) Groebner basis for I, then the set

$$\mathcal{B} := \{x^\alpha g_i : x^\alpha \in M \textit{ and } j < i \textit{ implies that } \mathrm{LM}(x^\alpha g_i) \textit{ is not a multiple of } \mathrm{LM}(g_j)\}$$

is a staggered linear basis for I.

Proof. ($\Rightarrow$). At first, we must prove that $\mathcal{G}$ is a nonempty finite set. Let $f \in B$ and suppose that $f \notin \mathcal{G}$, that is, there exists $g \in B$ such that $\mathrm{LM}(g)$ divides $\mathrm{LM}(f)$ and $f \neq g$ (in particular, $\mathrm{LM}(g) \leq \mathrm{LM}(f)$). Take $\mathrm{LM}(f)$ to be the smallest leading monomial of a polynomial in B with this property. If $\mathrm{LM}(g) = \mathrm{LM}(f)$, then $f = g$ since B is a staggered linear basis, which is absurd, thus $\mathrm{LM}(g) < \mathrm{LM}(f)$. By minimality, $g \in \mathcal{G}$ so this set is nonempty. On the other hand, $\mathcal{G}$ is finite since for any $f, g \in \mathcal{G}$, $f \neq g$

$$\langle \mathrm{LM}(f) \rangle \subsetneq \langle \mathrm{LM}(f), \mathrm{LM}(g) \rangle$$

and R is a Noetherian ring.

Now, we prove that $\mathcal{G}$ is a Groebner basis for I. Let $f \in I$, since B is a linear basis, f can be written as

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

where $c_i \in \mathbb{F}$ and $f_i \in B$ (assume that $i \neq j$ implies $f_i \neq f_j$). Let j be such that $\mathrm{LM}(f_j) > \mathrm{LM}(f_i)$ for all $i \neq j$, this j exists since B is staggered, then $\mathrm{LM}(f) = \mathrm{LM}(f_j)$. If $f_j \notin \mathcal{G}$, there must exist $h \in B$ such that $h \neq f_j$ and $\mathrm{LM}(h)$ divides $\mathrm{LM}(f_j)$, then $\mathrm{LM}(h) < \mathrm{LM}(f_j)$. By iterating this argument, we find eventually ($<$ is a well-ordering) $g \in B$ such that $g \in \mathcal{G}$ and $\mathrm{LM}(g)$ divides $\mathrm{LM}(f_j) = \mathrm{LM}(f)$, hence, $\mathcal{G}$ is a Groebner basis for I. Finally, it can be easily seen that G is minimal.

($\Leftarrow$). We begin by proving that $\mathcal{B}$ is staggered. If $\mathrm{LM}(x^\alpha g_i) = \mathrm{LM}(x^\beta g_j)$, both $i < j$ and $j < i$ yield a contradiction: $\mathrm{LM}(g_i)$ divides $\mathrm{LM}(x^\beta g_j)$ or $\mathrm{LM}(g_j)$ divides $\mathrm{LM}(x^\alpha g_i)$, hence $i = j$.

We now prove that $\mathcal{B}$ generates I. Suppose that $f \in I$ is such that f is not a linear combination of polynomials in $\mathcal{B}$, and assume f is one of the smallest polynomial with this property. Since $\mathcal{G}$ is a Groebner basis, there exists $g_i \in G$ such that $\mathrm{LM}(g_i)$ divides $\mathrm{LM}(f)$, say $x^\gamma \mathrm{LM}(g_i) = \mathrm{LM}(f)$; assume that i is the minimal index with this property, we claim that $x^\gamma \mathrm{LM}(g_i)$ lies in $\mathcal{B}$. Indeed, if this were not the case, there would exist $j < i$ and $x^\beta \in \mathcal{M}$ such that $x^\beta \mathrm{LM}(g_j) = \mathrm{LM}(x^\gamma g_i)$ and then $\mathrm{LM}(g_j)$ would divide $\mathrm{LM}(f)$, which contradicts the minimality of i. Since

$$\mathrm{LM}\left(f - \mathrm{LT}(f)\right) = \mathrm{LM}\left(f - \mathrm{LC}(f)x^\gamma \mathrm{LM}(g_i)\right) < \mathrm{LM}(f),$$

by minimality we have that $f - \mathrm{LC}(f)x^\gamma \mathrm{LM}(g_i)$ is a linear combination of elements in $\mathcal{B}$, which implies that f is too, a contradiction. □

Suppose that given a finite set of generators $\{f_1, \ldots, f_m\}$ of an ideal I we can get a staggered linear basis. Then, if this basis is written in terms of some parameters, we could use the previous proposition to get a Groebner basis of I. We refer the reader to [GM86] to more details on this approach.

Getting a staggered linear basis of an ideal I of R is not an easy task in general, as to begin with, I is an infinite dimensional $\mathbb{F}-$vector space. Moreover, even if we have a (infinite!) staggered linear basis of I, there is not a systematic way to obtain the finite set $\mathcal{G}$ from proposition 2.2.1. To bypass this issue, we replace I by the finite dimensional vector space $I_{\leq d}$. Finding a staggered linear basis for this vector space is an easy task, that can be accomplished by using the so-called Macaulay matrix in the homogeneous case and other similar techniques in the affine (not necessarily homogeneous) case.

Finding staggered linear bases for the vector spaces $I_{\leq d}$ is important since these yield to truncated Groebner bases of I, which are defined as follows.

Definition. Let I be a homogeneous ideal and $G = \{g_1, \ldots, g_m\} \subseteq I$ a subset of homogeneous polynomials. G is said to be a *$d-$Groebner basis* of I if

$$\mathrm{LM}(I) \cap R_{\leq d} \subseteq \langle \mathrm{LM}(g_1), \ldots, \mathrm{LM}(g_m) \rangle.$$

Note that every Groebner basis is a $d-$Groebner basis for all d, and that a $D-$Groebner basis is a $d-$Groebner basis for all $d \geq D$. Proposition 2.2.1 can be adapted as follows.

Proposition 2.2.2. *Let $d \geq 0$ and let B be a staggered linear basis for the $\mathbb{F}-$subspace $I_{\leq d}$, with I an ideal of R. Then, the set*

$$\mathcal{G} := \{f \in B : \textit{ for all } f \neq g \in B, \mathrm{LM}(g) \textit{ does not divide } \mathrm{LM}(f)\}$$

is a minimal $d-$Groebner basis for I. Conversely, if $G := \{g_1, \ldots, g_m\}$ is an (ordered) $d-$Groebner basis for I, then the set

$$\mathcal{B} := \{x^\alpha g_i : x^\alpha \in M \textit{ and } j < i \textit{ implies that } \mathrm{LM}(x^\alpha g_i) \textit{ is not a multiple of } \mathrm{LM}(g_j),\ \deg(x^\alpha g_i) = d\}$$

is a staggered linear basis for $I_{\leq d}$.

Due to this proposition we can always obtain, in a very efficient way, a $d-$Groebner basis from a staggered linear basis of $I_{\leq d}$, which we already know how to find.

To finish the exposition, we show why finding a $d-$Groebner basis for I is useful for obtaining a Groebner basis of I. This is basically due to the fact that for large enough d, a $d-$Groebner basis is actually a Groebner basis, as the following proposition shows.

Proposition 2.2.3. *Let I be an ideal of R. There exists an integer d_0 such that for all $d \geq d_0$, every $d-$Groebner basis of I is a Groebner basis of I*

Proof. Since R is Noetherian, there exists an integer d_0 such that the increasing sequence of ideals

$$\langle \mathrm{LM}(I) \cap R_0 \rangle \subseteq \langle \mathrm{LM}(I) \cap R_{\leq 1} \rangle \subseteq \cdots \subseteq \langle \mathrm{LM}(I) \cap R_{\leq d} \rangle \subseteq \cdots$$

stabilizes, hence

$$\langle \mathrm{LM}(I) \rangle = \langle \mathrm{LM}(I) \cap (\cup_{\ell=0}^{\infty} R_\ell) \rangle = \langle \mathrm{LM}(I) \cap (\cup_{\ell=0}^{d_0} R_\ell) \rangle = \langle \mathrm{LM}(I) \cap R_{\leq d_0} \rangle.$$

Let $d \geq d_0$ and $G = \{g_1, \ldots, g_m\}$ a $d-$Groebner basis of I, we have

$$\langle \mathrm{LM}(I) \rangle = \langle \mathrm{LM}(I) \cap R_{\leq d_0} \rangle \subseteq \langle \mathrm{LM}(I) \cap R_{\leq d} \rangle \subseteq \langle \mathrm{LM}(g_1), \ldots, \mathrm{LM}(g_m) \rangle$$

and hence, G is a Groebner basis of I. □

2.2.2 Homogeneous Lazard's Algorithm

We already saw how to obtain a $d-$Groebner basis of I assuming we had a linear basis of $I_{\leq d}$. Now we show how to find the latter. When all polynomials/ideals involved are homogeneous things are a lot easier to describe, so we will assume that this is the case here. In section 3.4 we discuss why is this assumption reasonable.

Definition. Given a finite subset $F = \{f_1, \ldots, f_m\} \subseteq R \setminus \{0\}$ of homogeneous polynomials and an integer $d \geq 0$, the *Maucalay matrix in degree d* associated to F, denoted by $\mathfrak{M}_d(F)$, is the matrix whose columns represent all the monomials of degree d and the rows represent each polynomial of the form $x^\alpha f_i$ with $|\alpha| + \deg(f_i) = d$, with the entry (f, x^α) being equal to $\mathrm{Coef}(f, \alpha)$, the coefficient of x^α in f. Conversely, given a matrix $M = [M_{i,\alpha}]_{i \in J, \alpha \in \mathcal{J}}$ whose columns are indexed by monomials, we define the *polynomial representation* of M as

$$\mathfrak{P}(M) := \left\{ \sum_{\alpha \in \mathcal{J}} (M_{i,\alpha}) x^\alpha : i \in J \right\} \setminus \{0\} \subseteq R.$$

In this way, to obtain a staggered linear basis for I_ℓ where $I = \langle f_1, \ldots, f_m \rangle$ and each f_i is a nonzero homogeneous polynomial, we perform Gaussian elimination on the matrix $\mathfrak{M}_\ell(\mathcal{F})$ with $F = \{f_1, \ldots, f_m\}$ to obtain the matrix M. It can be easily checked that $B = \mathfrak{P}(M)$ is a staggered linear basis of I_ℓ (the rows already generated this vector space, the reduced form ensures that this linear generating set becomes staggered). A

staggered linear basis of $I_{\leq d}$ will be simply the union of the previous staggered linear bases for $\ell = 0, \ldots, d$, which follows from the observation that

$$I_{\leq d} = \bigoplus_{0 \leq \ell \leq d} I_\ell$$

since I is a homogeneous ideal.

Using all the discussion above along with proposition 2.2.1, a very natural algorithm for computing $d-$Groebner bases comes to mind. This procedure is written formally as algorithm 5.

Algorithm 5: Homogeneous Lazard's algorithm

Input: $F = (f_1, \ldots, f_m) \in (R \setminus \{0\})^m$, a nonnegative integer d.
Output: G a $d-$Groebner basis of $I = \langle f_1, \ldots, f_m \rangle$.

1: $G := \emptyset$.
2: **for** $\ell = 0, \ldots, d$ **do**
3: $\quad M_\ell := \mathfrak{M}_\ell(F)$.
4: $\quad F_\ell := \mathfrak{P}\left(\mathrm{RowEchelonForm}(M_\ell)\right)$
5: $\quad G := G \cup \{f \in F_\ell : \text{ for all } g \text{ in } G, \mathrm{LM}(g) \text{ does not divide properly } \mathrm{LM}(f)\}$.
6: **return** G.

A brief words on the Macaulay matrix

The Macaulay matrix was introduced by Francis Macaulay in [Mac94] as a generalization of the Silvester matrix used in the computation of the resultant of two univariate polynomials. He used these matrices to define the resultant of n "generic" homogeneous polynomials $F_1, \ldots, F_n$ of degree d_i respectively in n variables. Just like the classical resultant, generic means that each polynomial has the form

$$F = \sum_{i_1 + \cdots + i_n = d} U_{i_1 \ldots i_n} x_1^{i_1} \cdots x_n^{i_n} \in \mathbb{F}\left[\{U_{i_1 \ldots i_n}\}_{i_1 + \cdots + i_n = d}\right][x_1, \ldots, x_n],$$

where each U_i is an indeterminate. Then, he constructs the Macaulay matrix of these polynomials in degree $d = \sum(d_i - 1) + 1$ and define the **resultant** to be the greatest common factor of the determinants of this matrix (when the resultant of certain specific polynomials is needed, it is just a matter of evaluating the indeterminates U_i at the coefficients of the polynomials).

The importance of the resultant is due to the following proposition.

Proposition 2.2.4. *[Mac02] The resultant is a homogeneous, irreducible polynomial with degree $d_1 \cdots d_{i-1} \cdot d_{i+1} \cdots d_n$ in the coefficients of the F_i. Also, a particular system $(F_1 = 0, \ldots, F_n = 0)$ has a nontrivial solution if and only if the (specialized) resultant equals zero.*

2.2.3 Affine (or General) Lazard's Algorithm

There are many reasons why, at least theoretically, we may only focus our attention to homogeneous polynomials/ideals (see section 3.4). Anyway, for completeness, we analyze the general form of Lazard's algorithm which includes the case in which the ideal may not be homogeneous.

The main drawback when considering nonhomogeneous ideals is that a linear basis of $I_{\leq d}$ is not easy to find. In one hand, since the ideal does not satisfy

$$I_{\leq d} = \bigoplus_{0 \leq \ell \leq d} I_\ell,$$

we can not rely on the I_ℓ's to obtain a linear basis of $I_{\leq d}$. In this case, we must find a linear basis of $I_{\leq d}$ directly, without using the smaller pieces I_ℓ. On the other hand, adapting Lazard's algorithm so that it yields a linear basis of $I_{\leq d}$ is not a simple nor efficient task.

The first modification we need to consider is that the columns of the Macaulay matrices must index all monomials of degree at most d, since a nonhomogeneous polynomial f has monomials of degree strictly smaller than $\deg(f)$. Like in homogeneous Lazard's algorithm, we iteratively construct and reduce Macaulay matrices with the rows representing polynomials of the form $x^\alpha f_i$ with $|\alpha| + \deg(f_i) = \ell$ for $\ell = 0, \ldots, d$, storing the polynomials obtained on each step. At this point, one may conjecture that the polynomials collected generate $I_{\leq d}$ as a vector space. This is false however since in our case there can be elements $\sum f_i x^{\alpha_i}$ in $I_{\leq d}$ for which some summands $f_i x^{\alpha_i}$ have degree strictly greater than d, so these will be obtained as linear combinations of the polynomials collected in a strictly larger degree. Fortunately, what is certainly true is that there exists $d' \geq d$ such that the polynomials collected in step d' generate a vector space $I_{\leq d'} \supseteq V \supseteq I_{\leq d}$. This immediately implies that this set is a $d-$Groebner basis

We now conclude that if we want a $d-$Groebner basis of $\langle f_1, \ldots, f_m \rangle$, we can run this algorithm up to certain step $d' \geq d$. Fortunately, experimental evidence shows that usually $d' - d$ is not so big, so we do not need to extend so much the computation with respect to the homogeneous case.

Many variants of this idea can be used. For example, one may not use d as a step indicator, we can form instead the matrix with polynomials of the form $x_j f_i$ and the f_i's themselves, reduce it, and then augment this matrix with polynomials of the form $x_j f$ and f, where the f's are the actual rows of the matrix. We then iterate this procedure and, after certain number of steps (when the rows span $I_{\leq d}$), we end up with a staggered linear basis of $I_{\leq d}$. This is the idea behind the XL algorithm [YC05], depicted in figure 2.1. Mutant MXL is another of these variants [Cab11].

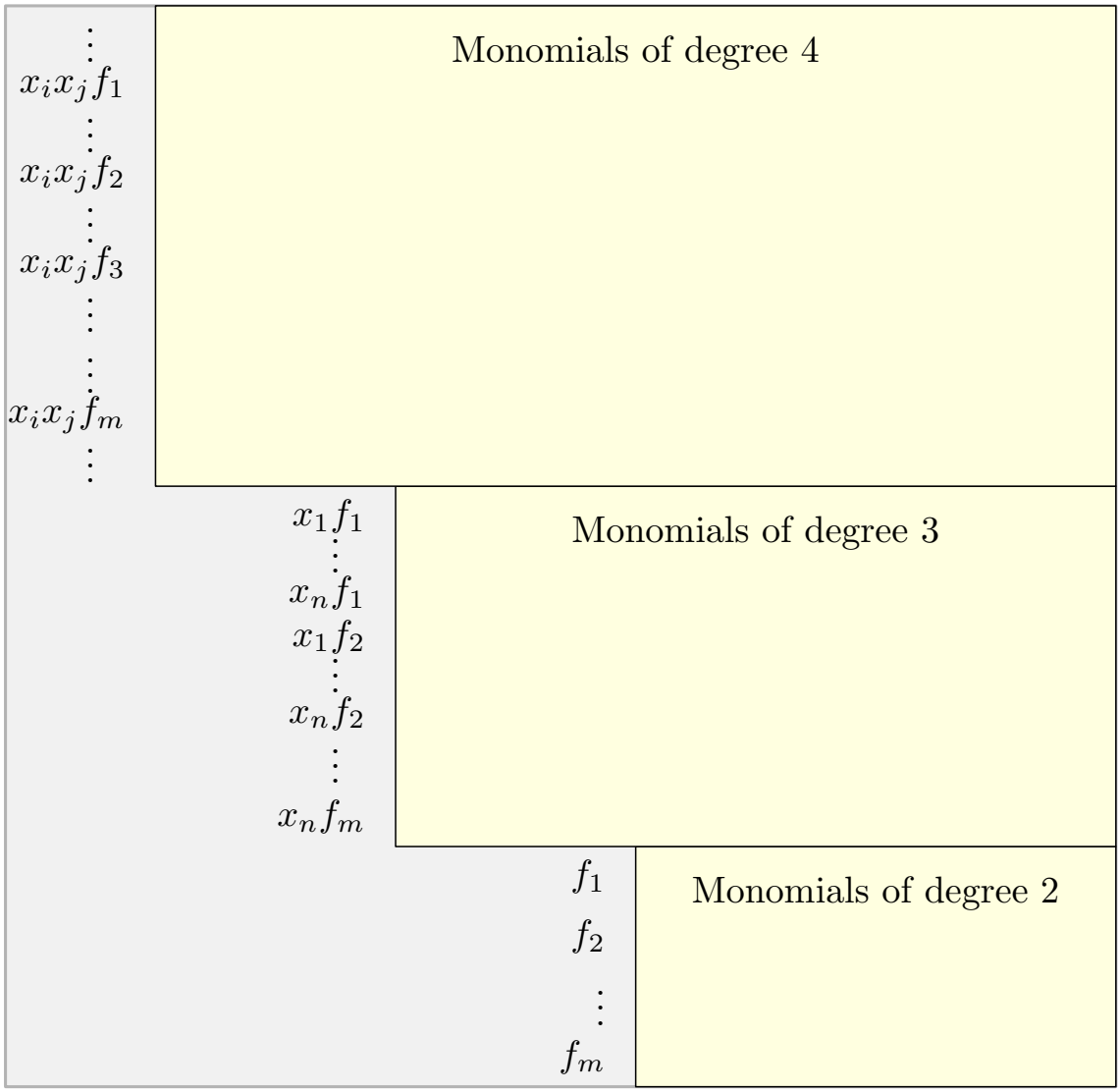

Figure 2.1: Matrix involved in affine Lazard's algorithm for $f_1, \ldots, f_m$ quadratic polynomials

2.2.4 Termination Criteria

We already know how to construct $d-$Groebner bases, and we know that these are Groebner bases for certain enough big d, but how do we know what d is this?

The first observation is the following. Although minimal Groebner bases are not unique, we saw in section 1.4 that their leading monomials are uniquely determined by the ideal so in particular the maximum degree among the leading monomials is the same on every minimal Groebner basis. If the monomial order refines the degree, then this is equivalent to saying that the maximum degree that appears in a minimal Groebner basis is a property inherent to the ideal. The second observation is that running homogeneous Lazard's algorithm up to step D always returns a set of polynomials whose biggest degree is exactly D. Now, since homogeneous Lazard's algorithm always ensures that the returned sets are minimal with respect to divisibility between leading monomials, we can be sure that whenever we end up with a Groebner basis, this will be minimal. The conclusion of this discussion is that if $\mathrm{MaxDeg}(I)$ is the maximum degree of the leading monomials in a reduced Groebner basis, running homogeneous Lazard's algorithm up to this degree will yield a minimal Groebner basis and moreover, this is the smallest step we can consider in order to obtain such.

Now, it may be tempting to eliminate the parameter d from homogeneous Lazard's algorithm and simply run it up to step $\mathrm{MaxDeg}(I)$. This would yield a very optimized version of the algorithm, where no more computation than needed is performed. Unfortunately, this is not possible since there is no efficient way of obtaining this number without actually computing the Groebner basis of I, so we have to look at another ways of deciding when to stop the algorithm.

In order to accomplish this, let's restrict ourselves to homogeneous zero-dimensional ideals for a moment. In this case, as we will show later, there exists certain D such that $I_d = R_d$ holds for all $d \geq D$ and therefore every $D-$Groebner basis $\{g_1, \ldots, g_m\}$ is a Groebner basis (this can be easily seen since $I_d = R_d$ implies that every monomial of degree d is in I, so $I_\ell = R_\ell$ for all $\ell \geq d$ and hence

$$\mathrm{LM}(I) \subseteq \langle \mathrm{LM}(I) \cap R_{\leq d} \rangle \subseteq \langle \mathrm{LM}(g_1), \ldots, \mathrm{LM}(g_m) \rangle).$$

Moreover, intuitively, getting a Groebner basis before this condition is met is an unusual situation inasmuch as $I_d \subsetneq R_d$ implies the existence of monomials in $\mathrm{LM}(I)$ of degree strictly greater than d that may not be divisible by any monomial in $\mathrm{LM}(I)$ of degree at most d, hence, the smallest d at which this condition is achieved is a plausible termination condition.

Definition. Given a zero-dimensional homogeneous ideal I, we define the *degree of regularity of* I as

$$\mathrm{d_{reg}}(I) := \min\{d \geq 0 : I_d = R_d\}.$$

In general, it is not true that $\mathrm{d_{reg}}(I)$ equals $\mathrm{MaxDeg}(I)$ (not even for regular sequences, see section 3.3), but these numbers are very similar. Even though we can not prove this assertion, we present experimental evidence of this fact in section 6.4.1 on the appendix, but this follows the intuitive idea presented before.

One may wonder why are we restricted solely to zero-dimensional ideals. In one hand, this condition is necessary since there exists a d such that $I_d = R_d$ if and only if the ideal is zero-dimensional. We will give a proof for this fact later, but an example can illustrate this in the mean time: if $n > 1$, then $I = \langle x_1 \rangle$ is not zero-dimensional and does not satisfy $I_d = R_d$ for any d, since $x_2^d \in R_d \setminus I_d$ for all d. On the other hand, for many cryptographic applications these are the only ideals of interest, so it makes sense to work in the zero-dimensional context.

The homogeneous condition could be thrown. However, many complexity results we will see later in this work would not remain valid. To be able to work in the affine context too, we will give later a different definition for affine systems. In the next section, we revisit the concept of degree of regularity in a more algebraic fashion.

2.3 Remarks about Computational Improvements

The algorithms presented in this chapter are very simple, clear and illustrate the basic ideas behind Groebner bases computation. However, these are not used in practice, at least not in the way we have presented them.

Regarding Buchberger's algorithm (which illustrates the basic idea of the computation of Groebner bases using Buchberger's criterion), there are a lot of improvements that can be done in order to gain efficiency. For instance, there are many pairs f, g for which we can *predict* that the remainder of $S(f, g)$ will be zero without actually doing the computation. See [CLO07, Ch. 2, §9] for more details about such improvements.

Turning our attention to Lazard's algorithm, one can observe that the matrices built have a huge rank defect, so a lot of rows will reduce to zero when we apply Gaussian elimination. The Lazard's based algorithm F4 developed by Faugère [Fau99] is used in practice and takes this into account by making some predictions about the rows that will reduce to zero, while combining the ideas of the $S-$ polynomials in Buchberger's algorithm. Moreover, F5 algorithm [Fau02] takes exactly the independent rows in the general case (regular sequences).

Besides these improvements, there are some computation strategies that are kept in mind.

Choice of the monomial order. Recall that all this theory depends on a monomial order chosen. The timings for Groebner bases algorithms have this dependence too, and the needed monomial order may not be the fastest for the computation to end. For instance, experimental evidence shows that the grevlex order is the most efficient for Groebner bases computation, however, many application does not involve this order, as for example finding the solutions of a zero-dimensional system is easy using a lex Groebner basis (algorithm 3), which is not easy to find in general. Moreover, the maximal degree of the polynomials in the Groebner basis computation with grevlex order is the smallest among all other possible orders [Bar04, §3.4]. In all these cases, the best approach is to find at first a grevlex Groebner basis and then use a change of order algorithm like FGML [FGLM93] in the zero-dimensional case and Groebner walk CKM97 in the general case [CKM97]. This is a very efficient process since a lex Groebner basis for

a zero-dimensional system with D solutions (counting multiplicities) can be computed from any Groebner basis (in particular, a grevlex Groebner basis) in $O(nD^3)$ arithmetic operations using FGML.

Choice of the critical pairs. In Buchberger's algorithm, there is not a particular order in which the $S-$polynomials are picked. Different choices of these may have a huge impact on the computation time. Although there is not a proof-based good strategy, experimental evidence shows that some strategies are better than others. In practice, the normal strategy is used in algorithm F4, which consists of considering at first pairs with smallest degree.

Homogenization. This will be discussed in more detail in section 3.4. As we will see there, many advantages arise when we consider only homogeneous polynomials, as for example, description of the arithmetic complexity becomes possible. Moreover, in 2.2.2 and 2.2.3 we saw that these advantages are not purely theoretical. Nonetheless, many affine polynomial systems arise in practice so we must deal with these at the end. One way to do this is by 'homogenizing' the polynomials involved, that is, obtaining some homogeneous polynomials that preserve certain desired property that depends on the application.

