

New candidates for multivariate trapdoor functions and new multivariate public key encryption schemes

Por

Jaiberth Porras Barrera

Trabajo presentado como requisito parcial para optar al título de

Doctor en Ciencias Matemáticas

Director
John Bayron Baena

Codirector
Jintai Ding
Universidad de Cincinnati

Universidad Nacional de Colombia
Sede Medellín

Facultad de Ciencias
Escuela de Matemáticas

Diciembre 2013

Abstract

In this thesis we present a new method for building pairs of HFE¹ polynomials of high degree, in such a way that the map constructed with this pair is easy to invert. The inversion is accomplished using a low degree polynomial of Hamming weight three, which is derived from a special reduction via Hamming weight three polynomials produced by these two HFE polynomials. This allows us to build new candidates for multivariate trapdoor functions in which we use the pair of HFE polynomials to fabricate the core map.

Using this new multivariate trapdoor function we derive an encryption scheme in a similar way as the HFE scheme is created. We show that this encryption scheme is relatively efficient and that it resists the attacks that have threatened the security of HFE. Finally, we propose parameters for a practical implementation of our cryptosystem.

¹HFE stands for Hidden Field Equations.

Resumen

En esta tesis presentamos un nuevo método para construir parejas de polinomios HFE² de grado alto, de tal manera que la función construida con esta pareja es fácil de invertir. La inversión se lleva a cabo utilizando un polinomio de grado bajo y de peso de Hamming tres, el cual se deriva por medio de una reducción especial, a través de polinomios de peso de Hamming tres producidos a partir de estos dos polinomios HFE. Esto nos permite construir nuevas candidatas para funciones de puerta trasera multivariadas, en las cuales utilizamos la pareja de polinomios HFE para construir la función central.

Utilizando esta nueva función de puerta trasera multivariada derivamos un esquema de cifrado de una manera similar a como se construye el esquema HFE. Demostramos que este esquema de cifrado es relativamente eficiente y que resiste los ataques que han amenazado la seguridad de HFE. Finalmente, proponemos parámetros para una aplicación práctica de nuestro criptosistema.

²HFE es la abreviación de Hidden Field Equations.

Acknowledgements

I want to thank my family for their unconditional support, without them this achievement would have been impossible.

I am very grateful with my advisor, Dr. John Baena, for everything that I learned from him in these years. His ideas and dedication were fundamental to accomplish this work.

Also, I am grateful to my second advisor, Dr. Jintai Ding, for his valuable ideas and his continued collaboration to make this research possible.

In addition, I want to thank Wael Mohamed and Daniel Cabarcas who ran many essential experiments to complete this thesis. I appreciate their generous collaboration.

Also, I want to thank José Manuel Gómez for giving a review of this thesis and for his valuable suggestions.

Finally, I want to thank the professors of the School of Mathematics of the National University of Colombia, Medellín Campus, for all their knowledge shared with me during my entire staying at the University.

Table of Contents

Abstract	ii
Acknowledgements	v
List of Tables	viii
List of Figures	ix
Chapter	
1. Introduction	1
1.1 Preface	1
1.2 Motivation	2
1.3 Our approach	3
1.4 Thesis structure	4
2. Some basic concepts	6
2.1 Elementary ideas in cryptography	6
2.2 Cryptanalysis	8
2.3 Public Key Cryptography	9
2.4 Multivariate Public key Cryptography	12
2.5 The big-field idea	14
3. Matsumoto-Imai scheme (MI)	18
3.1 Description of MI	18
3.2 Cryptanalysis of MI	20
3.3 Some variants of MI	21
4. Hidden Field Equations (HFE)	23
4.1 Description of HFE	23
4.2 Toy example	26
4.3 Algebraic attack	28
4.4 Kipnis-Shamir MinRank attack (KS attack)	30
4.5 Some variants of HFE	35
4.6 The history of HFE	37
5. The Zhuang-Zi algorithm (ZZ algorithm)	40
5.1 Description of ZZ	41
5.2 Toy example	43

5.3	Nontrivial examples	47
6.	New candidates for multivariate trapdoor functions	49
6.1	The Reduction method	49
6.2	Complexity of the reduction method and dimension of the solution space	53
6.3	How to build the trapdoor function	54
6.4	How to invert the trapdoor function	55
6.5	Toy example	57
6.6	Big examples	60
7.	New multivariate public key encryption schemes	64
7.1	The encryption scheme	64
7.2	Toy example	67
7.3	Algebraic attack	71
7.4	Kipnis-Shamir MinRank attack	81
8.	Conclusions and Future Work	89
	Bibliography	91
	Appendices	96

List of Tables

Table

7.1	Encryption and decryption time for the new encryption scheme, 100 messages were tested per key.	70
7.2	Algebraic attack against the new cryptosystem for $q = 2$ and $D_0 = 386$	73
7.3	Algebraic attack comparison between the new encryption scheme and a system of random equations for $q = 2$ and $D_0 = 386$	74
7.4	Algebraic attack against the new encryption scheme for $q = 7$ and $D_0 = 105$	75
7.5	Algebraic attack comparison between the new encryption scheme and a system of random equations for $q = 7$ and $D_0 = 105$	77
7.6	Algebraic attack against the new encryption scheme for $q = 17$ and $D_0 = 595$	78
7.7	Algebraic attack comparison between the new encryption scheme and a system of random quadratic equations for $q = 17$ and $D_0 = 595$	79
7.8	Algebraic attack for the new encryption scheme for several choices of (q, n, D_0)	81
7.9	KS attack against the new encryption scheme, for $q = 7$ and $D_0 = 105$	85
7.10	Time and memory needed to find the solution set for the KS attack against the new encryption scheme, for $q = 7$, $n = 8$ and $D_0 = 105$	86
7.11	Q-Rank(P) comparison between the new trapdoor function and random equations for $q = 7$	86
7.12	KS attack against a bounded Multi-HFE scheme for $q = 7$ and $\lceil \log_q D \rceil = 2$	87
A.1	Private key generation for $q = 2$ and $D_0 = 386$	97
A.2	Private key generation for $q = 7$ and $D_0 = 105$	98
B.1	KS attack against the new encryption scheme, for $q = 17$ and $D_0 = 595$	99
B.2	Time and memory needed to find the solution set for the KS attack against the new encryption scheme, for $q = 17$, $n = 8$ and $D_0 = 595$	99
B.3	Q-Rank(P) comparison between the new trapdoor function and random equations for $q = 17$	99

