Cubic Multivariate Cryptosystems based on Big Field Constructions and their Vulnerability to a Min-Rank Attack

Daniel Esteban Escudero Ospina

Tesis presentada como requisito
para optar por el título de
Master en Matemáticas

Asesor
Daniel Cabarcas Jaramillo

UNIVERSIDAD
NACIONAL
DE COLOMBIA

Escuela de Matemáticas
Facultad de Ciencias
Universidad Nacional de Colombia, Sede Medellín
Octubre, 2018

Contents

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 agreeing on a 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 the 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 her 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 cannot 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 on problems that can not be solved neither by classical computers nor by 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 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.

The importance of tensor theory within this field becomes apparent once we consider the fact that tensors are a natural generation of matrices, and allow for a general treatment in the case of higher dimensions. Moreover, especially relevant for cryptography is the existence of problems whose computational complexity is too hard for modern computers, so that secure schemes can be built. It is surprising that many simple problems regarding tensors of dimension higher than two are provably hard in the worst case, and with a thorough analysis it may be possible to argue that some of these problems are also hard in the average case. This would provide a new source of hard computational problems that could be used to build cryptographic primitives

Min-Rank Problem

The min-rank problem (MR) is, given k $m \times n$ matrices and a target rank r, to determine whether there exists a linear combination of the matrices of rank less or equal to r. Although NP-complete in its general setting, there are efficient algorithms to solve it for certain parameters. Indeed, Kipnis and Shamir modeled an attack on the HFE system as an MR problem and were able to break it. Since then, other multivariate public key schemes (MPK) have been subject to similar attacks. Rank defects also lead to other weakness such as a fixed degree of regularity in the algebraic attack on HFE [DH11].

The importance of the rank itself, and the prevalence of MR as an attack technique in MPK suggest a more central role as the underlying problem that supports security. For example, we can think of HFE as a way to construct low rank quadratic polynomials. Their low rank allows inversion, but it is insecure because the same low rank is preserved as a linear combination of the public key which can be efficiently solved through the Kipnis-Shamir modeling (KS) of MR.

Although the MR problem is stated for two-dimensional matrices, it can be naturally extended to d-dimensional matrices. It is particularly interesting to analyze it for three-dimensional matrices, since rank problems become much harder there. For example, simply determining the rank of a matrix is difficult for three-dimensional matrices, and it is not even known the maximum possible rank a matrix may have (see e.g. [HL13a]).

Three-dimensional matrices lead to cubic polynomials. They are less common than quadratic polynomials in MPKs for two reasons. First, they are larger thus less efficient than quadratics. But more important, if f is cubic, its differential $Df_{\mathbf{a}}(\mathbf{x}) := f(\mathbf{x}+\mathbf{a})-f(\mathbf{x})-f(\mathbf{a})$ is a quadratic map that preserves some of the properties of f. Thus, it is possible to extend rank analysis techniques from quadratics to cubics targeting the differential, c.f. [MPST17]. Yet one important question remains open: Is this a general property of any cubic map that dooms any such construction? In this thesis we address this question, by taking a general perspective not focused on a particular construction.

Main Contributions

In order to close the knowledge gap, we gather the appropriate literature to frame the discussion of the rank of cubic polynomials. We use the language of tensors that allows for very natural extensions of key concepts from two to d-dimensional matrices.

We extend the MR problem to three-dimensional matrices and we propose two ways to solve it, which naturally extend the KS modeling. Interestingly, if the rank is small, the complexity is even lower than for the quadratic case. However, the rank of a cubic polynomial in n variables can be larger than n, and in this case the attack is very inefficient.

Our results can be summarized as follows.

- A generalization of the Min-Rank problem to the 3-dimensional case and an algorithm to solve it
- Applications of the cubic Min-Rank problem to Multivariate Public Key Cryptography

- A new multivariate encryption scheme whose security arguments the point developed above

We also discuss the relevance of two other typical lines of attack for MPK in the context of cubic low rank polynomials, namely the algebraic and differential attacks. We show that the rank of the differential is not necessarily much smaller than the rank of the cubic polynomial, rendering this line of attack inefficient if the rank is large enough. Similarly, the algebraic attack is exponential in the rank, thus useless for high rank.

Although our approach is general, we provide a detailed example. We show how to efficiently construct cubic polynomials over a finite field from a weight three polynomial over a field extension, extending the so called big field idea. And then, we show that the rank is preserved by this construction in the sense that, a low rank core polynomial leads to a set of cubic polynomials with a low rank linear combination.

Part of the work presented in this thesis has been published and presented at the PQCrypto conference, in April 2018 [BCE+18], with the coauthors John Baena, Daniel Cabarcas, Karan Kathuria and Javier Verbel. The goal of this thesis is to extend that work.

Outline of the Document

This document is divided in three parts. In Part I we introduce all the necessary background for the treatment of the forthcoming sections. This includes tensor theory but also some algebraic geometry and Gröbner bases.

In Part II we discuss the main computational problem considered in this work: the Min-Rank problem. We introduce the Min-Rank problem in its original quadratic form and discuss some of the approaches for its solution considered in the literature. Then we discuss its generalization to the cubic case by using the theory introduced in the first part, and we show how to approach this computational problem in this new setting.

Finally, in Part III we show the applications of the cubic Min-Rank problem to MPKC, including our novel encryption scheme HiRaC.

Acknowledgments

I would like to thank Professor Daniel Cabarcas for his supervision throughout this work. Many thanks also to Professor John Bayron Baena, PhD student Karan Khathuria and PhD student Javier Verbel, for the fruitful seminar meetings we had at the university.

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

Part I

Min-rank Problem

Chapter 1

Algebraic Preliminaries

1.1 Basic Algebraic Structures

An abelian group is a non-empty set $\mathbb{G}$ equipped with some commutative, associative operation $+$ and with an element $0 \in \mathbb{G}$ such that for all $a \in \mathcal{G} : a + 0 = a$. A commutative ring R is a set with two different operations $+, \cdot$, such that for all $a, b, c \in \mathcal{G}$ it holds that $a \cdot (b + c) = a \cdot b + a \cdot c$ and $(R, +)$ and $(R, \cdot)$ are groups. The identity of the product $(\cdot)$ is denoted by 1. Finally, a field is a commutative ring $\mathbb{F}$ with the property that for every nonzero $a \in \mathbb{F}$ there exists an element $a^{-1} \in \mathbb{F}$ such that $a \cdot a^{-1} = 1$. In this work we focus solely on finite fields, i.e., fields with a finite number of elements. In the next section we describe particular properties of finite fields.

1.2 Finite Fields and Field Extensions

Let p be a prime and let $\mathbb{F}_p$ denote the ring of integers modulo p. It is easy to show that this is in fact a field. We denote by $\mathbb{F}_p[x]$ the ring of polynomials in the variable x with coefficients in $\mathbb{F}_p$. This is a ring under usual polynomial addition and multiplication. Notice that $\mathbb{F}_p$ is naturally embedded in $\mathbb{F}_p[x]$ as the subring of constant polynomials.

Let $h(x)$ be a polynomial in $\mathbb{F}_p[x]$. We say that $h(x)$ is irreducible if it is not divisible by a polynomial of degree strictly smaller than $\deg(h)$. We define the ideal generated by $h(x)$, denoted by $(h(x))$, as the subset of $\mathbb{F}_p[x]$ given by the polynomial multiples of $h(x)$. It can be shown than in fact this is a subring that is closed under multiplication by any polynomial. The quotient ring of $\mathbb{F}_p[x]$ and $(h(x))$, denoted by $\mathbb{F}_p[x]/(h(x))$, is defined as the set of equivalence classes of $\mathbb{F}_p[x]$ under the relation $f(x) \sim g(x)$ if and only if $h(x)$ divides $f(x) - g(x)$. For practical purposes this can be regarded as the set of polynomials in $\mathbb{F}_p[x]$ of degree strictly less than $\deg(h)$, with addition and multiplication performed modulo $h(x)$.

Using the fact that $\mathbb{F}_p[x]$ is a principal ideal domain it is easy to prove that $(h(x))$ is a maximal ideal if $h(x)$ is an irreducible polynomial, and therefore the quotient ring $\mathbb{F}_p[x]/(h(x))$ is in fact a field, which we denote by $\mathbb{F}_{p^n}$, where $n = \deg(h)$. This field, as the name suggests, has p^n elements. Moreover, it is isomorphic as a vector space to $\mathbb{F}_p^n = (\mathbb{F}_p)^n$

via the mapping that sends a polynomial of the degree at most $\deg(h) - 1$ to the vector holding its coefficients.

1.3 Vector Spaces

The only vector spaces we will be concerned with in this work are finite-dimensional vector spaces, therefore, we will not need a general treatment of such. Let $\mathbb{F}$ be a field. The vector space $\mathbb{F}^n$ is the additive group of vectors of length n, under point-wise addition. We also denote by $\mathbb{F}^{n\times m}$ the vector space of matrices over $\mathbb{F}$ of dimensions $n \times m$, and to extract the entry in position (i, j) we write $A[i, j]$, or sometimes $A_{i,j}$. Additionally, the i-th row of a matrix A (as a row vector) is denoted by $A[i, \cdot]$, and similarly $A[\cdot, j]$ for the j-th column (as a column vector).

Vectors are denoted by bold letters, e.g. $\mathbf{u}, \mathbf{v}$, and they are treated as column vectors by default unless stated otherwise. The vector $\mathbf{e}_i$ denotes the i-th canonical vector, i.e. the vector whose only non-zero entry is the i-th one, which is equal to 1. The i-th entry of a vector $\mathbf{u}$ is denoted by $\mathbf{u}[i]$, but sometimes we also use the non-bold version of the corresponding letter with subscript i: u_i.

A three dimensional matrix of dimensions $n \times m \times \ell$ is an array of elements in $\mathbb{F}$ indexed by tuples (i, j, k), where $1 \leq i \leq n$, $1 \leq j \leq m$ and $1 \leq k \leq \ell$. Notice that this is a natural extension of the usual (bidimensional) matrices. The vector space of these three-dimensional matrices is denoted, not surprisingly, by $\mathbb{F}^{n\times m\times \ell}$, and the entry indexed by (i, j, k) in a matrix $A \in \mathbb{F}^{n\times m\times \ell}$ will be denoted by $A[i, j, k]$. We denote by $A[i, \cdot, \cdot]$ the two-dimensional matrix whose entry (j, k) is given by $A[i, j, k]$, and similarly for $A[\cdot, j, \cdot]$ and $A[\cdot, \cdot, k]$.

1.3.1 Rank of a Matrix

We recall the definitions of the rank of a matrix $A \in \mathbb{F}^{n\times m}$. The following are equivalent definitions for $\mathsf{rank}(A)$:

Dimension of image: The dimension of the image of the linear map $f : \mathbb{F}^m \to \mathbb{F}^n$ given by $f(\mathbf{x}) = A\mathbf{x}$

Column rank: The maximal number of linearly independent columns of A

Row rank: The maximal number of linearly independent rows of A

Rank-Nullity Theorem $\mathsf{rank}(A) = m - \dim(K)$, where K is the kernel of A (i.e. the vector space formed by $\mathbf{u} \in \mathbb{F}^m$ such that $A\mathbf{u} = \mathbf{0}$)

Determinantal rank: The largest order of any non-zero minor in A

Factorization Rank The minumum number r such that A can be factored as $A = CF$ where $C \in \mathbb{F}^{n\times r}$ and $F \in \mathbb{F}^{r\times m}$.

Tensor rank: The minumum number r such that A can be written as $A = \sum_{i=1}^{r} \mathbf{u}_i\mathbf{v}_i^\intercal$, where $\mathbf{u}_i \in \mathbb{F}^{n\times 1}$ and $\mathbf{v}_i \in \mathbb{F}^{m\times 1}$.

Of particular interest to us is the last definition, since it is the one that is more naturally generalized to matrices of larger dimensions.

1.4 Polynomial Rings

During the rest of this document we assume that $\mathbb{F}$ is a finite field of characteristic not 2 nor 3.

We denote by $\mathbb{F}[x]$ the ring of univariate polynomials in x with coefficients in $\mathbb{F}$. Also, if $\mathbf{x} = (x_1, \ldots, x_n)$, we denote by $\mathbb{F}[\mathbf{x}]$ the ring of multivariate polynomials in the variables $x_1, \ldots, x_n$ with coefficients in $\mathbb{F}$. A non-zero polynomial in $\mathbb{F}[\mathbf{x}]$ has degree d if each of its monomials $x_1^{\alpha_1} \cdots x_n^{\alpha_n}$ satisfies $\alpha_1 + \cdots + \alpha_n \leq d$, and moreover, it is called homogeneous of degree d if equality holds for all monomials.

In this work we will be mostly dealing with quadratic and cubic polynomials, meaning that they have degree 2 and 3 respectively. Also, we will focus in homogeneous polynomials, although many of the arguments extend for the affine (i.e. non-homogeneous) case as well.

Any quadratic homogeneous polynomial $f(\mathbf{x}) \in \mathbb{F}[\mathbf{x}]$ has the form $f(\mathbf{x}) = \sum_{i,j=1}^{n} a_{i,j} x_i x_j$. This expression can be written as $f(\mathbf{x}) = \mathbf{x}^\intercal A \mathbf{x}$, where $A \in \mathbb{F}^{n \times n}$ is defined by $A[i,j] = a_{i,j}$. We will have more to say about the properties of this matrix in Section 2.3.

1.5 Lifting Polynomials

In this section we explore the relations between polynomial rings over different fields $\mathbb{F}$ and $\mathbb{K}$, where $\mathbb{K}$ is a field extension of $\mathbb{F}$. These results will be useful in the context of Multivariate Public Key Cryptography, where we construct encryption schemes using the so-called Lifting Idea, which involves polynomial rings over several fields and transformations among them.

1.5.1 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 satisfies $x^t = e$ 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 1.5.1. *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 an $\mathbb{F}$-linear transformation and therefore so is every polynomial of the form

$$\mathcal{F}(X) = \sum_{i=0}^{n-1} \alpha_i X^{q^i}, \tag{1.1}$$

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

$$\boldsymbol{\alpha} = \Delta \cdot \phi(\alpha) \tag{1.2}$$

where $\boldsymbol{\alpha}$ is the vector $[\alpha, \alpha^q, \alpha^{q^2}, \ldots, \alpha^{q^{n-1}}]^\intercal$ and Δ is the matrix involving y's above. It is easy to see that Δ is invertible [LN97] and therefore $\Delta^{-1} \cdot \boldsymbol{\alpha} = \phi(\alpha)$. If $M \in \mathbb{F}^{n \times n}$ is the matrix representing the linear transformation F, then $F \circ \phi(\alpha) = M \cdot \Delta^{-1} \cdot \boldsymbol{\alpha}$ and therefore $\phi^{-1} \circ F \circ \phi(\alpha)$ is the dot product between the vectors $[y^0, y^1, \ldots, y^{n-1}]^\intercal$ and $M \cdot \Delta^{-1} \cdot \boldsymbol{\alpha}$, which clearly has the shape in Equation (1.1).

We will generalize this result in the following.

1.5.2 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 1.5.2. (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*

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

whose inverse is

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

Before we get into the proof of this theorem, we will need the following lemmas.