For instance, in polynomial system solving, we may add what is known as a homogenization variable h and consider the polynomial

$$\widetilde{f}(x_1,\dots,x_n,h) = h^{\deg(f)} f\left(\frac{x_1}{h},\dots,\frac{x_n}{h}\right),$$

which is simply f with h^ℓ added in each monomial of strictly lower degree than $\deg(f)$ so that the degree of this monomial equals $\deg(f)$, f can be recovered then by evaluating $h=1$ (which is known as *specialization*). Two main drawbacks appear. In one hand, if one wishes to calculate a Groebner basis for $I=\langle f_1,\dots,f_m\rangle$ using the ideal $\widetilde{I} = \left\langle \widetilde{f_1},\dots,\widetilde{f_m}\right\rangle$, we need to make sure that specializing a Groebner basis of $\widetilde{I}$ at $h=1$ yields a Groebner basis of I; this holds for example if we use grevlex order with $x_1 \prec_{\text{grevlex}} \cdots \prec_{\text{grevlex}} x_n \prec_{\text{grevlex}} h$, and in general any elimination order will do the trick (see section 1.5.3).

On the other hand, if we want to find the solutions to $(f_1=0,\dots,f_m=0)$, we could solve the homogeneous system $(\widetilde{f_1}=0,\dots,\widetilde{f_1}=0)$ and then set $\left(\frac{a_1}{b},\dots,\frac{a_n}{b}\right)\in\mathbb{F}^n$ for each solution $(a_1,\dots,a_n,b)\in\mathbb{F}^{n+1}$ with $b\neq 0$ of this system, these will be the solutions to the original; the disadvantage with this method is the appearance of 'parasite' solutions, that is, those with $h=0$. Another technique that avoid this issue is using a weighted order, but this does not work for all systems.

Finally, as a homogeneous polynomial induced from an affine polynomial we can also consider its highest homogeneous degree part. This can be seen as adding the variable h as before, and specialize it at $h=0$. In this case, the solutions to a polynomial system are not related with this induced system, but Groebner bases do. In particular, as we will see in the next chapter, arithmetic complexity of Groebner bases computation is related, and therefore this induced system will be useful to analyze the computation process in the affine case.

Chapter 3

Complexity estimates

A complexity analysis of the algorithms presented before is included in this chapter. This will allow us to give a numerical conclusion about the efficiency of computing Groebner bases.

The contents of this chapter can be widely described in three parts. In section 3.2 we analyze an intrinsic property of polynomial systems that determines the arithmetic complexity for the computation of a Groebner basis using Lazard's algorithm. This property will be the so-called degree of regularity, and we will see that the mentioned complexity is exponential in this quantity. Then, in section 3.3, we study the behavior of this property in the "general" case (this will be made clear in that section) to have a reference point on how hard it is to compute a Groebner basis for most systems. At the end, we discuss how to measure the complexity for particular systems, where the degree of regularity is not easy to calculate. For these, the approach will be through the falling degree, which we explain in detail. All this work requires some concepts and techniques from Algebraic Geometry, which we introduce in the next section.

3.1 Some words on Algebraic Geometry and Commutative Algebra

(Classical) Algebraic Geometry is concerned with the study of zeros of polynomial systems with coefficients over a field. MPKC is naturally connected to algebraic geometry since it involves polynomial systems, and, as mentioned in the introduction, the hardness of finding zeros is critical for the security of MPKC cryptosystems.

We use the same notation as before: R denotes the polynomial ring $\mathbb{F}[x_1, \ldots, x_n]$. Additionally, we let $\overline{\mathbb{F}}$ and $\overline{R}$ denote the algebraic closure of $\mathbb{F}$ and the polynomial ring $\overline{\mathbb{F}}[x_1, \ldots, x_n]$ respectively.

3.1.1 Zariski topology

Definition. Let B be a subset of R and $\mathbb{K}$ an extension field of $\mathbb{F}$. We define the *algebraic set associated to B in $\mathbb{K}$* as

$$\mathbf{V}_{\mathbb{K}}(B) := \{a \in \mathbb{K}^n : f(a) = 0 \text{ for all } f \in B\}.$$

One easily checks that $\mathbf{V}_{\mathbb{K}}(B) = \mathbf{V}_{\mathbb{K}}(\langle B \rangle)$. In particular, by Hilbert Basis theorem, for each ideal I it holds that $\mathbf{V}_{\mathbb{K}}(I) = \mathbf{V}_{\mathbb{K}}(f_1, \ldots, f_m)$ for some $f_1, \ldots, f_m \in I$.

Computing $\mathbf{V}_{\mathbb{K}}(I)$ where $\mathbb{K}$ is an intermediate field between $\mathbb{F}$ and $\overline{\mathbb{F}}$ is not an easy task in general. For instance, finding the real solutions of a system with rational coefficients is not easy in general. However, when $\mathbb{F}$ is a finite field of q elements and we wish our solutions to lie in $\mathbb{F}$, we only need to add the so-called **field equations** to the system: $x_1^q = x_1, \ldots, x_n^q = x_n$. This holds since $\mathbf{V}_{\mathbb{K}}(x_1^q - x_1, \ldots, x_n^q - x_n) = \mathbb{F}^n$ for any field extension $\mathbb{K}$ of $\mathbb{F}$, so

$$\mathbf{V}_{\mathbb{K}}(f_1, \ldots, f_m, x_1^q - x_1, \ldots, x_n^q - x_n) = \mathbb{F}^n \cap \mathbf{V}_{\mathbb{K}}(f_1, \ldots, f_m) = \mathbf{V}_{\mathbb{F}}(f_1, \ldots, f_m).$$

We state this property (without proof) and some interesting facts about this ideal in the next proposition.

Proposition 3.1.1. *Let $\mathbb{F}$ be a finite field of size q and let $I = \langle f_1, \ldots, f_m \rangle$ be an ideal of $\mathbb{F}[x_1, \ldots, x_n]$. Let $J = I + \langle x_1^q - x_1, \ldots, x_n^q - x_n \rangle$, then*

$$\mathbf{V}_{\mathbb{F}}(I) = \mathbf{V}_{\overline{\mathbb{F}}}(I, x_1^q - x_1, \ldots, x_n^q - x_n).$$

Moreover, the following holds:

1. *The ideal J is radical;*
2. *$G = \{1\}$ if and only if the system $(f_1 = 0, \ldots, f_m = 0)$ has no solutions in $\mathbb{F}$;*
3. *$G = \{x_1 - a_1, \ldots, x_n - a_n\}$ if and only if the system $(f_1 = 0, \ldots, f_m = 0)$ has a unique solution $(a_1, \ldots, a_n)$ in $\mathbb{F}$.*

We will be mainly interested in the algebraic sets in $\overline{\mathbb{F}}$, so from now on during this section (unless otherwise is stated) we only consider these. The following is an easy-to-prove proposition.

Proposition 3.1.2. *Let I, J be ideals of $\overline{R}$. Then the following holds*

1. *$\mathbf{V}_{\overline{\mathbb{F}}}(\{0\}) = \overline{\mathbb{F}}$;*
2. *$\mathbf{V}_{\overline{\mathbb{F}}}(R) = \emptyset$;*
3. *$\mathbf{V}_{\overline{\mathbb{F}}}(IJ) = \mathbf{V}_{\overline{\mathbb{F}}}(I) \cup \mathbf{V}_{\overline{\mathbb{F}}}(J)$;*
4. *If $\{I_\alpha\}_\alpha$ is a family of ideals of $\overline{R}$, then $\mathbf{V}_{\overline{\mathbb{F}}}\left(\bigcup_\alpha I_\alpha\right) = \bigcap_\alpha \mathbf{V}_{\overline{\mathbb{F}}}(I_\alpha)$.*

Due to this proposition, we can define a topology in $\overline{\mathbb{F}}^n$ where the closed sets are the algebraic sets, that is, a set $X \subset \overline{\mathbb{F}}^n$ is closed in this topology if and only if $X = \mathbf{V}_{\overline{\mathbb{F}}}(B)$ for some $B \subseteq \overline{R}$. This toplogy is known as the **Zariski topology** over $\overline{\mathbb{F}}^n$.

Zariski topology has very interesting properties, in particular, we have the following.

Proposition 3.1.3. *Every nonempty open subset of $\overline{\mathbb{F}}^n$ is dense.*

This will allow us to define the concept of a generic property later on.

Definition. Let $\mathbb{K}$ be an extension field of $\mathbb{F}$ and $C \subseteq \mathbb{K}^n$, we define the *ideal associated to* C *in* $\mathbb{K}[x_1, \ldots, x_n]$ as

$$\mathbf{I}_{\mathbb{K}}(C) := \{f \in \mathbb{K}[x_1, \ldots, x_n] : f(c) = 0 \text{ for all } c \in C\}.$$

The operators $\mathbf{I}_{\overline{\mathbb{F}}}$ and $\mathbf{V}_{\overline{\mathbb{F}}}$ seem to be inverse of each other. However, if $I \subseteq R$ is a polynomial ideal of R, one may have polynomials f such that $f^r \in I$ but $f \notin I$, and in such case we would have that $f \in \mathbf{I}_{\overline{\mathbb{F}}}\left(\mathbf{V}_{\overline{\mathbb{F}}}(I)\right)$, so $\mathbf{I}_{\overline{\mathbb{F}}}\left(\mathbf{V}_{\overline{\mathbb{F}}}(I)\right)$ is not necessarily equal to I. However, the only way for a polynomial to be in $\mathbf{I}_{\overline{\mathbb{F}}}\left(\mathbf{V}_{\overline{\mathbb{F}}}(I)\right)$ is that some power of it is in I. In order to state more precisely this result, we need the following definition.

Definition. Let I be an ideal of a ring A. We define the *radical ideal* of I as the ideal

$$\sqrt{I} := \{a \in A : a^r \in I \text{ for some } r \in \mathbb{N}\}.$$

Theorem 3.1.4. (Hilbert Nullstelensatz). *[CLO07, p. 173] Let I be an ideal of $\overline{R}$, then*

$$\sqrt{I} = \mathbf{I}_{\overline{\mathbb{F}}}\left(\mathbf{V}_{\overline{\mathbb{F}}}(I)\right).$$

3.1.2 Systems with Finitely Many Solutions

We extend the concept of zero-dimensionality to an ideal I, saying that I is **zero-dimensional** if the associated algebraic set $\mathbf{V}_{\overline{\mathbb{F}}}(I)$ is finite.

Definition. Let A be a commutative ring. The *Krull dimension* of A, denoted by $\dim(A)$, is the number of inclusions in the longest strictly increasing chain of prime ideals of A.

Theorem 3.1.5. *Let I be a polynomial ideal of R. The following assertions are equivalent.*

(i) *I is zero-dimensional.*

(ii) *The dimension of R/I as a $\mathbb{F}-$vector space, denoted by $\dim_{\mathbb{F}}(R/I)$, is finite.*

(iii) *The Krull dimension of R/I, $\dim(R/I)$, is zero.*

(iv) *I is contained in only finitely many prime ideals.*

(v) *I is contained in only finitely many maximal ideals.*

(vi) *$I \cap \mathbb{F}[x_i] \neq \{0\}$ for each $i = 1, \ldots, n$.*

(vii) *For every Groebner basis G of I, there exists some element in $\mathrm{LM}(G)$ of the form x_i^ℓ for each $i \in \{1, \ldots, n\}$.*

3.1.3 Hilbert's Function and Hilbert's Series

Recall from section 2.2 that R is a graded $\mathbb{F}-$algebra

$$R = \bigoplus_{d\geq 0} R_d$$

with R_d the $\mathbb{F}-$vector space generated by all the monomials of degree d. Given a homogeneous ideal I, the quotient R/I can be regarded as a graded ring with gradation

$$R/I = \bigoplus_{d\geq 0}(R_d + I)/I.$$

Since we have a natural isomorphism $R_d/I_d \cong (R_d + I)/I$ sending $f + I_d$ to $(f + 0) + I$, we can regard R/I as

$$R/I = \bigoplus_{d\geq 0} R_d/I_d.$$

Definition. Let $A = \oplus_d A_d$ be a graded finite dimensional $\mathbb{F}-$algebra. The *Hilbert's function* of A is defined as $\mathrm{HF}_A(d) = \dim_{\mathbb{F}}(A_d)$. The *Hilbert series* of A is defined as the formal power series

$$\mathrm{HS}_A(t) = \sum_{d\geq 0} \mathrm{HF}_A(d)t^d.$$

In particular, if I is a homogeneous ideal of R, we have

$$\mathrm{HF}_{R/I}(d) = \dim_{\mathbb{F}}\left((R/I)_d\right) = \dim_{\mathbb{F}}(R_d/I_d) = \dim_{\mathbb{F}}(R_d) - \dim_{\mathbb{F}}(I_d)$$

and

$$\mathrm{HS}_{R/I}(t) = \sum_{d\geq 0} \mathrm{HF}_{R/I}(d)t^d.$$

Theorem 3.1.6. *[MS04, Thm 8.20] Given a homogeneous ideal $I \subseteq R$, there exists a polynomial $N(t) \in \mathbb{Z}[t]$ such that*

$$\mathrm{HS}_{R/I}(t) = \frac{N(t)}{(1-t)^n}.$$

From this theorem, we can easily prove the following

Theorem 3.1.7. *Let I be a homogeneous ideal of R. There exists $D > 0$ and a polynomial $p \in \mathbb{Z}[x]$ whose degree is $\dim(R/I) - 1$ such that $\mathrm{HF}_{R/I}(d) = p(d)$ for every $d \geq D$.*

Proof. We know from theorem 3.1.6 that

$$\mathrm{HS}_{R/I}(t) = \frac{N(t)}{(1-t)^\delta}$$

where $\delta \leq n$ and $N(1) \neq 0$, if we apply partial fraction expansion we obtain a polynomial $Q(t) \in \mathbb{Z}[t]$ and scalars $a_1, \ldots, a_\delta \in \mathbb{Z}$ with $a_\delta \neq 0$ such that

$$\mathrm{HS}_{R/I}(t) = Q(t) + \sum_{i=1}^{\delta} \frac{a_i}{(1-t)^i}.$$

Notice that, for each $d \in \mathbb{Z}^+$, the coefficient of t^d in $\frac{a_i}{(1-t)^i}$ is

$$\text{Coef}\left(\frac{a_i}{(1-t)^i}, d\right) = a_i \binom{i+d-1}{i-1}$$

which is a polynomial in d of degree $i-1$. Then, for all $d > \deg(Q)$ we have that

$$\text{HF}_{R/I}(d) := \text{Coef}\left(\text{HS}_{R/I}(t), d\right) = \sum_{i=1}^{\delta} a_i \binom{i+d-1}{i-1},$$

which is a polynomial in d. □

Notice that if two polynomials satisfy the previous theorem then they are necessarily equal. Moreover, it is proved in [CLO07, Ch. 9, §3, Thm. 11] that the degree of the polynomial p equals the projective dimension of I, which is $\dim(I) - 1$. Besides, from the proof of the previous theorem we see that the degree of p equals the degree of $\sum_{i=1}^{\delta} a_i \binom{i+d-1}{i-1}$, which is $\delta - 1$. This implies that

Corollary 3.1.8. *If $N(1) \neq 0$ and $\text{HS}_{R/I}(t) = \frac{N(t)}{(1-t)^\delta}$, then $\dim(R/I) = \delta$. In particular, if $\dim(R/I) = 0$, then $\text{HS}_{R/I}(t)$ is a polynomial and $\text{HS}_{R/I}(1) = \dim_{\mathbb{F}}(R/I)$.*

As a final note, from the proof of theorem 3.1.7 we notice that $\text{HF}_{R/I}(d)$ is equal to the polynomial p if and only if $d > \deg(Q)$. This observation will be important in the next section.

3.2 Degree of Regularity and Complexity of Lazard's Algorithm

As mentioned at the beginning of this chapter, a good indicator for the complexity of computing a Groebner basis for an ideal, is its degree of regularity. This will be defined in this section as an algebraic property of polynomials sequences, and we will see later in the next section its relation with the termination of Lazard's algorithm.

Definition. Let I be a homogeneous ideal of R. The smallest D satisfying theorem 3.1.7 is called the *index of regularity* of I, and we denote it by $\text{i}_{\text{reg}}(I)$. The corresponding polynomial p, denoted by $\text{HP}_{R/I}(d)$, is called the *Hilbert polynomial* of I.

Suppose that I is a homogeneous zero-dimensional ideal, hence $\dim(R/I) = 0$ due to proposition 3.1.5 and therefore, from corollary 3.1.8, $\text{HS}_{R/I}(t)$ is a polynomial. This implies that the Hilbert polynomial of I is the constant polynomial 0 and so the index of regularity of I is $\deg\left(\text{HS}_{R/I}(t)\right) + 1$. Furthermore, since $\text{HF}_{R/I}(d) = \text{HP}(d) = 0$ for all $d \geq \text{i}_{\text{reg}}(I)$, by definition of the Hilbert function we see that $I_d = R_d$ if and only if $d \geq \text{i}_{\text{reg}}(I)$. By the definition of degree of regularity given in section 2.2.4, we see that $\text{i}_{\text{reg}}(I) = \text{d}_{\text{reg}}(I)$. We will use this relation to extend the concept of degree of regularity to affine zero-dimensional ideals.

Given a polynomial $f \in R$, we denote by $f^{(h)}$ the homogeneous component of highest degree in f.

Proposition 3.2.1. *Let $F = (f_1, \ldots, f_m) \in R^m$ where each f_i is a polynomial, not necessarily homogeneous, and let $F^{(h)} = \left(f_1^{(h)}, \ldots, f_m^{(h)}\right)$. If $\left\langle F^{(h)} \right\rangle$ is zero-dimensional, then $\langle F \rangle$ is also zero-dimensional.*

Definition. In the context of the previous proposition, we define the *degree of regularity of F* as

$$\mathrm{d}_{\mathrm{reg}}(F) := \mathrm{i}_{\mathrm{reg}}\left(\left\langle F^{(h)} \right\rangle\right).$$

Remark. It is worth noticing that the degree of regularity defined in this fashion is not an invariant of the ideal $\langle F \rangle$ when the polynomials are not homogeneous. For instance, $\langle x \rangle = \langle x^2 + x, x^2 \rangle$ but $\mathrm{d}_{\mathrm{reg}}(x) = 1$ and $\mathrm{d}_{\mathrm{reg}}(x^2 + x, x^2) = \mathrm{i}_{\mathrm{reg}}(x^2) = 2$.

The relation between the degree of regularity and the arithmetic complexity of Lazard's algorithm when the input is homogeneous has already been made clear. However, in the affine context this relation will only be made evident in section 3.4.2.

For a zero-dimensional homogeneous ideal I, since algorithm 5 obtains a Groebner basis for I when $d = \mathrm{d}_{\mathrm{reg}}(I)$ we can bound the arithmetic complexity of this algorithm in terms of this value.

Theorem 3.2.2. *[Spa12, Thm. 1.72] Let $F = (f_1, \ldots, f_m) \in R^m$ be a family of homogeneous polynomials generating a zero-dimensional ideal. The arithmetic complexity of computing a Groebner basis of this ideal is bounded by*

$$O\left(\sum_{i=0}^{\mathrm{d}_{\mathrm{reg}}(F)} \left[\binom{n+i-1}{i}\left(\sum_{j=1}^{m}\binom{n+i-\deg(f_j)-1}{i-\deg(f_j)}\right)\left(\binom{n+i-1}{i} - \mathrm{HF}_{R/\langle F\rangle}(i)\right)^{\omega-2}\right]\right)$$
$$\leq O\left(m\binom{n+\mathrm{d}_{\mathrm{reg}}(F)}{\mathrm{d}_{\mathrm{reg}}(F)}^{\omega}\right),$$

where ω is the exponent for the complexity of matrix multiplication.

In the sequel we write $\mathrm{d}_{\mathrm{reg}}(f_1, \ldots, f_m)$ in terms of the degrees of the f_i's, at least in the general case (i.e., random polynomials) so that we obtain a more concrete bound on the arithmetic complexity of Lazard's algorithm.

3.3 Regular and Semi-Regular Sequences

We saw that when I is a homogeneous zero-dimensional ideal of R, the complexity for finding a Groebner basis for I using Lazard's algorithm is bounded in terms of $\mathrm{d}_{\mathrm{reg}}(I)$, which equals $\mathrm{i}_{\mathrm{reg}}(I)$. However, computing $\mathrm{i}_{\mathrm{reg}}(I)$ in general can be as hard as computing a Groebner basis of I itself. Fortunately, there are some families of polynomials whose index of regularity is well known, these are the regular and semi-regular sequences.

Definition. A sequence of non-zero homogeneous polynomials $F = (f_1, \ldots, f_m)$ is called *regular* if for all $i \in \{1, \ldots, m-1\}$, f_{i+1} does not divide 0 in the ring $R/\langle f_1, \ldots, f_i \rangle$.

Although it is not clear from the definition, there are no regular sequences when $m > n$. This is due to the following characterization.

Proposition 3.3.1. *Let $F = (f_1, \ldots, f_m)$ be a sequence of non-zero homogeneous polynomials. Then the sequence is regular if and only if the Hilbert series of $I = \langle f_1, \ldots, f_m \rangle$ is*

$$\mathrm{HS}_{R/I}(t) = \frac{\prod_{i=1}^{m} \left(1 - t^{\deg(f_i)}\right)}{(1-t)^n}.$$

Hence, from corollary 3.1.8, we see that every regular sequence of length m has dimension $n - m$, in particular, there are no regular sequences of length greater than n. It is worth mentioning that proposition 3.3.1 actually characterizes regular sequences:

Proposition 3.3.2. *Let $F = (f_1, \ldots, f_m)$ be a sequence of non-zero homogeneous polynomials. Then the sequence is regular if and only if the Krull dimension of $I = \langle f_1, \ldots, f_m \rangle$ is $n - m$.*

The main property we are going to use is that when I is zero-dimensional, the Hilbert series becomes a polynomial and its degree plus 1 will be the index of regularity. We already mentioned this, but the difference now is that for a zero-dimensional regular sequence $f_1, \ldots, f_n$ we have an explicit formula for the Hilbert series:

$$\frac{\prod_{i=1}^{n} \left(1 - t^{\deg(f_i)}\right)}{(1-t)^n}$$

so after taking the n common factors $(1 - t)$ in the numerator, we end up with a polynomial of degree $\sum_{j=1}^{n}(\deg(f_j) - 1)$, so we obtain the following result.

Theorem 3.3.3. *Given a regular sequence $f_1, \ldots, f_n$, the (degree) index of regularity of the ideal I generated by them is*

$$\mathrm{d}_{\mathrm{reg}}(I) = \mathrm{i}_{\mathrm{reg}}(I) = 1 + \sum_{j=1}^{n}(\deg(f_j) - 1),$$

which is known as the Macaulay bound.

Corollary 3.3.4. *For a regular quadratic system of equations $(p_1 = 0, \ldots, p_n = 0)$, the complexity of solving the polynomial system is exponential in n.*

Proof. In this case, the previous theorem shows that the degree of regularity is linear in n, so theorem 3.2.2 implies that the arithmetic complexity of computing a Groebner basis of this ideal is exponential in n. □

We already know the degree of regularity for regular sequences, but not every sequence is regular (for instance, $F = (x_1, x_1^2)$ is not regular). However, *almost every sequence of $m \leq n$ polynomials is regular*. This will be made more precise in the following.

3.3.1 Generic Properties

Let V be a finite dimensional vector space over $\mathbb{F}$, then we can identify V with $\mathbb{F}^D$, where $D = \dim_{\mathbb{F}}(V)$. Given a property that objects in V may or may not have, we say that this property is a **generic property** if it holds (under the previous identification) in a nonempty open set of $\mathbb{F}^D$, using the Zariski topology defined in 3.1.1. Since any nonempty Zariski-open subset is dense, a generic property holds in a dense set and therefore, *intuitively*, it should hold for almost every point in V.

Fix $m \leq n$ and non-negative integers $d_1, \ldots, d_m$. Consider V as the $\mathbb{F}-$vector space of all sequences $F = (f_1, \ldots, f_m)$ with $f_i \in R$ being a homogeneous polynomial of degree d_i. We identify each sequence as a tuple whose entries are the coefficients of the polynomials in the sequence. We aim to prove that being regular is a generic property in this vector space, and this will be actually a corollary of the following theorem.

Theorem 3.3.5. *Let V the $\mathbb{F}-$vector space defined above and let D be its dimension. Then, there exists a nonempty Zariski-open subset of $\mathbb{F}^D$ such that the Hilbert function of $\langle F \rangle$ is constant among all the sequences F in such set (under the identification).*

The proof can be found in [Par10]. As a corollary, using proposition 3.3.1, we see that the set of regular sequences is an open set, thus we only need to check it is nonempty. To accomplish this, it is easy to see that the following is a regular sequence

$$F = (x_1^{d_1}, \ldots, x_m^{d_m}).$$

3.3.2 Semi-Regular Sequences

Using Macaulay bound and theorem 3.2.2, we can bound the arithmetic complexity of Lazard's algorithm for computing a Groebner basis of a zero-dimensional ideal $\langle f_1, \ldots, f_m \rangle$, with $m \leq n$. However, the case $n > m$ still needs to be addressed. This case is important since, in many applications, an overdetermined system (more equations than variables) is needed to be solved; for instance, we can consider $\mathbb{F}$ to be a finite field of size q and work in its algebraic closure $\overline{\mathbb{F}}$ all the time, then if we want to solve a system $(f_1 = 0, \ldots, f_m = 0)$ where each polynomial lies in R we make computations (or apply any theorem involving algebraic closeness) in $\overline{R}$ and then add the so-called *field equations* $x_1^q = x_1, \ldots, x_n^q = x_n$ (which makes the system overdetermined) to force the solutions to lie in $\mathbb{F}^n$, and not in $\overline{\mathbb{F}}^n$.

Semi-regular sequences are a natural extension of regular sequences that allow us to preserve many properties of such in the $m > n$ case. These are introduced in chapter 3 of [Bar04], and we refer the reader to that work for further details. However, in order to give a general idea, we state the definition of semi-regular sequences for the zero-dimensional case below.

Definition. A sequence of non-zero homogeneous polynomials $F = (f_1, \ldots, f_m)$ is called *semi-regular* if $\langle f_1, \ldots, f_m \rangle \neq R$ and for all $i \in \{1, \ldots, m\}$, if $g_i f_i = 0$ in the ring $R/\langle f_1, \ldots, f_i \rangle$ and $\deg(g_i f_i) < \mathrm{i}_{\mathrm{reg}}(I)$, then $g_i = 0$ in $R/\langle f_1, \ldots, f_i \rangle$.