List of Figures

Figure

3.1	MI scheme.	19
4.1	HFE scheme.	24
6.1	New candidate for multivariate trapdoor function.	55
7.1	Algebraic attack against the new encryption scheme for $q = 2$ and $D_0 = 386$	73
7.2	Algebraic attack for the new encryption scheme for $q = 2$ and $D_0 = 386$	74
7.3	Algebraic attack comparison between the new encryption scheme and a system of random equations for $q = 2$ and $D_0 = 386$	75
7.4	Algebraic attack against the new encryption scheme for $q = 7$ and $D_0 = 105$	76
7.5	Algebraic attack for the new encryption scheme for $q = 7$ and $D_0 = 105$	76
7.6	Algebraic attack comparison between the new encryption scheme and a system of random equations for $q = 7$ and $D_0 = 105$	78
7.7	Algebraic attack against the new encryption scheme for $q = 17$ and $D_0 = 595$	79
7.8	Algebraic attack for the new encryption scheme for $q = 17$ and $D_0 = 595$	80
7.9	Algebraic attack comparison between the new encryption scheme and a system of random quadratic equations for $q = 17$ and $D_0 = 595$	80
A.1	Private key generation for $q = 2$ and $D_0 = 386$	97
A.2	Private key generation for $q = 7$ and $D_0 = 105$	98

Chapter 1

Introduction

1.1 Preface

The public-key cryptosystems currently being used in practice are based on the difficulty of factoring large integers or solving the Discrete Logarithm Problem. In 1996 P. Shor published an algorithm to solve both problems in polynomial time on a quantum computer [43]. Some experts argue that it is possible to build in the coming years a quantum computer, which is a threat to our modern communication system. This leads to the recent fast development of Post-Quantum Cryptography, which refers to the study of schemes that have the potential to resist the future quantum computer attacks [3].

Multivariate Public Key Cryptography is part of the Post-Quantum Cryptography. In a multivariate public key scheme (MPK scheme) the public key consists of a set of multivariate quadratic polynomials over a finite field. One of the main MPK schemes is named Hidden Field Equations (HFE), proposed by Patarin in 1996 [39]. The public key in HFE is formed by “hiding” a core polynomial F by two invertible affine transformations, and using the vector space structure of a field extension of the base field.

A crucial part in HFE is the choice of the degree D of the core polynomial F . The degree D cannot be too big otherwise the decryption process would not be efficient. The main attacks against HFE, direct algebraic attack and the Kipnis-Shamir MinRank attack,

exploit this fact. For characteristic 2, HFE is vulnerable to the direct algebraic attack [25]. Recently, some authors improved the KS attack and were able to break certain HFE systems, over both even and odd characteristic [4].

1.2 Motivation

There exist secure and efficient MPK signature schemes (one example is Rainbow [15]) but we do not know secure and efficient MPK encryption schemes, since all the ones proposed have been broken.

Some variants of HFE have been proposed as encryption schemes, but all of them have been proven to be insecure. Perhaps the reason for this fact is the low degree and the low rank of the polynomial used as core map for these systems. This degree cannot be too large because the decryption process would be very slow. One question arises:

- Is there any way to enlarge the degree of the core polynomial of an HFE encryption scheme without affecting the efficiency of the decryption process?

Using some especially constructed HFE polynomials as the core map in HFE, we give here an affirmative answer to this question. So far we know that no one has proposed any idea to use high degree polynomials for the core map in HFE or some of its variants. The low degree D of the core polynomial F was always considered an immovable. We developed a special reduction method that enables us to build multivariate trapdoor functions using core polynomials of high degree in an HFE scheme. From the trapdoor functions we derive a very efficient encryption scheme which resists the major currently known attacks against these kind of cryptosystems.

1.3 Our approach

The idea of our construction is inspired by the first steps of the ZZ algorithm [15]. Given a finite field k of size q and a field extension K of degree n , we consider two high degree HFE polynomials over K of the form $F(X) = \sum a_{ij}X^{q^i+q^j} + \sum b_iX^{q^i} + c$ and $\tilde{F}(X) = \sum \tilde{a}_{ij}X^{q^i+q^j} + \sum \tilde{b}_iX^{q^i} + \tilde{c}$, where the coefficients $a_{ij}, b_i, c, \tilde{a}_{ij}, \tilde{b}_i, \tilde{c} \in K$ are to be determined. The idea behind the method is to construct a low degree polynomial Ψ of Hamming weight three of the form

$$\Psi = X \left(\alpha_1 F_0 + \cdots + \alpha_n F_{n-1} + \beta_1 \tilde{F}_0 + \cdots + \beta_n \tilde{F}_{n-1} \right) + X^q \left(\alpha_{n+1} F_0 + \cdots + \alpha_{2n} F_{n-1} + \beta_{n+1} \tilde{F}_0 + \cdots + \beta_{2n} \tilde{F}_{n-1} \right),$$

where F_0, F_1, \dots, F_{n-1} are the Frobenius powers of F , and $\tilde{F}_0, \tilde{F}_1, \dots, \tilde{F}_{n-1}$ are the Frobenius powers of \tilde{F} .