Lemma 1.5.3. *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 $\phi(\alpha \cdot y) = [-a_0 b_{n-1}, b_0 - b_{n-1}a_1, \ldots, b_{n-2} - b_{n-1}a_{n-1}]^\intercal$, which is the same as $C \cdot \phi(\alpha)$ since $\phi(\alpha) = [b_0, b_1, \ldots, b_{n-1}]^\intercal$. □

Lemma 1.5.4. *Let $\mathcal{Q}(X), \mathcal{F}(X) \in \mathbb{K}[X]$ where $\mathcal{F}$ has the shape in equation (1.1). 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*

$$\phi(\mathcal{F}(X) \cdot \mathcal{Q}(X)) = \sum_{i=1}^{n} p_i(\phi(X)) \cdot C^{i-1} \cdot \phi(\mathcal{Q}(X))$$

Proof. Let $\mathbf{x} = \phi(X)$, hence

$$\begin{aligned}\mathcal{F}(X) = \mathcal{F}\left(\phi^{-1}(\mathbf{x})\right) &= \phi^{-1}\left(\phi \circ \mathcal{F} \circ \phi^{-1}(\mathbf{x})\right) \\ &= \phi^{-1}\left([p_1(\mathbf{x}), p_2(\mathbf{x}), \ldots, p_n(\mathbf{x})]^\intercal\right) = p_1(\mathbf{x}) + p_2(\mathbf{x})y + \cdots + p_n(\mathbf{x})y^{n-1}\end{aligned}$$

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 1.5.2. 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 $\mathsf{Drp}\left(\mathcal{F}\right) \in R_d^n$. Clearly, this is enough since lemma 1.5.3 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 $\mathsf{Drp}(\mathcal{F}) \in R_0^n$. 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 1.5.4 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

$$\begin{aligned}\mathsf{Drp}\left(\mathcal{F}\right)(\mathbf{x}) = \phi \circ \left(\mathcal{F}\left(\phi^{-1}(\mathbf{x})\right)\right) &= \sum_{i=1}^{n} p_i\left(\phi\left(\phi^{-1}(\mathbf{x})\right)\right) \cdot C^{i-1} \cdot \phi\left(\mathcal{Q}\left(\phi^{-1}(\mathbf{x})\right)\right) \\ = \sum_{i=1}^{n} p_i\left(\phi\left(\phi^{-1}(\mathbf{x})\right)\right) \cdot C^{i-1} \cdot \phi\left(\mathcal{Q}\left(\phi^{-1}(\mathbf{x})\right)\right) &= \sum_{i=1}^{n} p_i\left(\mathbf{x}\right) \cdot C^{i-1} \cdot \mathsf{Drp}\left(\mathcal{Q}\right)(\mathbf{x}),\end{aligned}$$

but using the induction hypothesis we see that $\mathsf{Drp}\left(\mathcal{Q}\right)(\mathbf{x})$ is a vector with n homogeneous polynomials of degree d, so $\mathsf{Drp}\left(\mathcal{F}\right)(\mathbf{x})$ 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, $\mathsf{Drp}(\mathcal{F}_\ell) \in R_\ell^n$ for each ℓ, since

$$F = \mathsf{Drp}(\mathcal{F}) = \sum_{\ell=0}^{d'} \mathsf{Drp}(\mathcal{F}_\ell)$$

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. □

1.5.3 Computation of Liftings and Droppings in the Quadratic Case

The results from the previous section show that given any polynomial system F of degree d over $\mathbb{K}$, we can obtain a univariate polynomial of weight d over $\mathbb{F}$ by computing $\mathsf{Lft}(F)$, and viceversa by using Drp. The proof of this fact we gave was not constructive. However, for computational purposes it is useful to have a more direct way for computing $\mathsf{Drp}(\mathcal{F})$ from $\mathcal{F}$ and $\mathsf{Drp}^{-1}(F)$ from F. In this section we provide expressions for achieving this in the quadratic case. This is well known due to its applications in MPKC, and we dedicate this section to this matter. In Section 6.1.1 we provide similar formulas for the cubic setting.

Let $p(x_1, \ldots, x_n) \in R$ be a quadratic polynomial, then p has the form

$$p(x_1, \ldots, x_n) = \sum_{i,j=1}^{n} a_{ij} x_i x_j + \sum_{i=1}^{n} b_i x_i + c$$

and therefore can be written as

$$p(x_1, \ldots, x_n) = \mathbf{x}^\intercal A \mathbf{x} + B\mathbf{x} + c$$

where $\mathbf{x} = [x_1, \ldots, x_n]^\intercal$, $A \in \mathbb{F}^{n \times n}$ is the matrix $[a_{ij}]_{ij}$ and $B \in \mathbb{F}^{1 \times n}$ 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

$$\mathcal{F}(X) = \sum_{i,j=1}^{n} \alpha_{ij} X^{q^{i-1}+q^{j-1}} + \sum_{i=1}^{n} \beta_i X^{q^{i-1}} + \gamma$$

and therefore can be written as

$$\mathcal{F}(X) = \mathbf{X}^\intercal M \mathbf{X} + N\mathbf{X} + \gamma$$

where $\mathbf{X} = [X^{q^0}, \ldots, X^{q^{n-1}}]^\intercal$, $M \in \mathbb{K}^{n \times n}$ is the matrix $[\alpha_{ij}]_{ij}$ and $N \in \mathbb{K}^{1 \times n}$ 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 $\mathsf{Drp}\,(\mathcal{F})$ from $\mathcal{F}$

Let $\mathcal{F}(X) \in \mathbb{K}[X]$ be a polynomial with weight at most 2 given by

$$\mathcal{F}(X) = \mathbf{X}^\intercal M\mathbf{X} + N\mathbf{X} + \gamma,$$

we will find an explicit description of the dropping $\mathsf{Drp}\,(\mathcal{F})$ in terms of the matrices M and N. If $\mathbf{x} = \phi(X)$, we have that

$$\begin{aligned}\mathcal{F}\left(\phi^{-1}(\mathbf{x})\right) &= \mathcal{F}(X) = \mathbf{X}^\intercal M\mathbf{X} + N\mathbf{X} + \gamma \\ &= (\Delta \cdot \phi(X))^\intercal M\,(\Delta \cdot \phi(X)) + N\,(\Delta \cdot \phi(X)) + \gamma = \mathbf{x}^\intercal \Delta^\intercal M\Delta\mathbf{x} + N\Delta\mathbf{x} + \gamma.\end{aligned}$$

By factoring each y^i from the matrices $\Delta^\intercal M\Delta$ and $N\Delta$, we can write

$$\Delta^\intercal M\Delta = \sum_{i=1}^{n} y^{i-1}A_i$$

and

$$N\Delta = \sum_{i=1}^{n} y^{i-1}B_i$$

where $A_i \in \mathbb{F}^{n\times n}$ and $B_i \in \mathbb{F}^{1\times n}$, and therefore, if $\gamma = c_1 + c_2 y + \cdots + c_n y^{n-1}$

$$\begin{aligned}\mathcal{F}\circ\phi^{-1}(\mathbf{x}) = \mathbf{x}^\intercal\left(\sum_{i=1}^{n} y^{i-1}A_i\right)\mathbf{x} + \left(\sum_{i=1}^{n} y^{i-1}B_i\right)\mathbf{x} + \sum_{i=1}^{n} c_i y^{i-1} \\ = \sum_{i=1}^{n} y^{i-1}\left(\mathbf{x}^\intercal A_i\mathbf{x} + B_i\mathbf{x} + c_i\right).\end{aligned}$$

Since for all i and particular $x_1, \ldots, x_n \in \mathbb{F}$ we have that $\mathbf{x}^\intercal A_i\mathbf{x} + B_i\mathbf{x} + c_i \in \mathbb{F}$, we conclude by the definition of ϕ that

$$\mathsf{Drp}\,(\mathcal{F})\,(\mathbf{x}) = \phi\circ\mathcal{F}\circ\phi^{-1}(\mathbf{x}) = [\mathbf{x}^\intercal A_1\mathbf{x} + B_1\mathbf{x} + c_1, \ldots, \mathbf{x}^\intercal A_n\mathbf{x} + B_n\mathbf{x} + c_n]^\intercal$$

Computation of $\mathsf{Lft}(F)$ 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

$$p(x_1, \ldots, x_n) = \mathbf{x}^\intercal A_i\mathbf{x} + B_i\mathbf{x} + c_i$$

where $A_i \in \mathbb{F}^{n\times n}$ and $B_i \in \mathbb{F}^{1\times n}$. We define $\gamma = c_1 + c_2 y + \cdots + c_n y^{n-1} \in \mathbb{K}$ and the matrices $M \in \mathbb{K}^{n\times n}, N \in \mathbb{K}^{1\times n}$ as

$$M = (\Delta^\intercal)^{-1}\left(\sum_{i=1}^{n} y^{i-1}A_i\right)\Delta^{-1}$$

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 $\mathsf{Lft}(F)$ is given by

$$\mathsf{Lft}\,(F)\,(X) = \phi^{-1}\circ\mathcal{F}\circ\phi(X) = \mathbf{X}^\intercal M\mathbf{X} + N\mathbf{X} + \gamma.$$

Chapter 2

Tensor Theory

2.1 Tensor Product

In this section we introduce the language of tensors, which will be our main object of study in the subsequent chapters. Tensor theory can be developed from a very abstract perspective. However, since we are using only finite fields and finitely dimensional vector spaces, we find it more fruitful for our purposes to take a more explicit but less general approach.

We begin by defining an operation between two vectors $\mathbf{u} \in \mathbb{F}^n$ and $\mathbf{v} \in \mathbb{F}^m$, which we denote by $\mathbf{u} \otimes \mathbf{v}$, and gives as a result a matrix in $\mathbb{F}^{n \times m}$ whose entry (i, j) is given by $u_i \cdot u_j$, i.e. $(\mathbf{u} \otimes \mathbf{v})[i, j] = \mathbf{u}[i] \cdot \mathbf{v}[j]$. We refer to this operation as the tensor product of $\mathbf{u}$ and $\mathbf{v}$.[1] It is easy to check that this operation can be seen also as $\mathbf{u} \otimes \mathbf{v} = \mathbf{u}\mathbf{v}^\intercal$.

Similarly, given $\mathbf{u} \in \mathbb{F}^n$, $\mathbf{v} \in \mathbb{F}^m$ and $\mathbf{w} \in \mathbb{F}^\ell$, we can define $\mathbf{u} \otimes \mathbf{v} \otimes \mathbf{w}$ to be the three-dimensional matrix in $\mathbb{F}^{n \times m \times \ell}$ whose entry indexed by (i, j, k) is given by $u_i \cdot v_j \cdot w_k$, i.e., $(\mathbf{u} \otimes \mathbf{v} \otimes \mathbf{w})[i, j, k] = \mathbf{u}[i] \cdot \mathbf{v}[j] \cdot \mathbf{w}[k]$.

Tensor theory is an exciting branch of mathematics with many applications to physics, chemistry, and engineering. Moreover, cryptography has also benefited from tensor theory. For example, in [Sch12] a new encryption scheme is proposed using the properties of cubic tensors (unfortunately, such a scheme turns out to be vulnerable to a Min-Rank attack). Also, more generally, all the multivariate schemes can be considered as tensor-based, since multivariate polynomials are ultimately some type of tensors (as we will see in Section 2.3). Additionally, a very recent and interesting application of tensor theory to Indistinguishability Obfuscation (iO) has been introduced [GJ18]. Such a primitive has proven to be very hard to construct. In plain terms, iO allows programs to be obfuscated so that they can be executed on arbitrary data without revealing the internals of the program itself. The fact that iO can be realized from tensor theory only shows how powerful the tensor problems can be, and how useful they could be for cryptography.

[1] It would be more adequate to call it Kronecker product, since the tensor product is technically reserved to an operation between vector spaces. However, for simplicity we will keep the term tensor product to denote the operation between vectors (and as we will see soon, between matrices)

2.2 Rank for Three-Dimensional Matrices

Recall that the rank of a matrix $A \in \mathbb{F}^{n\times m}$ can be defined as the minimum number of summands r required to write A as

$$A = \sum_{i=1}^{r} \mathbf{u}_i \mathbf{v}_i^\intercal,$$

where $\mathbf{u}_i \in \mathbb{F}^n$ and $\mathbf{v}_i \in \mathbb{F}^m$ for all $i = 1, \ldots, r$. Keeping in mind that $\mathbf{u}_i \mathbf{v}_i^\intercal = \mathbf{u}_i \otimes \mathbf{v}_i$, we can easily generalize this to the three-dimensional case (and in fact, to any dimension) by letting the rank of a three-dimensional matrix $A \in \mathbb{F}^{n\times m\times \ell}$ be the minimum number of summands r required to write A as

$$A = \sum_{i=1}^{r} \mathbf{u}_i \otimes \mathbf{v}_i \otimes \mathbf{w}_i,$$

where $\mathbf{u}_i \in \mathbb{F}^n$, $\mathbf{v}_i \in \mathbb{F}^m$ and $\mathbf{w}_i \in \mathbb{F}^\ell$ for all $i = 1, \ldots, r$. Similarly to the bidimensional case, We denote this number by $\mathsf{rank}(A)$.

And important remark is that $\mathsf{rank}(A)$ is always finite, and in fact it is upper bounded by n^2. To see this, begin by noticing that we can always write

$$A = \sum_{i,j,k} A[i,j,k] \cdot (\mathbf{e}_i \otimes \mathbf{e}_j \otimes \mathbf{e}_k) = \sum_i \mathbf{e}_i \otimes \left(\sum_{j,k} A[i,j,k] \cdot (\mathbf{e}_j \otimes \mathbf{e}_k) \right).$$

Now, each bidimensional matrix $\sum_{j,k} A[i,j,k] \cdot (\mathbf{e}_j \otimes \mathbf{e}_k)$ has rank at most n and therefore can be written as $\sum_{\ell=1}^{n} \mathbf{u}_\ell^i \otimes \mathbf{v}_\ell^i$, which means that we can write

$$A = \sum_{i,\ell} \mathbf{e}_i \otimes \mathbf{u}_\ell^i \otimes \mathbf{v}_\ell^i.$$

Since this summation has at most n^2 summands, we conclude that $\mathsf{rank}(A) \leq n^2$.

And interesting fact is that many of the computational problems related to the concept of rank that are trivial in the bidimensional setting become much harder in the three-dimensional one. Below we illustrate some examples.

- Computing the rank of a bidimensional matrix is simple using Guass-Jordan reduction and reading the rank from the number of non-zero rows. In the three-dimensional setting computing the rank is a much harder task, since it essentially involves solving a minimization problem. In fact, it can be shown that this problem is NP-complete [Hå90]. Moreover, to the best of our knowledge, there are no efficient algorithms *in the average* for computing such rank.

- In particular, constructing a cubic matrix of a desired rank is not an easy task. This will become a problem for us in Section 4.3.2, where we show experimental data about the rank of some special matrices. We overcome this issue by using some characterizations of rank which are much easier to deal with.

- We know that a matrix in $\mathbb{F}^{n\times n}$ can have rank at most n, and moreover, this maximum is attainable (for instance by the identity matrix). However, determining the maximal rank attainable by a cubic matrix is a hard, open problem. We have shown above that this maximum is at most n^2, However, it is quite surprising that this maximum is in fact strictly smaller than n^2. The best that is known is that the largest rank attainable by a matrix in $\mathbb{F}^{n\times n\times n}$ lies between $(1/3)n^2$ and $(3/4)n^2$ (see [How78, Theorem 7] for a proof of these bounds, and [Blä14] for some explicit constructions of high-rank tensors).

- There are many other problems over tensors which turn out to be computationally hard. See [HL13b] for a good compendium on some of these problems.