3.4 Homogeneous vs Affine Polynomial Systems

This is a good point to make clear why have we been assuming that the polynomials/ideals are homogeneous. We list at first some advantages of considering homogeneous polynomials/ideals:

- The theory developed in section 3.1.3 (necessary for finding the degree of regularity for regular sequences) is only valid for homogeneous ideals.
- When dealing with homogeneous polynomials $f_1, \ldots, f_m$ of degrees $d_1, \ldots, d_m$, we can be sure that any algebraic combination $\sum f_i g_i$ with each g_i homogeneous of degree $d - d_i$ for some d will have degree exactly d or will be zero. As we saw in section 2.2, this allowed Homogeneous Lazard's algorithm to be clearer and more efficient than its affine version. Also, this allowed us to give a concrete termination criteria for the former.

Additional to these, homogeneous polynomials are frequent in many applications, and they are used very often due to their simplicity. Now we wonder, are we losing any generality when we consider only homogeneous polynomials/ideals? at the end of the day we need to take affine systems into account. For example, they are involved when we append the field equations $x_1^q - x_1 = 0, \ldots, x_n^q - x_n = 0$, which are necessary in order to force the solutions to lie in the finite field $\mathbb{F}_q$ and not in its closure.

In the following we show how to move from affine systems to homogeneous systems, and how to obtain information of one from the other. The conclusion will be that if we want to study certain affine system, we can derive a homogeneous system from it and apply the already studied theory to it. Fortunately, this will give us information about the original system.

3.4.1 Homogenization and Specialization

There is a standard way to get a homogeneous polynomial from an affine one and vice versa.

Definition. Let $f \in R \setminus \{0\}$. The homogeneous polynomial

$$\widetilde{f}(x_1, \ldots, x_n, h) = h^{\deg(f)} f\left(\frac{x_1}{h}, \ldots, \frac{x_n}{h}\right) \in R[h] = \mathbb{F}[x_1, \ldots, x_n, h]$$

is called the *homogenization of* f, and is obtained by multiplying $h^{\deg(f) - \deg(x^\alpha)}$ to every monomial x^α in f. If f is the zero polynomial in R, we define $\widetilde{f}$ to be the zero polynomial in $R[h]$. Given an ideal $I = \langle f_1, \ldots, f_m \rangle$ of R, we define the *homogenization of* I as the ideal of $R[h]$

$$\widetilde{I} = \left\langle \widetilde{f_1}, \ldots, \widetilde{f_m} \right\rangle.$$

Remark. We must note that $\widetilde{I}$ depends on the generators chosen, that is, if $I = \langle f_1, \ldots, f_m \rangle = \langle f'_1, \ldots, f'_m \rangle$ it may be the case that

$$\left\langle \widetilde{f_1}, \ldots, \widetilde{f_m} \right\rangle \neq \left\langle \widetilde{f'_1}, \ldots, \widetilde{f'_m} \right\rangle.$$

When this notation is used, the generators for I will be clear from the context.

So, for any (possible affine) polynomial, we can get a homogeneous polynomial of the same degree with a new variable. Notice that this is an "invertible" operation, meaning with this that f and I can always be recovered from $\widetilde{f}$ and $\widetilde{I}$. For this matter, the following notation comes in handy.

Definition. Let $c \in \mathbb{F}$. Given $f \in R[h]$, the polynomial

$$f_{(h=c)}(x_1, \ldots, x_n) = f(x_1, \ldots, x_n, c) \in R$$

is called the *specialization of f at $h = c$*. Analogously, given J an ideal of $R[h]$ generated by $p_1, \ldots, p_m \in R[h]$, we define $J_{(h=c)}$ as the ideal of R generated by $(p_1)_{(h=c)}, \ldots, (p_m)_{(h=c)}$, and it is called the *specialization of J at $h = c$*.

Remark. As in the homogenization, $J_{(h=c)}$ depends on the generators chosen, so these must be clear from the context.

Using this definition, we get that for any $f \in R$ it holds that $\widetilde{f}_{(h=1)} = f$ and $\widetilde{f}_{(h=0)}$ is equal to the highest homogeneous part of f, observations that will be relevant in subsequent results. We now wonder what properties do $\widetilde{f}$ and $\widetilde{I}$ inherit from f and I and, conversely, what properties do $p_{(h=c)}$ and $J_{(h=c)}$ inherit from p and J.

Varieties

Regarding the varieties, it is the case that the variety of I is closely related to the variety of $\widetilde{I}$. More precisely, if $f_1, \ldots, f_m \in R$ and we want to find the solutions to $(f_1 = 0, \ldots, f_m = 0)$, we can solve the homogeneous system $(\widetilde{f}_1 = 0, \ldots, \widetilde{f}_1 = 0)$ and then for each solution $(a_1, \ldots, a_n, b) \in \mathbb{F}^{n+1}$ with $b \neq 0$, $\left(\frac{a_1}{b}, \ldots, \frac{a_n}{b}\right) \in \mathbb{F}^n$ is a solution to the original system.

Conversely, if $p_1, \ldots, p_m \in R[h]$ then the nonzero solutions of $(p_1 = 0, \ldots, p_m = 0)$ with nonzero last coordinate will have the form $(c \cdot a_1 \ldots, c \cdot a_n, c)$ for $c \in \mathbb{F} \setminus \{0\}$ and for every solution $(a_1, \ldots, a_n)$ to the system

$$\left(p_{1(h=1)} = 0, \ldots, p_{m(h=1)} = 0\right).$$

This shows that, at least from a theoretical point of view, we can solve the homogenized system and extract the desired solutions from there.

Groebner Bases

Lemma 3.4.1. *[Frö97, Lemma 29, Chap 8] Let I be a homogeneous ideal in R. Then the reduced Groebner basis of I in any order consists of homogeneous elements.*

Recall from section 1.1 that given any monomial order $\prec$ we denote by $\prec'$ the graded order of $\prec$ which is defined by: $\alpha \prec' \beta$ if and only if $|\alpha| < |\beta|$ or $|\alpha| = |\beta|$ and $\alpha \prec \beta$.

Proposition 3.4.2. *[Frö97, Prop. 30, Chap 8] Let I be a homogeneous ideal in R. Then the reduced Groebner basis of I with respect to the orders $\prec$ and $\prec'$ coincide.*

This shows that for homogeneous ideals we can consider, without loss of generality, the graded order of a given order, which is very useful since in this case the degree of f is the highest among all the monomials in $\mathrm{supp}(f)$.

One would expect that Groebner bases of $\widetilde{I}$ say something about Groebner bases of I. This is indeed the case, but in order to see this we need the following definition.

Definition. Let $\prec$ be a monomial order in $R = \mathbb{F}[x_1, \ldots, x_n]$. We say that a monomial order $\prec_1$ in $\mathbb{F}[x_1, \ldots, x_n, h]$ *extends* the order $\prec$ if $\mathrm{LM}(f) = \mathrm{LM}\left(\widetilde{f}\right)$ for any $f \in R$.

Remark. Recall that an elimination order of the variables $x_1, \ldots, x_n$ in $R[h]$ is a monomial order $\prec_1$ of $R[h]$ such that, for all monomials $x^{\alpha_1}h^{b_1}$ and $x^{\alpha_2}h^{b_2}$, if $x^{\alpha_1} \prec_1 x^{\alpha_2}$ then $x^{\alpha_1}h^{b_1} \prec_1 x^{\alpha_2}h^{b_2}$. It can be easily seen that if $\prec$ is the monomial order of R defined by restricting $\prec_1$, then $\prec_1$ extends $\prec$.

The following tells us that we can get a Groebner basis of an ideal by specializing a Groebner basis of its homogenization.

Proposition 3.4.3. *[Frö97, Prop. 34, Chap 8] Let $I = \langle f_1, \ldots, f_m \rangle$ be an ideal of R and $\prec$ be any monomial order in R. Let $\prec_1$ be a monomial order in $R[h]$ that extends $\prec$. If $G = \{g_1, \ldots, g_s\}$ is a Groebner basis of $\widetilde{I}$ with respect to $\prec_1$, then $\{g_{1(h=c)}, \ldots, g_{s(h=c)}\}$ is a Groebner basis of $\widetilde{I}_{(h=c)}$ for every $c \in \mathbb{F}$ with respect to $\prec$. In particular, $\{g_{1(h=1)}, \ldots, g_{s(h=1)}\}$ is a Groebner basis of I with respect to $\prec$.*

3.4.2 Arithmetical Complexity for Affine Systems

In this section we are going to assume that the monomial order in $R[h]$ is grevlex with $h \prec_{\text{grevlex}} x_n \prec_{\text{grevlex}} \ldots \prec_{\text{grevlex}} x_1$, so in particular the monomial order in R is also grevlex with $x_n \prec_{\text{grevlex}} \ldots \prec_{\text{grevlex}} x_1$. Notice that the former extends the latter, so we will be able to use the previous proposition. This is reasonable since in practice we use this order to compute Groebner bases, as mentioned in section 2.3.

In theorem 3.2.2 we gave a very explicit bound on the arithmetical complexity of Lazard's algorithm applied to a set of homogeneous polynomials generating a zero-dimensional ideal. Suppose that we have now $f_1, \ldots, f_m \in R$ not necessarily homogeneous (but still generating a zero-dimensional ideal). We now show how to apply the previous results to derive similar conclusions on these type of systems.

Considering the Highest Degree Homogeneous Part

A very common practice is to consider the homogeneous polynomials $f_1^{(h)}, \ldots, f_m^{(h)}$ where $f_i^{(h)} = \widetilde{f}_{i(h=0)}$. To justify this, we have the following proposition. Recall that $\mathrm{MaxDeg}(I)$ denotes the highest degree among all the monomials in the reduced Groebner basis of I.

Proposition 3.4.4. *Let I be the ideal of R generated by $f_1, \ldots, f_m$ and $J = \left\langle f_1^{(h)}, \ldots, f_m^{(h)} \right\rangle$. Then*

$$\mathrm{MaxDeg}(I) \leq \mathrm{MaxDeg}(J) \leq \mathrm{MaxDeg}\left(\widetilde{I}\right).$$

Proof. We begin by noticing that J equals $\widetilde{I}_{(h=0)}$. Let G, B and H be the reduced Groebner bases of I, J and $\widetilde{I}$ respectively. By proposition 3.4.3, we have that $B' = \left\{p_{(h=0)} : p \in H\right\}$ is a Groebner basis of J (not necessarily reduced). Nevertheless, B' contains a minimal Groebner basis of J and from it we can obtain B by applying algorithm 2. Since this operation involves only polynomial division and given that the order considered is graded (grevlex), we conclude that $\mathrm{MaxDeg}(J)$ is less than or equal to the highest degree of the polynomials in B'. On the other hand, for any polynomial $p \in H$ we have two cases: either h divides $\mathrm{LM}(p)$ and therefore (by the properties of the grevlex order) h divides p so $p_{(h=0)} = 0$, or h does not divide $\mathrm{LM}(p)$ in which case $\mathrm{LM}\left(p\right) = \mathrm{LM}\left(p_{(h=0)}\right)$. In any case, we see that $\deg\left(p_{(h=0)}\right) \leq \deg(p)$ for all $p \in H$ so we conclude that the highest degree of the polynomials in B' is at most $\mathrm{MaxDeg}(J)$, hence $\mathrm{MaxDeg}(J) \leq \mathrm{MaxDeg}\left(\widetilde{I}\right)$.

We omit the proof for the other inequality. □

Recall that we defined the degree of regularity of the polynomials $f_1, \ldots, f_m$ (generating a zero-dimensional ideal) as the degree of regularity of the homogeneous polynomials $f_1^{(h)}, \ldots, f_m^{(h)}$, which as we saw in section 2.2.4 is usually equal to the maximum degree of the Groebner basis of the ideal that these polynomials generate. Last proposition shows then that $\mathrm{d_{reg}}\left(f_1, \ldots, f_m\right)$ usually bounds the maximum of the degrees of the polynomials in the Groebner basis of the ideal they generate. We saw in section 2.2.4 that in the homogeneous case, this maximum degree actually determines the complexity of Lazard's algorithm, since it must run up to this step in order to obtain a Groebner basis. However, if we run affine Lazard's algorithm then this maximum degree does not tell the algorithm when to stop exactly, but as we saw in section 2.2.3 this maximum degree and the minimum number of steps needed to obtain a Groebner basis do not differ too much.

In conclusion, the degree of regularity of the polynomials $f_1, \ldots, f_m$ usually bounds the number of steps needed to compute a Groebner basis using Lazard's algorithm.

Considering the Homogenized System

It is not difficult to be convinced that computing Groebner bases for homogeneous ideals is easier than doing it for affine systems, the difference between homogeneous and affine Lazard's algorithms illustrate this. Due to proposition 3.4.3, given polynomials $f_1, \ldots, f_m \in R$ we can find a Groebner basis for the homogeneous ideal $\left\langle \widetilde{f_1}, \ldots, \widetilde{f_m} \right\rangle$ and then specialize it at $h = 1$ to obtain Groebner basis for $\langle f_1, \ldots, f_m \rangle$. Even though this may seem to be the best approach, one has to keep in mind that homogenizing adds a new variable and therefore the sizes of the Macaulay matrices involved in Lazard's algorithm grows in size from

$$\sum_{\ell=0}^{d'} \binom{n+\ell-1}{\ell}$$

(affine Lazard's algorithm in n variables) to

$$\binom{n+d}{d}$$

(homogeneous Lazard's algorithm, $d' \gtrsim d$).

3.5 Dimension 0 vs Positive Dimension

Additional to the homogeneous restriction we also have imposed constantly that the ideals must be zero-dimensional. The first reason to do this are the applications. We are studying Groebner bases as a mean to achieve an end: solving systems of polynomial equations. In this context, this only has sense if the system has a finite number of solutions so that we can list them all (unless we pursue a rational parametrization of the variety, which we do not consider here). Moreover, the fields of interest to us are finite fields and even if there may be systems with an infinite number of solutions, we will be only interested in the solutions lying in the finite field, and the number of such is clearly finite.

However, Groebner bases themselves are a tool for solving many other problems (as we could see in section 1.5), most of which do not require the ideal to be zero-dimensional. Even though the analysis we have performed so far using the index of regularity only applies to zero-dimensional ideals, it is surprising that this number is related to the complexity even when the dimension is positive.

We do not prove this fact here, but we give experimental evidence in the appendix. As can be seen there, $\mathrm{MaxDeg}(I)$ behaves just like in the zero-dimensional case, staying close to the $\mathrm{i}_{\mathrm{reg}}(I)$ (however, it is sometimes greater or smaller than this index).

We showed in theorem 3.3.3 that the degree of regularity of a regular sequence $f_1, \ldots, f_n$ is the Macaulay bound

$$1 + \sum_{j=1}^{n} (\deg(f_j) - 1).$$

We conjecture an extension of this result that seem to be true from the experiments.

Conjecture 3.5.1. *Given $m \leq n$ and a regular sequence $f_1, \ldots, f_m$, the degree (index) of regularity is given by*

$$1 - \underbrace{(n-m)}_{\dim(I)} + \sum_{j=1}^{m} (\deg(f_j) - 1)$$

whenever this value is nonnegative, and zero otherwise.

3.6 Falling Degree

We have seen how to predict the running time of Lazard's algorithm for almost all polynomial systems, but how do we measure this running time for a specific system? The degree of regularity as was defined before may be as difficult to compute as a Groebner Basis itself, therefore we need a different property from the system which is easier to measure and that gives an idea about the running time of Lazard's algorithm, this parameter is the Falling Degree. We introduce this concept by giving a motivation for

it. Suppose we have $f_1, \ldots, f_m \in R$ polynomials and we want to find a Groebner Basis for $I = \langle f_1, \ldots, f_m \rangle$, then we would proceed as depicted in figure 2.1 by considering the set of combinations $p_1 f_1 + \cdots + p_m f_m$ where each $p_i f_i$ has degree d. In general we have that $\dim(I_\ell) < \dim(R_\ell)$, and as discussed in section 2.2.4, this process will find a Groebner Basis once $I_\ell = R_\ell$. Suppose that at a degree $\ell < d$ we find a combination $f = p_1 f_1 + \cdots + p_m f_m$ where each $p_i f_i$ has degree d and such that f has degree ℓ, then, given that $f \in I$, this polynomial can be added to the Macaulay matrix at degree ℓ. If f was not already in this matrix, it will enlarge it and it will produce new polynomials that can be used for the degrees above ℓ. This effect will propagate among these degrees and we will obtain more of these polynomials that will help enlarging the corresponding matrices. The result of this is that $\dim(I_t) = \dim(R_t)$ will be achieved for a t close to d and therefore the algorithm will terminate at this step. The smallest degree d at which this situation occurs will be called Falling Degree.

We make this idea more precise in the following, but we begin by settling some notation used in our context.

3.6.1 Reduced Ring

In cryptography we are interested only in finite fields, so we will set for the rest of this work $\mathbb{F} = \mathbb{F}_q$ (finite field with q elements, where q is a prime number). Additionally, we are not interested in solutions that lie outside the field $\mathbb{F}$. A common practice to force this condition is to append the field equations $x_1^q - x_1 = 0, \ldots, x_n^q - x_n = 0$ to every polynomial system (see proposition 3.1.1). However, this can be seen equivalently as performing computations over a finitely generated algebra $\mathcal{R} = \mathbb{F}[\overline{x_1} \ldots, \overline{x_n}]$ with the relations $\overline{x_i}^q = \overline{x_i}$ for each i. This approach is more efficient and this is actually what is done in practice, but in order to work in this new algebra a formalization is required. More precisely, we will state the required notions to regard the elements of the algebra $\mathcal{R}$ as polynomials with algebraic variables while preserving some properties from transcendental variables like degree, homogeneity, specialization, among others.

Definition. We define the *algebra of functions* as

$$\mathcal{R} := \frac{\mathbb{F}[x_1, \ldots, x_n]}{\langle x_1^q - x_1, \ldots, x_n^q - x_n \rangle},$$

which is the image of R under the natural projection given by $f \in R \mapsto \pi(f) = f + \langle x_1^q - x_1, \ldots, x_n^q - x_n \rangle$, in particular, $\mathcal{R} = \mathbb{F}[\overline{x_1}, \ldots, \overline{x_n}]$ where $\overline{x_i} := \pi(x_i)$. We refer to the elements of this algebra as *functions*.

The name is due to the fact that every function $f : \mathbb{F} \to \mathbb{F}$ can be identified with a polynomial of $\mathcal{R}$, which is a consequence of the discussion below.

We now give a unique way of representing the elements of $\mathcal{R}$. This basically follows the intuitive idea that every polynomial in $\mathcal{R}$ can be written uniquely as a polynomial in the variables $\overline{x_1}, \ldots, \overline{x_n}$ with each variable having degree at most $q - 1$, performing reductions of powers modulo q. We need at first a very simple lemma.

Lemma 3.6.1. *[CLO07, prop. 4, §9, chap. 2] Let $G = \{g_1, \ldots, g_m\} \subseteq R$. If $\gcd(\mathrm{LM}(g_i), \mathrm{LM}(g_j)) = 1$ for all $i \neq j$, then G is a Groebner basis.*

Proposition 3.6.2. *Every polynomial in $\mathcal{R}$ can be written uniquely as a polynomial in the variables $\overline{x_1}, \ldots, \overline{x_n}$ with each variable having degree at most $q-1$.*

Proof. By the previous lemma we have that $\{x_1^q - x_1, \ldots, x_n^q - x_n\}$ is a Groebner basis (for every monomial order), so in particular by proposition 1.4.2 each element $f + \langle x_1^q - x_1, \ldots, x_n^q - x_n\rangle$ in the quotient $\mathcal{R}$ can be identified uniquely with its remainder. Finally, every remainder has the property that all of its monomials are not divisible by any $\mathrm{LM}(x_i^q - x_i) = x_i^q$, so in particular the degree of each variable is at most $q-1$. □

Let V be the subspace of R given by

$$V := \{f \in R : \text{ for all } x^\alpha \in \mathrm{supp}(f) \text{ and } i, x_i^q \nmid x^\alpha\}.$$

Due to the previous proposition for every $F \in \mathcal{R}$ there exist a unique $f \in V$ such that $\pi(f) = F$, and we denote this element by $\pi^{-1}(F)$. We define the degree of a function F in $\mathcal{R}$ to be the degree of the polynomial $\pi^{-1}(F)$ in R representing it.

Now that we have defined the concept of degree for elements in $\mathcal{R}$ we would like to obtain some properties of it. For instance, it is clear that the set of homogeneous functions in $\mathcal{R}$ of certain degree is a subspace of $\mathcal{R}$, and it is also true that every function in $\mathcal{R}$ can be decomposed uniquely as a sum of homogeneous functions. However, if we let $\mathcal{R}_d$ be the set of homogeneous functions in $\mathcal{R}$ of degree d, then $\mathcal{R} = \bigoplus_{d\in\mathbb{N}} \mathcal{R}_d$ but not as a graded algebra since it is not true that multiplications of functions of degree a and b have degree $a+b$ (for example, if $q=2$, $f = x_1$ has degree 1 and $f^2 = f$ has also degree 1). However, $\mathcal{R}$ inherits the structure of filtered algebra $\mathcal{R} = \cup_{d\geq 0}\mathcal{R}_{\leq d}$ with $\mathcal{R}_{\leq d} = \pi(R_{\leq d})$.

In the rest of this thesis (unless otherwise stated), we will always work in the ring $\mathcal{R} = \mathbb{F}[\overline{x_1}, \ldots, \overline{x_n}]$, and we will omit the bars in the variables and simply denote them by x_i, always assuming the relation $x_i^q - x_i$. The discussion from this section tells us that this does not affect basic concepts like degree, homogeneity, etc.

For simplicity in the sequel, we focus ourselves in quadratic polynomials. The definitions we will see can be easily extended to polynomials of any degree.

3.6.2 Degree Falls and Trivial Degree Falls

Let $f_1, \ldots, f_m \in \mathcal{R}$ be quadratic polynomials and suppose that $f = p_1 f_1 + \cdots + p_m f_m$ where each p_i has degree $d-2$. If $\deg(f) < d$ then the degree d homogeneous part of f, which is the same as that of $p_1^{(h)} f_1 + \cdots + p_m^{(h)} f_m$, is zero. This motivates the following definition.

Definition. Let $F = (f_1, \ldots, f_m)$ be a quadratic polynomial system. We say that $(h_1, \ldots, h_m) \in (\mathcal{R}_{d-2})^m$ is a *degree fall* in degree d of $f_1, \ldots, f_m$ if $h_1 f_1 + \cdots + h_m f_m$ has a degree strictly smaller than d.

Let $f_1, \ldots, f_m \in \mathcal{R}$ be quadratic polynomials and consider the $\mathbb{F}-$vector spaces homomorphism

$$\begin{aligned}\sigma_d(f_1, \ldots, f_m)\colon (\mathcal{R}_{d-2})^m &\longrightarrow \mathcal{R}_{\leq d}\\ (h_1, \ldots, h_m) &\longmapsto h_1 f_1 + \cdots + h_m f_m.\end{aligned}$$

There are always some predictable elements in the kernel of $\sigma_d(f_1, \ldots, f_m)$. For instance, when $d \geq 4$ the vector $s_{ij} := f_j \mathbf{e}_i - f_i \mathbf{e}_j$ always lies in this kernel since $f_j f_i - f_i f_j = 0$. This example may seem a little fake since this is yielding "trivially" zero, but a more interesting example can be found by considering the vectors $\left(f_j^{q-1} - 1\right) \mathbf{e}_j$, since $\left(f_j^{q-1} - 1\right) f_j = f_j^q - f_j = 0$. We notice that these relations are not using the structure of the polynomials at all, and they hold even if the p_j's are just the names of some variables. This motivates the following definition.

Definition. Consider the algebra $\mathcal{R}[y_1, \ldots, y_m]$ with the relations $y_j^q - y_j = 0$ and let $T_q(y_1, \ldots, y_m)$ denote the set of all tuples $(h_1, \ldots, h_m) \in (\mathcal{R}[y_1, \ldots, y_m])^m$ such that $h_1 y_1 + \cdots + h_m y_m = 0$. Given a quadratic polynomial system $(f_1, \ldots, f_m) \in \mathcal{R}^m$, the set of *trivial syzygies* of $f_1, \ldots, f_m$ is formed by all the polynomials in $T_q(y_1, \ldots, y_m)$ evaluated at $y_j = f_j$ for each j. We denote this set by $T_q(f_1, \ldots, f_m)$.

The degree $d-2$ homogeneous parts of the trivial syzygies of $f_1, \ldots, f_m$ are clearly degree falls in degree d of $f_1, \ldots, f_m$, and these are known as *trivial degree falls*. These degree falls are not interesting since they give no additional information about the polynomials, as shown in [DS13]. We are now ready to define the concept of falling degree.

Definition. Let $F = (f_1, \ldots, f_m) \in R^m$ be a quadratic polynomial system. We define its *falling degree* as the smallest d such that a non trivial degree fall of $f_1, \ldots, f_m$ exists in degree d.

Our conclusion is that the falling degree serves as a complexity parameter to bound the running time of Groebner basis algorithms, and this is the parameter we use to analyze the security of our proposals in the upcoming chapters. A final remark, which will be substantial in the mentioned analysis, is that the falling degree of a given system is invariant under linear combinations of the polynomials and linear changes of variables.

Remarks about Different Complexity Parameters

We discussed the falling degree as an alternative to the degree of regularity, which is a way of measuring the running time of Groebner basis algorithms. It is very common to find the concept of falling degree defined here being referred as degree of regularity on the literature. This, however, is a misunderstanding since

The falling degree and the degree of regularity are different concepts

Given an ideal I, it may be the case that its degree of regularity and falling degree differ. What is certainly true is that for almost all systems these concepts give an idea of arithmetic complexity for Groebner basis algorithms and therefore are very close in general, but they are not necessarily equal.

Part II

Applications to the security of MPK Cryptosystems

Chapter 4

Multivariate Public Key Cryptography

In this chapter we introduce basic ideas from Multivariate Public Key Cryptography, including basic constructions and examples. This will give the context to the New Alternatives we propose later on in this work

4.1 Preliminaries on Cryptography

We consider it appropriate to give a context on the general problem that is being addressed with MPKC, which is allowing a secret communication between two parties (usually referred as *Alice* and *Bob*).

In this section, we exhibit the problem of secret communication and the solution from Public Key Cryptography. We stress that we are going to keep an informal speech during this section, and we refer the reader to formal definitions when needed.

4.1.1 Public Key Cryptography

Suppose that **Alice** have a message m and she wants **Bob** to learn this message while guaranteeing that no one but Bob will be able to do so.

To solve this problem, suppose we have a function $\mathcal{P}$ such that

1. $\mathcal{P}$ is one-to-one[1]
2. $\mathcal{P}$ is very easy to evaluate for Alice (and in general for anyone who wishes to send a message to Bob)
3. $\mathcal{P}$ is not easy to invert for anybody who simply knows $\mathcal{P}$
4. Bob possesses some **secret information** that allows him to efficiently invert this function[2]

The first three properties ensure that $\mathcal{P}$ is a *One-way Function*, and the last one that it is a *Trapdoor Function*. See [KL07] for details on these concepts.