To obtain such a polynomial Ψ we need to determine the coefficients of F and \tilde{F} , also the scalars α_i and β_i , such that the degree of Ψ is less than or equal than a fixed positive integer D_0 (the integer D_0 is such that we can easily invert Ψ using Berlekamp's algorithm). To achieve this, we derive a system of equations from the vanishing coefficients of the terms in Ψ of degree higher than D_0 . After randomly choosing in this system the scalars α_i and β_i , we get a linear system with more variables than equations, and thus we can guarantee nontrivial solutions for it. This linear system has about n^3 variables and therefore we have to deal with huge matrices to reach large values of n . On the plus side we have that these matrices are sparse, which is an advantage in terms of efficiency.

The new multivariate trapdoor function is built in a similar way to the HFE scheme (composition with invertible affine linear transformations), except that now the core map is replaced by the map $G = (F, \tilde{F})$. The main part of the inversion of the trapdoor function is to invert the map G , which is achieved using the low degree polynomial Ψ of Hamming

weight three and the scalars α_i and β_i .

We use the new trapdoor function to construct a new encryption scheme. Since we are utilizing high degree HFE polynomials for the core map, we expect that the public key has high degree of regularity, very different from what was observed by Faugère and Joux [25] for a system of quadratic equations derived from a single HFE polynomial with low degree. Our extensive experiments confirmed that the public key of the new encryption scheme has high degree of regularity (it increases as n increases). This high degree of regularity shows that our new encryption scheme is secure against the direct algebraic attack.

For odd q , we implemented the KS MinRank attack and observed that the minimum rank increases as n increases. Therefore this attack would not work in this case against the new encryption scheme. For the case $q = 2$, we can give a theoretical argument to show why the KS MinRank attack does not work against the new encryption scheme, based on some results about the degree of regularity obtained by Ding and Hodges [17].

After testing the direct algebraic and KS MinRank attacks for several values of q and n , we suggest parameters for a secure and efficient encryption scheme.

The method described here was not our first attempt to reduce high degree HFE polynomials along the same line. Among failed attempts, we considered using a single polynomial F . However, the linear systems that we needed to solve had more equations than variables and then we could not guarantee nontrivial solutions for them. This lead us to use two HFE polynomials instead of one in order to get a linear system with more variables than equations and this gives the construction in this thesis.

1.4 Thesis structure

The rest of this thesis is organized as follows. In Chapter 2 we present some basic concepts about cryptography. In particular we discuss public key encryption schemes and multivariate

public key encryption schemes. Chapters 3 and 4 are devoted to the study of two of the main multivariate public key encryption schemes: MI and HFE cryptosystems. In Chapter 5 we study the Zhuang-Zi algorithm, a tool for solving multivariate systems over a finite field which has inspired this work. Chapters 6 and 7 contain the original work of this dissertation. In Chapter 6 we construct new candidates for multivariate trapdoor functions and we show how to invert these trapdoor functions. From these trapdoor functions we derive a new encryption scheme in Chapter 7. We present some conclusions and discuss future work in Chapter 8. The Appendix contains some additional data produced by the experiments performed with the new cryptosystem proposed in Chapter 7.

Chapter 2

Some basic concepts

We begin by presenting some basic concepts and procedures of cryptography. Then we discuss two of the main encryption schemes used today in public key cryptography. Finally, we introduce multivariate public key cryptography, the area in which our research is framed.

2.1 Elementary ideas in cryptography

Cryptography is the science that studies the methods to protect integrity and confidentiality of information. Cryptography has been used historically by secret communication agencies. Today, with the rise of the Internet, cryptography is essential to secure communication, digitally sign documents, access controls, E-commerce, electronic vote, among other applications.

One of the main activities in cryptography, although not the only one, is exchanging or storing information in a secure way. This task is accomplished by means of encryption schemes¹.

Definition 2.1. An *encryption scheme* or a *cryptosystem* is a tuple $(\mathcal{M}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ such that:

- \mathcal{M} is a set called the *plaintext space* and its elements are called *plaintexts*. \mathcal{C} is a set

¹Cryptography also studies other important topics as signature and authentication schemes

called the *ciphertext space* and its elements are called *ciphertexts*. \mathcal{K} is a set called the *key space* and its elements are called *keys*.

- $\mathcal{E} = \{E_e : e \in \mathcal{K}\}$ is a set of functions $E_e : \mathcal{M} \rightarrow \mathcal{C}$. The elements of \mathcal{E} are called encryption functions. For a plaintext (message) $m \in \mathcal{M}$ and a key $e \in \mathcal{K}$, $c = E_e(m)$ is called a ciphertext of the message m .
- $\mathcal{D} = \{D_d : d \in \mathcal{K}\}$ is a set of functions $D_d : \mathcal{C} \rightarrow \mathcal{M}$. The elements of \mathcal{D} are called decryption functions.
- For each $e \in \mathcal{K}$ there exists a unique $d \in \mathcal{K}$ such that $D_d(E_e(m)) = m$ for all $m \in \mathcal{M}$.

For an encryption function E_e the key e is called an *encryption key* and for a decryption function D_d the key d is called a *decryption key*. Notice that if an encryption key e is fixed, every plaintext $m \in \mathcal{M}$ is encrypted in the same ciphertext for anyone who uses the encryption function E_e . Therefore Definition 2.1 describes what is called a *deterministic encryption scheme*. One problem with deterministic schemes is that it is difficult to prove anything about their security. For this reason industrial applications usually use *probabilistic encryption schemes*, introduced in 1982 by Goldwasser and Micali [27]. In a probabilistic scheme three different probabilistic polynomial-time algorithms are used for key generation, encryption and decryption (for a precise definition see [31]). A disadvantage of probabilistic schemes is that some of them are vulnerable to certain *chosen-ciphertext attacks*.

According to the way their keys are shared, the encryption schemes are divided into two classes. The first class is called private key (symmetric) encryption schemes and the second one is called public key (asymmetric) encryption schemes. We now define these two classes.

Definition 2.2. A *private key (symmetric) encryption scheme* is a cryptosystem where for each pair of keys $(e, d) \in \mathcal{K} \times \mathcal{K}$, it is computationally feasible² to determine the decryption

²We will use the term “computationally feasible” for a problem that is easy to solve, in the sense that there exists a

key d from the encryption key e . In this case both keys are kept secret.