As we saw in Section 1.3.1, there are many characterizations for the rank of a matrix in $\mathbb{F}^{n\times n}$, some of them being more useful in certain contexts than others. However, after our discussion on the hardness of some computational problems on cubic tensor rank, it should not be a surprise that obtaining characterizations for the rank in the cubic setting is a much harder task. Below we will enunciate some of the characterizations that will be useful for us in upcoming sections.

We begin with this characterization of rank for cubic matrices, which will be important when we provide a generalization of the Kipnis-Shamir modeling for the Min-Rank problem in Section 4.2.2 (for a proof, see e.g. [Lan12]).

Theorem 2.2.1. *Given a three-dimensional matrix $A \in \mathbb{F}^{n\times m\times \ell}$, the rank of A is the minimal number r of rank one matrices $S_1, \ldots, S_r \in \mathbb{F}^{m\times \ell}$, such that, for all slices $A[i,\cdot,\cdot]$ of A, $A[i,\cdot,\cdot] \in \operatorname{span}(S_1, \ldots, S_r)$.*

Another useful characterization of rank is the one given by the Kruskal rank. The Kruskal rank of a matrix with columns $\mathbf{u}_1, \ldots, \mathbf{u}_m$, denoted by $\mathsf{KRank}(\mathbf{u}_1, \ldots, \mathbf{u}_m)$, is defined as the maximum integer k such that any subset of $\{\mathbf{u}_1, \ldots, \mathbf{u}_m\}$ of size k is linearly independent. The following theorem is a particular case of the known Kruskal's theorem [Kru77, Shm16].

Theorem 2.2.2. *Let $\mathbb{F}$ be a finite field, $\mathbf{u}_1, \ldots, \mathbf{u}_r \in U$ and $t_1, \ldots, t_r \in \mathbb{F}$. Suppose that $A = \sum_{i=1}^{r} t_i \mathbf{u}_i \otimes \mathbf{u}_i \otimes \mathbf{u}_i$ and that $2r + 2 \leq \mathsf{KRank}(t_1\mathbf{u}_1, \ldots, t_r\mathbf{u}_r) + 2 \cdot \mathsf{KRank}(\mathbf{u}_1, \ldots, \mathbf{u}_r)$. Then $\mathsf{rank}(A) = r$.*

Finally, it is important to remark that some properties of the quadratic rank still holds in the cubic setting. A particular property that will be very relevant for us when we study the Min-Rank attack in the cubic setting in Chapter 6 is that the cubic rank is invariant under invertible linear transformations. However, to make this more precise we need the machinery of trilinear forms, which will not be developed until we reach Section 2.3.2.

2.2.1 Symmetric Rank

Another useful notion is the concept of symmetric rank.

Definition. Let $S \in \mathbb{F}^{n\times n\times n}$ be a three-dimensional symmetric matrix.[2] We define the symmetric rank of S as the minimum number of summands s required to write S as

$$S = \sum_{i=1}^{s} t_i \mathbf{u}_i \otimes \mathbf{u}_i \otimes \mathbf{u}_i,$$

where $\mathbf{u}_i \in \mathbb{F}^n$, $t_i \in \mathbb{F}$. If such decomposition does not exist, this number is defined to be ∞. We denote this number by $\mathsf{SRank}(S)$.

It is clear by the definitions of rank and symmetric rank that for every symmetric matrix $A \in \mathbb{F}^{n\times n\times n}$ it holds that $\mathsf{rank}(A) \leq \mathsf{SRank}(A)$. It can be shown that these two numbers coincide in a number of cases, but not always [CGLM08].

The following proposition gives us a sufficient condition over $\mathbb{F}$ to guarantee that for all symmetric matrices in $\mathbb{F}^{n\times n\times n}$ the symmetric rank is finite. A more general result is shown in [SgS13, Proposition 7.2].

Proposition 2.2.3. *Let $\mathbb{F}$ be a finite field with $|\mathbb{F}| \geq 3$. Then each three-dimensional symmetric matrix $S \in \mathbb{F}^{n\times n\times n}$ can be written as $S = \sum_{i=1}^{s} t_i \mathbf{u}_i \otimes \mathbf{u}_i \otimes \mathbf{u}_i$, where $\mathbf{u}_i \in \mathbb{F}^n$ and $t_i \in \mathbb{F}$.*

2.3 Bilinear and Trilinear Maps

2.3.1 Bilinear Maps

A bilinear map $B : \mathbb{F}^n \times \mathbb{F}^n \to \mathbb{F}$ is a map that is linear in each argument, that is, $B(\mathbf{x_0} + \lambda\mathbf{x_1}, \mathbf{y}) = B(\mathbf{x_0}, \mathbf{y}) + \lambda B(\mathbf{x_1}, \mathbf{y})$ for all $\mathbf{x_0}, \mathbf{x_1}, \mathbf{y} \in \mathbb{F}^n$ and $\lambda \in \mathbb{F}$, and similarly for the second coordinate. It is easy to check that if we define the matrix $A \in \mathbb{F}^{n\times n}$ by $A[i,j] = B(\mathbf{e}_i, \mathbf{e}_j)$, then for all $\mathbf{x}, \mathbf{y} \in \mathbb{F}^n$ it holds that

$$B(\mathbf{x}, \mathbf{y}) = \mathbf{x}^\intercal A \mathbf{y}, \tag{2.1}$$

which is a more compact representation of B.

We see that a bilinear map can be represented by a matrix $B \in \mathbb{F}^{n\times n}$, and it is easy to see that one bilinear map can come from only one matrix, i.e. bilinear maps and matrices in $\mathbb{F}^{n\times n}$ are in a one-to-one correspondence. To see why this is the case simply notice that if two matrices can give rise to the same bilinear form, i.e. $\mathbf{x}^\intercal A_0 \mathbf{y} = \mathbf{x}^\intercal A_1 \mathbf{y}$ for all $\mathbf{x}, \mathbf{y} \in \mathbb{F}^n$, then $\mathbf{x}^\intercal (A_0 - A_1)\mathbf{y} = \mathbf{0}$. This implies that the matrix $A_0 - A_1$ represents the null bilinear map, but only the zero matrix can represent such map since any non-zero entry $A[i,j]$ yields a non-zero value $B(\mathbf{e}_i, \mathbf{e}_j) = A[i,j] \neq 0$.

Given a bilinear map B we can obtain a quadratic homogeneous polynomial $f(\mathbf{x}) \in \mathbb{F}[\mathbf{x}]$ by defining $f(\mathbf{x}) := B(\mathbf{x}, \mathbf{x})$. In particular, f can be expressed as $f(\mathbf{x}) = \mathbf{x}^\intercal A \mathbf{x}$ for some matrix $A \in \mathbb{F}^{n\times n}$. Moreover, as we saw in Section 1.4, every such polynomial can be represented in such way. Nevertheless, unlike the case of bilinear maps, different matrices

[2] A cubic symmetric matrix is a matrix that is invariant under any permutation of its indexes. A more precise definition will be given in Section 2.3.2

A can give rise to the same quadratic polynomial. For instance, if $A' \in \mathbb{F}^{n\times n}$ is skew-symmetric, meaning that $A'[i,j] = -A'[j,i]$ for all i, j, it can be seen that $A + A'$ represents the same quadratic polynomial as A. In fact, the converse holds: A_0 and A_1 represent the same quadratic polynomial if and only if $A_0 - A_1$ is an skew-symmetric matrix. In particular, A and A^T represent the same quadratic polynomial.

Even though the quadratic polynomials are not in one-to-one correspondence with matrices in $\mathbb{F}^{n\times n}$, we can obtain such correspondence by restricting $\mathbb{F}^{n\times n}$. Let $\mathcal{S}^{n\times n} \subseteq \mathbb{F}^{n\times n}$ denote the set of symmetric matrices, we claim that $\mathcal{S}^{n\times n}$ is in one-to-one correspondence with the quadratic homogeneous polynomials in $\mathbb{F}[\mathbf{x}]$. To prove this we just need to show two things. First, that every quadratic polynomial $f(\mathbf{x})$ can be written as $\mathbf{x}^\intercal A\mathbf{x}$ where $A \in \mathcal{S}^{n\times n}$, and secondly that such representation is unique. To see the first fact let us begin by writting $f(\mathbf{x}) = \mathbf{x}^\intercal A'\mathbf{x}$, where $A' \in \mathbb{F}^{n\times n}$ (not necessarily symmetric). Define $A = \frac{1}{2}(A' + A'^\intercal)$ (recall that we are in characteristic not 2 nor 3), then A is symmetric and $\mathbf{x}^\intercal A\mathbf{x} = \frac{1}{2}(\mathbf{x}^\intercal A'\mathbf{x} + \mathbf{x}^\intercal A'^\intercal\mathbf{x}) = \frac{1}{2}(f(\mathbf{x}) + f(\mathbf{x})) = f(\mathbf{x})$. To prove the second claim simply notice that if A, A' are symmetric matrices then $A - A'$ cannot be skew-symmetric unless $A = A'$, so two different symmetric matrices cannot represent the same quadratic polynomial.

Let R_2 denote the set of quadratic homogeneous polynomials in $\mathbb{F}[\mathbf{x}]$. Also, let us say that a bilinear map B is symmetric if for all $\mathbf{a}, \mathbf{b} \in \mathbb{F}^n$ it holds that $B(\mathbf{a}, \mathbf{b}) = B(\mathbf{b}, \mathbf{a})$, which is equivalent to the unique matrix representing B being symmetric. We have seen that R_2 is in correspondence with $\mathcal{S}^{n\times n}$, and the latter is in correspondence with the symmetric bilinear maps. Given a polynomial $f(\mathbf{x}) \in R_2$, we can get the corresponding matrix $A \in \mathcal{S}^{n\times n}$ by letting $A[i,j] = \frac{1}{2}f_{i,j}$ if $i \neq j$, and $A[i,j] = f_{i,j}$ when $i = j$, where $f_{i,j} \in \mathbb{F}$ is the coefficient of $x_i x_j$ in f. Then, to get the corresponding symmetric bilinear map we can define $B(\mathbf{x}, \mathbf{y}) = \mathbf{x}^\intercal A\mathbf{y}$. Interestingly, there is a more direct way of getting this map from the polynomial f: The symmetric bilinear map B can be computed as $B(\mathbf{x}, \mathbf{y}) := \frac{1}{2}\left(f(\mathbf{x} + \mathbf{y}) - f(\mathbf{x}) - f(\mathbf{y})\right)$. This obervation will prove to be useful when we consider the differential of the polynomial f.

2.3.2 Trilinear Maps

Once we have the background from Bilinear Maps, extending these to Trilinear Maps is not very difficult. A trilinear map $T : \mathbb{F}^n \times \mathbb{F}^n \times \mathbb{F}^n \to \mathbb{F}$ is a map that is linear in each argument. Similar to the bilinear case, there is a way to write it as some product related to the values of T on the canonical vectors. However, since in three dimensions there is no concept of matrix multiplication as such, we must take a slightly different approach. T can be written as

$$T(\mathbf{x}, \mathbf{y}, \mathbf{z}) = \sum_{i,j,k\in[n]} x_i y_j z_k \cdot \alpha_{i,j,k}$$

where $\alpha_{i,j,k} := T(\mathbf{e}_i, \mathbf{e}_j, \mathbf{e}_k)$. Therefore, T can be represented by means of the matrix $A \in \mathbb{F}^{n\times n\times n}$ such that $A[i,j,k] = \alpha_{i,j,k}$ and this representation is unique.

Given a trilinear form T we can obtain a homogeneous cubic polynomial $f(\mathbf{x}) \in \mathbb{F}[\mathbf{x}]$ defined as $f(\mathbf{x}) := T(\mathbf{x}, \mathbf{x}, \mathbf{x})$. Just like in the bilinear case, many different matrices can give rise to the same polynomial. However, unlike the bilinear case, two matrices represent

the same polynomial if their difference is skew-symmetric but the reverse direction is not necessarily true. The definition for skew-symmetric in the three-dimensional case is much trickier, and it forces us to introduce some concepts before we dive into it.

Let Π_ℓ denote the set of permutations in the set $\{1,2,3\}$. Given $\pi \in \Pi_3$ and $A \in \mathbb{F}^{n\times n\times n}$, we write $\pi(A)$ to denote the matrix in $\mathbb{F}^{n\times n\times n}$ resulting of permuting the indexes of A according to π, i.e. $\pi(A)[i,j,k] = A[\pi(i,j,k)]$. We then say that A is skew-symmetric if $\pi(A) = (-1)^{\mathsf{sign}(\pi)}A$ for all permutations $\pi \in \Pi_3$, where $\mathsf{sign}(\pi)$ is the sign of the permutation π. Notice that this naturally extends the concept of skew-symmetry for bidimensional matrices.

Now we show that A and A' represent the same cubic polynomial if $A - A'$ is skew-symmetric. For this it suffices to show that if B is skew-symmetric then B represents the null polynomial. Let $B \in \mathbb{F}^{n\times n\times n}$. The coefficient of $x_i x_j x_k$ in the polynomial represented by B is given by $\sum_{\pi\in\Pi_3} B[\pi(i,j,k)]$. If B is skew-symmetric, then it holds that $B[\pi(i,j,k)] = (-1)^{\mathsf{sign}(\pi)}B[i,j,k]$, so this coefficient is given by $B[i,j,k]\sum_{\pi\in\Pi_3}(-1)^{\mathsf{sign}(\pi)}$. It is easy to check then via a group-theoretic argument that this coefficient equals 0,[3] so the matrix B represents is the null polynomial.

An important observation is that once we restrict to symmetric matrices, the representation of a cubic polynomial via a matrix is unique. A matrix $A \in \mathbb{F}^{n\times n\times n}$ is symmetric if $\pi(A) = A$ for all $\pi \in \Pi_3$. A bit more explicitly, A is symmetric if

$$A[i,j,k] = A[i,k,j] = A[j,i,k] = A[k,i,j] = A[j,k,i] = A[k,j,i]$$

for all i,j,k. We denote by $\mathcal{S}^{n\times n\times n}$ the subset of $\mathbb{F}^{n\times n\times n}$ formed by the symmetric matrices.

To prove that any cubic homogeneous polynomial is representable in a unique manner by a symmetric matrix in $\mathcal{S}^{n\times n\times n}$, we consider an arbitrary such polynomial $f(\mathbf{x}) = \sum_{i,j,k} f_{i,j,k}x_i x_j x_k$. If we define the matrix A' such that $A'[i,j,k] = f_{i,j,k}$, we see that the associated trilinear form gives rise to the polynomial f. However, A' might not be symmetric. To turn A' into a symmetric matrix representing the same cubic polynomial we define $A = \frac{1}{3!}\left(\sum_{\pi\in\Pi_3}\pi(A')\right)$. It is easy to see that this matrix is symmetric and that it represents the same cubic polynomial. Finally, the fact that the difference of two different symmetric matrices cannot be skew-symmetric implies that this representation is unique.

As a note, observe that $\mathsf{rank}(A) \leq 3! \cdot (\mathsf{rank}(A'))$ since each $\pi(A')$ has the same rank as A' and A is the sum of $3!$ such matrices.

We have shown that in analogy to the bilinear case, the polynomials in R_3 are in a one-to-one correspondence with $\mathcal{S}^{n\times n\times n}$, which in turn are in one-to-one correspondence with the symmetric trilinear maps, defined simply as the trilinears map for which the underlying matrix is symmetric. An important fact is that given a homogeneous polynomial f of degree 3 we can obtain the corresponding symmetric trilinear form by defining

$$T(\mathbf{x},\mathbf{y},\mathbf{z}) = \frac{1}{3!}(f(\mathbf{x}+\mathbf{y}+\mathbf{z}) - f(\mathbf{y}+\mathbf{z}) - f(\mathbf{x}+\mathbf{z}) \\ - f(\mathbf{x}+\mathbf{y}) + f(\mathbf{x}) + f(\mathbf{y}) + f(\mathbf{z})). \quad (2.2)$$

For a cubic homogeneous polynomial $f \in \mathbb{F}[\mathbf{x}]$, we define its rank, denoted by $\mathsf{rank}(f)$, as the rank of the corresponding three-dimensional symmetric matrix.