[1] we will see that many of our constructions satisfy a more relaxed condition which can be stated as being "few-to-one", that is, every element in the range of the function has "few" preimages

[2] from the properties it can be seen that necessarily this secret information can not be found from $\mathcal{P}$ since in this case, anyone with access to this function would be able to invert just like Bob

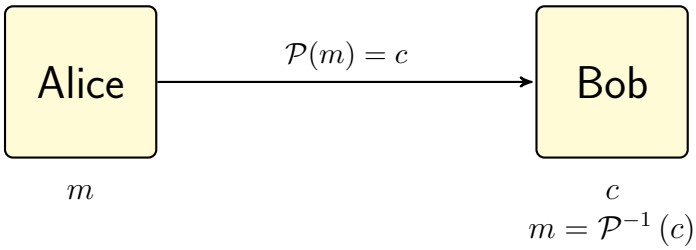

Figure 4.1: Protocol that allows Alice send her message to Bob securely

What Alice can do in order to solve her issue is evaluating m at $\mathcal{P}$, obtaining $\mathcal{P}(m)$. Then she can send this value to Bob, due to our assumptions about $\mathcal{P}$, no one is able to learn m from this value. Once Bob receives this value, he can use his secret information to invert the function and therefore finding m. Figure 4.1 pictures this idea.

We now introduce some notation common in Cryptography

- The function $\mathcal{P}$ described above, along with all other information necessary to evaluate it are often referred as the **Public Key**, since this is "public" for anyone who wishes to send a message to Bob[3];
- The secret information possessed by Bob is the **Secret Key**;
- Every possible message m in the domain of $\mathcal{P}$ is called a **Plaintext**, and every element of the range of this function is known as a **Ciphertext**;
- **Encryption** is the act of evaluating the function $\mathcal{P}$ and **Decryption** is the act of inverting it.

The general way that trapdoor functions are constructed is by means of a procedure Gen that takes the secret information sk and outputs the correspondent trapdoor function P that can be inverted with the secret key sk. It is clear that the procedure Gen can not be invertible because in this case one would be able to recover the secret information from the function, therefore violating its properties.

Example. (RSA) Consider two large prime numbers p and q, e some positive integer and d such that $ed \equiv 1 \mod \phi(N)$ with $N = pq$, where ϕ is the Euler's totient function. With this setting basic number theory can show that for every integer m between 0 and $n-1$ we have that

$$(m^e)^d \equiv m^{ed} \equiv m^1 \mod N.$$

Let $\mathcal{P}$ be the function that takes m and raise it to the e-th power and takes modulo N. It is widely assumed that computing $m \mod N$ from $\mathcal{P}(m)$ is a difficult task without additional information, but as we have seen, we can achieve this by having knowledge of d since we simply compute $m \equiv \mathcal{P}(m)^d \mod N$. If we keep d secret, then only someone with this information will be able to decrypt; moreover, we found d by means

[3] if you are familiar with cryptography, then you probably regard the public key as some parameter pk which is fed to a function $\mathrm{Enc}_{pk}(\cdot)$; here we regard the public key $\mathcal{P}$ as this function itself, which is an equivalent and more convenient approach

of p and q, so at the end what must be kept secret is the prime factorization of N, so the security of this cryptosystem heavily relies on the problem of factoring large numbers. See [Sho05] for details on this cryptosystem.

4.1.2 Post-Quantum Cryptography

The development of Quantum-Computers is a very big research field with a lot of investment, and expert estimate that within the next two decades these computers could be built. This may seem like good news, but this is a concern for the security of communications.

RSA example we saw before is not merely a theoretical Public Key Cryptosystem, many of our communications today actually use this cryptosystem to ensure privacy. As we noticed there, an attacker would be able to learn the secret information if he can factor large numbers into primes. Even though this is widely believed to be a hard problem in a classical computers, an algorithm for quantum-computers developed by Peter Shor [Sho99] can perform this task in only polynomial time.

The latter shows that cryptosystems based on problems like factoring (or finding discrete logarithms, which is another widely used technique and can be also broken with Shor's algorithm) will not be secure in the near future, hence, we need to develop new schemes whose security rely in different problems that can not be solved efficiently even by a quantum computer. One of these problems is the MQ-problem, related to polynomial system solving. We will discuss this in detail.

4.2 Multivariate Public Key Cryptosystems

During the rest of this work $\mathbb{F}$ will denote a finite field with q elements (q a prime number) and $\mathbb{K}$ will denote a field extension of $\mathbb{F}$ of degree n (see section 6.1.1 in the appendix for more details on these concepts). Recall that $R_{\leq d}$ is the set of polynomials in $R = \mathbb{F}[x_1, \ldots, x_n]$ of degree at most d. Elements in $R_{\leq 2}$ are known as *quadratic polynomials*. A function $F : \mathbb{F}^n \to \mathbb{F}^m$ is called a *regular function* if it is given by m multivariate polynomials (actually, one can easily prove that every function $\mathbb{F}^n \to \mathbb{F}^m$ is regular once we impose the relations $x_i^q = x_i$, see section 3.6.1), and it is *quadratic* if each component is a quadratic polynomial.

Consider the following computational problem.

MQ Problem Let $f_1, \ldots, f_n \in R$ be quadratic multivariate polynomials chosen uniformly at random. Find $(a_1, \ldots, a_n) \in \mathbb{F}^n$, if there is any, such that for all $i = 1, \ldots, n$

$$f_i(a_1, \ldots, a_n) = 0.$$

There are many reasons to believe that this problem is hard, even for quantum computers. From the theoretical point of view, it has been proved that the problem of deciding whether or not a given polynomial system has a solution or not is NP-complete [GJ90]. This is valuable since we do not expect NP to be equal to P even in the quantum model of computation. However, there may be NP-complete problems whose difficulty

in the average case is not that hard. Nonetheless, this is not the case with the MQ problem since there are not known better techniques for polynomial systems over finite fields than the general ones we have studied in the first part of this work, and we have seen there that random systems are ought to behave as regular sequences and therefore, the best approach to this problem is exponential in n (see Theorem 3.2.2). Moreover, nowadays there is no known polynomial-time quantum algorithm to solve the problem.

This problem will be the starting point for us to build the so-called Multivariate Public Key Cryptosystems. For these schemes, the trapdoor function is a function $\mathcal{P} : \mathbb{F}^n \to \mathbb{F}^m$ where each coordinate is given by a polynomial, and the secret key is some secret information allowing us to invert this function.

Assumption Given $F : \mathbb{F}^n \to \mathbb{F}^n$ defined by n quadratic polynomials chosen uniformly at random and given $\mathbf{c}$ in the range of F, it is difficult to find $\mathbf{a} \in \mathbb{F}^n$ such that $F(\mathbf{a}) = \mathbf{c}$.

Remark. To find such $\mathbf{a}$ one must solve the system of equations $p_1(\mathbf{x}) = c_1, \ldots, p_n(\mathbf{x}) = c_n$, where the p_i's are the quadratic polynomials defining F and $\mathbf{c} = (c_1, \ldots, c_n)$. By defining the quadratic polynomials $q_i(\mathbf{x}) := p_i(\mathbf{x}) - c_i$, this is the same as solving the system $q_1(\mathbf{x}) = 0, \ldots, q_n(\mathbf{x}) = 0$. This may look the same as the MQ problem, but the difference here is that the q_i's are not chosen at random! for instance, we know a priori that the system possesses at least one solution, which is not the general case in the MQ problem. However, experimental evidence shows that it does not hurt to assume that the latter problem is difficult too, which is the assumption we need to make in order to build our trapdoor functions.

What we have so far is that if we pick a random function from the set of all quadratic regular functions $\mathbb{F}^n \to \mathbb{F}^n$, the chances are that this function is not easy to invert. Moreover, another reasonable assumption is that regular functions $\mathbb{F}^n \to \mathbb{F}^n$ chosen at random are very likely to be "few-to-one".

In order to construct trapdoor functions, we only need to describe a generation procedure Gen that picks some secret information and outputs a regular quadratic function which looks like random and is easy to invert using this secret information.

In what follows we describe the generation procedure that outputs regular functions easy to invert with the secret information. Notwithstanding, there is not a known way today we can ensure that these functions are easy to invert only if the secret information is possessed (which is the property we need on a trapdoor function). In fact, for many constructions today either the generation procedure is invertible (that is, the secret information can be recovered from the regular function) or the behavior of the resulting regular functions is not like that of random ones, resulting in easier to invert functions.

As a final note, we extend our constructions to trapdoor functions $\mathbb{F}^n \to \mathbb{F}^m$, where m may be different than n. The first observation is that m must be at least n since otherwise our functions would not be "few-to-one". On the other hand, if m is very large with respect to n, theory developed in [Bar04] shows that our systems may be easier to solve, yet it is not harming if $m = O(n)$.

Also, please note that although the assumption is stated for quadratic polynomials, it can be easily generalized for degree $d \geq 2$ polynomials without loss on the hardness. Given this, we will not restricts ourselves to quadratic polynomials in the exposition of the general constructions.

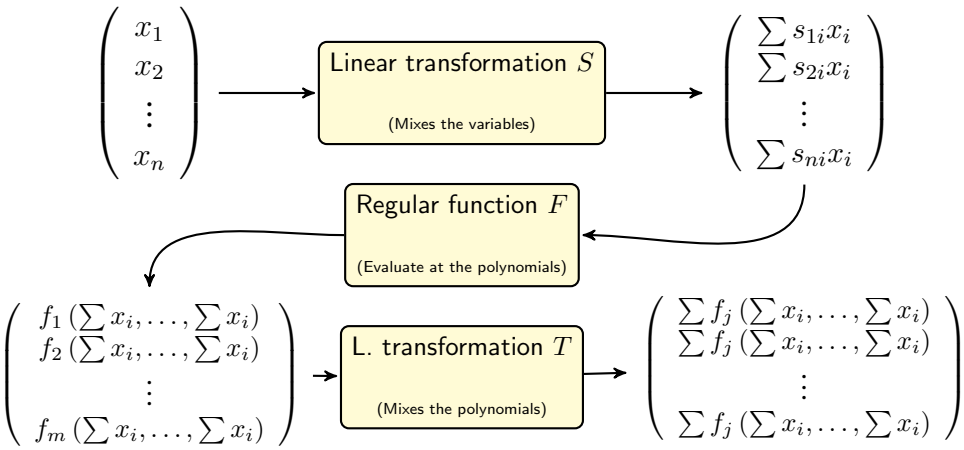

Figure 4.2: Construction of MPK Cryptosystems from easy-to-invert regular functions

4.2.1 First Reduction: Bipolar Construction

Definition. Given a regular function $F : \mathbb{F}^n \to \mathbb{F}^m$, $S : \mathbb{F}^n \to \mathbb{F}^n$ and $T : \mathbb{F}^m \to \mathbb{F}^m$ linear transformations, we define the *bipolar construction* of F, S and T as the regular function $P : \mathbb{F}^n \to \mathbb{F}^m$ given by $P = T \circ F \circ S$.

It can be easily seen that if each polynomial in F has degree d, then each polynomial in P also has degree d.

Assume now that we have a regular function $F : \mathbb{F}^n \to \mathbb{F}^m$ with the following property: Any equation $F(x_1, \ldots, x_n) = (c_1, \ldots, c_n)$ where $(c_1, \ldots, c_n) \in F(\mathbb{F}^n)$ can be efficiently solved[4]. Clearly, F would not serve as a public key itself since anyone is able to invert it, however, we can create a MPK Cryptosystem from F by choosing uniformly at random two linear transformations $S : \mathbb{F}^n \to \mathbb{F}^n$ and $T : \mathbb{F}^m \to \mathbb{F}^m$ and considering $P = T \circ F \circ S$, the bipolar construction of F, S and T. The idea with this construction is that S mixes the variables and T mixes the equation, therefore hiding the structure of the function F. Figure 4.2 shows how the process work.

An important property of this construction is that someone who knows F, S and T can easily invert any equation of the form $P(x_1, \ldots, x_n) = (c_1, \ldots, c_n)$ where $(c_1, \ldots, c_n) \in P(\mathbb{F}^n)$ since $P^{-1} = S^{-1} \circ F^{-1} \circ T^{-1}$ and we are assuming that F is easy to invert (here, we must notice that $T^{-1}(c_1, \ldots, c_n) \in F(\mathbb{F}^n)$). Therefore, it makes sense to consider F, S and T as secret information and P as the public information. From the security point of view, we want to make sure that someone who simply sees P is not able to recover F, S and T, which is some kind of "factoring problem" for maps. This problem is assumed to be hard in general, and is closely related to the Jacobian conjecture on Invertible Polynomial Maps. Unfortunately, there may be some F's for which this problem is not difficult, and this may lead to attacks like MinRank attack.

On the other hand, an important concern is that we can not ensure that the only way to invert the function P is by making use of F, S and T. For instance, if F is linear

[4]we restrict ourselves to only inverting the function where there is indeed a preimage of the element involved. This makes sense since we only want to decrypt valid ciphertexts. Some of the cryptosystems we will encounter only allow us to invert in this situation, and they would fail to decrypt if a non-valid ciphertext is asked for decryption

then P is linear as well, and then of course everyone can invert the function P without having any knowledge of F, S nor T. It is clear that one would not take F to be linear for this construction, but deeper conditions can be found, for example, F is easy to invert if it has a low falling degree since Lazard's algorithm finishes at an early stage, however, we saw in section 3.6 that bipolar constructions inherit the falling degree from F and hence P would be easy to invert for anyone as well. The precise requirement for F so that the bipolar construction P is not easy to invert is not clear. In fact, many of the defeated MPK Cryptosystems are in such a status due to the fact that the function P has a low falling degree and therefore is easy to invert.

In any case, in many cases this can be assumed to be a hard problem and therefore it makes sense to look at easy-to-invert regular functions $F : \mathbb{F}^n \to \mathbb{F}^m$ to build trap-door functions by doing the Bipolar Construction, and now we focus on the problem of finding such F's. We stress that we do not know yet a sufficient condition on F that guarantees that the bipolar construction is difficult to factorize or more generally to invert.

4.2.2 Second Reduction: Lifting Idea

According to the previous section, now we need to focus in building regular functions $F : \mathbb{F}^n \to \mathbb{F}^m$ that are easy to invert. The method we will use for this is known as the *lifting idea*, and involves an extension field of $\mathbb{F}$ and univariate polynomials over this extension.

Consider a field extension $\mathbb{K}$ of $\mathbb{F}$ of degree n, and consider $\phi : \mathbb{K} \to \mathbb{F}^n$ to be the natural linear transformation between these vector spaces (see section 6.1.1 for more details on this). Recall our notation $R := \mathbb{F}[x_1, \ldots, x_n]$. Given a nonzero natural number b, any other nonzero natural number a can be written uniquely as $a = c_1 b^0 + c_2 b^1 + \cdots + c_\ell b^{\ell-1}$ where $0 \leq c_i < b$ for all i. We say that $(c_1, \ldots, c_\ell)$ is the expansion of a in base b, and we refer to $d = \sum_{i=1}^{\ell} c_i$ as the $b-$Hamming weight of a. In order to extend the definition we define the $b-$Hamming weight of $a = 0$ to be 0. To illustrate the concept, a has $q-$Hamming weight 2 if and only if it has the form $a = q^i + q^j$.

Definition. The *weight* of a monomial $X^a \in \mathbb{K}[X]$ is the $q-$Hamming weight of a. A polynomial $\mathcal{F}(X) \in \mathbb{K}[X]$ is said to be *homogeneous of weight* d if all of its monomials have weight d, and it is said to have *weight* d if all of its monomials have weight at most d.

The importance of the concept of *weight* is that it corresponds to *degree* on multivariate polynomials under what we call Lifting and Droppings, as we can see in the following theorem.

Theorem 4.2.1. (Correspondence of Polynomials). *Let $d \geq 0$ be an integer, let $\mathbb{K}[X]_d$ denote the set of homogeneous polynomials in $\mathbb{K}[X]$ of weight d and let $(R_d)^n = R_d^n$ denote the set of all functions $F : \mathbb{F}^n \to \mathbb{F}^n$ where each coordinate is a homogeneous polynomial in $\mathbb{F}[x_1, \ldots, x_n]$ of degree d, these sets are naturally $\mathbb{F}$-vector spaces. The following is a well-defined bijective linear transformation*

$$\begin{aligned} \mathrm{Drp}\colon \mathbb{K}[X]_d &\longrightarrow R_d^n \\ \mathcal{F} &\longmapsto \phi \circ \mathcal{F} \circ \phi^{-1}. \end{aligned}$$

whose inverse is

$$\begin{aligned} \mathrm{Lft}\colon R_d^n &\longrightarrow \mathbb{K}[X]_d \\ F &\longmapsto \phi^{-1} \circ F \circ \phi. \end{aligned}$$

The proof of this theorem can be found in section 6.2 in the appendix. The names Lft (lifting) and Drp (dropping) arise from the following commutative diagram, which illustrates the correspondence.

$$\begin{array}{ccc} \mathbb{K} & \xrightarrow{\mathcal{F}} & \mathbb{K} \\ {\scriptstyle \mathrm{Lft}(F)}\uparrow \quad \uparrow{\scriptstyle \phi^{-1}} & & \downarrow{\scriptstyle \phi} \quad \downarrow{\scriptstyle \mathrm{Drp}(\mathcal{F})} \\ \mathbb{F}^n & \xrightarrow{F} & \mathbb{F}^n \end{array}$$

Clearly, F is invertible if and only if $\mathcal{F}$ is, so we can focus now in finding easy-to-invert univariate polynomials $\mathcal{F}(X) \in \mathbb{K}[X]$ with weight at most d. Even though this correspondence exists for degree higher than 2, it has been used so far only for the quadratic case. Section 6.3 shows that this procedure is very efficient.

As a comment, there is not a restriction in using only one polynomial. In the summary below we state the generalization of this.

4.2.3 General Construction

To sum up, we describe the general procedure to build a trapdoor function $P : \mathbb{F}^n \to \mathbb{F}^m$ where $m = tn$.

1. Choose some secret linear transformations $S, T_1, \ldots, T_t : \mathbb{F}^n \to \mathbb{F}^n$
2. Find t univariate polynomials $\mathcal{F}_1, \ldots, \mathcal{F}_t \in \mathbb{K}[X]$ having weight at most d such that system of equations $(\mathcal{F}_1(X) = Y_1, \ldots, \mathcal{F}_t(X) = Y_t)$ where $Y_i \in \mathcal{F}_i(\mathbb{K})$ can be efficiently solved
3. The trapdoor function is $P : \mathbb{F}^n \to \mathbb{F}^m$ given by $P = (P_1, \ldots, P_t)$ with $P_i = T_i \circ \mathrm{Drp}\,(\mathcal{F}) \circ S$

This construction is depicted in figure 4.3.

So far we have considered degree d polynomials, with $d \geq 2$; however, many of the constructions so far involve only quadratic polynomials. This makes sense due to the following considerations

- There are $\binom{n+d-1}{d} = O(n^d)$ monomials of degree d, so we need $O(mn^d)$ elements from the field $\mathbb{F}$ to store m polynomials in R of degree d. If $d = 2$ then this is a manageable size, by raising d one gets sizes beyond practical applications.[5]
- In order for this construction to be efficient one needs to be able to compute $\mathrm{Drp}(\mathcal{F})$ from $\mathcal{F}$ in an efficient manner. This is well known in the quadratic case, as we describe in section 6.3.

[5] $d = 3$ is still manageable, which is the starting point for our contributions in next section

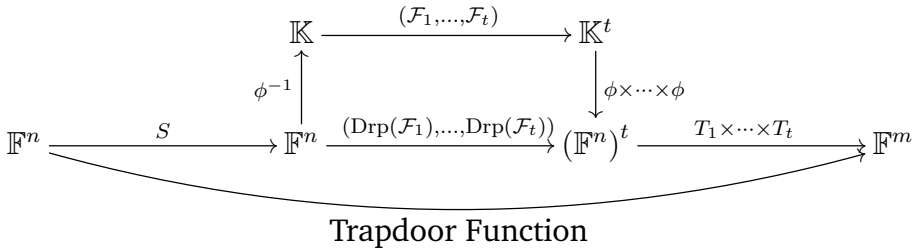

Figure 4.3: General Construction of Multivariate Trapdoor Functions

4.3 Examples: HFE and ZHFE

We now discuss two examples of MPK Cryptosystems: Hidden Field Equations (HFE) and ZHFE. The former was proposed by Patarin in 1996 [PG97], and was a good alternative until Kipnis and Shamir proposed the so-called MinRank attack [KS99]. It was a theoretical attack back then, but subsequent work by L. Perret et al [BFP13] improved this attack for any set of practical parameters.

On the other hand, ZHFE was proposed as an alternative to avoid the MinRank attack. It was presented in 2014 by Porras et al. [PBD15] and it was well received by the MPKC community for its new and creative idea. Unfortunately, it had efficiency issues in its very beginning. Almost one year after its release, an improvement on the efficiency of ZHFE and a security analysis based in the min-rank were published [BCE+16, PS16]. Although the former gave a hope on the future of ZHFE as a usable primitive, the latter showed a weakness on the cryptosystem that led to the necessity of reformulating it.

4.3.1 HFE

Recall that we need to find polynomials $\mathcal{F}_1(X),\ldots,\mathcal{F}_t(X) \in \mathbb{K}[X]$ which are, in conjunction, easy to invert. In finite fields, just like in the field of real numbers, we have algorithms that can efficiently find the roots of a given univariate polynomial if its degree is small enough (e.g. Berlekamp and Cantor-Zassenhaus algorithms, see [LN97]). Given this, it is natural to consider low degree polynomials since these are naturally invertible.

Definition

In HFE, the core function is given by a low degree polynomial of weight two. More precisely, fix a parameter D and consider a polynomial of the form

$$\mathcal{F}(X) = \sum_{q^i+q^j \leq D} \alpha_{ij} X^{q^i+q^j}$$

(for illustrative reasons we assume $\mathcal{F}$ is homogeneous). If D is low enough, this function is easy to invert. The trapdoor function is built then by choosing some secret linear transformations $S, T : \mathbb{F}^n \to \mathbb{F}^n$ and computing $P = T \circ \phi \circ F \circ \phi^{-1} \circ S$.

Security Analysis

The HFE Cryptosystem has a vulnerability against what is known as a MinRank attack. Since we will encounter the same type of attack in the next chapter, it is worth to see the most relevant aspects of it. At first, write the polynomial $\mathcal{F}$ as

$$\mathcal{F}(X) = \begin{pmatrix} X^{q^0} & X^{q^1} & \cdots & X^{q^{n-1}} \end{pmatrix} \begin{pmatrix} * & \cdots & * & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ * & \cdots & * & 0 & \cdots & 0 \\ 0 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & \cdots & 0 \end{pmatrix} \begin{pmatrix} X^{q^0} \\ X^{q^1} \\ \vdots \\ X^{q^{n-1}} \end{pmatrix}$$

where only the $r \times r$ square on the top left of this matrix is nonzero ($r = \lfloor \log_q D \rfloor$). This should look familiar to the representation of quadratic forms in several variables but using the "variables" X^{q^i} instead (recall that $X^{q^n} = X$ for any particular $X \in \mathbb{K}$, so we only need to consider these powers up to $X^{q^{n-1}}$). Notice that the inner matrix has a low rank r (since D is small, by construction).

Proposition 4.3.1. *Let $P_i \in \mathcal{M}_{n\times n}(\mathbb{F})$ be the matrix representing the ith quadratic polynomial of the trapdoor function $P = T \circ \phi \circ F \circ \phi^{-1} \circ S$ (that is, each component of P is $p_i(\mathbf{x}) = \mathbf{x}^T P_i \mathbf{x}$), then there exist $\lambda_1, \ldots, \lambda_n \in \mathbb{K}$ such that the matrix $\sum_{i=1}^n \lambda_i P_i$ has low rank r.*

Given matrices $P_1, \ldots, P_n$, the problem of finding scalars $\lambda_1, \ldots, \lambda_n \in \mathbb{K}$ such that the matrix $\sum_{i=1}^n \lambda_i P_i$ has low rank is known as the MinRank problem, and it is in general a very hard computational problem. However, this is easy to solve in our case since we know that there is at least one solution for it at a very low rank r, and these scalars can be found for instance by methods like KS modeling or Minors modeling [KS99]. The first negative implication of this property is that the trapdoor functions from HFE are distinguishable from random regular functions, which is undesirable. Moreover, one is able to build (equivalent) secret keys that allow for decryption using the scalars that one obtain directly from the trapdoor function, so our construction is not secure.

4.3.2 ZHFE

It is worth mentioning ZHFE, which appeared as an alternative to overcome the MinRank attack. The basic construction for the core polynomial is as follows. Just like in HFE, we begin by fixing a small parameter D that will allow us to invert. Then we look for scalars $\alpha_1, \ldots, \alpha_{2n}, \beta_1, \ldots, \beta_{2n} \in \mathbb{K}$ and two weight 2 polynomials $\mathcal{F}(X)$ and $\widetilde{\mathcal{F}}(X)$ satisfying that the polynomial

$$\Psi(X) = X\left(\alpha_1 \mathcal{F}^{q^0} + \cdots + \alpha_n \mathcal{F}^{q^{n-1}} + \beta_1 \widetilde{\mathcal{F}}^{q^0} + \cdots + \beta_n \widetilde{\mathcal{F}}^{q^{n-1}}\right) + X^q\left(\alpha_{n+1} \mathcal{F}^{q^0} + \cdots + \alpha_{2n} \mathcal{F}^{q^{n-1}} + \beta_{n+1} \widetilde{\mathcal{F}}^{q^0} + \cdots + \beta_{2n} \widetilde{\mathcal{F}}^{q^{n-1}}\right),$$

has low degree D (it is important to note that a weight two polynomial raised to a Frobenius power q^i is again weight 2). These are obtained by solving sparse linear systems of equations (see [BCE+16] for more details on this). Our central function will be $\mathcal{G} = (\mathcal{F}, \widetilde{\mathcal{F}})$. To invert this function, suppose we are given $(Y_0, Y_1) \in \mathcal{G}(\mathbb{K})$, we want to find X such that $\mathcal{G}(X) = (Y_0, Y_1)$, that is, $\mathcal{F}(X) = Y_0$ and $\widetilde{\mathcal{F}}(X) = Y_1$. Clearly, such X will also satisfy the low degree polynomial equation

$$\Psi(X) = X\left(\alpha_1 Y_0 + \cdots + \alpha_n Y_0^{q^{n-1}} + \beta_1 Y_1 + \cdots + \beta_n Y_1^{q^{n-1}}\right) + X^q\left(\alpha_{n+1} Y_0 + \cdots + \alpha_{2n} Y_0^{q^{n-1}} + \beta_{n+1} Y_1 + \cdots + \beta_{2n} Y_1^{q^{n-1}}\right).$$

which we can solve, finding therefore the preimages of (Y_0, Y_1).