In a symmetric cryptosystem we usually have that the encryption key e and the decryption key d are the same. If Alice wants to communicate with Bob using a symmetric cryptosystem, she must previously exchange with him the private (secret) key e . One secure way to share this private key is using an asymmetric encryption scheme.

Definition 2.3. A *public key (asymmetric) encryption scheme* is a cryptosystem where for each pair of keys $(e, d) \in \mathcal{K} \times \mathcal{K}$, it is computationally infeasible to determine the decryption key d from the encryption key e . The encryption key e is made public and the decryption key d is kept private. In this way, anyone can encrypt a message and send it to the owner of the private, who can use this private key to recover the original plaintext.

The encryption/decryption process is generally slower for public key schemes than for symmetric schemes. However, public key schemes have the advantage that the keys do not need to be exchanged previously. Therefore, in practice, a hybrid system is often used, in which a public key scheme is utilized to exchange the private key for a symmetric scheme, and the latter is used to share the information.

2.2 Cryptanalysis

In this section we discuss briefly the security of a cryptosystem. The study of the attacks against cryptosystems is known as *cryptanalysis*. An *attack* on a cryptosystem is a method that seeks to obtain the plaintext from the ciphertext. If in addition the attack tries to find the decryption key, it is called a *key-recovery attack* or a *structural attack*. We say that an attack breaks a cryptosystem if it allows to recover the plaintext from the ciphertext efficiently. Depending on the knowledge and the abilities required to perform an attack, these can be classified as follows.

polynomial time algorithm to solve it using a low amount of time. And we will use the term “computationally infeasible” for a problem for which there are no known efficient algorithms to solve it in realistic time.

Ciphertext-only attack. The attacker only knows a ciphertext. An example of this attack is the *exhaustive search*, which tests all possible keys or plaintexts until the correct one is found.

Known-plaintext attack. The attacker knows a pair plaintext/ciphertext (or several such pairs). For example, many letters begin with the word “hello” and therefore an attacker can know the ciphertext corresponding to this word. With this information the attacker can try to decrypt other ciphertexts.

Chosen-plaintext attack (CPA). The attacker is able to encrypt plaintexts of his choice. This attack is always a threat for public key cryptosystems because the encryption key is known by anyone.

Chosen-ciphertext attack (CCA). The attacker is able to decrypt ciphertexts of his choice. This attack is a threat for some probabilistic encryption schemes. This attack gives birth to a type of security analysis for a cryptosystem called a *CCA-secure test* (see [31]).

2.3 Public Key Cryptography

In Section 2.1 we have defined public key encryption schemes (see Definition 2.3). However, public key cryptography comprises a more general set up, in which we can mention, for example, public key encryption schemes, signature schemes and authentication schemes. In this section we review some general ideas which are the foundation of any construction related to public key cryptography.

In 1976 Diffie and Hellman introduced the basic idea of public key cryptography [13]. Two important concepts in public key cryptography are one-way and trapdoor functions. We give here informal definitions for these notions. For more precise definitions see [31].

Definition 2.4. Let \mathcal{M} and \mathcal{C} be two sets. A function $f : \mathcal{M} \rightarrow \mathcal{C}$ is called a *one-way function* if it is easy to compute but it is hard to “invert”. By easy to compute we mean

that there exists a polynomial time algorithm to compute $f(m)$ for any $m \in \mathcal{M}$. By hard to invert we mean that there are no known efficient algorithms to compute pre-images of a randomly selected $c \in \text{Im } f \subseteq \mathcal{C}$.

In public key cryptography we are interested in a kind of functions that are one-way from the point of view of an attacker, but that are easy to invert for a legitimate user, who knows some additional information related to the construction of the function. Such maps are called trapdoor functions.

Definition 2.5. Let \mathcal{M} and \mathcal{C} be two sets. A function $f : \mathcal{M} \rightarrow \mathcal{C}$ is called a *trapdoor function* if it is easy to compute but it is hard to invert without the knowledge of some private information called the *trapdoor*. The function must be easy to invert for everyone who knows the trapdoor information.

One of the most important trapdoor functions in cryptography comes from number theory and it is the following.

Example 2.6. Randomly choose two large prime numbers $p \neq q$ of the same size (512 or 1024 bits). Set $n = pq$ and select a positive integer e relatively prime to $\phi(n) = (p-1)(q-1)$. Let $d = e^{-1} \pmod{\phi(n)}$, $\mathcal{M} = \mathcal{C} = \mathbb{Z}/n\mathbb{Z}$ and define the function $f : \mathcal{M} \rightarrow \mathcal{C}$ by

$$f(x) = x^e \pmod{n}.$$

Modular exponentiation is easy to compute because it can be done in polynomial time. Inverting f is a simple task if you know d because $x^{ed} \equiv x \pmod{n}$.

Why is f a trapdoor function? The computation of d requires the knowledge of $\phi(n)$. Computing $\phi(n)$ is computationally equivalent to factoring $n = pq$, and factoring large integers is currently computationally infeasible. Therefore, f is a trapdoor function and the trapdoor information is the integer d .

Although Diffie and Hellman introduced the notion of public key cryptography in 1976, the first public key encryption scheme was only proposed until 1978 by Rivest, Shamir and Adleman [42], which was named after them: RSA. This encryption scheme is constructed using the trapdoor function f introduced in Example 2.6. The keys for RSA can be summarized as follows:

- Public key: the trapdoor function f .
- Private key: the integer d .

If Bob wants to use RSA to send a message $m \in \mathcal{M}$ to Alice, he utilizes Alice's public key f to calculate the ciphertext $c = f(m) = m^e \bmod n$ and send this ciphertext to Alice. To recover the plaintext from the ciphertext, Alice uses her private key d to compute $m = c^d \bmod n$.