[3] One way to see this is recalling that exactly half of the elements in Π_n have sign $+1$, while the other half has sign -1

2.4 Rank is Invariant under Invertible Linear Transformations

Let $S \in \mathbb{F}^{n\times n}$ be an invertible matrix, and let $A \in \mathbb{F}^{n\times n\times n}$. Consider T as the trilinear map associated to A. If we regard S as a function $\mathbb{F}^n \to \mathbb{F}^n$, we then can define the function $T' : \mathbb{F}^n \times \mathbb{F}^n \times \mathbb{F}^n \to \mathbb{F}$ given by $T'(\mathbf{x}, \mathbf{y}, \mathbf{z}) = T(S\mathbf{x}, S\mathbf{y}, S\mathbf{z})$. It can be easily seen that T' is a trilinear form and therefore it has a matrix $A' \in \mathbb{F}^{n\times n\times n}$ associated to it. In this section we will prove that $\mathsf{rank}(A) = \mathsf{rank}(A')$. This is a natural generalization of the fact that for quadratic matrices multiplication by an invertible matrix does not change the rank.

We begin by noticing that it suffices to show that $\mathsf{rank}(A') \leq \mathsf{rank}(A)$ since the reverse inequality can be obtained by applying the same argument and considering S^{-1} instead. Let $r = \mathsf{rank}(A)$ and write A as $A = \sum_{\ell=1}^{r} \mathbf{u}_\ell \otimes \mathbf{v}_\ell \otimes \mathbf{w}_\ell$. A simple calculation shows that $A' = \sum_{\ell=1}^{r} S\mathbf{u}_\ell \otimes S\mathbf{v}_\ell \otimes S\mathbf{w}_\ell$, which concludes the claim.

Chapter 3

Quadratic Min-Rank Problem

In this chapter we introduce the Min-Rank problem in its quadratic version. This computational problem was introduced by Buss et al. [BFS99] in the context of linear algebra and proved its NP-completeness.

Many applications of this problem to cryptography have been seen throughout the years. A zero-knowledge proof system was devised in [Cou01], and many modern constructions of code-based schemes using the rank metric have a relation with the Min-Rank problem (see for example [GRS+]). However, it can be said that its major applications lie in the domain of cryptanalysis of multivariate public key cryptosystems. The Min-Rank problem in the context of multivariate cryptography first appeared as part of an attack against the HFE cryptosystem by Kipnis and Shamir [KS99a]. Some generalizations of this attack have been seen during the subsequent years, like the generalization to Multi-HFE [BFP13a] or ZHFE [CSTV17], among others.

The last aforementioned applications will be explored in a bit more detail in Section 5.3. For now we will restrict ourselves to exploring the problem from a purely algebraic perspective, and we will discuss some algorithms that can lead to its solution.

3.1 Basic Definitions

The Min-Rank problem is defined as follows.[1]

(Quadratic) Min-Rank problem, decisional version

Given positive integers n, r, k, and matrices $M_1, \ldots, M_k \in \mathbb{F}^{n \times n}$, determine whether there exist $\lambda_1, \ldots, \lambda_k \in \mathbb{F}$ such that the rank of $\sum_{i=1}^{k} \lambda_i M_i$ is less than or equal to r.

It can be shown that this problem is NP-complete (see for example [HL13b]). A bit more precisely, this means that, unless the computational classes P and NP are equal, there is no polynomial-time algorithm that can solve this problem for *any* choice of parameters. However, this does not rule out the existence of some algorithms that can solve this

[1] We restrict ourselves to the scenario in which the matrices are square, even though the problem can be defined for more general rectangular matrices

problem for specific parameters. In fact, we will discuss below some algorithms that have a reasonable performance when the rank r is small enough.

We will be mostly dealing with the search version of this problem, stated below.

(Quadratic) Min-Rank problem, search version

Given positive integers n, r, k, and matrices $M_1, \ldots, M_k \in \mathbb{F}^{n \times n}$, find $\lambda_1, \ldots, \lambda_k \in \mathbb{F}$ such that the rank of $\sum_{i=1}^{k} \lambda_i M_i$ is less than or equal to r, if they exist.

3.2 Some Algorithms

The basic approach to solve the (search) Min-Rank problem is to regard each coefficient λ_i as an unknown and then use the fact that the rank of $\sum_{i=1}^{k} \lambda_i M_i$ must be smaller than r coupled with some characterization of rank to derive some equations whose solution yield the desired coefficients. These equations will be multivariate polynomials, and solving this type of equations is not necessarily an easier task at all. However, different approaches yield different set of parameters for these equations, which may be efficiently solvable in some specific scenarios.

To distinguish between the actual solutions $\lambda_i \in \mathbb{F}$ and the variables used to set up the systems of equations, we write $A = \sum_{i=1}^{k} t_i M_i$ where each t_i is a variable, so that each entry of A lies in the polynomial ring $\mathbb{F}[t_1, \ldots, t_k]$.

3.2.1 The Kipnis-Shamir Algorithm

We know from the rank-nullity characterization of rank that $\mathsf{rank}(A) \leq r$ if and only if the dimension of its right kernel is at least $n - r$. This is equivalent to the existence of $n - r$ linearly independent vectors in the kernel of A, which by taking the appropriate linear combinations can be assumed to have the form of the columns of the following matrix

$$K = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \\ y_{11} & y_{12} & \cdots & y_{1,n-r} \\ y_{21} & y_{22} & \cdots & y_{2,n-r} \\ \vdots & \vdots & \ddots & \vdots \\ y_{r1} & y_{r2} & \cdots & y_{r,n-r} \end{pmatrix} \tag{3.1}$$

where the y's are unknowns/variables.

Since these vectors are in the kernel of A, this gives rise to the matrix equation $A \cdot K = 0$, which can be translated (by looking at each entry) to $n \cdot (n - r)$ quadratic equations in the t's and y's, for a total of $k + r \cdot (n - r)$ variables.

3.2.2 Guessing Kernel Vectors

As with any system of equations, it is possible to guess some variables in (3.1) and solve for the others. Because of the structure of this system, it is particularly appealing to guess kernel vectors (i.e. the $y_{i,j}$ variables) and solve the resulting linear system in the t_i variables, as proposed in [GC00] (in fact, since the linear system is very overdetermined, it is enough to guess k/n kernel vectors). The complexity of such attack is dominated by the guessing part and depends on the probability of a correct guess. A tight bound on this probability can be significantly improved by understanding the structure of the solution space, e.g. by exploiting the interlinked kernel structure [YC05] or by using the subspace differential invariant structure [MPST14].

3.2.3 Minors Modeling

In [FLdVP08], Faugère et al. introduced the minors method approach to solve the MinRank problem and in [BFP13b] they improved the MinRank attack on HFE using this modeling. This approach uses the characterization of rank using the minors: the rank of A is at most r if and only if every minor of size $r+1$ is zero (recall that a minor of size ℓ of a matrix is the determinant of an $\ell \times \ell$ submatrix obtained by taking a subset of ℓ rows and ℓ columns).

By applying this characterization to the matrix A as defined above, we can derive one equation for each minor by setting it equal to zero. Each of these equations is homogeneous of degree $r+1$, and the number of $(r+1)$-minors in A is $\binom{n}{r+1}^2$.

3.2.4 Using Tensor Decomposition

Recall that the rank of $A \in \mathbb{F}^{n\times n}$ can be defined as the minimum number r such that we can write A as $A = \sum_{i=1}^{r} \mathbf{u}_i \mathbf{v}_i^\intercal$. Treating each entry of each $\mathbf{u}_i, \mathbf{v}_i$ as variables, we get a quadratic system of n^2 equations and $2 \cdot r \cdot n + k$ variables.

3.2.5 Using the Factorization Rank

Recall that $\mathsf{rank}(A)$ can be defined as the minumum number r such that A can be factored as $A = C \cdot F$ where $C \in \mathbb{F}^{n\times r}$ and $F \in \mathbb{F}^{r\times n}$. By column-reducing the matrix C it is possible to assume that the upper $r \times r$ block of C is the identity matrix. We can then let the coefficients of C and F be unkowns and solve the matrix equation $A = C \cdot F$.

If $A \in \mathbb{F}^{n\times n}$ is symmetric then the decomposition boils down to $A = R^\intercal S R$ where $R \in \mathbb{F}^{r\times n}$ and $S \in \mathbb{F}^{r\times r}$ is an invertible matrix.

Chapter 4

Cubic Min-Rank Problem

In the previous chapter we focused on the Min-Rank problem in its quadratic version. As we will see in Section 5.3, this is an important tool in the analysis of some multivariate cryptographic schemes.

However, it is natural to wonder if this computational problem also makes sense in the cubic case. In this chapter we show a natural extension of this problem to the three-dimensional setting by using our generalized definition of rank for these type of matrices from Section 2.2. Then we introduce some algorithms to solve this problem.

We will see in Section 6 some applications of this problem to the cryptanalysis of multivariate public key cryptosystems. The main takeaway is that shifting from a quadratic setting to a cubic one does not rule out completely the possibility of an attack involving the Min-Rank problem.

4.1 Basic Definitions

The three-dimensional Min-Rank problem is defined as follows.

Three-Dimensional Min-Rank problem, decisional version

Given positive integers n, r, k, and matrices $M_1, \ldots, M_k \in \mathbb{F}^{n \times n \times n}$, determine whether there exist $\lambda_1, \ldots, \lambda_k \in \mathbb{F}$ such that the rank of $\sum_{i=1}^{k} \lambda_i M_i$ is less than or equal to r.

It can also be shown that this problem is NP-complete (See for example [HL13b]).

Just like in the quadratic case, we will be mostly dealing with the search version of this problem:

Three-Dimensional Min-Rank problem, search version

Given positive integers n, r, k, and matrices $M_1, \ldots, M_k \in \mathbb{F}^{n \times n \times n}$, find $\lambda_1, \ldots, \lambda_k \in \mathbb{F}$ such that the rank of $\sum_{i=1}^{k} \lambda_i M_i$ is less than or equal to r, if they exist.

4.2 Solving the Three-Dimensional Min-Rank Problem

Our approach for solving the three-dimensional min-rank problem is essentially the same we took in Section 3.2 for the quadratic version of the problem: we consider each unknown λ_i as a variable t_i and then use some equivalent definitions of cubic rank to obtain a system of equations. However, the problem in this setting is that such characterizations are scarce and also not very friendly computational-wise.

4.2.1 Using the Tensor-Rank Definition

The first natural approach is to use the definition of cubic rank directly. This is akin to the algorithm we provided in Section 3.2.4. Let $A = t_1 M_1 + \cdots + t_k M_k$, where the t_i's are variables (so each entry of A is a linear polynomial in the t_i's). We know by the definition of rank that $\mathsf{rank}(A) \leq r$ if and only if there exist $\mathbf{u}_i, \mathbf{v}_i, \mathbf{w}_i \in \mathbb{F}^n$ for $i = 1, \ldots, r$ such that $A = \sum_{i=1}^{r} \mathbf{u}_i \otimes \mathbf{v}_i \otimes \mathbf{w}_i$. By regarding each $\mathbf{u}_i, \mathbf{v}_i, \mathbf{w}_i$ as a vector of unknowns we obtain a system of n^3 cubic equations. The total number of variables is $3 \cdot n \cdot r + k$.

4.2.2 A Generalization of the Kipnis-Shamir Modeling

We know from Theorem 2.2.1 that A is of rank r, if and only if, there exist rank one matrices $S_1, \ldots, S_r \in \mathbb{F}^{n \times n}$, such that, for $i = 1, \ldots, n$, $A[i, \cdot, \cdot] \in \mathrm{span}(S_1, \ldots, S_r)$. Since each S_i matrix is of rank one, we can write it as $S_i = \mathbf{u}_i \mathbf{v}_i^T$ for some vectors $\mathbf{u}_i, \mathbf{v}_i \in \mathbb{F}^n$. Considering the entries of the $\mathbf{u}_i$'s, $\mathbf{v}_i$'s, and the linear combination coefficients as variables yields a cubic system of n^3 equations in $r(2n) + rn + k = 3rn + k$ variables

$$\sum_{j=1}^{r} \alpha_{ij} \mathbf{u}_j \mathbf{v}_j^T = A[i, \cdot, \cdot], \text{ for } i = 1, \ldots, n. \tag{4.1}$$

To the best of our knowledge, the complexity of solving a system such as (4.1) has not been studied. It can be seen as a multi-homogeneous system of multi-degree $(1, 1, 1, 1)$, i.e. a tetra-linear system, and assuming some notion of tetra-regularity, analyze it using the techniques in [FDS11]. It should be noticed that the techniques in [FDS11] do not address the semi-regularity inherent to such an overdetermined system. Alternatively, the techniques in [BFSY05] could be used to establish the asymptotic behavior of an upper bound of the degree of regularity based on the semi-regularity assumption. Although a complete asymptotic analysis is outside the scope of this thesis, Table 4.1 shows the growth of such bound for selected parameters.

n	r	vars	eqns	d-reg	cpx
10	10	310	1000	67	699
11	11	374	1331	74	798
12	12	444	1728	81	899
13	13	520	2197	89	1010
14	14	602	2744	97	1123
15	15	690	3375	105	1240

Table 4.1: Complexity of MR by KS modeling for cubic system. For different values of n, KS yields a cubic system of n^3 equations in $(3r+1)n$ variables (assuming $k=n$). The d-reg column gives the degree of regularity for such a semi-regular system without field equations. The complexity column, gives the log base 2 of $\binom{vars+d-1}{d}^{2.8}$.

4.2.3 Improvement of KS for $r \ll n$

If $r \ll n$ we can do much better. In that case, for most such rank r matrices A, the first r slices $A[1,\cdot,\cdot],\ldots,A[r,\cdot,\cdot]$ are linearly independent. In this case, $\mathrm{span}(S_1,\ldots,S_r) = \mathrm{span}(A[1,\cdot,\cdot],\ldots,A[r,\cdot,\cdot])$. Then, for $i=r+1,\ldots,n$, $A[i,\cdot,\cdot] \in \mathrm{span}(A[1,\cdot,\cdot],\ldots,A[r,\cdot,\cdot])$. Considering the coefficients of the linear combinations as variables yields a system of $n^2(n-r)$ quadratic equations in $(n-r)r+k$ variables

$$\sum_{j=1}^{r} \alpha_{ij} A[j,\cdot,\cdot] = A[i,\cdot,\cdot], \text{ for } i=r+1,\ldots,n. \tag{4.2}$$

Notice that the converse is not necessarily true. A solution to the system in (4.2) does not necessarily implies the existence of the rank one S_i matrices, neither that A has rank r. However, this is a very overdetermined system, hence a solution is very unlikely, unless indeed A has rank r.

The system in (4.2) has $\mathcal{O}(n^3)$ quadratic equations in $\mathcal{O}(n)$ variables. Since the number of degree two monomials is $\mathcal{O}(n^2)$ the system can be solved by relinearization at degree 2, which reduces to solving a $\mathcal{O}(n^2)$ square matrix. Notice that this is much faster than the KS approach in the two-dimensional case since in this case we have many more equations. This is very surprising, since it essentially says that the Min-Rank problem becomes easier in degree 3 if the rank is small enough.