Security Analysis

Write

$$\Psi = \underbrace{X\left[L_0\left(F, \widetilde{F}\right)\right]}_{\Psi_0} + \underbrace{X^q\left[L_1\left(F, \widetilde{F}\right)\right]}_{\Psi_1}.$$

and recall that there are no terms of degree higher than D in Ψ. However, many of these terms come either from Ψ_0 or Ψ_1 (not both!). From this observation it can be seen that the matrices representing the quadratic forms $L_0\left(F, \widetilde{F}\right)$ and $L_1\left(F, \widetilde{F}\right)$ have the following shape

$$\begin{pmatrix} * & * & * & & * & & \\ * & * & * & \cdots & * & * & \cdots & * \\ * & * & * & & * & & \\ & \vdots & & \ddots & & & \\ * & * & * & & * & & \\ & * & & & & & \\ & \vdots & & & & & \\ & * & & & & & \end{pmatrix}, \quad \begin{pmatrix} * & * & * & & * & * & \cdots & * \\ * & * & * & \cdots & * & & & \\ * & * & * & & * & & & \\ & \vdots & & \ddots & & & & \\ * & * & * & & * & & & \\ * & & & & & & & \\ \vdots & & & & & & & \\ * & & & & & & & \end{pmatrix}$$

where each block on the top-left is $r \times r$, with $r = \lceil \log_q D \rceil$. Hence, these matrices have a low rank of $r + 1$. This may seem as the attack on HFE, but the main difference is that the low rank is possessed by L_0 and L_1, not $\mathcal{F}$ and $\widetilde{\mathcal{F}}$. However, it has been discovered that this is not a barrier for a similar attack to that on HFE [PS16], and this cryptosystem is unfortunately insecure.

Chapter 5

New Alternatives Using Cubic Polynomials

Ideas for MPK Cryptosystems are presented in this chapter. We also explore some attacks that partially break these schemes.

Many of the constructions seen so far un MPKC use quadratic polynomials. This makes sense since our assumptions say that these systems are difficult to solve, and from a theoretical point of view every polynomial system can be made quadratic by adding enough equations and renaming monomials. Another advantage of considering these systems is that it takes $O(mn^2)$ elements from the field $\mathbb{F}$ to store m quadratic polynomials, which is a reasonable number.

Our contribution is related to the use of cubic polynomials instead of quadratic. This will give us more flexibility but we will need $O(mn^3)$ elements from $\mathbb{F}$ to store m of these polynomials. However, this number is still manageable, and the possible advantages of using these may overcome the bottlenecks.

In the first section, we describe a first proposal for a Trapdoor function that happens to be breakable by a MinRank attack similar to that on HFE. Then we move to a variation of this proposal which seems to avoid this problem, but then a deeper analysis using the Falling Degree will expose a weakness in terms of a Direct Algebraic Attack. Finally, we explore the possibility of artificially raising the falling degree, but, as we will see, an "intelligent" attack can overcome this method.

5.1 Multivariate Noisy Encryption Scheme

We describe the first proposal for the cryptosystem, and we describe the attack that led to a reformulation of it.

5.1.1 Description

We will work in the same context as in the previous chapter: q is a prime number, n a positive integer, $\mathbb{F}$ a finite field of size q and $\mathbb{K}$ a field extension of $\mathbb{F}$ of degree n. For

our trapdoor function we will need a small parameter r which we will use for inverting the central function.

To build the central function, we begin by picking completely at random a weight 2 polynomial $\mathcal{F} \in \mathbb{K}[X]$. We also choose at random for each $j = 0, \ldots, r$, a $q-$weight 1 polynomial $\mathcal{M}_j \in \mathbb{K}[X]$ and a weight 3 polynomial $\mathcal{G}(X) \in \mathbb{K}[X]$ whose biggest power is $3q^r$. As usual, we choose two invertible linear transformations $S, T : \mathbb{F}^n \to \mathbb{F}^n$. Finally, we consider the weight 3 polynomial $\mathcal{H} : \mathbb{K} \to \mathbb{K}$ given by

$$\mathcal{H}(X) = \sum_{j=0}^{r} X^{q^j} \mathcal{M}_j \left(\mathcal{F}'(X)\right) + \mathcal{G}(X) \tag{5.1}$$

where $\mathcal{F}' = \mathcal{F} \circ \phi^{-1} \circ S^{-1} \circ \phi$.

The trapdoor function is then $P : \mathbb{F}^n \to \mathbb{F}^{2n}$ given by

$$P = (\phi \circ \mathcal{F} \circ \phi^{-1}, T \circ \phi \circ \mathcal{H} \circ \phi^{-1} \circ S),$$

while the secret information is $(\mathcal{F}, \mathcal{M}_i, \mathcal{G}, \mathcal{H}, S, T)$.

We refer to $\mathcal{G}$ as the *noise*, since it is intended to hide the structure $\sum X^{q^j} \mathcal{M}_j \left(\mathcal{F}'(X)\right)$

Remark. Since $\mathcal{F}$ is chosen completely at random, we do not need to apply the linear transformation T at the end. In addition to this, one may apply S on the right to $\mathcal{F}$ and by doing so one can use $\mathcal{F}$ directly on equation (5.1) rather than $\mathcal{F}'$. However, we keep the construction in this fashion to stress that the left part of the public key is completely random.

To invert P we proceed as follows. Suppose that we are being given $\mathbf{c} = (c_1, \ldots, c_{2n})$ in the range of P, we want to solve the simultaneous equations $\mathcal{F}\left(\phi^{-1}(\mathbf{x})\right) = Z_1$ and $\mathcal{H}\left(\phi^{-1}(S\mathbf{x})\right) = Z_2$ where $Z_1 = \phi^{-1}(c_1, \ldots, c_n)$ and $Z_2 = \phi^{-1} \circ T^{-1}(c_{n+1}, \ldots, c_{2n})$. By setting $X = \phi^{-1}(S\mathbf{x})$, this is the same as $\mathcal{F}'\left(X\right) = Z_1$ and $\mathcal{H}\left(X\right) = Z_2$. Any solution to this system will also satisfy the polynomial equation

$$Z_2 = \sum_{j=0}^{r} X^{q^j} \mathcal{M}_j \left(Z_1\right) + \mathcal{G}(X),$$

and the parameter r is chosen small enough so that this equation can be solved.

5.1.2 Computation of Cubic Droppings

If we want to work with weight three polynomials, then we must be able to compute $T \circ \phi \circ \mathcal{H} \circ \phi^{-1} \circ S$ from S, T and $\mathcal{H}$. As it has been mentioned before, this is well known in the quadratic case, but we needed to make an effort to develop similar theory for the cubic case.

We begin by introducing some notation. We denote the $i-$th row of a matrix M as $M_{(i)}$ (as a row vector) and its entry s, t as $M_{(s)}^{(t)}$. We write $[\alpha_{ij}]_{ij}$ for the matrix whose i, j entry is α_{ij}. We denote by C the companion matrix of the irreducible polynomial that

produces the field $\mathbb{K}$ so that for any $\alpha \in \mathbb{K}$ we have that $\phi(\alpha y^j) = C^j \cdot \phi(\alpha)$ (see lemma 6.2.2). Additionally, let D_j denote the matrix

$$\left[\begin{array}{c|c} 0 & I_{n-j} \\ \hline I_j & 0 \end{array}\right]$$

where I_ℓ is the $\ell \times \ell$ identity matrix.

Finally, we let $\mathbf{x}$ and $\mathbf{X}$ denote the vectors $(x_1, \ldots, x_n)$ and $(X^{q^0}, \ldots, X^{q^{n-1}})$, respectively, which are often considered as $n \times 1$ matrices.

Proposition 5.1.1. *If $\mathcal{F}(X) = \boldsymbol{X}^T F \boldsymbol{X}$, $\mathcal{G}(X) = \sum_{j=0}^{r} X^{q^j} \left(\boldsymbol{X}^T G_j \boldsymbol{X}\right)$ and for each $j = 0, \ldots, r$: $\mathcal{M}_j(X) = \sum_{i=0}^{n-1} m_{j,i} X^{q^i}$ where $F, G_j \in \mathcal{M}_{n\times n}(\mathbb{K})$ and $m_{j,i} \in \mathbb{K}$, then we can write $\mathcal{H}(X)$ as*

$$\mathcal{H}(X) = \sum_{j=0}^{r} X^{q^j} \left(\boldsymbol{X}^T A_j \boldsymbol{X}\right) \tag{5.2}$$

with

$$A_j = \sum_{i=0}^{n-1} m_{j,i} F_i + G_j,$$

where $F_i := D_i^T \left[\left(F_{(s)}^{(t)}\right)^{q^i}\right]_{st} D_i$.

Remark. The polynomial $\mathcal{G}$ has a degree of at most $3q^r$, hence, the matrix G_j is zero for $j > r$. Besides, for $j \leq r$, the entry s, t of the matrix G_j is zero whenever $r < \max(s-1, t-1)$.

Proof. By definition, we have

$$\begin{aligned}
\mathcal{H}(X) &= \sum_{j=0}^{r} X^{q^j} \mathcal{M}_j\left(\mathcal{F}(X)\right) + \mathcal{G}(X) \\
&= \sum_{j=0}^{r} X^{q^j} \left[\sum_{i=0}^{n-1} m_{j,i} \left(\mathcal{F}(X)\right)^{q^i}\right] + \sum_{j=0}^{r} X^{q^j} \left(\mathbf{X}^T G_j \mathbf{X}\right) \\
&= \sum_{j=0}^{r} X^{q^j} \left[\sum_{i=0}^{n-1} m_{j,i} \left(\mathcal{F}(X)\right)^{q^i} + \mathbf{X}^T G_j \mathbf{X}\right].
\end{aligned}$$

For each i, $(\mathcal{F}(X))^{q^i}$ is again a quadratic form and it is represented by a cyclic shift of the matrix F, and raising its entries to the power q^i. More specifically, this matrix may be written as

$$\left[\left(F_{(s-i \mod n)}^{(t-i \mod n)}\right)^{q^i}\right]_{st},$$

but it can be checked that this matrix is equal to F_i, hence

$$\begin{aligned}
\mathcal{H}(X) &= \sum_{j=0}^{r} X^{q^j} \left[\sum_{i=0}^{n-1} m_{j,i} \mathbf{X}^T F_i \mathbf{X} + \mathbf{X}^T G_j \mathbf{X}\right] \\
&= \sum_{j=0}^{r} X^{q^j} \left(\mathbf{X}^T \left[\sum_{i=0}^{n-1} m_{j,i} F_i + G_j\right] \mathbf{X}\right).
\end{aligned}$$

□

Let $\overline{A}_j := S^T \Delta^T A_j \Delta S$ where Δ is the matrix from section 6.1.2 and write the entries of this matrix as a linear combination of powers of y and distribute with respect to these powers, we obtain an expression of the form

$$\overline{A}_j = y^0 B_{0,j} + \cdots + y^{n-1} B_{n-1,j}$$

where each B matrix lies in $\mathcal{M}_{n\times n}(\mathbb{F})$.

Proposition 5.1.2. *The second component from the public key can be calculated as*

$$T \circ \phi \circ \mathcal{H} \circ \phi^{-1} \circ S(\mathbf{x}) = T \sum_{j=0}^{r} C_j \overline{D}_j \mathbf{x}.$$

where

$$C_j := \left[\left(\mathbf{x}^T B_{0,j} \mathbf{x}\right) \cdot C^0 + \cdots + \left(\mathbf{x}^T B_{n-1,j} \mathbf{x}\right) \cdot C^{n-1} \right] \in \mathcal{M}_{n\times n}\left(\mathbb{F}[x_1, \ldots, x_n]\right)$$

and $\overline{D}_j := \Delta^{-1} D_j \Delta S \in \mathcal{M}_{n\times n}(\mathbb{F})$

Proof. Replacing X by $\phi^{-1}(S\mathbf{x})$ in (5.2) and using the properties of Δ discussed in section 6.1.2, we have

$$\mathcal{H}\left(\phi^{-1}(S\mathbf{x})\right) = \sum_{j=0}^{r} \left(\Delta_{(j+1)} S\mathbf{x}\right) \cdot \left(\mathbf{x}^T S^T \Delta^T A_j \Delta S \mathbf{x}\right).$$

Since $S^T \Delta^T A_j \Delta S$ is equal to $y^0 B_{0,j} + \cdots + y^{n-1} B_{n-1,j}$, we obtain

$$\begin{aligned}
\mathcal{H}\left(\phi^{-1}(S\mathbf{x})\right) &= \sum_{j=0}^{r} \left(\Delta_{(j+1)} S\mathbf{x}\right) \cdot \left(y^0 \mathbf{x}^T B_{0,j} \mathbf{x} + \cdots + y^{n-1} \mathbf{x}^T B_{n-1,j} \mathbf{x}\right) \\
&= \sum_{j=0}^{r} \left(y^0 \cdot \left(\Delta_{(j+1)} S\mathbf{x}\right) \cdot \left(\mathbf{x}^T B_{0,j} \mathbf{x}\right) + \cdots + y^{n-1} \cdot \left(\Delta_{(j+1)} S\mathbf{x}\right) \cdot \left(\mathbf{x}^T B_{n-1,j} \mathbf{x}\right)\right).
\end{aligned}$$

We now apply ϕ to the previous expression, which yields

$$\begin{aligned}
\phi\left(\mathcal{H}\left(\phi^{-1}(S\mathbf{x})\right)\right) &= \phi\left(\sum_{j=0}^{r} \left[y^0 \cdot \left(\Delta_{(j+1)} S\mathbf{x}\right) \cdot \underbrace{\left(\mathbf{x}^T B_{0,j} \mathbf{x}\right)}_{\in \mathbb{F}} + \cdots + y^{n-1} \cdot \left(\Delta_{(j+1)} S\mathbf{x}\right) \cdot \underbrace{\left(\mathbf{x}^T B_{n-1,j} \mathbf{x}\right)}_{\in \mathbb{F}} \right]\right) \\
&= \sum_{j=0}^{r} \left[\left(\mathbf{x}^T B_{0,j} \mathbf{x}\right) \cdot \phi\left(y^0 \left(\Delta_{(j+1)} S\mathbf{x}\right)\right) + \cdots + \left(\mathbf{x}^T B_{n-1,j} \mathbf{x}\right) \cdot \phi\left(y^{n-1} \left(\Delta_{(j+1)} S\mathbf{x}\right)\right) \right] \\
&= \sum_{j=0}^{r} \left[\left(\mathbf{x}^T B_{0,j} \mathbf{x}\right) \cdot C^0 \phi\left(\Delta_{(j+1)} S\mathbf{x}\right) + \cdots + \left(\mathbf{x}^T B_{n-1,j} \mathbf{x}\right) \cdot C^{n-1} \phi\left(\Delta_{(j+1)} S\mathbf{x}\right) \right] \\
&= \sum_{j=0}^{r} \left[\left(\mathbf{x}^T B_{0,j} \mathbf{x}\right) \cdot C^0 + \cdots + \left(\mathbf{x}^T B_{n-1,j} \mathbf{x}\right) \cdot C^{n-1} \right] \phi\left(\Delta_{(j+1)} S\mathbf{x}\right).
\end{aligned}$$

On the other hand, we have

$$\begin{aligned}\phi\left(\Delta_{(j+1)}S\mathbf{x}\right) &= \Delta^{-1}\left((\Delta_{(j+1)}S\mathbf{x})^{q^0}, (\Delta_{(j+1)}S\mathbf{x})^{q^1}, \ldots, (\Delta_{(j+1)}S\mathbf{x})^{q^{n-1}}\right)^T \\ &= \Delta^{-1}\left(\Delta_{(j+1)}S\mathbf{x}, \Delta_{(j+2)}S\mathbf{x}, \ldots, \Delta_{(n)}S\mathbf{x}, \Delta_{(1)}S\mathbf{x}, \ldots, \Delta_{(j)}S\mathbf{x}\right)^T \\ &= \Delta^{-1}\begin{bmatrix}\Delta_{(j+1)}\\ \Delta_{(j+2)}\\ \vdots\\ \Delta_{(n)}\\ \Delta_{(1)}\\ \vdots\\ \Delta_{(j)}\end{bmatrix} S\mathbf{x} = \Delta^{-1}\left[\begin{array}{c|c} 0 & I_{n-j}\\ \hline I_j & 0\end{array}\right]\Delta S\mathbf{x} = \overline{D}_j\mathbf{x}.\end{aligned}$$

so

$$\phi\circ\mathcal{H}\circ\phi^{-1}(S\mathbf{x}) = \sum_{j=0}^{r}\underbrace{\left[\left(\mathbf{x}^T B_{0,j}\mathbf{x}\right)\cdot C^0 + \cdots + \left(\mathbf{x}^T B_{n-1,j}\mathbf{x}\right)\cdot C^{n-1}\right]}_{C_j}\overline{D}_j\mathbf{x}.$$

Finally, we apply T to obtain

$$T\circ\phi\circ\mathcal{H}\circ\phi^{-1}(S\mathbf{x}) = T\sum_{j=0}^{D} C_j\overline{D}_j\mathbf{x}.$$

□

This proposition gives a closed expression of the public key polynomials that arise from $T\circ\phi\circ\mathcal{H}\circ\phi^{-1}\circ S$. However, to find these polynomials, we need to make computations over $\mathbb{F}[x_1,\ldots,x_n]$. To avoid this computation, we represent each one of those n cubic homogeneous polynomials $q_1,\ldots,q_n$ (following the same idea of the representation of $\mathcal{H}$) as

$$q_i(x_1,...,x_n) = \sum_{j=1}^{n} x_i\left(\mathbf{x}^T Q_{i,j}\mathbf{x}\right)$$

where each $Q_{i,j}$ is a $n\times n$ matrix over $\mathbb{F}$ and we give closed formulas for these matrices, which will be useful for the implementation.

Proposition 5.1.3. *Let $q_1,...,q_n$ be the cubic homogeneous polynomials of the composition $T\circ\phi\circ\mathcal{H}\circ\phi^{-1}\circ S$, then, for all s we have*

$$q_s\left(\mathbf{x}\right) = \sum_{t=1}^{n} x_t\left(\mathbf{x}^T Q_{s,t}\mathbf{x}\right)$$

where

$$Q_{s,t} = \sum_{j=0}^{r}\sum_{i=0}^{n-1} B_{i,j}\left[T\cdot C^i\cdot\overline{D}_j\right]_{(s)}^{(t)}.$$

Proof. Let

$$\mathcal{A} := T \sum_{j=0}^{D} C_j \overline{D}_j \in \mathcal{M}_{n\times n}\left(\mathbb{F}[x_1, \ldots, x_n]\right),$$

then by the previous proposition we have that

$$\mathcal{A}_{(s)}^{(t)} = \mathbf{x}^T Q_{s,t} \mathbf{x},$$

so getting a hold of each $\mathcal{A}_{(s)}^{(t)}$ will give us the matrices $Q_{s,t}$.

From its definition, it can be easily seen that the entry s, t of the matrix C_j is given by

$$\mathbf{x}^T \left(\sum_{i=0}^{n-1} B_{i,j} \left(C^i\right)_{(s)}^{(t)} \right) \mathbf{x},$$

so

$$\begin{aligned}
\mathcal{A} &= T \sum_{j=0}^{r} C_j \overline{D}_j \\
&= T \sum_{j=0}^{r} \left[\mathbf{x}^T \left(\sum_{i=0}^{n-1} B_{i,j} \left(C^i\right)_{(s)}^{(t)} \right) \mathbf{x} \right]_{st} \left[\left(\overline{D}_j\right)_{(s)}^{(t)} \right]_{st} \\
&= T \sum_{j=0}^{r} \left[\sum_{\ell=1}^{n} \left(\mathbf{x}^T \left(\sum_{i=0}^{n-1} B_{i,j} \left(C^i\right)_{(s)}^{(\ell)} \right) \mathbf{x} \right) \left(\overline{D}_j\right)_{(\ell)}^{(t)} \right]_{st} \\
&= T \sum_{j=0}^{r} \left[\mathbf{x}^T \left(\sum_{\ell=1}^{n} \sum_{i=0}^{n-1} B_{i,j} \left(C^i\right)_{(s)}^{(\ell)} \left(\overline{D}_j\right)_{(\ell)}^{(t)} \right) \mathbf{x} \right]_{st} \\
&= T \sum_{j=0}^{r} \left[\mathbf{x}^T \left(\sum_{i=0}^{n-1} B_{i,j} \sum_{\ell=1}^{n} \left(C^i\right)_{(s)}^{(\ell)} \left(\overline{D}_j\right)_{(\ell)}^{(t)} \right) \mathbf{x} \right]_{st} \\
&= T \sum_{j=0}^{r} \left[\mathbf{x}^T \left(\sum_{i=0}^{n-1} B_{i,j} \left(C^i \cdot \overline{D}_j\right)_{(s)}^{(t)} \right) \mathbf{x} \right]_{st} \\
&= \left[T_{(s)}^{(t)}\right]_{st} \left[\mathbf{x}^T \left(\sum_{j=0}^{r} \sum_{i=0}^{n-1} B_{i,j} \left(C^i \cdot \overline{D}_j\right)_{(s)}^{(t)} \right) \mathbf{x} \right]_{st} \\
&= \left[\sum_{\ell=1}^{n} T_{(s)}^{(\ell)} \mathbf{x}^T \left(\sum_{j=0}^{r} \sum_{i=0}^{n-1} B_{i,j} \left(C^i \cdot \overline{D}_j\right)_{(\ell)}^{(t)} \right) \mathbf{x} \right]_{st} \\
&= \left[\mathbf{x}^T \left(\sum_{j=0}^{r} \sum_{i=0}^{n-1} B_{i,j} \sum_{\ell=1}^{n} T_{(s)}^{(\ell)} \left(C^i \cdot \overline{D}_j\right)_{(\ell)}^{(t)} \right) \mathbf{x} \right]_{st} \\
&= \left[\mathbf{x}^T \left(\sum_{j=0}^{r} \sum_{i=0}^{n-1} B_{i,j} \left(T \cdot C^i \cdot \overline{D}_j\right)_{(s)}^{(t)} \right) \mathbf{x} \right]_{st}
\end{aligned}$$

and this concludes the result. □

For the implementation, we will give an alternative form of the matrices $Q_{s,t}$.

Proposition 5.1.4. *The s'-th row of each matrix $Q_{s,t}$ is given by*

$$(Q_{s,t})_{(s')} = \sum_{j=0}^{D} \left[(T \cdot C^0 \cdot \overline{D}_j)_{(s)}^{(t)}, \cdots, (T \cdot C^{n-1} \cdot \overline{D}_j)_{(s)}^{(t)} \right]_{1\times n} \left[\phi\left((\overline{A}_j)_{(s')}^{(1)} \right), \cdots, \phi\left((\overline{A}_j)_{(s')}^{(n)} \right) \right]_{n\times n}$$

Proof. This is basically a consequence of noticing that by construction

$$\left((B_{0,j})_{(s')}^{(t')}, \cdots, (B_{n-1,j})_{(s')}^{(t')} \right)^T = \phi\left((\overline{A}_j)_{(s')}^{(t')} \right)$$

□

5.1.3 Performance

What we have seen in the previous section makes feasible the idea of using cubic droppings. In table 6.7 in section 6.4.2 we can see the timings for the key generation process using this idea given the secret key, along with encryption and decryption times for several sets of parameters.

5.1.4 Security analysis

Some of the motivations for using cubic polynomials instead of quadratic is that here we have more "freedom" to produce polynomials. For instance, we can multiply linear and quadratic polynomials together, among several other interesting operations. Another observation is that the MinRank attack on HFE and ZHFE works because we have a very clear way of representing quadratic forms, which has very nice algebraic properties; this is not the case with cubic forms, where there is not a clear way of handling them.

Like in HFE and ZHFE we have in our first attempt a small parameter r, which may look like trouble. We were confident at the beginning with the lack of a way of representing cubic forms to make use of this small value, however, by representing the cubic forms in certain way, one can develop an attack that recover the secret information.

Consider the following way of representing the weight 3 polynomial $\mathcal{H}$

$$\mathcal{H}(X) = \underbrace{\left[X^{q^0}, \ldots, X^{q^{n-1}} \right]}_{\mathbf{x}^T} A \underbrace{\begin{bmatrix} X^{q^0}X^{q^0} \\ X^{q^0}X^{q^1} \\ \vdots \\ X^{q^0}X^{q^{n-1}} \\ \hline X^{q^1}X^{q^0} \\ \vdots \\ X^{q^1}X^{q^{n-1}} \\ \hline \vdots \\ \hline X^{q^{n-1}}X^{q^0} \\ \vdots \\ X^{q^{n-1}}X^{q^{n-1}} \end{bmatrix}}_{\widetilde{\mathbf{x}}} \tag{5.3}$$

where A is a $n \times n^2$ matrix over $\mathbb{K}$. From equation (5.1), since the maximum degree of the q-weight 3 polynomial $\mathcal{G}(X)$ is $q^r + q^r + q^r$, we have that $\mathcal{H}(X)$ has no monomials of the form $X^{q^i+q^j+q^k}$ with at least two of i, j, k greater than r. According to this, one of the possible A's representing $\mathcal{H}$ is such that all but its first $r+1$ rows are zero. In particular, A has a low rank of at most $r+1$. We can apply the same sort of representation to each cubic form of the public key. Surprisingly, as in the quadratic case, the low rank is inherited by these matrices and unfortunately this yields an attack that allows us to recover S, T and $\mathcal{H}$.