ElGamal [23] is another important public key scheme that is based on number theory. Its security relies on the difficulty of solving the Discrete Logarithm Problem, i.e., to solve for x in the congruence

$$a^x \equiv b \pmod{n},$$

for given integers a , b , and a large positive integer n .

Nowadays RSA, ElGamal, and other public key cryptosystems – that are based on number theory – are widely used in multiple applications. However, cryptographers are still in the search of new schemes, mainly due to two reasons. The first reason is to find more efficient schemes and the second one is because of the threats against existing schemes. One of these threats is the possible emergence of quantum computers. Such computers can factor large integers and solve the discrete logarithm problem in polynomial time [43].

Post-Quantum Cryptography refers to cryptosystems that are resistant to quantum computer attacks. The major research in this area focuses on Hash-based cryptography, Code-based cryptography, Lattice-based cryptography and Multivariate public key cryptography.

For a review in Post-Quantum Cryptography see [3].

The new encryption scheme that we propose in this thesis belongs to Multivariate Public Key Cryptography. Because of that we dedicate the rest of this chapter to describe these schemes.

2.4 Multivariate Public key Cryptography

As an alternative to existing public key schemes, *Multivariate Public Key Cryptography* has emerged [15]. The security of a multivariate public key encryption scheme (MPK encryption scheme) is suggested by the fact that solving a randomly system of multivariate quadratic polynomial equations over a finite field is an NP-hard problem [26]. This is known as the *MQ-problem* (multivariate quadratic problem). Moreover, it seems that quantum computers have no advantage over the traditional computers attacking this problem. Before we describe an MPK encryption scheme we establish some terminology used in this work.

The terminology given in this section is used throughout this work, unless otherwise noted. By k we denote a finite field of size q . We consider a degree n irreducible polynomial $g(y) \in k[y]$, and let K be the degree n extension field $k[y]/(g(y))$, which has q^n elements. We denote by φ the standard k -linear isomorphism between K and k^n given by

$$\varphi(u_1 + u_2y + \dots + u_ny^{n-1}) = (u_1, u_2, \dots, u_n).$$

We know that the ring of functions from k^n to k is isomorphic to the quotient ring

$$k[x_1, \dots, x_n] / (x_1^q - x_1, \dots, x_n^q - x_n).$$

Abusing the notation we identify the elements of this ring with the polynomials in $k[x_1, \dots, x_n]$.

In particular, the ring of functions from K to K is isomorphic to the ring $K[X] / (X^{q^n} - X)$ and we identify this ring with the ring of polynomials $K[X]$.

Besides the standard degree of a polynomial in $K[X]$ we will need to define its q -weight.

Definition 2.7. We say that a polynomial $F(X) \in K[X]$ has q -weight W (or Hamming weight W) if the maximum of the q -Hamming weights of all its exponents is W . The q -Hamming weight of a non-negative integer is the sum of the q -digits of its q -nary expansion.

For example, if $q = 3$, the 3-weight of the polynomial $F(X) = X^{27} + X^{20} + X^{11} + X^2 + 1$ is $W = 4$, since we can write $F(X) = X^{3^3} + X^{2 \times 3^0 + 2 \times 3^2} + X^{2 \times 3^0 + 3^2} + X^{2 \times 3^0} + 1$.

To build a MPK encryption scheme we use a multivariate trapdoor function $(p_1, \dots, p_m) : k^n \rightarrow k^m$, where each $p_i : k^n \rightarrow k$ can be represented as a nonlinear polynomial over the small field k . Therefore the public key is a sequence of nonlinear (usually quadratic) polynomials:

$$P = (p_1(x_1, \dots, x_n), \dots, p_m(x_1, \dots, x_n)),$$

where $p_i \in k[x_1, \dots, x_n]$. Some trapdoor information is saved as private key in order to invert P , i.e., to find pre-images of P . The public key polynomials are chosen quadratic to reduce the public key size and also for efficiency reasons in the encryption/decryption process.

If Bob wishes to send a plaintext $(x_1, \dots, x_n) \in k^n$ to Alice, he calculates the ciphertext

$$(y_1, \dots, y_m) = (p_1(x_1, \dots, x_n), \dots, p_m(x_1, \dots, x_n)),$$

and then he sends to Alice the ciphertext (y_1, \dots, y_m) across some possibly insecure channel.

To recover the plaintext Alice uses her private key to invert P .

Naturally, a random set of quadratic polynomials would not have a trapdoor and therefore would not be usable for building a MPK encryption scheme. Nevertheless, there exist several ways to construct MPK encryption schemes whose public key polynomials are not exactly random but are expected to behave as if they were. One of these ways is known as the big-field idea.

2.5 The big-field idea

The big-field idea is one of the several ways to generate a MPK encryption scheme. Select a q -weight two polynomial $F(X) \in K[X]$ and choose two invertible affine transformations S and T from k^n to k^n . Let $\varphi : K \rightarrow k^n$ be the standard k -linear isomorphism between K and k^n . The public key polynomials are generated by hiding the polynomial $F(X)$ by means of the affine transformations S and T . More precisely, the public key is the map $P : k^n \mapsto k^n$ given by

$$P(x_1, \dots, x_n) = T \circ \varphi \circ F \circ \varphi^{-1} \circ S(x_1, \dots, x_n).$$

We refer to F as the *core map* of the scheme. The public key P constructed in this way turns out to be a set of multivariate quadratic polynomials. This fact was proven in [32], but we present here a matrix-based and constructive proof. First, as in [4], we write the isomorphism φ and its inverse φ^{-1} in matrix form.

For a basis $\theta_1, \dots, \theta_n$ of K over k , consider the matrix Δ whose columns are the Frobenius powers of the basis elements, i.e.,

$$\Delta = \begin{pmatrix} \theta_1 & \theta_1^q & \cdots & \theta_1^{q^{n-1}} \\ \theta_2 & \theta_2^q & \cdots & \theta_2^{q^{n-1}} \\ \vdots & \vdots & \ddots & \vdots \\ \theta_n & \theta_n^q & \cdots & \theta_n^{q^{n-1}} \end{pmatrix}.$$

The matrix Δ is invertible because $\theta_1, \dots, \theta_n$ is a basis of K over k ([33], Pag. 109).