4.3 Slices and Differentials

It is reasonable to wonder if a cubic instance of the Min-Rank problem can be transformed into a quadratic instance, for which the algorithms from Section 3 can be applied. One way to achieve this is by taking the slices $M_\ell[i,\cdot,\cdot]$ of each matrix in the instance. In this section we explore how viable this approach is by first showing that it is directly related to the concept of the differential of a polynomial. Then we show that in general this differential does not necessarily preserve the rank from the original instance, which will render this approach useless for most of the cases.

4.3.1 Relation between Slices and Differentials

Let $A \in \mathbb{F}^{n\times n\times n}$ be a symmetric matrix and let $f(\mathbf{x}) \in \mathbb{F}[\mathbf{x}]$ be the homogeneous cubic polynomial represented by A, i.e. $f(\mathbf{x}) = \sum_{i,j,k} A[i,j,k] \cdot x_i x_j x_k$. We begin by showing that the slices $A[i,\cdot,\cdot]$ have a direct relation with the differential of f. Such differential is defined by $D_{\mathbf{a}}f(\mathbf{x}) := f(\mathbf{x}+\mathbf{a}) - f(\mathbf{x}) - f(\mathbf{a})$.

In general, if f is a cubic homogeneous polynomial, then $D_{\mathbf{a}}f(\mathbf{x})$ is a quadratic polynomial in $\mathbf{x}$, but not necessarily homogeneous. Let us write $D_{\mathbf{a}}f(\mathbf{x}) = g(\mathbf{x}) + h(\mathbf{x})$ where g is quadratic homogeneous and h is linear.[1] On the other hand, consider the symmetric matrix $A' \in \mathbb{F}^{n\times n}$ representing the polynomial $g(\mathbf{x})$. As we saw in Section 2.3.1, the symmetric bilinear map $B(\mathbf{x},\mathbf{y})$ associated to A' can be computed as $B(\mathbf{x},\mathbf{y}) = \frac{1}{2}(g(\mathbf{x}+\mathbf{y}) - g(\mathbf{x}) - g(\mathbf{y}))$. Moreover, using the fact that $g(\mathbf{x}) = D_{\mathbf{a}}f(\mathbf{x}) - h(\mathbf{x})$ and that $h(\mathbf{x})$ is linear, it can be obtained that $B(\mathbf{x},\mathbf{y}) = \frac{1}{2}(D_{\mathbf{a}}f(\mathbf{x}+\mathbf{y}) - D_{\mathbf{a}}f(\mathbf{x}) - D_{\mathbf{a}}f(\mathbf{y}))$. Finally, by unfolding the definition of the differential in terms of f, we see that

$$B(\mathbf{x},\mathbf{y}) = \frac{1}{2}(f(\mathbf{x}+\mathbf{y}+\mathbf{a}) - f(\mathbf{x}+\mathbf{y}) - f(\mathbf{x}+\mathbf{a}) - f(\mathbf{y}+\mathbf{a}) + f(\mathbf{x}) + f(\mathbf{y}) + f(\mathbf{a})).$$

On the other hand, if $T(\mathbf{x},\mathbf{y},\mathbf{a})$ is the trilinear form associated to the matrix A, we know from Section 2.3.2 that T can be computed from f as

$$T(\mathbf{x},\mathbf{y},\mathbf{a}) = \frac{1}{3!}(f(\mathbf{x}+\mathbf{y}+\mathbf{a}) - f(\mathbf{x}+\mathbf{y}) - f(\mathbf{x}+\mathbf{a}) - f(\mathbf{y}+\mathbf{a}) + f(\mathbf{x}) + f(\mathbf{y}) + f(\mathbf{a})).$$

By joining these two expressions we obtain that $B(\mathbf{x},\mathbf{y}) = 3 \cdot T(\mathbf{x},\mathbf{y},\mathbf{a})$. When translating this in terms of slices, keeping in mind that $A[i,j,k] = T(\mathbf{e}_i,\mathbf{e}_j,\mathbf{e}_k)$, we obtain that $A[i,j,k] = \frac{1}{3}B(\mathbf{e}_i,\mathbf{e}_j)$. In other words, this means that the bidimensional matrix representing the quadratic homogeneous part of the differential of f at $\mathbf{a} = \mathbf{e}_k$ is precisely the slice $A[\cdot,\cdot,k]$ up to a factor of 3. For the general case $\mathbf{a} \in \mathbb{F}^n$ we simply notice that $D_{\mathbf{a}}f(\mathbf{x}) = \sum_{i=1}^n a_i D_{\mathbf{e}_i}f(\mathbf{x})$, so the matrix representing the quadratic homogeneous part of $D_{\mathbf{a}}f(\mathbf{x})$ is given by $\sum_{k=1}^n a_k \cdot A[\cdot,\cdot,k]$.

Notice that in the previous argument we used the fact that A was symmetric to argue that A was the matrix representing the symmetric trilinear form T. For a general A, we would have that $\frac{1}{3!}\left(\sum_{\pi\in\Pi_3}\pi(A)\right)$ is the actual symmetric matrix representing the trilinear form T, so the argument would be applied to this matrix instead.

4.3.2 Rank of the Differential

Assume that $A \in \mathbb{F}^{n\times n\times n}$ is a symmetric matrix. From the previous section we see that the rank of the quadratic part of the differential of a cubic polynomial f at $\mathbf{a}$ is the rank of $\sum_{k=1}^n a_k \cdot A[\cdot,\cdot,k]$. Moreover, the latter has rank upper-bounded by $\mathsf{rank}(A)$. This holds since, if $A = \sum_{\ell=1}^{\mathsf{rank}(A)} \mathbf{u}_\ell \otimes \mathbf{v}_\ell \otimes \mathbf{w}_\ell$, then

$$\sum_{k=1}^n a_k A[\cdot,\cdot,k] = \sum_{k=1}^n a_k \sum_{\ell=1}^{\mathsf{rank}(A)} (\mathbf{u}_\ell \otimes \mathbf{v}_\ell) \cdot \mathbf{w}_\ell[k] = \sum_{\ell=1}^{\mathsf{rank}(A)} (\mathbf{u}_\ell \otimes \mathbf{v}_\ell) \cdot \left(\sum_{k=1}^n a_k \mathbf{w}_\ell[k]\right).$$

[1] The free coefficient of $D_{\mathbf{a}}f(\mathbf{x})$ must be equal to 0 since $D_{\mathbf{a}}f(\mathbf{0}) = f(\mathbf{0}+\mathbf{a}) - f(\mathbf{0}) - f(\mathbf{a}) = 0$

All in all, we conclude that the rank of the differential is upper-bounded by the rank of A, so taking the differential cannot increase the rank. This shows that a good strategy for solving the three-dimensional Min-Rank problem when $r \ll n$ is taking the differential of each of the matrices of the instance and solving the resulting quadratic min-rank problem (however, recall from Section 4.2.3 that in this case it is more efficient to run directly on the cubic instance). Nevertheless, this is prohibitive if r is close to or greater than n, and if the differential happens to have a rank close to this. It is natural to ask then if this is the case: Is the rank of the differential much smaller than the rank of the original three-dimensional matrix? In what is left of the section we will answer this question heuristically by showing experimental results that indicate that in general the rank of the differential tends to stay close to the original rank. This shows that in general, transforming a cubic instance of the min-rank problem into a quadratic one by applying the differential does not necessarily yield an easier computational problem.

Lower Bound for the Rank of the Differentials

The first thing to note is that a general good lower bound cannot be provided, since there are cubic matrices for which the rank of the differential drops by an order of a square root. For example, let $A' \in \mathbb{F}^{r\times r\times r}$ be a matrix of maximal rank. As we saw in Section 2.2, it is known that the rank of such a matrix is of the order $O(r^2)$. Then, consider the matrix $A \in \mathbb{F}^{n\times n\times n}$ given by $A[i,j,k] = A'[i,j,k]$ if $i,j,k \leq r$, and 0 otherwise. It can be seen that $\mathsf{rank}(A) = \mathsf{rank}(A') = O(r^2)$. However, a slice of A is just a matrix with only an $r \times r$ non-zero block in the upper-left corner, so its rank is upper-bounded by r. As we will see in Section 6.3, this is precisely the situation in the cryptosystem HFE, and this is the reason why this encryption scheme, even in its cubic form, is vulnerable to a quadratic Min-Rank attack.

Given the above, our analysis must be probabilistic, in the sense that we should consider the average case instead of the worst case. Therefore, we formulate our question as follows: given a random homogeneous cubic polynomial $f \in \mathbb{F}[\mathbf{x}]$ of rank r, we want to estimate the rank of the quadratic part of its differential $D_{\mathbf{a}}f(\mathbf{x}) = f(\mathbf{x}+\mathbf{a}) - f(\mathbf{x}) - f(\mathbf{a})$.

The first and main problem that we face in our analysis is: given an integer r, how can we generate random homogeneous cubic polynomials of rank r? Or equivalently, how can we generate random symmetric three-dimensional matrices of rank r? To address these questions, we use the concept of symmetric rank. We then choose random polynomials and use Kruskal's theorem to guarantee that those polynomials have certain symmetric rank.

By Proposition 2.2.3, if $|\mathbb{F}| \geq 3$, any homogeneous cubic polynomial f can be written as $\sum_{i=1}^{k} t_i u_i(\mathbf{x}) u_i(\mathbf{x}) u_i(\mathbf{x})$, where each $u_i(\mathbf{x})$ is a homogeneous linear polynomial and k depends on f. Furthermore, the symmetric rank of a homogeneous cubic $f \in \mathbb{F}[\mathbf{x}]$, denoted by $\mathsf{SRank}(f)$ and defined as the symmetric rank of its symmetric matrix representation, does exist.

The symmetric rank is a good parameter to consider because it is a bound for the rank of the differential.

Proposition 4.3.1. *Let $f \in \mathbb{F}[\mathbf{x}]$ be a homogeneous cubic polynomial. If g is the quadratic homogeneous part of $Df_{\mathbf{a}}(\mathbf{x})$, then $\mathsf{rank}(g) \leq \mathsf{SRank}(f)$.*

Proof. If f can be written as $f(\mathbf{x}) = \sum_{i=1}^{r} t_i u_i(\mathbf{x}) u_i(\mathbf{x}) u_i(\mathbf{x})$, then for any $\mathbf{a} \in \mathbb{F}^n$ the quadratic part of $Df_{\mathbf{a}}(\mathbf{x})$ is $\sum_{i=1}^{r} 3t_i u_i(\mathbf{a}) u_i(\mathbf{x}) u_i(\mathbf{x})$. □

Let $U = \mathbb{F}^n$. Clearly, each symmetric matrix $A \in \mathbb{F}^{n \times n \times n}$ with symmetric rank r can be written as a sum of exactly r terms of the form $t\mathbf{u} \otimes \mathbf{u} \otimes \mathbf{u}$, where $t \in \mathbb{F} - \{0\}$ and $\mathbf{u} \in U$.

Let $\mathcal{S}_r$ be the function which outputs $A = \sum_{i=1}^{r} t_i \mathbf{u}_i \otimes \mathbf{u}_i \otimes \mathbf{u}_i$, for $t_i \in \mathbb{F} - \{0\}$ and $\mathbf{u}_i \in U$. By Proposition 2.2.3, if $|\mathbb{F}| \geq 3$, then each symmetric matrix $A \in \mathbb{F}^{n \times n \times n}$ with symmetric rank equal to r is in the codomain of $\mathcal{S}_r$. But some symmetric matrices having symmetric rank less than r can also be there.

Now we will use Theorem 2.2.2 to argue that if $t_i \in \mathbb{F} - \{0\}$ and $\mathbf{u}_i \in U$ are chosen uniformly at random, then with high probability the corresponding output A of $\mathcal{S}_r$ has symmetric rank equal to r. Moreover, by Kruskal's theorem with high probability $\mathsf{rank}(A) = \mathsf{SRank}(A)$. The argument is as follows. Suppose $2 \leq r \leq n$. If $\mathbf{u}_1, \ldots, \mathbf{u}_r \in U$ are chosen uniformly at random, then with high probability a matrix with columns $\mathbf{u}_1, ..., \mathbf{u}_r$ has full rank. If a matrix with columns $\mathbf{u}_1, \ldots, \mathbf{u}_r \in U$ is full rank, then $\mathsf{KRank}(\mathbf{u}_1, \ldots, \mathbf{u}_r) = r$ and $\mathsf{KRank}(t_1\mathbf{u}_1, \ldots, t_r\mathbf{u}_r) = r$, for any $t_1, \ldots, t_r \in \mathbb{F} - \{0\}$. In this case, by Theorem 2.2.2 the corresponding output A of $\mathcal{S}_r$ is such that $\mathsf{rank}(A) = \mathsf{SRank}(A) = r$.

We experimentally analyze the behavior of the rank of the differential of a polynomial that is the output of $\mathcal{S}_{r^2}$. The experimental results are shown in Figure 4.1, where each curve represents the percentage of times that a rank is obtained, over 100,000 iterations.

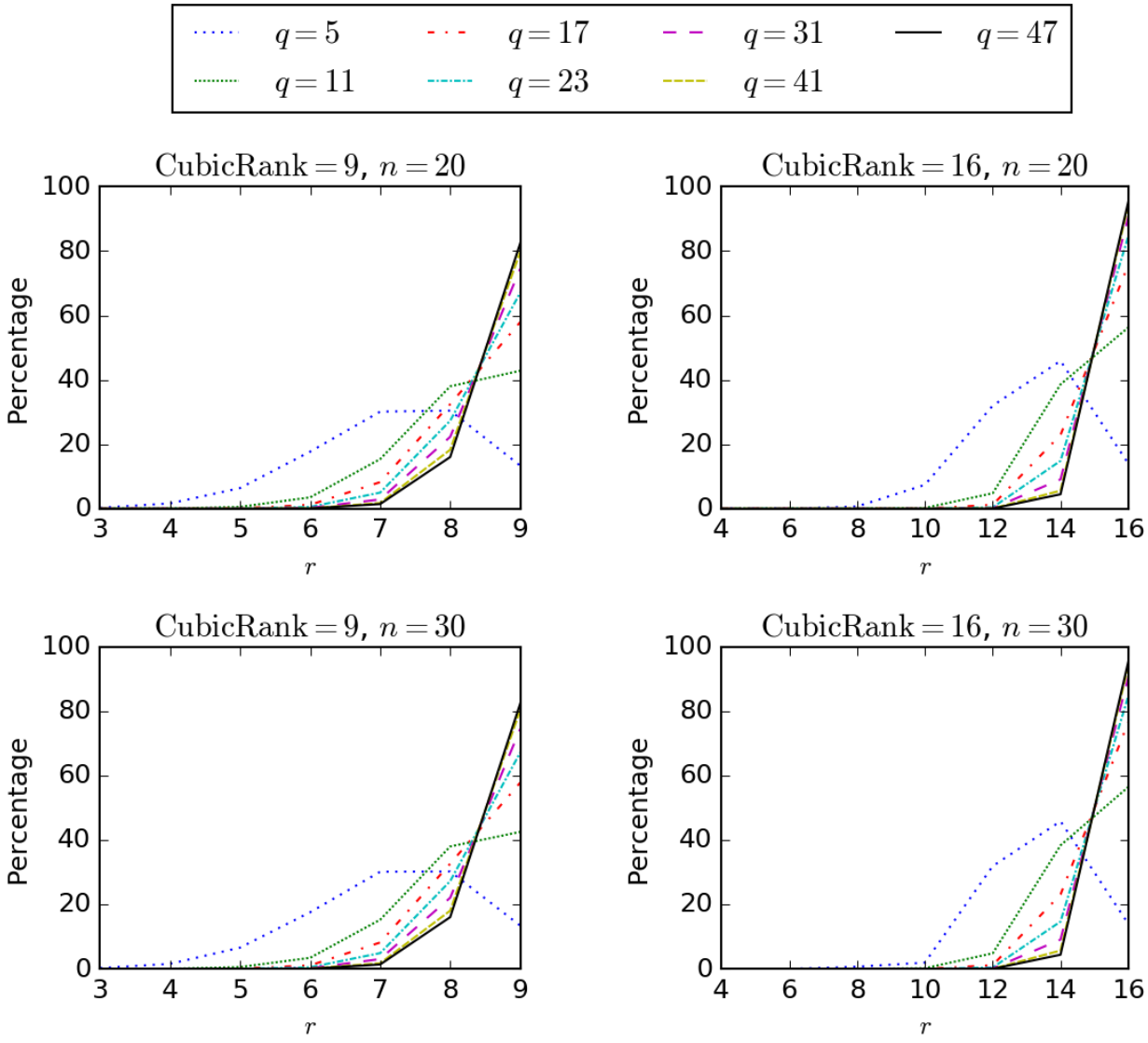


Figure 4.1: For different values of q, CubicRank, and n, a polynomial f is chosen according to $\mathcal{S}_{\text{CubicRank}}$, the $\mathsf{rank}(Df_{\mathbf{a}})$ is computed for a random $\mathbf{a} \in \mathbb{F}^n$. Each graph represents the percentage of times that a particular value $\mathsf{rank}(Df_{\mathbf{a}})$ is obtained over 100,000 iterations.