Step 1: Representing the polynomials from P

In this section we will encounter several matrices with n^2 rows or columns. When this is the case, these rows or columns will be indexed by tuples

$$(1,1), \ldots, (1,n), (2,1), \ldots, (2,n), \ldots, (n,1), \ldots, (n,n).$$

Considering that $\mathbf{x}^T \Delta_{(i)}^T = \Delta_{(i)}\mathbf{x} = X^{q^{i-1}}$ and $\Delta\mathbf{x} = \mathbf{X}$ where $\phi(X) = \mathbf{x}$, we get from (5.3) that

$$\mathcal{H}\left(\phi^{-1}(\mathbf{x})\right) = (\Delta\mathbf{x})^T A \begin{bmatrix} \mathbf{x}^T\Delta_{(1)}^T\Delta_{(1)}\mathbf{x} \\ \mathbf{x}^T\Delta_{(1)}^T\Delta_{(2)}\mathbf{x} \\ \vdots \\ \mathbf{x}^T\Delta_{(1)}^T\Delta_{(n)}\mathbf{x} \\ \hline \mathbf{x}^T\Delta_{(2)}^T\Delta_{(1)}\mathbf{x} \\ \vdots \\ \mathbf{x}^T\Delta_{(2)}^T\Delta_{(n)}\mathbf{x} \\ \hline \vdots \\ \hline \mathbf{x}^T\Delta_{(n)}^T\Delta_{(1)}\mathbf{x} \\ \vdots \\ \mathbf{x}^T\Delta_{(n)}^T\Delta_{(n)}\mathbf{x} \end{bmatrix} = \mathbf{x}^T\Delta^T A \cdot \begin{bmatrix} \mathbf{x}^T\Delta_{(1)}^T\Delta_{(1)}\mathbf{x} \\ \mathbf{x}^T\Delta_{(1)}^T\Delta_{(2)}\mathbf{x} \\ \vdots \\ \mathbf{x}^T\Delta_{(1)}^T\Delta_{(n)}\mathbf{x} \\ \hline \mathbf{x}^T\Delta_{(2)}^T\Delta_{(1)}\mathbf{x} \\ \vdots \\ \mathbf{x}^T\Delta_{(2)}^T\Delta_{(n)}\mathbf{x} \\ \hline \vdots \\ \hline \mathbf{x}^T\Delta_{(n)}^T\Delta_{(1)}\mathbf{x} \\ \vdots \\ \mathbf{x}^T\Delta_{(n)}^T\Delta_{(n)}\mathbf{x} \end{bmatrix}.$$

Let $\widetilde{\Delta}$ be the $n^2 \times n^2$ matrix whose $(i,j),(k,\ell)$ entry equals $\left(\Delta_{(i)}^T\Delta_{(j)}\right)_{(k)}^{(\ell)} = \Delta_{(i)}^{(k)} \cdot \Delta_{(j)}^{(\ell)}$ so that $\mathbf{x}^T\Delta_{(i)}^T\Delta_{(j)}\mathbf{x} = \widetilde{\Delta}_{(i,j)}\widetilde{\mathbf{x}}$ where

$$\widetilde{\mathbf{x}} = \left[x_1x_1 \cdots x_1x_n \mid x_2x_1 \cdots x_2x_n \mid \cdots \mid x_nx_1 \cdots x_nx_n\right]^T,$$

then

$$\mathcal{H}\left(\phi^{-1}(\mathbf{x})\right) = \mathbf{x}^T\Delta^T A\widetilde{\Delta}\widetilde{\mathbf{x}}.$$

Now let $\widetilde{S}$ be the $n^2 \times n^2$ matrix whose $(i,j),(k,\ell)$ entry equals $\left(S_{(i)}^T S_{(j)}\right)_{(k)}^{(\ell)}$, we can verify then that $\widetilde{(S\mathbf{x})} = \widetilde{S}\widetilde{\mathbf{x}}$ and therefore

$$\mathcal{H}\left(\phi^{-1}(S\mathbf{x})\right) = (S\mathbf{x})^T\Delta^T A\widetilde{\Delta}\widetilde{S}\widetilde{\mathbf{x}} = \mathbf{x}^T S^T\Delta^T A\widetilde{\Delta}\widetilde{S}\widetilde{\mathbf{x}}.$$

Since A has a rank of at most $r+1$, it is very likely that the matrix $S^T\Delta^T A\widetilde{\Delta}\widetilde{S}$ will have this rank too. Now write the matrix $\Delta^T A\widetilde{\Delta}$ as $y^0R_1+\cdots+y^{n-1}R_n$ where each R_i's has its entries in $\mathbb{F}$, then

$$\begin{aligned}\mathcal{H}\left(\phi^{-1}(S\mathbf{x})\right) &= \mathbf{x}^T S^T\Delta^T A\widetilde{\Delta}\widetilde{S}\widetilde{\mathbf{x}}\\ &= \mathbf{x}^T S^T\left(y^0R_1+\cdots+y^{n-1}R_n\right)\widetilde{S}\widetilde{\mathbf{x}}\\ &= y^0\left(\mathbf{x}^T S^T R_1\widetilde{S}\widetilde{\mathbf{x}}\right)+\cdots+y^{n-1}\left(\mathbf{x}^T S^T R_n\widetilde{S}\widetilde{\mathbf{x}}\right),\end{aligned}$$

since each matrix $S^T R_1\widetilde{S}$ has its entries in $\mathbb{F}$, we conclude that

$$\phi\circ\mathcal{H}\circ\phi^{-1}\circ S\left(\mathbf{x}\right)=\left(\mathbf{x}^T S^T R_1\widetilde{S}\widetilde{\mathbf{x}},\ldots,\mathbf{x}^T S^T R_n\widetilde{S}\widetilde{\mathbf{x}}\right)^T,$$

and hence, after we apply T we get

$$T\circ\phi\circ\mathcal{H}\circ\phi^{-1}\circ S\left(\mathbf{x}\right)=\left(\sum_{j=1}^{n}T_{(1)}^{(j)}\mathbf{x}^T S^T R_j\widetilde{S}\widetilde{\mathbf{x}},\ldots,\sum_{j=1}^{n}T_{(n)}^{(j)}\mathbf{x}^T S^T R_j\widetilde{S}\widetilde{\mathbf{x}}\right)^T,$$

For each i, let $\mathcal{Q}_i\in\mathcal{M}_{n\times n^2}\left(\mathbb{F}\right)$ be a matrix that represents the $i-$th component of the composition $T\circ\phi\circ\mathcal{H}\circ\phi^{-1}\circ S$ in a similar manner as (5.3), that is, the $i-$th component of this composition is given by $\mathbf{x}^T\mathcal{Q}_i\widetilde{\mathbf{x}}$. The previous expression shows that these matrices can be taken as

$$\mathcal{Q}_i=\sum_{j=1}^{n}T_{(i)}^{(j)}S^T R_j\widetilde{S}.$$

Notice that the $\mathcal{Q}_i$'s are public.

On the other hand, if $z_1,\ldots,z_n\in\mathbb{K}$ satisfy $T^T\cdot\left(z_1,\ldots,z_n\right)^T=\left(y^0,\ldots,y^{n-1}\right)^T$, then it can be seen that

$$z_1\mathcal{Q}_1+\cdots+z_n\mathcal{Q}_n=S^T\left(\sum_{j=1}^{n}y^{j-1}R_j\right)\widetilde{S}=S^T\Delta^T A\widetilde{\Delta}\widetilde{S},\tag{5.4}$$

which is a linear combination of the $\mathcal{Q}_i$'s of rank at most $r+1$.

Step 2: Recovering T

From the previous step we know that there exists a linear combination of the public matrices $\mathcal{Q}_i$'s of rank at most $r+1$. Using the KS modeling [KS99], we can compute at least one of these combinations. Assume that we get exactly the combination shown in (5.4), then we know that $T^T\cdot\left(z_1,\ldots,z_n\right)^T=\left(y^0,\ldots,y^{n-1}\right)^T$ and therefore for each row j of T^T we have the following equation

$$T_{(1)}^{(j)}z_1+\cdots+T_{(n)}^{(j)}z_n=y^{j-1}.$$

By iteratively raising this equation to the qth power and recalling that the entries of T lie on $\mathbb{F}$, we get the following n linear equations

$$\begin{cases} T_{(1)}^{(j)} z_1 + \cdots + T_{(n)}^{(j)} z_n & = & y^{j-1} \\ T_{(1)}^{(j)} z_1^q + \cdots + T_{(n)}^{(j)} z_n^q & = & y^{q(j-1)} \\ \vdots & & \vdots \\ T_{(1)}^{(j)} z_1^{q^{n-1}} + \cdots + T_{(n)}^{(j)} z_n^{q^{n-1}} & = & y^{q^{n-1}(j-1)} \end{cases}$$

that we can use to efficiently find the j-th column of T. We proceed in a similar fashion to obtain T itself.

Step 3: Recovering S and $\mathcal{H}$

From (5.4), assuming again that we get exactly this combination, we have knowledge of the matrix $(\Delta S)^T A \widetilde{\Delta} \widetilde{S}$. We will use this matrix to recover S.

Let M denote the matrix ΔS, we will write the product $\widetilde{\Delta}\widetilde{S}$ in terms of this matrix. The entry in position $(k,\ell),(s,t)$ of $\widetilde{\Delta}\widetilde{S}$ is given by

$$\left(\widetilde{\Delta}\widetilde{S}\right)_{(k,\ell)}^{(s,t)} = \sum_{i=1}^{n}\sum_{j=1}^{n} \left(\Delta_{(k)}^T \Delta_{(\ell)}\right)_{(i)}^{(j)} \cdot \left(S_{(i)}^T S_{(j)}\right)_{(s)}^{(t)},$$

but considering that $\left(S_{(i)}^T S_{(j)}\right)_{(s)}^{(t)} = S_{(i)}^{(s)} \cdot S_{(j)}^{(t)}$ and $\left(\Delta_{(k)}^T \Delta_{(\ell)}\right)_{(i)}^{(j)} = \Delta_{(k)}^{(i)} \cdot \Delta_{(\ell)}^{(j)}$, we can write this as

$$\left(\widetilde{\Delta}\widetilde{S}\right)_{(k,\ell)}^{(s,t)} = \left(\sum_{i=1}^{n} \Delta_{(k)}^{(i)} \cdot S_{(i)}^{(s)}\right) \cdot \left(\sum_{j=1}^{n} \Delta_{(\ell)}^{(j)} \cdot S_{(j)}^{(t)}\right) = (\Delta S)_{(k)}^{(s)} \cdot (\Delta S)_{(\ell)}^{(t)} = \left(M_{(k)}^T M_{(\ell)}\right)_{(s)}^{(t)}.$$

It makes sense then to denote $\widetilde{\Delta}\widetilde{S}$ by $\widetilde{M}$, and the main observation is that it can be computed once we get M.

Let R denote the (known) matrix $(\Delta S)^T A \widetilde{\Delta}\widetilde{S} = M^T A \widetilde{M}$, hence $A\widetilde{M} = WR$ where $W = \left(M^T\right)^{-1}$. Due to the fact that only the first $r+1$ rows of A are nonzero, the matrix $A\widetilde{M}$ shares this shape and therefore so does the matrix WR. This allows us to find the rows $W_{(j)}$ with $j > r+1$ as follows: for each of these rows consider the n^2 linear equations

$$W_{(j)}^{(1)} R_{(1)}^{(s,t)} + \cdots + W_{(j)}^{(n)} R_{(n)}^{(s,t)} = \left(A\widetilde{M}\right)_{(j)}^{(s,t)} = 0$$

from which we can obtain $W_{(j)}$.

On the other hand, since $M = \Delta S$ it can be seen that M has the shape

$$M = \begin{bmatrix} v_1 & v_2 & \cdots & v_{n-1} & v_n \\ (v_1)^{q^1} & (v_2)^{q^1} & \cdots & (v_{n-1})^{q^1} & (v_n)^{q^1} \\ (v_1)^{q^2} & (v_2)^{q^2} & \cdots & (v_{n-1})^{q^2} & (v_n)^{q^2} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ (v_1)^{q^{n-1}} & (v_2)^{q^{n-1}} & \cdots & (v_{n-1})^{q^{n-1}} & (v_n)^{q^{n-1}} \end{bmatrix}$$

and therefore it suffices to find its first row to find M itself. Since $WM^T = \mathrm{Id}$ and given that we know the last $n-(D+1)$ rows W, we get enough weight 1 equations on the v_i's (that can be turned into linear equations by raising to adequate q-th powers) that should allow us to find M.

Once we get M, we compute $\widetilde{M}$ and recover A by computing $A = \left(M^T\right)^{-1} R \left(\widetilde{M}\right)^{-1}$. S can be recovered as $S = \Delta^{-1} M$.

Clearly, by knowing S, T and the public trapdoor function one can recover $\mathcal{H}$.

Remarks about the Attack

This attack has been developed only in theory and it has not been implemented yet. There are some concerns to be considered

- As we saw in equation (5.3), one of the possible $n \times n^2$ matrices representing $\mathcal{H}$ has low rank of at most $r+1$. However, unlike the quadratic case, we do not know yet if any other representation of the same form will have the same rank (this is not substantial for the attack though, since we only need a low rank combination to exist, and this shows existence).

- One can show that for any solution $(z_1, \ldots, z_n)$ to the MinRank problem in (5.4), we get $n-1$ solutions corresponding to the Frobenius powers $(z_1^{q^i}, \ldots, z_n^{q^i})$ for $i = 0, \ldots, n-1$. MinRank attack on HFE and ZHFE only assume that one of these solutions is found, while here we use the fact that we find exactly

$$(z_1^{q^i}, \ldots, z_n^{q^i}) = \left(y^0, \ldots, y^{n-1}\right) T^{-1}.$$

  The main issue with this assumption is that an attacker does not know a priori whether or not he got the right scalars, however, we expect our attack to be efficient enough so that it can be run using all solutions to the MinRank problem, our experiments show that we only get n of these (corresponding to the n Frobenius powers of the solution described in the attack).

- We do not know yet the viability of finding S at the end of the attack. In any case, at that point we have already recovered T, which imposes a vulnerability. Moreover, the fact that a low rank linear combination exists makes the trapdoor function distinguishable from random, which an issue from the security point of view (for instance, we can not ensue that its behavior against Groebner bases and other attacks are like that for random systems).

- Suppose in the worst case that an attacker is able to recover S, T and $\mathcal{H}$. Since he has knowledge of $\mathcal{F}$, he will be able to find $\mathcal{G}$ and the $\mathcal{M}_j$'s from equation (5.1), therefore inverting the trapdoor function. An alternative to avoid this consequence is to hide $\mathcal{F}$ using a different transformation T'. We do not know yet the level of security of this approach.

5.2 "Non-noisy" Version

Due to this attack, a reformulation that allowed r to be larger became necessary. We now present a variant of the first idea that allows r to be as big as we want (we take it as $n-1$). Unfortunately, this variant can be broken with a direct algebraic attack, which we describe in full detail.

5.2.1 Description

The idea here is to consider exactly the same construction than the previous approach, but fixing $\mathcal{G}(X)$ to be the zero polynomial (that is, removing the *noise*). More precisely, pick completely at random a weight 2 polynomial $\mathcal{F} \in \mathbb{K}[X]$ and for each $j = 0, \ldots, n-1$, a weight 1 polynomial $\mathcal{M}_j \in \mathbb{K}[X]$. Naturally, choose two invertible linear transformations $S, T : \mathbb{F}^n \to \mathbb{F}^n$. Finally, we consider the weight 3 polynomial $\mathcal{H} : \mathbb{K} \to \mathbb{K}$ given by

$$\mathcal{H}(X) = \sum_{j=0}^{n-1} X^{q^j} \mathcal{M}_j\left(\mathcal{F}'(X)\right) \tag{5.5}$$

where $\mathcal{F}' = \mathcal{F} \circ \phi^{-1} \circ S^{-1} \circ \phi$. The trapdoor function is then $P : \mathbb{F}^n \to \mathbb{F}^{2n}$ given by

$$P = (\phi \circ \mathcal{F} \circ \phi^{-1}, T \circ \phi \circ \mathcal{H} \circ \phi^{-1} \circ S),$$

while the secret information is $(\mathcal{F}, \mathcal{M}_i, \mathcal{H}, S, T)$.

One can easily see that we can not invert P as we did before since the resulting polynomial will have high degree, the approach here is a bit different. Given $\mathbf{c} = (c_1, \ldots, c_{2n})$ in the range of P, we want to solve the simultaneous equations $\mathcal{F}\left(\phi^{-1}(\mathbf{x})\right) = Z_1$ and $\mathcal{H}\left(\phi^{-1}(S\mathbf{x})\right) = Z_2$ where $Z_1 = \phi^{-1}(c_1, \ldots, c_n)$ and $Z_2 = \phi^{-1} \circ T^{-1}(c_{n+1}, \ldots, c_{2n})$. This is the same as $\mathcal{F}'\left(\phi^{-1}(S\mathbf{x})\right) = Z_1$ and $\mathcal{H}\left(\phi^{-1}(S\mathbf{x})\right) = Z_2$, and any solution to this system will also satisfy

$$Z_2 = \mathcal{L}\left(\phi^{-1}(S\mathbf{x})\right)$$

where $\mathcal{L}(X) = \sum_{j=0}^{n-1} X^{q^j} M_j\left(Z_1\right)$. Since $\mathcal{L}$ has weight 1 and using theorem 4.2.1, by applying ψ to both sides of the previous expression this is the same as solving the linear system over $\mathbb{F}$ given by

$$\phi(Z_2) = \phi\left(\mathcal{L}\left(\phi^{-1}(S\mathbf{x})\right)\right).$$

5.2.2 Security analysis

The first thing we must notice is that the MinRank attack presented before will not work with this scheme. This is basically due to the fact that the matrices representing $\mathcal{H}$ this time have a large rank, as D can be as large as we want. However, this second approach is susceptible to a Direct Algebraic Attack.

More precisely, if $(f_1, \ldots, f_n, h_1, \ldots, h_n)$ is a trapdoor function constructed as in the previous section and if $(c_1, \ldots, c_{2n})$ is in the range of this function, then the polynomial system $(f_1 - c_1, \ldots, f_n - c_n, h_1 - c_{n+1}, \ldots, h_n - c_{2n})$ has a fixed falling degree independent of n (which is not a natural behavior with random systems). Moreover, this falling

degree is low enough for this system to be efficiently solved. We focus our efforts in showing why is this the case.

Correspondence of polynomials (theorem 4.2.1) shows that the concept of *weight* on $\mathbb{K}[X]$ is analogous to the degree in $\mathbb{F}[x_1, \ldots, x_n]$. By imposing the relation $X^{q^{n-1}} = X$, in analogy to section 3.6 it is natural to make the following definition.

Definition. A *degree fall* in degree d of a tuple $(\mathcal{F}_1, \ldots, \mathcal{F}_m)$ of weight 2 polynomials in $\mathbb{K}[X]$ is a tuple $(\mathcal{H}_1, \ldots, \mathcal{H}_m) \in (\mathbb{K}[X]_{d-2})^m$ such that $\mathcal{H}_1\mathcal{F}_1 + \cdots + \mathcal{H}_m\mathcal{F}_m$ has a weight strictly smaller than d.

We can define trivial degree falls for polynomials in $\mathbb{K}[X]$ in a completely analogous way as trivial degree falls were defined in section 3.6. It is natural to regard the *falling degree* of $(\mathcal{F}_1, \ldots, \mathcal{F}_m)$ as the smallest d such that a non-trivial degree fall of $\mathcal{F}_1, \ldots, \mathcal{F}_m$ exists in degree d. As in section 3.6, these ideas extend naturally to non-quadratic polynomials.

The importance of extending this concept to polynomials in $\mathbb{K}[X]$ is that the falling degree is preserved under the correspondence of polynomials, according to the following proposition.

Proposition 5.2.1. *Non-trivial degree falls on $\mathcal{P}, \mathcal{Q} \in \mathbb{K}[X]$ are in correspondence with non-trivial degree falls on $(\phi \circ \mathcal{P} \circ \phi^{-1}, \phi \circ \mathcal{Q} \circ \phi^{-1})$*

This important property can be found in [DG10] as *property 5*. We now show that our polynomials $\mathcal{F}'(X)$ and $\mathcal{H}(X)$ have a low falling degree.

Theorem 5.2.2. *Let $(Y_0, Y_1) \in \mathcal{G}(\mathbb{K})$ where $\mathcal{G}(X) = (\mathcal{F}'(X), \mathcal{H}(X))$, then $(\mathcal{F}'(X) - Y_0, \mathcal{H}(X) - Y_1)$ has a degree fall at degree 3*

Proof. Let $\Psi(X) := \sum X^{q^j} \mathcal{M}_j(Y_0)$, then according to equation (5.5) for all X with $\mathcal{F}(X) = Y_0$ it holds that $\mathcal{H}(X) = \Psi(X)$, hence $\mathcal{F}(X) - Y_0$ divides $\mathcal{H}(X) - \Psi(X)$ and therefore there exists $\mathcal{P}(X) \in \mathbb{K}[X]$ such that $\mathcal{H}(X) - \Psi(X) = \mathcal{P}(X)(\mathcal{F}(X) - Y_0)$. It follows that

$$(\mathcal{H}(X) - Y_1) - \mathcal{P}(X)(\mathcal{F}(X) - Y_0) = \Psi(X) - Y_1.$$

□

Of course, this degree fall is not expected to be trivial. Since the falling degree remains unchanged under linear changes of variables and $\mathcal{F} = \mathcal{F}' \circ \phi^{-1} \circ S \circ \phi$, the previous results show that the falling degree of the system

$$P = (\phi \circ \mathcal{F} \circ \phi^{-1}, T \circ \phi \circ \mathcal{H} \circ \phi^{-1} \circ S)$$

(subtracting the corresponding ciphertext) is at most 3, moreover, equality is expected, and it is the case according to our experiments in the appendix. In particular, the falling degree does not grow with n as expected from theorem 3.2.2. This imposes a weakness by itself on the cryptosystem since its security does not grow along with n.

Moreover, 3 is a very low falling degree that allows an attacker to invert the public key directly. In section 6.4.2 in the appendix we can find the experimental evidence that this attack is completely efficient.

5.3 Two-Layer Construction

To overcome this issue we present here an alternative that, expectedly, can raise the falling degree according to a predefined parameter d. The idea is inspired in the ABC Cryptosystem developed by Jintai Ding [Din12].

5.3.1 Description

Let n be a square number, say $n = s^2$, consider a trapdoor function

$$(\phi \circ \mathcal{F} \circ \phi^{-1}, T \circ \phi \circ \mathcal{H} \circ \phi^{-1} \circ S)$$

built as in the previous section and let $\mathbf{f} = (f_1, \ldots, f_n) = \phi \circ \mathcal{F} \circ \phi^{-1}$. Choose $2n$ linear transformations $A_1, \ldots, A_n, C_1 \ldots, C_n : \mathbb{F}^n \to \mathbb{F}$ uniformly at random and consider the matrices $A, C \in \mathcal{M}_{s\times s}(R)$ given by

$$A = \begin{bmatrix} A_1(\mathbf{f}) & A_2(\mathbf{f}) & \cdots & A_s(\mathbf{f}) \\ A_{s+1}(\mathbf{f}) & A_{s+2}(\mathbf{f}) & \cdots & A_{2s}(\mathbf{f}) \\ \vdots & \vdots & \ddots & \vdots \\ A_{n-s+1}(\mathbf{f}) & A_{n-s+2}(\mathbf{f}) & \cdots & A_n(\mathbf{f}) \end{bmatrix},$$

$$C = \begin{bmatrix} C_1(\mathbf{f}) & C_2(\mathbf{f}) & \cdots & C_s(\mathbf{f}) \\ C_{s+1}(\mathbf{f}) & C_{s+2}(\mathbf{f}) & \cdots & C_{2s}(\mathbf{f}) \\ \vdots & \vdots & \ddots & \vdots \\ C_{n-s+1}(\mathbf{f}) & C_{n-s+2}(\mathbf{f}) & \cdots & C_n(\mathbf{f}) \end{bmatrix}.$$

On the other hand, choose $b_1, \ldots, b_n \in R$ monomials of degree d and consider the matrix $B \in \mathcal{M}_{s\times s}(R)$ given by

$$B = \begin{bmatrix} b_1 & b_2 & \cdots & b_s \\ b_{s+1} & b_{s+2} & \cdots & b_{2s} \\ \vdots & \vdots & \ddots & \vdots \\ b_{n-s+1} & b_{n-s+2} & \cdots & b_n \end{bmatrix}.$$

Finally, construct the matrices $E_1, E_2 \in \mathcal{M}_{s\times s}(R)$ given by $E_1 = BC$ and $E_2 = AB$. Note that each entry f of E_1 or E_2 is a polynomial of degree $d + 2$, but it is very *sparse* (few monomials are required to represent it) due to the following: f has the form $f = \sum_{i=1}^{s} \widetilde{f}_i \cdot m_i$ where each $\widetilde{f}_i$ is quadratic and each m_i is one of the n monomials of degree d in B. Since there are $O(n^2)$ monomials of degree 2, there are $O(n^3)$ monomials of the form $\mathbf{x}^\alpha \cdot b_i$, where $|\alpha| = 2$ and the b_i's are the monomials from B; these are the monomials needed to represent each entry of E_1 and E_2.

Finally, we consider the trapdoor function as

$$P = \left(E_1, E_2, T \circ \phi \circ \mathcal{H} \circ \phi^{-1} \circ S\right),$$

where E_1 and E_2 are not understood as matrices but rather the n polynomials that constitute them.

To invert the trapdoor function, suppose that we are given $Y_1 = E_1(\mathbf{a})$, $Y_2 = E_2(\mathbf{a})$ and $Y_3 = T \circ \phi \circ \mathcal{H} \circ \phi^{-1} \circ S(\mathbf{a})$ (where $E_i(\mathbf{a})$ is the matrix of all entries of E_i evaluated at $\mathbf{a}$), we derive $\mathbf{f}(\mathbf{a})$ from Y_1, Y_2 as follows.

Recall that $E_1 = BC$ and $E_2 = AB$, hence $AE_1 = E_2C = ABC$ and after evaluation we have that

$$A(\mathbf{a}) \cdot Y_1 = Y_2 \cdot C(\mathbf{a}). \tag{5.6}$$

Let y_i denote the unknown value $f_i(\mathbf{a})$, then, if $\mathbf{y} = (y_1, \ldots, y_n)$, what we have is that

$$\begin{bmatrix} A_1(\mathbf{y}) & A_2(\mathbf{y}) & \cdots & A_s(\mathbf{y}) \\ A_{s+1}(\mathbf{y}) & A_{s+2}(\mathbf{y}) & \cdots & A_{2s}(\mathbf{y}) \\ \vdots & \vdots & \ddots & \vdots \\ A_{n-s+1}(\mathbf{y}) & A_{n-s+2}(\mathbf{y}) & \cdots & A_n(\mathbf{y}) \end{bmatrix} \cdot Y_1 = Y_2 \cdot \begin{bmatrix} C_1(\mathbf{y}) & C_2(\mathbf{y}) & \cdots & C_s(\mathbf{y}) \\ C_{s+1}(\mathbf{y}) & C_{s+2}(\mathbf{y}) & \cdots & C_{2s}(\mathbf{y}) \\ \vdots & \vdots & \ddots & \vdots \\ C_{n-s+1}(\mathbf{y}) & C_{n-s+2}(\mathbf{y}) & \cdots & C_n(\mathbf{y}) \end{bmatrix}$$

Since Y_1, Y_2 are known, this yields n linear equations in the variables $y_1, \ldots, y_n$ which we can solve, hence obtaining the values $f_1(\mathbf{a}), \ldots, f_n(\mathbf{a})$. At this point, we can obtain $\mathbf{a}$ just like we did in section 5.2.1.