Lemma 2.8. *Let $X = x_1\theta_1 + \cdots + x_n\theta_n$ be an element of K . Then we have*

$$(x_1, \dots, x_n)\Delta = (X, X^q, \dots, X^{q^{n-1}}).$$

Proof. Since the Frobenius powers are linear transformations and they fix the elements of

the small field k , we have that

$$\begin{aligned}
(x_1, \dots, x_n)\Delta &= \left(\sum_{i=1}^n x_i \theta_i^{q^0}, \dots, \sum_{i=1}^n x_i \theta_i^{q^{n-1}} \right) \\
&= \left(\left(\sum_{i=1}^n x_i \theta_i \right)^{q^0}, \dots, \left(\sum_{i=1}^n x_i \theta_i \right)^{q^{n-1}} \right) \\
&= \left(X, X^q, \dots, X^{q^{n-1}} \right).
\end{aligned}$$

□

As a consequence, we can write the isomorphism φ and its inverse φ^{-1} in matrix form.

Corollary 2.9. *For $X = x_1\theta_1 + \dots + x_n\theta_n \in K$ we have*

$$\varphi(X) = \left(X, X^q, \dots, X^{q^{n-1}} \right) \Delta^{-1},$$

and

$$\varphi^{-1}(x_1, \dots, x_n) = ((x_1, \dots, x_n)\Delta)[1] = X,$$

where $((x_1, \dots, x_n)\Delta)[1]$ denotes the first component of the vector $(x_1, \dots, x_n)\Delta$.

As it was pointed out in [4], the matrix form of φ and φ^{-1} allows us to lift a quadratic multivariate system to a q -weight two polynomial in a more simple and efficient way. Moreover, the public key of a scheme derived from the big-field idea can be generated using matrix products. These two facts are the content of the proof that we give here for the following theorem. For simplicity, in the next theorem \vec{x} denotes the vector (x_1, \dots, x_n) and \vec{X} denotes the vector $(X, X^q, \dots, X^{q^{n-1}})$.

Theorem 2.10. *Consider a map $f : k^n \rightarrow k^n$. Then, each component of f is a quadratic polynomial over k if and only if the lifting $F = \varphi^{-1} \circ f \circ \varphi$ is a q -weight two polynomial in $K[X]$.*

Moreover, we can give the exact matrix representation of F : if each component f_i of f is

of the form $f_i(\vec{x}) = \vec{x}A_i\vec{x}^t + B_i\vec{x}^t + c_i$, where $A_i \in \mathcal{M}_{n \times n}(k)$, $B_i \in \mathcal{M}_{1 \times n}(k)$, and $c_i \in k$, $i = 1, \dots, n$, then its lifting is the q -weight two polynomial

$$F = \vec{X}A\vec{X}^t + B\vec{X}^t + c,$$

where

$$A = \theta_1\Delta^{-1}A_1(\Delta^{-1})^t + \dots + \theta_n\Delta^{-1}A_n(\Delta^{-1})^t,$$

$$B = \theta_1B_1(\Delta^{-1})^t + \dots + \theta_nB_n(\Delta^{-1})^t,$$

$$c = (\theta_1, \dots, \theta_n)(c_1, \dots, c_n)^t.$$

In particular, if each component of f is linear, its lifting is the q -weight one polynomial

$$F = B\vec{X}^t + c.$$

Proof. (\Rightarrow) For simplicity we suppose that $f_i(\vec{x}) = \vec{x}A_i\vec{x}^t$, where $A_i \in \mathcal{M}_{n \times n}(k)$, $i = 1, \dots, n$. By Corollary 2.9 we have $\varphi(X) = \vec{X}\Delta^{-1}$, and so

$$\begin{aligned} f(\varphi(X)) &= f(\vec{X}\Delta^{-1}) \\ &= \left(\vec{X}\Delta^{-1}A_1(\Delta^{-1})^t\vec{X}^t, \dots, \vec{X}\Delta^{-1}A_n(\Delta^{-1})^t\vec{X}^t \right) \\ &= (y_1, \dots, y_n), \end{aligned}$$

where $y_i = \vec{X}\Delta^{-1}A_i(\Delta^{-1})^t\vec{X}^t$. Using one more time Corollary 2.9 we get

$$\varphi^{-1}(y_1, \dots, y_n) = ((y_1, \dots, y_n)\Delta)[1].$$

Therefore

$$\begin{aligned} F(X) &= \varphi^{-1}(f(\varphi(X))) \\ &= \varphi^{-1}(y_1, \dots, y_n) \\ &= ((y_1, \dots, y_n)\Delta)[1] \\ &= (y_1, \dots, y_n)(\theta_1, \dots, \theta_n)^t \\ &= \vec{X}\left(\theta_1\Delta^{-1}A_1(\Delta^{-1})^t + \dots + \theta_n\Delta^{-1}A_n(\Delta^{-1})^t\right)\vec{X}^t. \end{aligned}$$

Thus, F is a q -weight two polynomial with associated matrix

$$A = \theta_1 \Delta^{-1} A_1 (\Delta^{-1})^t + \cdots + \theta_n \Delta^{-1} A_n (\Delta^{-1})^t.$$

(\Leftarrow) For simplicity we suppose that the lifting $F = \varphi^{-1} \circ f \circ \varphi$ is a q -weight two polynomial of the form $F = \vec{X} A \vec{X}^t$, where $A \in \mathcal{M}_{n \times n}(K)$. Notice that $f = \varphi \circ F \circ \varphi^{-1}(\vec{x})$. If we set $X = \varphi^{-1}(\vec{x})$, then by Lemma 2.8 we infer that $\vec{X} = \vec{x} \Delta$. Hence,

$$F(\varphi^{-1}(\vec{x})) = F(X) = \vec{X} A \vec{X}^t = \vec{x} \Delta A \Delta^t \vec{x}^t = \sum_{j=1}^{n-1} \sum_{i=1}^{n-1} a_{ij} x_i x_j,$$

where the scalars $a_{ij} \in K$ are the components of the matrix $\Delta A \Delta^t$.