Part II

Applications to Multivariate Public-Key Cryptography

Chapter 5

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 necessary context to the New Alternatives we propose later on in this work.

5.1 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.

5.1.1 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]

[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 cannot be found from $\mathcal{P}$ since in this case, anyone with access to this function would be able to invert just like Bob.

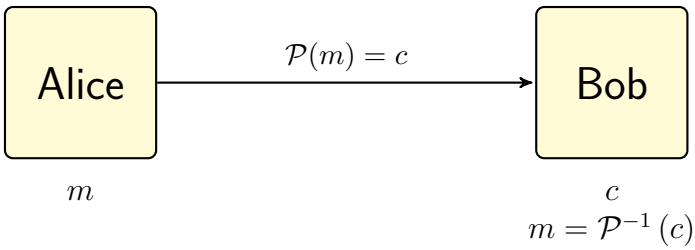


Figure 5.1: Protocol that allows Alice send her message to Bob securely

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.

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 5.1 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 to 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 in which these 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 cannot 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

[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

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 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.

5.1.2 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.

The 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 on different problems that cannot 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.

5.2 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. We denote by $R_{\leq d}$ 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 [Esc16]), 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 illustrated in the introduction which make use of Groebner bases. It can be shown that for random systems this approach has an expected exponential complexity in n (see for example [Spa12]). Moreover, nowadays there are no known polynomial-time quantum algorithms 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 from n. The 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 harmful if $m = O(n)$.

Also, 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 restrict ourselves to quadratic polynomials in the exposition of the general constructions.

5.2.1 First Reduction: Bipolar Construction

Definition. Given a regular function $F : \mathbb{F}^n \to \mathbb{F}^m$, invertible linear transformations $S : \mathbb{F}^n \to \mathbb{F}^n$ and $T : \mathbb{F}^m \to \mathbb{F}^m$, 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 equations, therefore hiding the structure of the function F. Figure 5.2 shows how the process works.

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 mappings. This problem is assumed to be

[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

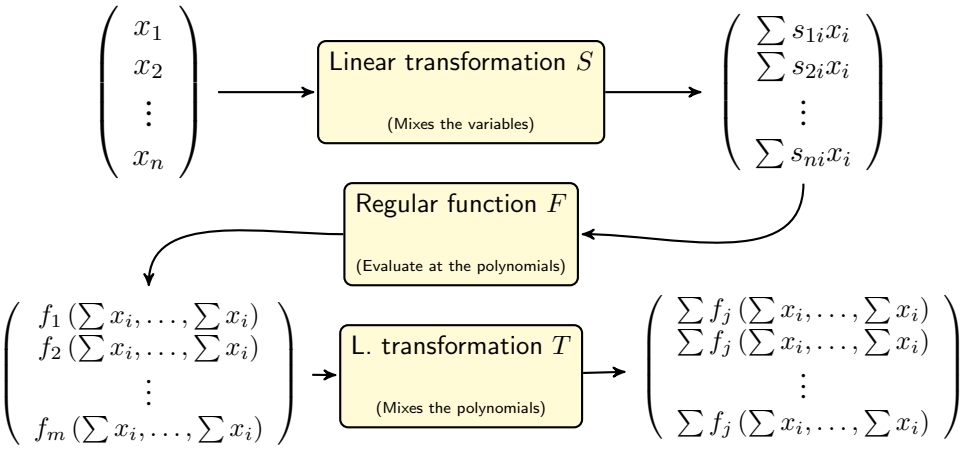


Figure 5.2: Construction of MPK Cryptosystems from easy-to-invert regular functions

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 the Min-Rank attack.

An important concern is that we cannot 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 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, bipolar constructions inherit the falling degree from F and hence P would be easy to invert for anyone as well (see [Esc16] for the details on this). 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 it is easy to invert.

In any case, in many scenarios 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 trapdoor 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.

5.2.2 Second Reduction: Lifting Idea

According to the previous section, now we need to focus on 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 Chapter 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. As an example, notice that 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 5.2.1. (Correspondence of Polynomials, restated). *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

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

whose inverse is

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

The proof of this theorem can be found in Section 1.5.2.

The names Lft (lifting) and Drp (dropping) arise from the following commutative diagram, which illustrates the correspondence.

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

Clearly, F is invertible if and only if $\mathcal{F}$ is, so we can focus now on 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 1.5.3 shows that this procedure can be done very efficiently.

5.2.3 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 invertible 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 the 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 $P_i = T_i \circ \mathsf{Drp}(\mathcal{F}) \circ S$.

This construction is depicted in Figure 5.3.

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 to a much larger value one gets sizes beyond practical applications.[5]

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

- In order for this construction to be efficient one needs to be able to compute $\mathsf{Drp}(\mathcal{F})$ from $\mathcal{F}$ in an efficient manner. This is well known in the quadratic case, as we described in Section 1.5.3.

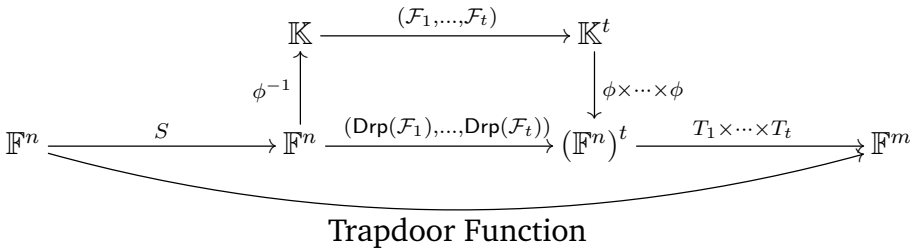


Figure 5.3: General Construction of Multivariate Trapdoor Functions

5.3 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 Min-Rank attack [KS99b]. It was a theoretical attack back then, but subsequent work by L. Perret et al [BFP13a] improved this attack for any set of practical parameters.

On the other hand, ZHFE was proposed as an alternative to avoid the Min-Rank 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.

5.3.1 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 easily 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 invertible linear transformations $S, T : \mathbb{F}^n \to \mathbb{F}^n$ and computing $P = T \circ \phi \circ \mathcal{F} \circ \phi^{-1} \circ S$.

Security Analysis

The HFE Cryptosystem has a vulnerability against what is known as a Min-Rank attack. This attack reduces the problem of finding the secret key[6] to the Min-Rank problem. Since we will encounter the same type of attack in the next chapter when we generalize it to the cubic setting, it is worth to see the most relevant aspects of it.

We begin by writing 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 $X^{q^{n-1}}$). Let us denote the matrix in the middle by $M \in \mathbb{K}^{n \times n}$. Notice that M has a low rank r (since D is small, by construction).

Now, recall from Section 1.5.3 that if $A_i \in \mathbb{F}^{n \times n}$ is the quadratic matrix representing the i-th component of $\mathsf{Drp}(\mathcal{F}) = \phi \circ \mathcal{F} \circ \phi^{-1}$, then $\Delta^\intercal M \Delta = \sum_{i=1}^n y^{i-1} A_i$. Moreover, it is easy to check that the effect of composing with the matrix $S \in \mathbb{F}^{n \times n}$ from the right gives $\phi \circ \mathcal{F} \circ \phi^{-1} \circ S = (\Delta S)^\intercal M (\Delta S)$.[7] Finally, the matrix P_i representing the i-th component of $P = T \circ \phi \circ \mathcal{F} \circ \phi^{-1} \circ S$ is given by $P_i = \sum_{j=1}^n T[i,j] \cdot (S^\intercal A_j S)$.

As an important consequence, we see that the following relation holds

$$(\Delta S)^\intercal M (\Delta S) = \sum_{i=1}^n \lambda_i P_i, \tag{5.1}$$

where $(\lambda_1, \ldots, \lambda_n) = (y^0, \ldots, y^{n-1}) T^{-1}$. This holds since

$$\sum_{i=1}^n \lambda_i P_i = \sum_{i=1}^n \lambda_i \left(\sum_{j=1}^n T[i,j] \cdot (S^\intercal A_j S) \right) = \sum_j S^\intercal A_j S \sum_i \lambda_i T[i,j] = \sum_j y^{j-1} S^\intercal A_j S.$$

The important observation about Equation (5.1) is that the left-hand side (and therefore, the right-hand side) has the same rank as M, which is at most r and that has to be

[6] In fact, an *equivalent* secret key is recovered, which is a secret key that may not be the one used originally to create the given public key, but that also works for decryption. This will not matter much for our discussion.

[7] This follows from the fact that composing a quadratic polynomial represented by a matrix A with a linear transformation S gives again a quadratic polynomial represented by $S^\intercal A S$

low in order to keep decryption efficient. This gives an instance of the Min-Rank problem that can be tackled using the techniques from Chapter 3. The first negative implication of this property is that the trapdoor functions from HFE are distinguishable from random regular functions, which is undesirable. Secondly, this attack allows an attacker to find the coefficients $\lambda_1, \ldots, \lambda_n$ and it turns out that this is enough to build an equivalent secret key (for the details see for example [BFP13b]).

5.3.2 ZHFE

It is worth mentioning ZHFE, which appeared in the literature as an alternative to overcome the Min-Rank 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).

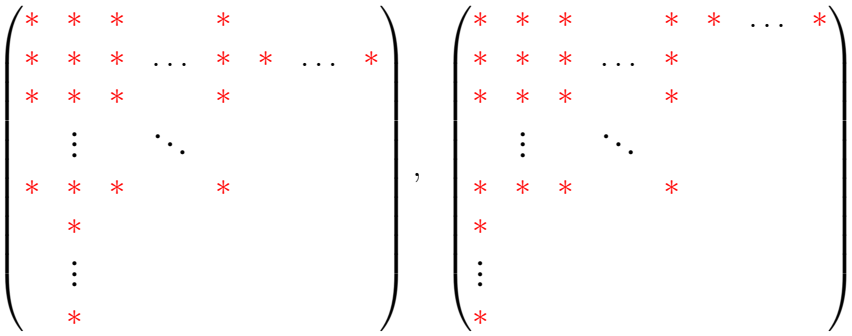
Security Analysis

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



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, CSTV17], and this cryptosystem is unfortunately insecure.

Chapter 6

Rank Analysis of Cubic Polynomials

6.1 Min-Rank Analysis for Cubic Big Field Constructions

As we pointed out in Section 5.2, the Big Field Idea has been a basis to propose quadratic multivariate encryption schemes for decades. Nevertheless, Theorem 1.5.2 is not restricted to any particular degree, which means that this approach works to generate polynomials of any degree, in particular degree 3. In this section we show that if the central map is a low rank cubic polynomial, then, as in the quadratic case, there must exists a low-rank linear combination of the polynomials of the public key. In particular, we obtain an instance of the cubic Min-Rank problem which can be solved using the techniques presented in Section 4.2.

6.1.1 Big Field Idea for Cubic Polynomials

Let $\mathcal{F} \in \mathbb{K}[X]$ be a homogeneous weight 3 polynomial given by

$$\mathcal{F}(X) = \sum_{1 \leq i,j,k \leq n} \alpha_{i,j,k} X^{q^{i-1}+q^{j-1}+q^{k-1}}$$

and $S, T \in \mathbb{F}^{n \times n}$ invertible matrices. Our first goal is to give an explicit expression for the multivariate cubic polynomials of the composition $T \circ \phi \circ \mathcal{F} \circ \phi^{-1} \circ S$. We begin by representing the map $\mathcal{F}$ as $\mathcal{F}(X) = \mathcal{T}(\mathbf{X}, \mathbf{X}, \mathbf{X})$ where $\mathbf{X} = (X^{q^0}, \ldots, X^{q^{n-1}})^\intercal$ and $\mathcal{T} : \mathbb{K}^n \times \mathbb{K}^n \times \mathbb{K}^n \to \mathbb{K}$ is the trilinear form given by

$$\mathcal{T}(\boldsymbol{\beta}, \boldsymbol{\delta}, \boldsymbol{\gamma}) = \sum_{1 \leq i,j,k \leq n} \alpha_{i,j,k} \cdot (\beta_i \delta_j \gamma_k).$$

Let A be the three-dimensional matrix whose entry (i, j, k) is given by $\alpha_{i,j,k}$, and assume without loss of generality that the matrix A is symmetric (otherwise we can take the matrix whose (i, j, k) entry is given by $\frac{1}{3!} \cdot (A[i, j, k] + A[i, k, j] + A[j, i, k] + A[j, k, i] + A[k, i, j] + A[k, j, i])$, which represents the same trilinear form $\mathcal{T}$).

Let $\mathcal{T}' : \mathbb{K}^n \times \mathbb{K}^n \times \mathbb{K}^n \to \mathbb{K}$ be the trilinear form given by $\mathcal{T}'(\boldsymbol{\beta}, \boldsymbol{\delta}, \boldsymbol{\gamma}) = \mathcal{T}(\Delta S\boldsymbol{\beta}, \Delta S\boldsymbol{\delta}, \Delta S\boldsymbol{\gamma})$, then we can write this trilinear form as

$$\mathcal{T}'(\boldsymbol{\beta}, \boldsymbol{\delta}, \boldsymbol{\gamma}) = \sum_{1 \leq i,j,k \leq n} \alpha'_{i,j,k} \cdot (\beta_i \delta_j \gamma_k)$$

where $\alpha'_{i,j,k} = \mathcal{T}'(\mathbf{e}_i, \mathbf{e}_j, \mathbf{e}_k) = \mathcal{T}(\Delta S\mathbf{e}_i, \Delta S\mathbf{e}_j, \Delta S\mathbf{e}_k)$.

Let A' be the three-dimensional matrix whose entry (i, j, k) is given by $\alpha'_{i,j,k}$. Notice that this is the cubic version of the matrix $(\Delta S)^\intercal A(\Delta S)$ in Equation 5.1 from Section 5.3. It is easy to see that the matrix A' is symmetric since the matrix A is.