We call this construction *Two-Layer* since it consists of two stages, being the first one the proposal on section 5.2 and the second the construction we have just seen in this section.

5.3.2 Security Analysis

Direct Algebraic Attack

It is not difficult to see from equation (5.6) that $(E_1 - Y_1, E_2 - Y_2)$ has a falling degree of at least $d + 2$: we have that

$$\begin{aligned} A \cdot (E_1 - Y_1) - (E_2 - Y_2) \cdot C & \\ = (A \cdot E_1 - E_2 \cdot C) - (A \cdot Y_1 - Y_2 \cdot C) & \\ = A(\mathbf{x}) \cdot Y_1 - Y_2 \cdot C(\mathbf{x}), & \end{aligned} \tag{5.7}$$

which, after performing matrix multiplication, translates into a degree fall in degree $d + 2$. Now, since we have chosen d, we can make it grow along with n (for instance $d = O(n)$ or $d = O(\sqrt{n})$) to increase the falling degree of our system asymptotically in an artificial manner. What is expected is that the complexity of a direct algebraic attack increases. Even though our experiments showed this to be true, we could not conclude that the scheme is secure. In fact, a more "intelligent" algebraic attack can break the scheme, as we now show.

Consider equation (5.7). It shows that there is a polynomial combination of $E_1 - Y_1$ and $E_2 - Y_2$ that yields the degree 2 polynomials in $A(\mathbf{x}) \cdot Y_1 - Y_2 \cdot C(\mathbf{x})$. Moreover, the polynomials in A and C are quadratic so this combination can be found in degree $d + 2$, this means that by running Lazard's algorithm in two steps (which is possible since the Macaulay matrices constructed involve linear and quadratic monomials only) we will find the polynomials in $A(\mathbf{x}) \cdot Y_1 - Y_2 \cdot C(\mathbf{x})$ (these will be the resulting degree 2 polynomials after reduction). This by itself is not a problem since the polynomials are quadratic so this system is difficult to solve by itself. The security issue relies on the fact that each of these polynomials is a linear combination of $(f_1, \ldots, f_n)$ and therefore, by

appending the block of the public key involving $\mathcal{H}$ and subtracting the corresponding ciphertext we see that the resulting system has 3 as falling degree (due to the analysis on the previous scheme and the invariance of the falling degree under linear transformations).

The importance of this scheme is that it shows that we must be careful when analyzing the security of a MPK Cryptosystem by means of its falling degree: it can be as large as we wish but the complexity of the algebraic attack will remain constant. The intuition is that we do not make the task of finding a Groebner basis harder by simply raising the degree of the polynomials involved since the Macaulay matrices used only require degree 2 monomials. This gives the suggestion that it is not the degree fall what must be large, but the difference between the degree of the polynomials and the degree fall (that is, the sizes of the Macaulay matrices that need to be constructed).

Other Attacks

Besides the algebraic attack, we contemplate the possibility of two attacks that can apply to this cryptosystem. The first one is the Linearization Equations Attack, considered at first as an attack to the Matsumoto-Imai Cryptosystem [DGS06]. Secondly, we consider what we call a Combinatorial attack, which attempts to exploit the sparsity of the polynomials in the public key to recover the polynomials from A and B.

Linearization Equations Attack. Consider equation (5.6), which holds for any plaintext-ciphertext pair $\mathbf{a}, (Y_1, Y_2)$. If we treat the coefficients of the quadratic polynomials in A and C as variables, each time we have a valid plaintext-ciphertext pair we have a linear equation on these variables arising from equation (5.6). Since we have $2n \times O(n^2) = O(n^3)$ of these variables, having knowledge of $O(n^3)$ valid plaintext-ciphertext pairs will gives us enough equations to find the values of these variables and therefore finding A and C. Once in this position one can find preimages of ciphertexts by performing an algebraic attack identical to that on section 5.2.2.

To avoid this issue, we propose to compose E_1 and E_2 with a secret linear transformation $T' : \mathbb{F}^{2n} \to \mathbb{F}^{2n}$, which hides the matrix multiplicative structure and still allows the legitimate user to obtain Y_1 and Y_2.

Combinatorial Attack. In order to keep the sizes of both public and secret key as low as possible, we propose to make B public and fixed, however, doing so implies that we have to choose carefully which monomials to use.

Recall that the polynomials from BC and AB are part of the public key, it may be the case that some information about the polynomials $f_1, \ldots, f_n$ can be obtained from them. For instance, if the parameter d is at least 3 and we let each entry b_i of the matrix B be equal to x_i^d, then we can factor out each x_i^d on the polynomials from E_1 and E_2 to obtain the polynomials in A and C, which of course would lead to a security issue. This can be regarded as a combinatorial problem, where the objective is to use the structure of the monomials in B along with E_1 and E_2 to obtain secret information.

To avoid this issue, we propose to use square-free monomials (monomials of the form $x_{i_1} \cdots x_{i_d}$ with $i_s \neq i_t$ for $s \neq t$), which clearly complicates the combinatorial

problem. However, we have not studied in detail yet the feasibility of this attack, even under this choice.

5.3.3 Importance of using both Layers

Recall that in this construction the first layer is the Second Attempt described in section 5.2. We saw there that it was not secure to use this directly as a trapdoor function since it possesses a low Falling Degree and therefore it is easy to invert. The second layer takes the first part of this construction: $(f_1, \ldots, f_n)$, and obtain some polynomials (E_1, E_2) having a high Falling Degree that allow us to recover the evaluations $f_1(\mathbf{a}), \ldots, f_n(\mathbf{a})$ from $E_1(\mathbf{a}), E_2(\mathbf{a})$. We notice here that this construction is completely independent from $(f_1, \ldots, f_n)$.

One may ask at this point why not simply choosing $f_i(\mathbf{x}) = x_i$ (which are linear) so that we can recover $\mathbf{a}$ from $E_1(\mathbf{a}), E_2(\mathbf{a})$. By doing this the second layer would be itself a trapdoor function and it makes no sense to use both layers. Unfortunately (or fortunately for this work), extensive experiments we ran show that even though the Falling Degree is high (as expected according to our previous discussion), a Direct Algebraic Attack is completely efficient. We still do not know why is this the case, and why is not the Falling Degree saying anything about the complexity of Lazard's algorithm, but we conjecture that this is due to the fact that the f_i's are linear. We stress that experiments show that this is not an issue if the f_i's are quadratic, which is the case in our Two-Layer construction.

Chapter 6

Appendix

6.1 Preliminaries

6.1.1 Finite Fields and Field Extensions

We begin by considering prime fields. Consider n a natural number and let $\mathbb{Z}_n$ denote the ring of integers modulo n. It is well known that this set is a field when n is a prime p, and in this case we denote $\mathbb{Z}_p$ by $\mathbb{F}_p$ (or $\mathrm{GL}(p)$, making reference to *Galois field of order* p). These fields are known as *Prime Fields*.

A *Finite Field* is simply a field with a finite number of elements. In the following we describe with certain detail how finite fields look like.

Given any field $\mathbb{F}$, we say that a field $\mathbb{K}$ is a field extension of $\mathbb{F}$ if $\mathbb{K} \supseteq \mathbb{F}$. If $\mathbb{K}$ is a field extension of $\mathbb{F}$ then $\mathbb{K}$ is naturally a $\mathbb{F}-$vector space with scalar multiplication given by field multiplication, in this case, the dimension of $\mathbb{K}$ as a $\mathbb{F}-$vector space is known as the *degree* of the extension. Given this, if $\mathbb{K}$ is a field extension of $\mathbb{F}$ of degree n, we have a $\mathbb{F}$-vector spaces isomorphism

$$\phi : \mathbb{K} \to \mathbb{F}^n, \tag{6.1}$$

in particular, if $s < \infty$ is the size of the field $\mathbb{F}$, then $\mathbb{K}$ has s^n elements.

On the other hand, if $\mathbb{F}$ is a finite field it is well known that its characteristic (the smallest $c \in \mathbb{N}$ such that $c \cdot 1 = 0$) must be a prime number q, hence, the field $\mathbb{F}_q = \{1, 2 \cdot 1, \ldots, (q-1) \cdot 1\}$ is contained in $\mathbb{F}$ and in particular $\mathbb{F}$ is an extension field of $\mathbb{F}_q$. Due to our previous observations, we conclude that the number of elements in $\mathbb{F}$ is q^n where q is the characteristic of $\mathbb{F}$ and n is the dimension of $\mathbb{F}$ as a $\mathbb{F}_q-$vector space.

So far we have seen that the number of elements in every finite field is a prime power. During the rest of this work we will only be concerned with finite fields.

Field Extensions

We now study field extensions with more detail. Let $\mathbb{F}$ be a finite field, a polynomial $g(y) \in \mathbb{F}[y]$ is said to be *irreducible* if it can not be factored into the product of two non-constant polynomials. An interesting fact is that, if $g(y)$ is an irreducible polynomial of degree n, the quotient ring $\mathbb{F}/\langle g(y) \rangle$ constitutes a field $\mathbb{K}$ which is a field extension of $\mathbb{F}$

of degree n (identifying $a \in \mathbb{F}$ with $a + \langle g(y) \rangle \in \mathbb{K}$), and even more interesting is the fact that all field extensions have this form (quotient of $\mathbb{F}$ by a irreducible polynomial). If $g(y) = y^n + a_{n-1}y^{n-1} + \cdots + a_1 y^1 + a_0$, since $g(y) = 0$ in $\mathbb{K}$ we can regard elements of this field as polynomials in the variable y having degree at most $n-1$, reducing y^n according to the rule $y^n = -a_{n-1}y^{n-1} - \cdots - a_1 y^1 - a_0$.

Due to this fact, if $\mathbb{K}$ is an extension of $\mathbb{F}$ of this type we can take the isomorphism in equation (6.1) to be

$$b_0 + b_1 y^1 + \cdots + b_{n-1}y^{n-1} \in \mathbb{K} \longmapsto (b_0, b_1, \ldots, b_{n-1}) \in \mathbb{F}^n$$

A very important fact is that any field extension $\mathbb{K}$ of $\mathbb{F}$ has the form mentioned before , so the expression above is the isomorphism ϕ between $\mathbb{K}$ and $\mathbb{F}^n$ for any field extension of degree n of a finite field $\mathbb{F}$.

6.1.2 Frobenius Powers

Let $\mathbb{K}$ be a field extension of $\mathbb{F}$, where $\mathbb{F}$ is a finite field of characteristic q. Recall that every finite group with t elements satisfy $x^t = e$ for all x in the group, where e is the identity of such. If $\mathbb{F}$ is a field, then every nonzero element of $\mathbb{F}$ admits a multiplicative inverse and therefore $\mathbb{F}^* := \mathbb{F} \setminus \{0\}$ is a multiplicative group with identity 1. Since every finite field has q^n elements where q is its characteristic, we conclude that $x^{q^n-1} = 1$ for all $x \in \mathbb{F}^*$, and therefore $x^{q^n} = x$ for all $x \in \mathbb{F}$. In particular, $x^q = x$ for all $x \in \mathbb{F}_q$ (these are the so-called *Field Equations*).

Recall that $\mathbb{F}$ is a field extension of $\mathbb{F}_q$ and therefore a $\mathbb{F}_q$-vector space, the following is a very important proposition.

Proposition 6.1.1. *The function $\mathbb{F} \to \mathbb{F}$ defined by $x \mapsto x^q$ is a $\mathbb{F}_q$-linear transformation, that is, $(ax + z)^q = ax^q + z^q$ for all $a \in \mathbb{F}_q$, $x, z \in \mathbb{F}$.*

This linear transformation is known as a *Frobenius Transformation*, and its importance will become clearer in the next few sections.

Linear Combinations of Frobenius Powers

Consider a field extension $\mathbb{K}$ of $\mathbb{F}$ of degree n. So far we have seen that every element in $\alpha \in \mathbb{K}$ can be written as $\alpha = b_0 + b_1 y^1 + \cdots + b_{n-1}y^{n-1}$, and this defines the bijective $\mathbb{F}$-linear transformation

$$\begin{aligned} \phi \colon \mathbb{K} &\longrightarrow \mathbb{F}^n \\ b_0 + b_1 y^1 + \cdots + b_{n-1}y^{n-1} &\longmapsto (b_0, b_1, \ldots, b_{n-1}). \end{aligned}$$

We know that the Frobenius transformation $X \mapsto X^q$ for $X \in \mathbb{K}$ is a $\mathbb{F}$-linear transformation and therefore so is every polynomial of the form

$$\mathcal{F}(X) = \sum_{i=0}^{n-1} \alpha_i X^{q^i} \tag{6.2}$$

implying that the composition $\phi \circ \mathcal{F} \circ \phi^{-1} : \mathbb{F}^n \to \mathbb{F}^n$ is $\mathbb{F}$-linear as well, that is, it is given by n polynomials, each one homogeneous of degree 1. On the other hand, one can show that if $F : \mathbb{F}^n \to \mathbb{F}^n$ is a linear transformation, then $\mathcal{F}(X) = \phi^{-1} \circ F \circ \phi(X)$ has the shape above.

In fact, let $\alpha = b_0 + b_1 y^1 + \cdots + b_{n-1} y^{n-1} \in \mathbb{K}$, then for each $i = 0, \ldots, n-1$ it is clear that $\alpha^{q^i} = b_0 + b_1 \left(y^1\right)^{q^i} + \cdots + b_{n-1} \left(y^{n-1}\right)^{q^i}$ (since $b_i^q = b_i$), and therefore

$$\begin{bmatrix} \alpha \\ \alpha^q \\ \alpha^{q^2} \\ \vdots \\ \alpha^{q^{n-1}} \end{bmatrix} = \begin{bmatrix} y^0 & y^1 & \cdots & y^{n-2} & y^{n-1} \\ (y^0)^{q^1} & (y^1)^{q^1} & \cdots & (y^{n-2})^{q^1} & (y^{n-1})^{q^1} \\ (y^0)^{q^2} & (y^1)^{q^2} & \cdots & (y^{n-2})^{q^2} & (y^{n-1})^{q^2} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ (y^0)^{q^{n-1}} & (y^1)^{q^{n-1}} & \cdots & (y^{n-2})^{q^{n-1}} & (y^{n-1})^{q^{n-1}} \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ \vdots \\ b_{n-1} \end{bmatrix}.$$

Since $\phi(\alpha) = [b_0, b_1, \ldots, b_{n-1}]^T$, we have that

$$\vec{\alpha} = \Delta \cdot \phi(\alpha) \tag{6.3}$$

where $\vec{\alpha}$ is the vector $[\alpha, \alpha^q, \alpha^{q^2}, \ldots, \alpha^{q^{n-1}}]^T$ and Δ is the matrix involving y's above. It is easy to see that Δ is invertible [LN97] and therefore $\Delta^{-1} \cdot \vec{\alpha} = \phi(\alpha)$. If $M \in \mathcal{M}_{n \times n}(\mathbb{F})$ is the matrix representing the linear transformation F, then $F \circ \phi(\alpha) = M \cdot \Delta^{-1} \cdot \vec{\alpha}$ and therefore $\phi^{-1} \circ F \circ \phi(\alpha)$ is the dot product between the vectors $[y^0, y^1, \ldots, y^{n-1}]^T$ and $M \cdot \Delta^{-1} \cdot \vec{\alpha}$, which clearly has the shape in equation (6.2).

We will generalize this result in the following.

6.2 Correspondence of Polynomials

Given a nonzero natural number b, any other nonzero natural number a can be written uniquely as $a = c_1 b^0 + c_2 b^1 + \cdots + c_\ell b^{\ell-1}$ where $0 \leq c_i < b$ for all i. We say that $(c_1, \ldots, c_\ell)$ is the expansion of a in base b, and we refer to $d = \sum_{i=1}^{\ell} c_i$ as the $b-$Hamming weight of a. In order to extend the definition we define the $b-$Hamming weight of $a = 0$ to be 0. To illustrate the concept, a has $q-$Hamming weight 2 if and only if it has the form $a = q^i + q^j$.

Definition. The *weight* of a monomial $X^a \in \mathbb{K}[X]$ is the $q-$Hamming weight of a. A polynomial $\mathcal{F}(X) \in \mathbb{K}[X]$ is said to be *homogeneous of weight* d if all of its monomials have weight d, and it is said to have *weight* d if all of its monomials have weight at most d.

We aim to prove the following theorem, which will be the heart of what we will develop next. Recall our notation $R := \mathbb{F}[x_1, \ldots, x_n]$.

Theorem 6.2.1. (Correspondence of Polynomials). *Let $d \geq 0$ be an integer, let $\mathbb{K}[X]_d$ denote the set of homogeneous polynomials in $\mathbb{K}[X]$ of weight d and let $(R_d)^n = R_d^n$ denote the set of all functions $F : \mathbb{F}^n \to \mathbb{F}^n$ where each coordinate is a homogeneous polynomial*

in $\mathbb{F}[x_1, \ldots, x_n]$ of degree d, these sets are naturally $\mathbb{F}$-vector spaces. The following is a well-defined bijective linear transformation

$$\begin{aligned} \mathrm{Drp}\colon \mathbb{K}[X]_d &\longrightarrow R_d^n \\ \mathcal{F} &\longmapsto \phi \circ \mathcal{F} \circ \phi^{-1}. \end{aligned}$$

whose inverse is

$$\begin{aligned} \mathrm{Lft}\colon R_d^n &\longrightarrow \mathbb{K}[X]_d \\ F &\longmapsto \phi^{-1} \circ F \circ \phi. \end{aligned}$$

Before we get into the proof of this theorem, we will need the following lemmas.

Lemma 6.2.2. *Let $\mathbb{K} = \mathbb{F}[y]/\langle g(y)\rangle$ where $g(y) = y^n + a_{n-1}y^{n-1} + \cdots + a_1 y^1 + a_0$ is an irreducible polynomial over $\mathbb{F}$. Let*

$$C = \begin{bmatrix} 0 & 0 & \cdots & 0 & -a_0 \\ 1 & 0 & \cdots & 0 & -a_1 \\ 0 & 1 & \cdots & 0 & -a_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & -a_{n-1} \end{bmatrix},$$

then for any $\alpha \in \mathbb{K}$ we have that $\phi(\alpha y^j) = C^j \cdot \phi(\alpha)$.

Proof. It suffices to show the result for $j = 1$ since the general case follows from an iteration of this case. Let $\alpha = b_0 + b_1 y^1 + \cdots + b_{n-1}y^{n-1} \in \mathbb{K}$, then

$$\begin{aligned} \alpha \cdot y &= b_0 y + b_1 y^2 + \cdots + b_{n-2}y^{n-1} + b_{n-1}y^n \\ &= b_0 y + b_1 y^2 + \cdots + b_{n-2}y^{n-1} + b_{n-1}(-a_{n-1}y^{n-1} - \cdots - a_1 y^1 - a_0) \\ &= -a_0 b_{n-1} + (b_0 - b_{n-1}a_1)y^1 + \cdots + (b_{n-2} - b_{n-1}a_{n-1})y^{n-1} \end{aligned}$$

hence $\phi(\alpha \cdot y) = [-a_0 b_{n-1}, b_0 - b_{n-1}a_1, \ldots, b_{n-2} - b_{n-1}a_{n-1}]^T$, which is the same as $C \cdot \phi(\alpha)$ since $\phi(\alpha) = [b_0, b_1, \ldots, b_{n-1}]^T$. □

Lemma 6.2.3. *Let $\mathcal{Q}(X), \mathcal{F}(X) \in \mathbb{K}[X]$ where $\mathcal{F}$ has the shape in equation (6.2). We already know that in this case $\phi \circ \mathcal{F} \circ \phi^{-1}$ is given by n homogeneous degree 1 polynomials $p_1, \ldots, p_n \in R$. Then, for all $X \in \mathbb{K}$ we have that*

$$\phi\left(\mathcal{F}(X) \cdot \mathcal{Q}(X)\right) = \sum_{i=1}^{n} p_i\left(\phi(X)\right) \cdot C^{i-1} \cdot \phi\left(\mathcal{Q}(X)\right)$$

Proof. Let $\mathbf{x} = \phi(X)$, hence

$$\begin{aligned} \mathcal{F}(X) &= \mathcal{F}\left(\phi^{-1}(\mathbf{x})\right) = \phi^{-1}\left(\phi \circ \mathcal{F} \circ \phi^{-1}(\mathbf{x})\right) \\ &= \phi^{-1}\left([p_1(\mathbf{x}), p_2(\mathbf{x}), \ldots, p_n(\mathbf{x})]^T\right) = p_1(\mathbf{x}) + p_2(\mathbf{x})y + \cdots + p_n(\mathbf{x})y^{n-1} \end{aligned}$$

and therefore, since $p_i(\mathbf{x}) \in \mathbb{F}$, due to the previous lemma we have that

$$\begin{aligned}\phi\left(\mathcal{F}(X)\cdot\mathcal{Q}(X)\right) &= \phi\left(p_1(\mathbf{x})\mathcal{Q}(X) + p_2(\mathbf{x})y\mathcal{Q}(X) + \cdots + p_n(\mathbf{x})y^{n-1}\mathcal{Q}(X)\right)\\ &= p_1(\mathbf{x})\phi\left(\mathcal{Q}(X)\right) + p_2(\mathbf{x})\phi\left(y\mathcal{Q}(X)\right) + \cdots + p_n(\mathbf{x})\phi\left(y^{n-1}\mathcal{Q}(X)\right)\\ &= p_1(\mathbf{x})\phi\left(\mathcal{Q}(X)\right) + p_2(\mathbf{x})C\mathcal{Q}(X) + \cdots + p_n(\mathbf{x})C^{n-1}\mathcal{Q}(X)\\ &= \sum_{i=1}^{n} p_i\left(\phi(X)\right)\cdot C^{i-1}\cdot\phi\left(\mathcal{Q}(X)\right).\end{aligned}$$

□

Proof of Theorem 6.2.1. We begin with the proof that this function is well defined by proving that for every monomial $\mathcal{F}(X) = X^a \in \mathbb{K}[X]_d$ it holds that $\mathrm{Drp}\left(\mathcal{F}\right) \in R_d^n$. Clearly, this is enough since lemma 6.2.2 ensures that this is true for terms αX^a and therefore it is true for any homogeneous polynomial of weight d since Drp is a composition operation so it is additively homomorphic. The claim is clear for $d = 0$ since in this case $a = 0$ and therefore the polynomial $\mathcal{F}(X) = \alpha$ is constant, as well as $\mathrm{Drp}(\mathcal{F}) \in R_0^n$. Let's assume the claim holds for d and let's prove it holds for $d+1$ as well. Since a has weight $d+1$ it can be written as $a = b+q^i$ where b has weight d so $\mathcal{F}(X) = X^a = X^{q^i}X^b$. By lemma 6.2.3 with $\mathcal{Q}(X) = X^b$ we have that

$$\phi\left(\mathcal{F}(X)\right) = \phi\left(X^{q^i}\mathcal{Q}(X)\right) = \sum_{i=1}^{n} p_i\left(\phi(X)\right)\cdot C^{i-1}\cdot\phi\left(\mathcal{Q}(X)\right)$$

where each p_i is a homogeneous degree 1 polynomial, therefore

$$\begin{aligned}\mathrm{Drp}\left(\mathcal{F}\right)(\mathbf{x}) &= \phi\circ\left(\mathcal{F}\left(\phi^{-1}(\mathbf{x})\right)\right) = \sum_{i=1}^{n} p_i\left(\phi\left(\phi^{-1}(\mathbf{x})\right)\right)\cdot C^{i-1}\cdot\phi\left(\mathcal{Q}\left(\phi^{-1}(\mathbf{x})\right)\right)\\ &= \sum_{i=1}^{n} p_i\left(\phi\left(\phi^{-1}(\mathbf{x})\right)\right)\cdot C^{i-1}\cdot\phi\left(\mathcal{Q}\left(\phi^{-1}(\mathbf{x})\right)\right) = \sum_{i=1}^{n} p_i\left(\mathbf{x}\right)\cdot C^{i-1}\cdot\mathrm{Drp}\left(\mathcal{Q}\right)(\mathbf{x}),\end{aligned}$$

but using the induction hypothesis we see that $\mathrm{Drp}\left(\mathcal{Q}\right)(\mathbf{x})$ is a vector with n homogeneous polynomials of degree d, so $\mathrm{Drp}\left(\mathcal{F}\right)(\mathbf{x})$ is a vector with n homogeneous polynomials of degree $d+1$.

Proving that Drp is bijective is not a problem now. Let $F \in R_d^n$, then $\mathcal{F} = \phi^{-1}\circ F\circ\phi$ is a polynomial in $\mathbb{K}[X]$ (every function $\mathbb{K}\to\mathbb{K}$ is a polynomial function), which we can write as

$$\mathcal{F} = \sum_{\ell=0}^{d'}\mathcal{F}_\ell$$

where each $\mathcal{F}_\ell \in \mathbb{K}[X]$ is homogeneous of weight ℓ. Due to what we have proved, $\mathrm{Drp}\left(\mathcal{F}_\ell\right) \in R_\ell^n$ for each ℓ, since

$$F = \mathrm{Drp}\left(\mathcal{F}\right) = \sum_{\ell=0}^{d'}\mathrm{Drp}\left(\mathcal{F}_\ell\right)$$

and $F \in R_d^n$, we conclude that $\mathcal{F}_\ell = 0$ for all $\ell \neq d$ and $\mathcal{F} = \mathcal{F}_d \in \mathbb{K}[X]_d$. This shows that $F \mapsto \phi^{-1}\circ F\circ\phi$ is the inverse of Drp. □

6.3 Computation of Liftings and Droppings in the Quadratic Case

For computational purposes it will be useful to have a more direct way for computing $\mathrm{Drp}\,(\mathcal{F})$ from $\mathcal{F}$ and $\mathrm{Drp}^{-1}(F)$ from F in the quadratic case. This is well known due to its applications in MPKC, and we dedicate this section to this matter.