If we write each scalar a_{ij} in terms of the basis $\theta_1, \dots, \theta_n$, we get that $F(\varphi^{-1}(\vec{x})) = f_1(\vec{x}) \theta_1 + \cdots + f_n(\vec{x}) \theta_n$, where f_i is a quadratic polynomial in $k[x_1, \dots, x_n]$. Therefore, $f = \varphi(F(\varphi^{-1}(\vec{x}))) = (f_1(\vec{x}), \dots, f_n(\vec{x}))$ is a quadratic system from k^n to k^n .

□

The procedure in the proof of the implication (\Rightarrow) of Theorem 2.10 can be used to lift a quadratic system $f : k^n \rightarrow k^n$ to a polynomial $F : K \rightarrow K$ (see Section 5.2). Moreover, in Chapter 7 we will use the procedure that we just showed in the proof of the implication (\Leftarrow) to generate efficiently the public key of the new encryption scheme.

In the next two chapters we discuss two encryption schemes based on the big-field idea, which are related to our work: the Matsumoto Imai and the Hidden Field Equations schemes.

Chapter 3

Matsumoto-Imai scheme (MI)

The first relevant MPK encryption scheme was proposed in 1988 by Matsumoto and Imai [35]. They named their scheme C^* but it was later known as the MI cryptosystem. MI is built via the big-field idea with a specific and invertible core polynomial F . Although MI was broken, it has inspired many of the proposed schemes in multivariate public key cryptography.

3.1 Description of MI

Let k be a finite field of characteristic 2 and size q , and let $g(y) \in k[y]$ be an irreducible polynomial of degree n . We define the quotient $K = k[y]/(g(y))$, which is an extension field of degree n of k . Let $\varphi : K \rightarrow k^n$ be the k -linear isomorphism between K and k^n given by

$$\varphi(u_1 + u_2y + \dots + u_ny^{n-1}) = (u_1, u_2, \dots, u_n).$$

Choose an integer θ such that $0 < \theta < n$ and $\gcd(q^\theta + 1, q^n - 1) = 1$. Consider the function $F : K \rightarrow K$ defined by

$$F(X) = X^{q^\theta + 1}.$$

The conditions assumed on θ guarantee that the function F is invertible. In fact

$$F^{-1}(X) = X^t,$$

where t is a positive integer which satisfies the condition

$$t(1 + q^\theta) \equiv 1 \pmod{(q^n - 1)}.$$

Now choose two invertible affine transformations S and T over k^n . The public key of the MI scheme is the trapdoor function $P : k^n \rightarrow k^n$ given by

$$P(x_1, \dots, x_n) = T \circ \varphi \circ F \circ \varphi^{-1} \circ S(x_1, \dots, x_n).$$

Notice that P is an n -tuple

$$P(x_1, \dots, x_n) = (p_1(x_1, \dots, x_n), \dots, p_n(x_1, \dots, x_n)).$$

The private key consists of the transformations S and T (see Figure 3.1). Since F is a q -weight two polynomial, Theorem 2.10 assures that the polynomials p_1, \dots, p_n are quadratic in the variables x_1, \dots, x_n .

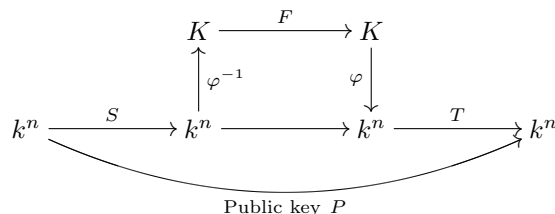


Figure 3.1: MI scheme.

Neither the privacy of the isomorphism φ nor the privacy of the parameter θ contribute to the security of MI. Since all finite fields of size q^n are isomorphic, hiding the isomorphism φ has no effect on the security of MI. Moreover, there are fewer than n choices for θ that satisfy the conditions above, so keeping the parameter θ private does not greatly affect the complexity of the attacks against MI. Therefore the keys for MI are as follows:

Public key of MI. The public key of the MI encryption scheme includes:

- The field k and its structure.

- The trapdoor function $P(x_1, \dots, x_n) = (p_1(x_1, \dots, x_n), \dots, p_n(x_1, \dots, x_n))$.

Private key of MI. The private key of the MI encryption scheme includes:

- The two invertible affine transformations S and T .

Encryption and decryption in MI. To encrypt a plaintext $(x_1, \dots, x_n) \in k^n$ we simply evaluate the public key P in the plaintext to obtain the ciphertext

$$(y_1, \dots, y_n) = T \circ \varphi \circ F \circ \varphi^{-1} \circ S(x_1, \dots, x_n) \in k^n.$$

The plaintext can be recovered from the ciphertext inverting each component of P , i.e.,

$$(x_1, \dots, x_n) = S^{-1} \circ \varphi \circ F^{-1} \circ \varphi^{-1} \circ T^{-1}(y_1, \dots, y_n).$$

3.2 Cryptanalysis of MI

In 1995 Patarin broke the MI scheme in [38]. His key observation was that the core map $Y = X^{q^\theta + 1}$ satisfies the equation

$$XY^{q^\theta} - YX^{q^{2\theta}} = 0,$$

and therefore the public key $(y_1, \dots, y_n) = (p_1(x_1, \dots, x_n), \dots, p_n(x_1, \dots, x_n))$ satisfies the equations

$$(3.1) \quad \sum_{i=1}^n \sum_{j=1}^n a_{ij} x_i y_j + \sum_{i=1}^n b_i x_i + \sum_{j=1}^n c_j y_j + d = 0,$$

where $a_{ij}, b_i, c_j, d \in k$.