Let $\mathbf{a} \in \mathbb{F}^n$ and let $\alpha = \phi^{-1}(S\mathbf{a})$, we know that $\mathsf{Frob}(\alpha) = \Delta \cdot \phi(\alpha) = \Delta S \cdot \mathbf{a}$ and therefore

$$\begin{aligned}\mathcal{F} \circ \phi^{-1}(S\mathbf{a}) = \mathcal{F}(\alpha) &= \mathcal{T}(\mathsf{Frob}(\alpha), \mathsf{Frob}(\alpha), \mathsf{Frob}(\alpha)) = \mathcal{T}(\Delta S \cdot \mathbf{a}, \Delta S \cdot \mathbf{a}, \Delta S \cdot \mathbf{a}) \\ &= \mathcal{T}'(\mathbf{a}, \mathbf{a}, \mathbf{a}) = \sum_{1 \le i,j,k \le n} \alpha'_{i,j,k} \cdot (a_i a_j a_k).\end{aligned}$$

Let $R_1, \ldots, R_n \in \mathbb{F}^{n \times n \times n}$ be three-dimensional symmetric matrices such that $A' = y^0 \cdot R_1 + y^1 \cdot R_2 + \cdots + y^{n-1} \cdot R_n$, where $y^0, y^1 \ldots y^{n-1}$ is the basis of $\mathbb{K}$ over $\mathbb{F}$. Then

$$\begin{aligned}\mathcal{F} \circ \phi^{-1} \circ S(\mathbf{a}) &= \sum_{1 \le i,j,k \le n} \alpha'_{i,j,k} \cdot (a_i a_j a_k) \\ &= \sum_{1 \le i,j,k \le n} \left(\sum_{\ell=1}^{n} y^{\ell-1} R_\ell[i,j,k] \right) \cdot (a_i a_j a_k) \\ &= \sum_{\ell=1}^{n} y^{\ell-1} \underbrace{\left(\sum_{1 \le i,j,k \le n} R_\ell[i,j,k] \cdot (a_i a_j a_k) \right)}_{t_\ell}.\end{aligned}$$

Since $t_\ell \in \mathbb{F}$, we obtain that $\phi \circ \mathcal{F} \circ \phi^{-1} \circ S(\mathbf{a}) = (t_1, \ldots, t_\ell)^\intercal$, therefore, each cubic polynomial in the composition $\phi \circ \mathcal{F} \circ \phi^{-1} \circ S$ is given by $f_\ell(\mathbf{x}) = \sum_{1 \le i,j,k \le n} R_\ell[i,j,k] \cdot (x_i x_j x_k)$. Finally, when we apply the transformation T we obtain that each cubic polynomial in the composition $P = T \circ \phi \circ \mathcal{F} \circ \phi^{-1} \circ S$ is given by

$$p_\ell(\mathbf{x}) = \sum_{1 \le i,j,k \le n} \left(\sum_{t=1}^{n} T[\ell, t] \cdot R_t[i,j,k] \right) \cdot (x_i x_j x_k).$$

As a conclusion, if we let A_ℓ be the matrix whose entry (i, j, k) is given by $\sum_{t=1}^{n} T[\ell, t] \cdot R_t[i, j, k]$ then we obtain that this is the symmetric matrix corresponding to the ℓ-th polynomial in P. In particular, this shows we can compute efficiently the composition $T \circ \phi \circ \mathcal{F} \circ \phi^{-1} \circ S$ from S, T and $\mathcal{F}$.

6.1.2 Existence of Low Rank Linear Combination

Let us continue with the same setting as before, and let r be the rank of A, which in particular means that A can be written as $\sum_{\ell=1}^{r} \mathbf{u}_\ell \otimes \mathbf{v}_\ell \otimes \mathbf{w}_\ell$. Suppose that $r \ll n$. In this section we prove that there exists a low-rank linear combination of the three-dimensional matrices representing the composition P, and in Section 4.2 we showed how to find such combination.

Recall that the matrix A' was defined as $A'[i, j, k] = \mathcal{T}(\Delta S\mathbf{e}_i, \Delta S\mathbf{e}_j, \Delta S\mathbf{e}_k)$. We claim that the rank of this matrix is at most the rank of A. We show this by exhibiting vectors

$\mathbf{u}'_\ell, \mathbf{v}'_\ell, \mathbf{w}'_\ell \in \mathbb{K}^n$ such that $A' = \sum_{\ell=1}^{r} \mathbf{u}'_\ell \otimes \mathbf{v}'_\ell \otimes \mathbf{w}'_\ell$. Let M be the matrix ΔS, we define $\mathbf{u}'_\ell = \sum_{i=1}^{n} \mathbf{u}_\ell[i] \cdot M[i,\cdot]$, $\mathbf{v}'_\ell = \sum_{i=1}^{n} \mathbf{v}_\ell[i] \cdot M[i,\cdot]$ and $\mathbf{w}'_\ell = \sum_{i=1}^{n} \mathbf{w}_\ell[i] \cdot M[i,\cdot]$, then

$$
\begin{aligned}
&A'[i',j',k'] \\
&\quad = \mathcal{T}'(M\mathbf{e}_{i'}, M\mathbf{e}_{j'}, M\mathbf{e}_{k'}) \\
&\quad = \sum_{1\le i,j,k\le n} A[i,j,k] \cdot \big((M\mathbf{e}_{i'})[i] \cdot (M\mathbf{e}_{j'})[j] \cdot (M\mathbf{e}_{k'})[k]\big) \\
&\quad = \sum_{1\le i,j,k\le n} \left(\sum_{\ell=1}^{r} \mathbf{u}_\ell[i] \cdot \mathbf{v}_\ell[j] \cdot \mathbf{w}_\ell[k]\right) \big((M[i,\cdot]\mathbf{e}_{i'}) \cdot (M[j,\cdot]\mathbf{e}_{j'}) \cdot (M[k,\cdot]\mathbf{e}_{k'})\big) \\
&\quad = \sum_{\ell=1}^{r} \sum_{1\le i,j,k\le n} \big(\mathbf{u}_\ell[i]M[i,\cdot]\mathbf{e}_{i'}\big)\big(\mathbf{v}_\ell[j]M[j,\cdot]\mathbf{e}_{j'}\big)\big(\mathbf{w}_\ell[k]M[k,\cdot]\mathbf{e}_{k'}\big) \\
&\quad = \sum_{\ell=1}^{r} \left(\sum_{i=1}^{n} \mathbf{u}_\ell[i]M[i,\cdot]\mathbf{e}_{i'}\right)\left(\sum_{j=1}^{n} \mathbf{v}_\ell[j]M[j,\cdot]\mathbf{e}_{j'}\right)\left(\sum_{k=1}^{n} \mathbf{w}_\ell[k]M[k,\cdot]\mathbf{e}_{k'}\right) \\
&\quad = \sum_{\ell=1}^{r} \left[(\mathbf{u}'_\ell)\,\mathbf{e}_{i'}\right]\left[(\mathbf{v}'_\ell)\,\mathbf{e}_{j'}\right]\left[(\mathbf{w}'_\ell)\,\mathbf{e}_{k'}\right] \\
&\quad = \sum_{\ell=1}^{r} \mathbf{u}'_\ell[i'] \cdot \mathbf{v}'_\ell[j'] \cdot \mathbf{w}'_\ell[k'].
\end{aligned}
$$

From this we conclude that $A' = \sum_{\ell=1}^{r} \mathbf{u}'_\ell \otimes \mathbf{v}'_\ell \otimes \mathbf{w}'_\ell$ and hence $\mathsf{rank}(A') \le r$.

Now let $(\lambda_1, \ldots, \lambda_n) = (y^0, \ldots, y^{n-1}) \cdot T^{-1}$, then

$$
\sum_{i=1}^{n} \lambda_i A_i = \sum_{i=1}^{n} \lambda_i \left(\sum_{j=1}^{n} T[i,j] \cdot R_j\right) = \sum_{j=1}^{n} R_j \sum_{i=1}^{n} T[i,j] \cdot \lambda_i = \sum_{j=1}^{n} R_j \cdot y^{j-1} = A'.
$$

This shows that there is a linear combination of the matrices representing the public key whose result is a low rank three-dimensional matrix. This yields directly an instance of the cubic Min-Rank problem which can be solved for instance with the extension of the Kipnis-Shamir modeling presented in Section 4.2.2. As we mentioned before, this is by itself a weakness of the scheme, as it allows distinguishing public keys from random polynomial systems and also has implications on the degree of regularity of the system, as stated in Section 6.2. Moreover, the coefficients we have obtained here carry the same information about the secret key as those in the original (quadratic) Min-Rank attack, and this can be used in a similar way to construct equivalent keys.

6.2 Direct Algebraic Attack

The direct algebraic attack, or simply the direct attack, refers to the case when an attacker aims to find the plaintext associated with a ciphertext $(c_1, \ldots, c_n)$ directly from the public multivariate equations $p_1 = c_1, \ldots, p_n = c_n$, without the knowledge of any other information of the system. In almost all the cases, the most efficient way to perform this attack

is to compute a Gröbner basis of the ideal I generated by the multivariate polynomials $p_1 - c_1, \ldots, p_n - c_n$.

Gröbner bases have played an important role not only in multivariate cryptography, but also in coding theory and lattices [ABBQBTP16, ASP11]. There is a general consensus that when computing a Gröbner basis over a finite field, one of the most efficient ways to do it is to use the F_4 or F_5 algorithms [Fau99, Fau02]. In a recent work [MS17], the authors used their M4GB algorithm to solve some of Fukuoka's MQ challenges within 11 days. The complexity of all these algorithms depends on the *degree of regularity* of the system. Since the degree of regularity is hard to determine, it is usually approximated by its *first fall degree*, defined as the first degree at which non-trivial relations between the polynomials $p_1, \ldots, p_n$ occur. For a more thorough survey of the complexity of computing Gröbner bases and an analysis of the different parameters used to study it, see [Esc16].

We now want to derive an upper bound for the first fall degree of the system. Before we do that, we need the following definition.

Definition. The LRank of a homogeneous $\lambda \in \mathbb{F}[x_1, \ldots, x_n]$ is the smallest integer s such that there exist linear homogeneous $\mu_1, \ldots, \mu_s \in \mathbb{F}[x_1, \ldots, x_n]$ with λ contained in the algebra $\mathbb{F}[\mu_1, \ldots, \mu_s]$.

Hodges et al. [HPS14] proved that for a scheme with core polynomial of weight 3, its first fall degree $D_{\text{ff}}(p_1, \ldots, p_n)$ is bounded by

$$D_{\text{ff}}(p_1, \ldots, p_n) \leq \frac{\mathsf{LRank}(P_0)(q-1)+5}{2}.$$

Here P_0 is the homogeneous part of highest degree of the core polynomial $\mathcal{F}$ seen as an element of the graded algebra $\mathbb{K}[X_0, \ldots, X_{n-1}]/\left(X_0^q, \ldots, X_{n-1}^q\right)$, where X_i corresponds to X^{q^i}, for $i = 0, \ldots, n-1$. In our case

$$P_0 = \sum_{1 \leq i,j,k \leq n} \alpha_{i,j,k} X_{i-1} X_{j-1} X_{k-1}.$$

If we take α_{ijk} uniformly at random, then with high probability $\mathsf{LRank}(P_0) \leq \mathsf{rank}(P_0)$, so

$$D_{\text{ff}}(p_1, \ldots, p_n) \leq \frac{\mathsf{rank}(\mathcal{F})(q-1)+5}{2}, \tag{6.1}$$

since $\mathsf{rank}(P_0) = \mathsf{rank}(\mathcal{F})$.

In addition, in [HPS14] the authors show that if $\deg \mathcal{F} = D$, then $\mathsf{rank}(\mathcal{F}) \leq \lfloor \log_q(D-2) \rfloor + 1$, and hence

$$D_{\text{ff}}(p_1, \ldots, p_n) \leq \frac{(q-1)\lfloor \log_q(D-2) \rfloor + 4 + q}{2}. \tag{6.2}$$

We now want to experimentally study the tightness of the bound (6.2), as they did in [HPS14] for different parameters[1]. In Table 6.1 we present some of the results obtained

[1]Table 1 in [HPS14] do not include the values for the parameters we are interested in, so we constructed our own version of it.

for different values of the parameters q, n and t, where t is the smallest integer such that $D \leq q^t - 1$. The value B corresponds to the bound given by equation (6.2), and D_{ff} is the first fall degree of the system for each choice of the parameters, which is read from Magma's verbose output. All the polynomials used in the attack were built as it was explained in Section 6.1.1, and for all cases we have included the field equations, i.e., $x_i^q - x_i$ for $i = 1, \ldots, n$.

q	t	n	B	D_{ff}
5	3	8	8	8
5	3	9	8	8
5	3	10	8	8
5	4	8	10	9
5	4	9	10	9
5	4	10	10	10
5	5	8	12	9
5	5	9	12	9
5	5	10	12	10

q	t	n	B	D_{ff}
7	3	8	11	10
7	3	9	11	10
7	3	10	11	10
7	4	8	14	10
7	4	9	14	11
7	4	10	14	12
7	5	8	17	10
7	5	9	17	11
7	5	10	17	12

q	t	n	B	D_{ff}
11	3	8	17	13
11	3	9	17	14
11	3	10	17	15
11	4	8	22	13
11	4	9	22	14
11	4	10	22	15
11	5	8	27	13
11	5	9	27	14
11	5	10	27	15

q	t	n	B	D_{ff}
17	3	8	26	17
17	3	9	26	18
17	3	10	26	18
17	4	8	34	17
17	4	9	34	18
17	4	10	34	18
17	5	8	42	17
17	5	9	42	18
17	5	10	42	18

Table 6.1: Experimental results to study the tightness of the bound for D_{ff} given by (6.2), for different choices of the parameters q, t and n. The value of D_{ff} is read from Magma's verbose output.

We notice that the bound given by (6.2) is very tight for small values of q and t, and that it starts to widen considerably as q increases, and with a smaller pace as t increases. We also observe that for fixed q and t, the bound gets tighter as n increases. It is very clear that the bound needs to be improved for larger values of q.

On the other hand, the complexity of finding a Groebner basis $\mathcal{G}$ for the ideal I is bounded by

$$O\left(\binom{n + D_{\text{ff}}}{D_{\text{ff}}}^{\omega}\right),$$

where $2 \leq \omega \leq 3$ is the linear algebra constant. When n grows to infinity, the complexity [2] becomes $O\left(n^{\omega D_{\text{ff}}}\right)$. Therefore, according to the bound in (6.1), the complexity of finding $\mathcal{G}$ is bounded by

$$O\left(n^{\omega \frac{\mathsf{rank}(\mathcal{F})(q-1)+5}{2}}\right).$$

Thus, if q and $\mathsf{rank}(\mathcal{F})$ are constant, then the complexity of finding $\mathcal{G}$ is polynomial in the number of variables n. We also observe that the complexity is exponential in $\mathsf{rank}(\mathcal{F})$.