Representation of Quadratic and Linear Forms

Let $p(x_1,\ldots,x_n) \in R$ be a quadratic polynomial, then p has the form

$$p(x_1,\ldots,x_n) = \sum_{i,j=1}^{n} a_{ij}x_ix_j + \sum_{i=1}^{n} b_ix_i + c$$

and therefore can be written as

$$p(x_1,\ldots,x_n) = \mathbf{x}^T A\mathbf{x} + B\mathbf{x} + c$$

where $\mathbf{x} = [x_1,\ldots,x_n]^T$, $A \in \mathcal{M}_{n\times n}(\mathbb{F})$ is the matrix $[a_{ij}]_{ij}$ and $B \in \mathcal{M}_{1\times n}(\mathbb{F})$ is the matrix $[b_i]_{1i}$.

It is interesting that we can have the same sort of representation with polynomials in $\mathbb{K}[X]$ having weight at most 2. These have the shape

$$\mathcal{F}(X) = \sum_{i,j=1}^{n} \alpha_{ij}X^{q^{i-1}+q^{j-1}} + \sum_{i=1}^{n} \beta_iX^{q^{i-1}} + \gamma$$

and therefore can be written as

$$\mathcal{F}(X) = \mathbf{X}^T M\mathbf{X} + N\mathbf{X} + \gamma$$

where $\mathbf{X} = [X^{q^0},\ldots,X^{q^{n-1}}]^T$, $M \in \mathcal{M}_{n\times n}(\mathbb{K})$ is the matrix $[\alpha_{ij}]_{ij}$ and $N \in \mathcal{M}_{1\times n}(\mathbb{K})$ is the matrix $[\beta_i]_{1i}$.

For the following we need to recall the invertible matrix

$$\Delta = \begin{bmatrix} y^0 & y^1 & \cdots & y^{n-2} & y^{n-1} \\ (y^0)^{q^1} & (y^1)^{q^1} & \cdots & (y^{n-2})^{q^1} & (y^{n-1})^{q^1} \\ (y^0)^{q^2} & (y^1)^{q^2} & \cdots & (y^{n-2})^{q^2} & (y^{n-1})^{q^2} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ (y^0)^{q^{n-1}} & (y^1)^{q^{n-1}} & \cdots & (y^{n-2})^{q^{n-1}} & (y^{n-1})^{q^{n-1}} \end{bmatrix}$$

which satisfies

$$\mathbf{X} = \Delta \cdot \phi(X)$$

Computation of $\mathrm{Drp}\,(\mathcal{F})$ from $\mathcal{F}$

Let $\mathcal{F}(X) \in \mathbb{K}[X]$ be a polynomial with weight at most 2 given by

$$\mathcal{F}(X) = \mathbf{X}^T M \mathbf{X} + N\mathbf{X} + \gamma,$$

we will find an explicit description of the dropping $\mathrm{Drp}\,(\mathcal{F})$ in terms of the matrices M and N. If $\mathbf{x} = \phi(X)$, we have that

$$\begin{aligned}\mathcal{F}\left(\phi^{-1}(\mathbf{x})\right) &= \mathcal{F}(X) = \mathbf{X}^T M \mathbf{X} + N\mathbf{X} + \gamma \\ &= (\Delta \cdot \phi(X))^T M (\Delta \cdot \phi(X)) + N(\Delta \cdot \phi(X)) + \gamma = \mathbf{x}^T \Delta^T M \Delta \mathbf{x} + N\Delta\mathbf{x} + \gamma.\end{aligned}$$

By factoring each y^i from the matrices $\Delta^T M \Delta$ and $N\Delta$, we can write

$$\Delta^T M \Delta = \sum_{i=1}^{n} y^{i-1} A_i$$

and

$$N\Delta = \sum_{i=1}^{n} y^{i-1} B_i$$

where $A_i \in \mathcal{M}_{n\times n}(\mathbb{F})$ and $B_i \in \mathcal{M}_{1\times n}(\mathbb{F})$, and therefore, if $\gamma = c_1 + c_2 y + \cdots + c_n y^{n-1}$

$$\begin{aligned}\mathcal{F} \circ \phi^{-1}(\mathbf{x}) = \mathbf{x}^T \left(\sum_{i=1}^{n} y^{i-1} A_i\right)\mathbf{x} + \left(\sum_{i=1}^{n} y^{i-1} B_i\right)\mathbf{x} + \sum_{i=1}^{n} c_i y^{i-1} \\ = \sum_{i=1}^{n} y^{i-1}\left(\mathbf{x}^T A_i \mathbf{x} + B_i \mathbf{x} + c_i\right).\end{aligned}$$

Since for all i and particular $x_1, \ldots, x_n \in \mathbb{F}$ we have that $\mathbf{x}^T A_i \mathbf{x} + B_i \mathbf{x} + c_i \in \mathbb{F}$, we conclude by the definition of ϕ that

$$\mathrm{Drp}\,(\mathcal{F})\,(\mathbf{x}) = \phi \circ \mathcal{F} \circ \phi^{-1}(\mathbf{x}) = \left[\mathbf{x}^T A_1 \mathbf{x} + B_1 \mathbf{x} + c_1, \ldots, \mathbf{x}^T A_n \mathbf{x} + B_n \mathbf{x} + c_n\right]^T$$

Computation of $\mathrm{Lft}(F)$ from F

Let $F : \mathbb{F}^n \to \mathbb{F}^n$ given by n quadratic polynomials $p_1, \ldots, p_n \in R$, where each polynomial is written as

$$p(x_1, \ldots, x_n) = \mathbf{x}^T A_i \mathbf{x} + B_i \mathbf{x} + c_i$$

where $A_i \in \mathcal{M}_{n\times n}(\mathbb{F})$ and $B_i \in \mathcal{M}_{1\times n}(\mathbb{F})$. We define $\gamma = c_1 + c_2 y + \cdots + c_n y^{n-1} \in \mathbb{K}$ and the matrices $M \in \mathcal{M}_{n\times n}(\mathbb{K}), N \in \mathcal{M}_{1\times n}(\mathbb{K})$ as

$$M = \left(\Delta^T\right)^{-1}\left(\sum_{i=1}^{n} y^{i-1} A_i\right)\Delta^{-1}$$

and

$$N = \left(\sum_{i=1}^{n} y^{i-1} B_i\right)\Delta^{-1},$$

by reverting the steps in the previous section we can see that $\mathrm{Lft}(F)$ is given by

$$\mathrm{Lft}\,(F)\,(X) = \phi^{-1} \circ \mathcal{F} \circ \phi(X) = \mathbf{X}^T M \mathbf{X} + N\mathbf{X} + \gamma$$

6.4 Experimental Data

This section presents all experimental data cited in this work. Computations were done using software Magma V2.21-1 [BCP97] on a server with a eight core Intel(R) Xeon(R) CPU E5-2609 running each at 2.40GHz.

6.4.1 Groebner Bases Computation

The following table resumes several experiments for computation of Groebner bases using homogeneous Lazard's algorithm. For this matter, m **quadratic** homogeneous polynomials $f_1, \ldots, f_m$ are chosen at random

q	$\dim(I)$	$\mathrm{i}_{\mathrm{reg}}(I)$	$\mathrm{MaxDeg}(I)$	MB	q	$\dim(I)$	$\mathrm{i}_{\mathrm{reg}}(I)$	$\mathrm{MaxDeg}(I)$	MB
2	1	10	10	11	5	0	11	11	11
2	1	10	10	11	5	0	11	11	11
2	1	10	10	11	5	0	11	11	11
2	1	10	10	11	5	1	10	10	11
2	0	11	11	11	5	1	10	10	11
2	1	10	10	11	5	0	11	11	11
2	1	10	10	11	5	0	11	11	11
2	1	10	10	11	5	0	11	11	11
2	0	11	10	11	5	0	11	11	11
2	0	11	11	11	5	0	11	10	11
2	0	11	11	11	5	1	10	10	11
2	0	11	10	11	5	1	10	10	11
2	1	10	10	11	5	0	11	11	11
2	1	10	10	11	5	0	11	10	11
2	1	10	10	11	5	0	11	10	11
2	0	11	10	11	5	0	11	11	11
2	1	10	10	11	5	0	11	11	11
2	0	11	11	11	5	0	11	11	11
2	1	10	10	11	5	0	11	11	11
2	0	11	10	11	5	0	11	11	11

Table 6.1: Experiments of Groebner bases computation, $n = 10, m = 10$

q	$\dim(I)$	$\mathrm{i}_{\mathrm{reg}}(I)$	$\mathrm{MaxDeg}(I)$	MB	q	$\dim(I)$	$\mathrm{i}_{\mathrm{reg}}(I)$	$\mathrm{MaxDeg}(I)$	MB
2	5	1	6	6	5	5	1	6	6
2	5	1	6	6	5	5	1	5	6
2	5	1	6	6	5	5	1	6	6
2	5	1	5	6	5	5	1	5	6
2	5	1	6	6	5	5	1	7	6
2	5	1	6	6	5	5	1	6	6
2	5	1	6	6	5	5	1	6	6
2	5	1	7	6	5	5	1	6	6
2	5	1	6	6	5	5	1	6	6
2	5	1	5	6	5	5	1	6	6
2	5	1	6	6	5	5	1	6	6
2	5	1	6	6	5	5	1	6	6
2	5	1	7	6	5	5	1	6	6
2	5	1	5	6	5	5	1	6	6
2	5	1	5	6	5	5	1	6	6
2	5	1	7	6	5	5	1	5	6
2	5	1	6	6	5	5	1	6	6
2	5	1	6	6	5	5	1	6	6
2	5	1	6	6	5	5	1	5	6
2	5	1	6	6	5	5	1	6	6

Table 6.2: Experiments of Groebner bases computation, $n = 10, m = 5$

q	$\dim(I)$	$\mathrm{i}_{\mathrm{reg}}(I)$	$\mathrm{MaxDeg}(I)$	MB	q	$\dim(I)$	$\mathrm{i}_{\mathrm{reg}}(I)$	$\mathrm{MaxDeg}(I)$	MB
2	7	0	5	4	5	7	0	3	4
2	7	0	4	4	5	7	0	4	4
2	7	0	3	4	5	7	0	4	4
2	7	0	3	4	5	7	0	4	4
2	7	0	5	4	5	7	0	4	4
2	7	0	4	4	5	7	0	4	4
2	7	0	4	4	5	7	0	4	4
2	7	0	3	4	5	7	0	4	4
2	7	0	4	4	5	7	0	4	4
2	7	0	3	4	5	7	0	4	4
2	7	0	4	4	5	7	0	4	4
2	7	0	3	4	5	7	0	4	4
2	7	0	4	4	5	7	0	3	4
2	7	0	3	4	5	7	0	3	4
2	7	0	5	4	5	7	0	4	4
2	7	0	7	4	5	7	0	4	4
2	7	0	4	4	5	7	0	4	4
2	7	0	3	4	5	7	0	4	4
2	7	0	4	4	5	7	0	4	4
2	7	0	2	4	5	7	0	5	4

Table 6.3: Experiments of Groebner bases computation, $n = 10, m = 3$

q	$\dim(I)$	$\mathrm{i}_{\mathrm{reg}}(I)$	MaxDeg(I)	MB	q	$\dim(I)$	$\mathrm{i}_{\mathrm{reg}}(I)$	MaxDeg(I)	MB
2	2	7	10	9	5	2	7	9	9
2	2	7	9	9	5	2	7	9	9
2	2	7	9	9	5	2	7	9	9
2	2	7	8	9	5	2	7	9	9
2	2	7	9	9	5	2	7	9	9
2	2	7	8	9	5	2	7	9	9
2	2	7	10	9	5	2	7	9	9
2	2	7	12	9	5	2	7	9	9
2	2	7	8	9	5	2	7	8	9
2	2	7	9	9	5	2	7	9	9
2	2	7	9	9	5	2	7	9	9
2	2	7	9	9	5	2	7	8	9
2	2	7	9	9	5	2	7	9	9
2	2	7	8	9	5	2	7	9	9
2	2	7	10	9	5	2	7	9	9
2	2	7	11	9	5	2	7	8	9
2	2	7	9	9	5	2	7	9	9
2	2	7	11	9	5	2	7	9	9
2	2	7	12	9	5	2	7	8	9
2	2	7	8	9	5	2	7	9	9

Table 6.4: Experiments of Groebner bases computation, $n = 10, m = 8$

q	$\dim(I)$	$\mathrm{i}_{\mathrm{reg}}(I)$	MaxDeg(I)	MB	q	$\dim(I)$	$\mathrm{i}_{\mathrm{reg}}(I)$	MaxDeg(I)	MB
2	0	7	7	12	5	0	7	7	12
2	0	6	6	12	5	0	6	6	12
2	0	6	6	12	5	0	6	6	12
2	0	7	6	12	5	0	6	6	12
2	0	6	6	12	5	0	7	7	12
2	0	6	6	12	5	0	6	6	12
2	0	6	6	12	5	0	6	6	12
2	0	7	7	12	5	0	6	6	12
2	0	7	7	12	5	0	6	6	12
2	0	6	6	12	5	0	6	6	12
2	1	7	7	12	5	0	6	6	12
2	0	6	6	12	5	0	6	6	12
2	1	6	6	12	5	0	7	7	12
2	0	7	6	12	5	0	6	6	12
2	0	6	6	12	5	0	6	6	12
2	0	7	7	12	5	0	6	6	12
2	0	7	6	12	5	0	7	7	12
2	0	6	6	12	5	0	6	6	12
2	1	6	6	12	5	0	6	6	12
2	0	7	6	12	5	1	6	6	12

Table 6.5: Experiments of Groebner bases computation, $n = 10, m = 9$

q	$\dim(I)$	$\mathrm{i}_{\mathrm{reg}}(I)$	$\mathrm{MaxDeg}(I)$	MB	q	$\dim(I)$	$\mathrm{i}_{\mathrm{reg}}(I)$	$\mathrm{MaxDeg}(I)$	MB
2	0	4	4	21	5	0	4	4	21
2	0	4	4	21	5	0	4	4	21
2	0	4	4	21	5	0	4	4	21
2	0	4	4	21	5	0	4	4	21
2	0	4	4	21	5	0	4	4	21
2	0	4	4	21	5	0	4	4	21
2	0	4	4	21	5	0	4	4	21
2	0	4	4	21	5	0	4	4	21
2	0	4	4	21	5	0	4	4	21
2	0	4	4	21	5	0	4	4	21
2	0	4	4	21	5	0	4	4	21
2	0	4	4	21	5	0	4	4	21
2	0	4	4	21	5	0	4	4	21
2	0	4	4	21	5	0	4	4	21
2	0	4	4	21	5	0	4	4	21
2	0	4	4	21	5	0	4	4	21
2	0	4	4	21	5	0	4	4	21
2	0	4	4	21	5	0	4	4	21
2	0	4	4	21	5	0	4	4	21
2	0	4	4	21	5	0	4	4	21
2	0	4	4	21	5	0	4	4	21
2	0	4	4	21	5	0	4	4	21
2	0	4	4	21	5	0	4	4	21

Table 6.6: Experiments of Groebner bases computation, $n = 10, m = 20$

6.4.2 New Alternatives

Multivariate Noisy Encryption Scheme

q	n	r	Plaintext space size $\approx$	Degree of $\mathcal{G}(X)$	Public key generation [s]	Encryption [s]	Decryption [s]
2	50	5	2^{50}	96	3.424	0.024	0.019
2	50	6	2^{50}	192	3.804	0.024	0.038
2	50	7	2^{50}	384	4.194	0.026	0.107
2	50	8	2^{50}	768	4.640	0.027	0.254
2	50	9	2^{50}	1536	5.387	0.026	0.629
2	50	10	2^{50}	3072	5.480	0.028	2.847
2	100	3	2^{100}	24	27.110	0.131	0.017
2	100	4	2^{100}	48	30.757	0.132	0.034
2	100	5	2^{100}	96	34.153	0.132	0.081
2	150	3	2^{150}	24	124.268	0.402	0.038
2	150	4	2^{150}	48	135.726	0.392	0.070
2	150	5	2^{150}	96	142.668	0.398	0.144
3	31	3	2^{50}	81	1.114	0.030	0.074
3	31	4	2^{50}	243	1.340	0.032	0.384
3	31	5	2^{50}	729	1.293	0.032	2.078
3	31	6	2^{50}	2187	1.453	0.032	7.214
3	63	2	2^{100}	27	10.274	0.238	0.034
3	63	3	2^{100}	81	11.650	0.238	0.168
3	63	4	2^{100}	243	12.788	0.236	0.834
3	63	5	2^{100}	729	14.080	0.240	4.516
3	94	2	2^{150}	27	65.796	1.838	0.128
3	94	3	2^{150}	81	73.036	1.836	0.542
3	94	4	2^{150}	243	79.340	1.834	2.886
5	21	2	2^{50}	75	0.254	0.006	0.026
5	21	3	2^{50}	375	0.358	0.006	0.288
5	21	4	2^{50}	1875	0.370	0.004	3.812
5	43	2	2^{100}	75	4.588	0.070	0.436
5	43	3	2^{100}	375	5.305	0.068	3.852
5	43	4	2^{100}	1875	6.000	0.070	28.940
5	64	2	2^{150}	75	8.236	0.356	0.248
5	64	3	2^{150}	375	10.010	0.354	3.068
5	64	4	2^{150}	1875	11.735	0.352	37.242
7	17	2	2^{50}	147	0.162	0.004	0.132
7	17	3	2^{50}	1029	0.200	0.004	1.844
7	17	4	2^{50}	7203	0.200	0.005	19.275
7	35	2	2^{100}	147	1.155	0.040	0.225
7	35	3	2^{100}	1029	1.370	0.040	4.850
7	35	4	2^{100}	7203	1.605	0.035	50.460
7	53	2	2^{150}	147	11.675	0.135	1.760
7	53	3	2^{150}	1029	13.230	0.140	22.545
11	14	2	2^{50}	363	0.090	0.010	0.415
11	14	3	2^{50}	3993	0.085	0.005	7.440
11	29	2	2^{100}	363	1.125	0.025	1.460
11	29	3	2^{100}	3993	1.325	0.025	29.570
11	43	2	2^{150}	363	5.635	0.080	3.665
11	43	3	2^{150}	3993	6.550	0.080	81.060
17	12	2	2^{50}	867	0.055	0.000	0.990
17	12	3	2^{50}	14739	0.040	0.005	27.680
17	24	2	2^{100}	867	0.390	0.010	3.840
17	24	3	2^{100}	14739	0.475	0.015	87.375
17	36	2	2^{150}	867	1.875	0.075	10.375

Table 6.7: Experiments of Public Key generation, encryption and decryption, for different values of q, n and D

"Non-noisy" Version

q	n	Falling Degree	Time of Algebraic Attack [s]
3	30	3	0.800
3	30	3	0.820
3	30	3	0.800
3	45	3	6.880
3	45	3	6.580
3	45	3	7.000
3	60	3	8.230
3	60	3	8.260
3	60	3	8.300

Table 6.8: Algebraic attack for different values of q and n

Bibliography

[Bar04] M. Bardet. *Étude des systèmes algébriques surdéterminés. Applications aux codes correcteurs et à la cryptographie.* PhD thesis, Université Paris 6, 2004.

[BBD08] Daniel J. Bernstein, Johannes Buchmann, and Erik Dahmen. *Post-Quantum Cryptography*. Springer Publishing Company, Incorporated, 1st edition, 2008.

[BCE+16] John B. Baena, Daniel Cabarcas, Daniel E. Escudero, Jaiberth Porras-Barrera, and Javier A. Verbel. *Efficient ZHFE Key Generation*, pages 213–232. Springer International Publishing, Cham, 2016.

[BCP97] Wieb Bosma, John Cannon, and Catherine Playoust. The Magma algebra system. I. The user language. *J. Symbolic Comput.*, 24(3-4):235–265, 1997. Computational algebra and number theory (London, 1993).

[BFP13] Luk Bettale, Jean-Charles Faugère, and Ludovic Perret. Cryptanalysis of HFE, multi-HFE and variants for odd and even characteristic. *Designs, Codes and Cryptography*, 69(1):1–52, 2013.

[Buc65] B. Buchberger. *Ein Algorithmus zum Auffinden der Basiselemente des Restklassenringes nach einem nulldimensionalen Polynomideal (An Algorithm for Finding the Basis Elements in the Residue Class Ring Modulo a Zero Dimensional Polynomial Ideal)*. PhD thesis, Mathematical Institute, University of Innsbruck, Austria, 1965. English translation in J. of Symbolic Computation, Special Issue on Logic, Mathematics, and Computer Science: Interactions. Vol. 41, Number 3-4, Pages 475–511, 2006.

[Cab11] Daniel Cabarcas. *Groebner Bases Computation and Mutant Polynomials*. PhD thesis, University of Cincinnati, 2011.

[CKM97] S. Collart, M. Kalkbrener, and D. Mall. Converting bases with the gröbner walk. *Journal of Symbolic Computation*, 24(3):465 – 469, 1997.

[CLO07] David A. Cox, John Little, and Donal O'Shea. *Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra, 3/e (Undergraduate Texts in Mathematics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2007.

[DG10] Vivien Dubois and Nicolas Gama. *The Degree of Regularity of HFE Systems*, pages 557–576. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

[DGS06] Jintai Ding, Jason E. Gower, and Dieter S. Schmidt. *Multivariate public key cryptosystems*, volume 25 of *Advances in Information Security*. Springer, New York, 2006.

[Din12] Jintai Ding. A simple provably secure key exchange scheme based on the learning with errors problem. *IACR Cryptology ePrint Archive*, 2012:688, 2012.

[DS05] Jintai Ding and Dieter Schmidt. Rainbow, a new multivariable polynomial signature scheme. In John Ioannidis, Angelos Keromytis, and Moti Yung, editors, *Applied Cryptography and Network Security*, volume 3531 of *Lecture Notes in Computer Science*, pages 164–175. Springer Berlin Heidelberg, 2005.

[DS13] Jintai Ding and Dieter Schmidt. *Solving Degree and Degree of Regularity for Polynomial Systems over a Finite Fields*, pages 34–49. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[Fau99] Jean-Charles Faugère. A new efficient algorithm for computing Gröbner bases (F_4). *J. Pure Appl. Algebra*, 139(1-3):61–88, 1999. Effective methods in algebraic geometry (Saint-Malo, 1998).

[Fau02] Jean-Charles Faugère. A new efficient algorithm for computing Gröbner bases without reduction to zero (F5). In *Proceedings of the 2002 international symposium on Symbolic and algebraic computation*, ISSAC '02, pages 75–83, New York, NY, USA, 2002. ACM.

[FGLM93] J.C. Faugère, P. 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.

[FJ03] Jean-Charles Faugère and Antoine Joux. Algebraic cryptanalysis of hidden field equation (HFE) cryptosystems using Gröbner bases. In *Advances in cryptology—CRYPTO 2003*, volume 2729 of *Lecture Notes in Computer Science*, pages 44–60. Springer, Berlin, 2003.

[Frö97] Ralf Fröberg. *An Introduction to Gröbner Bases*. Wiley, 1 edition, 1997.

[GJ90] Michael R. Garey and David S. Johnson. *Computers and Intractability; A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA, 1990.

[GM86] Rüdiger Gebauer and H. Michael Möller. Buchberger's algorithm and staggered linear bases. In *Proceedings of the Fifth ACM Symposium on Symbolic and Algebraic Computation*, SYMSAC '86, pages 218–221, New York, NY, USA, 1986. ACM.

[GM89] Patrizia M. Gianni and Teo Mora. Algebraic solution of systems of polynomial equations using groebner bases. In *Proceedings of the 5th International*

Conference on Applied Algebra, Algebraic Algorithms and Error-Correcting Codes, AAECC-5, pages 247–257, London, UK, UK, 1989. Springer-Verlag.

[KL07] Jonathan Katz and Yehuda Lindell. *Introduction to Modern Cryptography (Chapman & Hall/Crc Cryptography and Network Security Series)*. Chapman & Hall/CRC, 2007.

[KS99] Aviad Kipnis and Adi Shamir. Cryptanalysis of the HFE public key cryptosystem by relinearization. In *Advances in cryptology—CRYPTO '99 (Santa Barbara, CA)*, volume 1666 of *Lecture Notes in Computer Science*, pages 19–30. Springer, Berlin, 1999.

[Laz83] D. Lazard. *Gröbner bases, Gaussian elimination and resolution of systems of algebraic equations*, pages 146–156. Springer Berlin Heidelberg, Berlin, Heidelberg, 1983.

[LN97] Rudolf. Lidl and Harald Niederreiter. *Finite fields / Rudolf Lidl, Harald Niederreiter ; foreword by P.M. Cohn*. Cambridge University Press Cambridge ; New York, 2nd ed. edition, 1997.

[Mac02] F. S. MacAulay. Some formulæ in elimination. *Proceedings of the London Mathematical Society*, s1-35(1):3–27, 1902.

[Mac94] F.S. Macaulay. *The Algebraic theory of modular systems*. Cambridge mathematical library. Cambridge University Press, Cambridge, New York, Melbourne, 1994.

[MS04] Ezra Miller and Bernd Sturmfels. Combinatorial commutative algebra, 2004.

[Par10] Keith Pardue. Generic sequences of polynomials. *Journal of Algebra*, 324(4):579 – 590, 2010.

[Pat96] Jacques Patarin. Hidden Field Equations (HFE) and Isomorphisms of Polynomials (IP): Two new families of asymmetric algorithms. In Ueli Maurer, editor, *Advances in Cryptology—EUROCRYPT 96*, volume 1070 of *Lecture Notes in Computer Science*, pages 33–48. Springer-Verlag, 1996.

[PBD15] Jaiberth Porras, John Baena, and Jintai Ding. New candidates for multivariate trapdoor functions. *Revista Colombiana de Matemáticas*, 49:57–76, 06 2015.

[PG97] Jacques Patarin and Louis Goubin. Trapdoor one-way permutations and multivariate polynominals. In *ICICS '97: Proceedings of the First International Conference on Information and Communication Security*, pages 356–368, London, UK, 1997. Springer-Verlag.

[PS16] Ray A. Perlner and Daniel Smith-Tone. Security analysis and key modification for ZHFE. In *Post-Quantum Cryptography - 7th International Conference, PQCrypto 2016, Fukuoka, Japan, February 24-26, 2016. Proceedings*, 2016.

[Sho99] Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Rev.*, 41(2):303–332 (electronic), 1999.

[Sho05] Victor Shoup. *A Computational Introduction to Number Theory and Algebra*. Cambridge University Press, New York, NY, USA, 2005.

[Spa12] P-J. Spaenlehauer. *Solving multi-homogeneous and determinantal systems. Algorithms - Complexity - Applications.* PhD thesis, Université Paris 6, 2012.

[YC05] Bo-Yin Yang and Jiun-Ming Chen. *All in the XL Family: Theory and Practice*, pages 67–86. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.