6.3 Example: HFE Cubic

Here we present the natural generalization of the HFE cryptosystem discussed in Section 5.3. This is a natural scheme to which our new Min-Rank attack might apply. The secret

[2] Notice that we are using an upper bound to estimate the complexity. This is a customary usage for this kind of attacks. In practice, it has been observed [Spa12] that, on average, this bound is not too far from the actual complexity.

key consist of two invertible matrices $S, T \in \mathbb{F}^{n\times n}$ and a univariate polynomial $\mathcal{F} \in \mathbb{K}[X]$ of the form

$$\mathcal{F}(X) = \sum_{1\leq i,j,k\leq r} \alpha_{i,j,k} X^{q^{i-1}+q^{j-1}+q^{k-1}},$$

where $\alpha_{i,j,k} \in \mathbb{K}$. Notice that the degree of this polynomial is at most $3q^{r-1}$. Due to Theorem 1.5.2 we have that the composition $F = \phi \circ \mathcal{F} \circ \phi^{-1} : \mathbb{F}^n \to \mathbb{F}^n$ can be expressed as evaluation of n homogeneous cubic polynomials $f_1, \ldots, f_n \in \mathbb{F}[x_1, \ldots, x_n]$. Therefore the composition $T \circ F \circ S = T \circ \phi \circ \mathcal{F} \circ \phi^{-1} \circ S$ can also be seen as evaluation of n homogeneous cubic polynomials $p_1, \ldots, p_n \in \mathbb{F}[x_1, \ldots, x_n]$. These polynomials constitute the public key. Encryption and decryption is performed just as in HFE, which is possible since (as we did there) we take r to be small enough so that the polynomial $\mathcal{F}$ is easy to invert.

Min-Rank Analysis

Let A be the three-dimensional matrix whose entry (i, j, k) is equal to $\alpha_{i,j,k}$ if $i, j, k \leq r$, and 0 otherwise. As we have done before, we can assume, without loss of generality, that this matrix is symmetric. To see that our Min-Rank attack applies to this scheme, we only need to show that the three-dimensional matrix A has low rank. We claim that the matrix A has rank at most $(3/4)r^2$. This can be seen since the rank of the matrix A is the same as the rank of the matrix $A' \in \mathbb{F}^{r\times r\times r}$ defined by $A'[i, j, k] = A[i, j, k]$, and the latter is bounded by $(3/4)r^2$, as seen in Section 2.2.

It is important to remark that we considered this attack just as an example and it is not by any means the most efficient attack on this scheme. For instance, this scheme counts with a structural weakness: when the differential is applied the rank drops from $O(r^2)$ to r. As we saw in Section 4.3.2, this is not a typical behavior, and it only happens due to the underlying structure of the matrix involved.

Chapter 7

HiRaC: High Rank Cryptosystem

In this chapter we present a new proposal for a Multivariate Public Key Encryption Scheme. We call it HiRaC, standing for High Rank Cryptosystem. HiRaC uses cubic polynomials, and its name is motivated from the apparent fact that our scheme has a high rank and therefore is not vulnerable directly to the attacks sketched in the previous chapter.

Many of the constructions seen so far in 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.

7.1 Description of HiRaC

Let q be a prime number greater than 3, 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

$$\mathcal{H}(X) = \sum_{j=0}^{r} X^{q^j} \mathcal{M}_j \left(\mathcal{F}'(X)\right) + \mathcal{G}(X) \tag{7.1}$$

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 (7.1) 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 given $\mathbf{c} = (c_1, \ldots, c_{2n})$ in the range of P, and we want to solve the simultaneous equations $\mathcal{F}\left(\phi^{-1}(\mathbf{x})\right) = Z_1$, $\mathcal{H}\left(\phi^{-1}(S\mathbf{x})\right) = Z_2$ 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.

In Table 7.1 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.

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 7.1: Experiments of Public Key generation, encryption and decryption, for different values of q, n and r

7.2 Min-Rank Analysis

Now we apply our framework to analyze the vulnerability of our scheme with respect to the Min-Rank attack. More specifically, we explore the viability of performing the attack on the composition $T \circ \phi \circ \mathcal{H} \circ \phi^{-1} \circ S$. In order to achieve this, we must find an upper bound on the rank of the symmetric matrix representing the polynomial $\mathcal{H}$.

From equation (7.1) we can write $\mathcal{H}(X)$ as

$$\mathcal{H}(X) = \sum_{t=1}^{r} X^{q^{t-1}} \cdot \mathcal{F}_t(X)$$

where $\mathcal{F}_t \in \mathbb{K}[X]$ is some weight 2 polynomial. We can write this as $\mathcal{H}(X) = \mathcal{T}(\mathbf{x}, \mathbf{x}, \mathbf{x})$ where $\mathbf{x} = (X^{q^0}, \ldots, X^{q^{n-1}})^\intercal$ and $\mathcal{T} : \mathbb{K}^n \times \mathbb{K}^n \times \mathbb{K}^n \to \mathbb{K}$ is the trilinear form given by

$$\mathcal{T}(\boldsymbol{\beta}, \boldsymbol{\delta}, \boldsymbol{\gamma}) = \sum_{t=1}^{r} \beta_t \cdot \mathcal{T}_t(\boldsymbol{\delta}, \boldsymbol{\gamma})$$

with $\mathcal{F}_t(X) = \mathcal{T}_t(\mathbf{x}, \mathbf{x})$. Assume without loss of generality that each $\mathcal{T}_t$ is symmetric. Notice that $\mathcal{T}$ is not symmetric in general. To obtain the symmetric trilinear form associated to $\mathcal{H}$ we compute

$$\frac{1}{3!}\left(\mathcal{T}(\boldsymbol{\beta}, \boldsymbol{\delta}, \boldsymbol{\gamma}) + \mathcal{T}(\boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\delta}) + \mathcal{T}(\boldsymbol{\delta}, \boldsymbol{\beta}, \boldsymbol{\gamma}) + \mathcal{T}(\boldsymbol{\delta}, \boldsymbol{\gamma}, \boldsymbol{\beta}) + \mathcal{T}(\boldsymbol{\gamma}, \boldsymbol{\beta}, \boldsymbol{\delta}) + \mathcal{T}(\boldsymbol{\gamma}, \boldsymbol{\delta}, \boldsymbol{\beta})\right).$$

However, since each $\mathcal{T}_t$ is symmetric, any permutation of the two last inputs do not change the trilinear form, which leaves us with

$$\frac{1}{3}\left(\mathcal{T}(\boldsymbol{\beta}, \boldsymbol{\delta}, \boldsymbol{\gamma}) + \mathcal{T}(\boldsymbol{\delta}, \boldsymbol{\beta}, \boldsymbol{\gamma}) + \mathcal{T}(\boldsymbol{\gamma}, \boldsymbol{\beta}, \boldsymbol{\delta})\right).$$

As a conclusion, the rank of $\mathcal{H}$ is at most three times the rank of $\mathcal{T}$. Now we claim the latter rank is at most $r \cdot n$. Let T_t be the two-dimensional matrix associated with the bilinear form $\mathcal{T}_t$. Assuming that T_t has full rank, we can write $T_t = \sum_{\ell=1}^{n} \mathbf{v}_\ell^{(t)} \otimes \mathbf{w}_\ell^{(t)}$, then the matrix associated to $\mathcal{T}$ is given by

$$A = \sum_{t=1}^{r} \mathbf{e}_t \otimes \left(\sum_{\ell=1}^{n} \mathbf{v}_\ell^{(t)} \otimes \mathbf{w}_\ell^{(t)}\right) = \sum_{t=1}^{r}\sum_{\ell=1}^{n} \mathbf{e}_t \otimes \mathbf{v}_\ell^{(t)} \otimes \mathbf{w}_\ell^{(t)},$$

which has rank at most $r \cdot n$.

We conclude that the rank of the polynomial $\mathcal{H}$ is at most $3 \cdot r \cdot n$. It is important to notice that if $r = O(1)$, then this rank is not asymptotically maximal since $3 \cdot r \cdot n = O(n)$ and we know that the maximal rank for the given dimensions is $O(n^2)$. In particular, there is indeed a rank defect, meaning that our central map has a rank that is not maximal. However, if the rank happens to be close to $O(n)$ then all the approaches to the underlying cubic Min-Rank problem become inefficient as their complexity using Gröbner bases is exponential (see for example [Esc16] or [Spa12]).

The above argument shows that the overall structure of the scheme does not imply a low rank. However, since we have not provided a lower bound, the rank of the central map

could still be low due to some other structural weakness. We have run extensive experiments that seem to indicate this is not the case: We generated several HiRaC instances and considered the differential of the central polynomial. We calculated then its rank (recall that the differential of a cubic polynomial is a quadratic polynomial), and in all of our experiments we found that this rank was n. Since, as we saw in Section 4.3.2, the rank of the differential is less than or equal to the rank of the original polynomial, it follows that our central map has, for the experiments executed, rank greater than n.

Finally, we stress out that this argument does not rule out any structural attack like the one we showed at the end of Section 6.3, or like the one found in ZHFE.

Bibliography

[ABBQBTP16] Ismara Alvarez-Barrientos, Mijail Borges-Quintana, Miguel Angel Borges-Trenard, and Daniel Panario. Computing Gröbner Bases Associated with Lattices. *Adv. in Math. of Comm.*, 10(4):851–860, 2016.

[ASP11] M. Aliasgari, M. R. Sadeghi, and D. Panario. Gröbner Bases for Lattices and an Algebraic Decoding Algorithm. In *2011 49th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 1414–1415, Sept 2011.

[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.

[BCE+18] John Baena, Daniel Cabarcas, Daniel E Escudero, Karan Khathuria, and Javier Verbel. Rank analysis of cubic multivariate cryptosystems. In *International Conference on Post-Quantum Cryptography*, pages 355–374. Springer, 2018.

[BFP13a] 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.

[BFP13b] 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, Oct 2013.

[BFS99] J. F. Buss, G. S. Frandsen, and J. O. Shallit. The Computational Complexity of Some Problems of Linear Algebra. *Journal of Computer and System Sciences*, 58(3):572 – 596, 1999.

[BFSY05] M. Bardet, J.-C. Faugère, B. Salvy, and B.-Y. Yang. Asymptotic Behaviour of the Degree of Regularity of Semi-Regular Polynomial Systems. In *MEGA*

2005. Eighth International Symposium on Effective Methods in Algebraic Geometry, pages 1–14, 2005.

[Blä14] Markus Bläser. *Explicit Tensors*, pages 117–130. Springer International Publishing, Cham, 2014.

[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.

[CGLM08] Pierre Comon, Gene Golub, Lek-Heng Lim, and Bernard Mourrain. Symmetric tensors and symmetric tensor rank. *SIAM Journal on Matrix Analysis and Applications*, 30(3):1254–1279, jan 2008.

[Cou01] Nicolas T Courtois. Efficient zero-knowledge authentication based on a linear algebra problem minrank. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 402–421. Springer, 2001.

[CSTV17] Daniel Cabarcas, Daniel Smith-Tone, and Javier A Verbel. Key recovery attack for zhfe. In *International Workshop on Post-Quantum Cryptography*, pages 289–308. Springer, 2017.

[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.

[DH11] Jintai Ding and Timothy J. Hodges. Inverting HFE Systems is Quasi-Polynomial for All Fields. In Phillip Rogaway, editor, *Advances in Cryptology – CRYPTO 2011*, volume 6841 of *Lecture Notes in Computer Science*, pages 724–742. Springer Berlin Heidelberg, 2011.

[Esc16] Daniel Escudero. Groebner Bases and Applications to the Security of Multivariate Public Key Cryptosystems. Available online at http://cs.au.dk/~escudero/files/TDG.pdf, 2016. Accessed: 2017-11-25.

[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.

[FDS11] J.-C. Faugère, M. S. El Din, and P.-J. Spaenlehauer. Gröbner bases of bihomogeneous ideals generated by polynomials of bidegree (1,1): Algorithms and complexity. *Journal of Symbolic Computation*, 46(4):406 – 437, 2011.

[FLdVP08] Jean-Charles Faugère, Françoise Levy-dit Vehel, and Ludovic Perret. Cryptanalysis of Minrank. In *Advances in Cryptology – CRYPTO 2008*, pages 280–296, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

[GC00] Louis Goubin and Nicolas T. Courtois. *Cryptanalysis of the TTM Cryptosystem*, pages 44–57. Springer Berlin Heidelberg, Berlin, Heidelberg, 2000.

[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.

[GJ18] Craig Gentry and Charanjit S. Jutla. Obfuscation using tensor products. Cryptology ePrint Archive, Report 2018/756, 2018. https://eprint.iacr.org/2018/756.

[GRS+] Philippe Gaborit, Olivier Ruatta, Julien Schrek, Jean-Pierre Tillich, and Gilles Zémor. Rank based cryptography: a credible post-quantum alternative to classical cryptography.

[HL13a] Christopher J. Hillar and Lek-Heng Lim. Most Tensor Problems are NP-Hard. *J. ACM*, 60(6):45:1–45:39, November 2013.

[HL13b] Christopher J. Hillar and Lek-Heng Lim. Most tensor problems are np-hard. *J. ACM*, 60(6):45:1–45:39, November 2013.

[How78] Thomas D. Howell. Global properties of tensor rank. *Linear Algebra and its Applications*, 22(Supplement C):9 – 23, 1978.

[HPS14] Timothy J. Hodges, Christophe Petit, and Jacob Schlather. First Fall Degree and Weil Descent. *Finite Fields Appl.*, 30:155–177, November 2014.

[Hå90] Johan Håstad. Tensor rank is np-complete. *Journal of Algorithms*, 11(4):644 – 654, 1990.

[KL07] Jonathan Katz and Yehuda Lindell. *Introduction to Modern Cryptography (Chapman & Hall/Crc Cryptography and Network Security Series)*. Chapman & Hall/CRC, 2007.

[Kru77] Joseph B. Kruskal. Three-way arrays: rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics. *Linear Algebra and its Applications*, 18(2):95 – 138, 1977.

[KS99a] Aviad Kipnis and Adi Shamir. *Cryptanalysis of the HFE Public Key Cryptosystem by Relinearization*, pages 19–30. Springer Berlin Heidelberg, Berlin, Heidelberg, 1999.

[KS99b] 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.

[Lan12] Joseph M Landsberg. *Tensors: geometry and applications*, volume 128 of *Graduate Studies in Mathematics*. American Mathematical Society, 2012.

[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.

[MPST14] Dustin Moody, Ray Perlner, and Daniel Smith-Tone. *An Asymptotically Optimal Structural Attack on the ABC Multivariate Encryption Scheme*, pages 180–196. Springer International Publishing, Cham, 2014.

[MPST17] Dustin Moody, Ray Perlner, and Daniel Smith-Tone. Key recovery attack on the cubic abc simple matrix multivariate encryption scheme. In Roberto Avanzi and Howard Heys, editors, *Selected Areas in Cryptography – SAC 2016*, pages 543–558, Cham, 2017. Springer International Publishing.

[MS17] Rusydi H. Makarim and Marc Stevens. M4GB: An Efficient Gröbner-Basis Algorithm. In *Proceedings of the 2017 ACM on International Symposium on Symbolic and Algebraic Computation*, ISSAC '17, pages 293–300, New York, NY, USA, 2017. ACM.

[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.

[Sch12] Leonard J. Schulman. Cryptography from tensor problems. Cryptology ePrint Archive, Report 2012/244, 2012. https://eprint.iacr.org/2012/244.

[SgS13] Friedland. Shmuel and Małgorzata Stawiska. Best Approximation on Semi-Algebraic Sets and k-border Rank Approximation of Symmetric Tensors. *arxiv.org/pdf/1311.1561*, November 2013.

[Shm16] Friedland Shmuel. Remarks on the Symmetric Rank of Symmetric Tensors. *arxiv.org/pdf/1505.00860*, January 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, PhD thesis, Université Paris 6, 2012.

[YC05] Bo-Yin Yang and Jiun-Ming Chen. Building secure tame-like multivariate public-key cryptosystems: The new tts. In Colin Boyd and Juan Manuel González Nieto, editors, *Information Security and Privacy*, pages 518–531, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.