The last equations are known as the *linearization equations*. The set of all linearization equations is a vector space over k . Notice that for each ciphertext (y_1, \dots, y_n) we obtain a set of linear equations in the variables x_i for which the plaintext is a solution. Since an

attacker knows the public key, she can generate all of these equations. This leads us to the following question. How many of these equations can be generated from a given ciphertext that are linearly independent? The following theorem was proved by Patarin in [38].

Theorem 3.1. *Let (p_1, \dots, p_n) be the MI public key and fix a ciphertext $\vec{z} = (z_1, \dots, z_n)$. Let $L_{\vec{z}}$ be the space of linear equations derived by replacing the variable y_i by z_i in each linearization equation. Then*

$$\dim L_{\vec{z}} \geq n - \gcd(\theta, n) \geq \frac{2n}{3}.$$

Therefore, for a given plaintext an attacker can generate many linear equations for which the plaintext is a solution. If he can derive a system of linear equations with unique solution, this should be the plaintext. Even if the attacker cannot find directly the plaintext from these linear equations, he can add these linear equations to the quadratic public key equations derived from the ciphertext to make the system of equations much easier to solve.

3.3 Some variants of MI

In order to prevent the linearization equations attack and other subsequent attacks, a lot of MI variants have been proposed. In [41] some MI variants were proposed, among which we find the Minus and the Plus methods. The Minus and the Plus methods are not exclusive for MI, they can be applied to other multivariate public key schemes.

The Minus method. Consider a multivariate scheme with public key $P = (p_1, \dots, p_m)$. The Minus method simply removes the last r polynomials of P and thus it leads to a new public key $P^- = (p_1, \dots, p_{m-r})$. This causes the new system to be highly non injective and hence the Minus method is only suitable to create a signature scheme.

The Minus variant of MI is denoted by MI^- . SFlash [40] is a version of MI^- with the special parameters $q = 2^7$, $n = 37$, $\theta = 11$ and $r = 11$. A version of SFlash, called SFlash^{v2},

was accepted in 2004 by the New European Schemes for Signatures, Integrity and Encryption (NESSIE), for short digital signatures. Unfortunately, SFlash was broken in 2007 by Dubois et al. [21].

The Plus method. Given a multivariate scheme with public key $P = (p_1, \dots, p_m)$, the Plus method adds to P a number s of randomly chosen polynomials f_1, \dots, f_s . Then these polynomials are mixed into the public key using an invertible affine transformation L . Therefore, the new public key is the map $P^+ : k^n \mapsto k^{m+s}$ given by

$$P^+ = L \circ (p_1, \dots, p_m, f_1, \dots, f_s).$$

The Plus method does not try to fix the security of the original system, but rather to make a new injective map P^+ from a non injective map P . In this direction, the Plus method is frequently used together with the Minus method to obtain the so called Minus-Plus method which can be utilized for encryption.

The Minus-Plus variant of MI is denoted by MI^\pm . One problem of this scheme is the choice of the parameters r and s . For security reasons r cannot be too small and for efficiency reasons s cannot be too big.

The MI encryption scheme is a particular case of the HFE encryption scheme that we will discuss in the next chapter. Therefore the KS MinRank attack that we will present in Section 4.4 also breaks MI and the above variants.

Chapter 4

Hidden Field Equations (HFE)

The Hidden Field Equations scheme (HFE) was proposed by Patarin in 1996 [39] as a generalization of the MI scheme. Unlike MI, the base field in HFE does not need to be of characteristic 2. Moreover, the core map in HFE is not as specific as in MI and it is not one-to-one in general.

4.1 Description of HFE

Let k be a finite field of size q . Fix $n \in \mathbb{N}$ and take an irreducible polynomial g over k of degree n . Consider the field extension $K = k[y]/(g(y))$. Then $K \cong k^n$, via the isomorphism $\varphi: K \rightarrow k^n$ defined by $\varphi(u_1 + u_2y + \dots + u_ny^{n-1}) = (u_1, u_2, \dots, u_n)$. For building an HFE scheme we require some special weight two polynomials over K .

Definition 4.1. Let $F: K \rightarrow K$ be a q -weight two polynomial of the form

$$F(X) = \sum_{0 \leq j \leq i}^{n-1} a_{ij} X^{q^i + q^j} + \sum_{i=0}^{n-1} b_i X^{q^i} + c,$$

where the coefficients a_{ij} , b_i and c are choosing randomly in K . We say that F is an *HFE polynomial*. If in addition, we require that $\deg(F) \leq D$, where D is a fixed positive integer, we say that F is an *HFE polynomial with bound D* .

For a fixed D , an HFE scheme is built as follows. First, we randomly choose an HFE polynomial with bound D , say $F: K \rightarrow K$. Then, we randomly choose two invertible affine

transformations S and T over k^n . The public key P is the composition of F with the transformations S and T , together with the isomorphism φ . More precisely,

$$P(x_1, \dots, x_n) = T \circ \varphi \circ F \circ \varphi^{-1} \circ S(x_1, \dots, x_n).$$

Notice that P is an n -tuple

$$P(x_1, \dots, x_n) = (p_1(x_1, \dots, x_n), \dots, p_n(x_1, \dots, x_n)),$$

where each p_i is a multivariate polynomial.

The private key consists of the core map F together with the transformations S and T (see Figure 4.1).

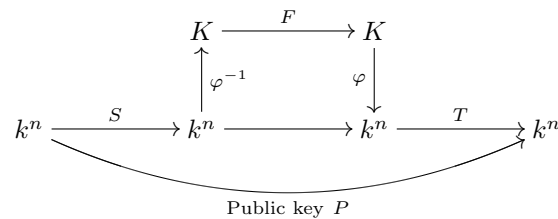


Figure 4.1: HFE scheme.

To construct an HFE scheme we need to be very careful with the choice of the degree bound D . This bound cannot be too high since this would affect the decryption process. Also, D cannot be too small because this would make the system vulnerable to attacks explained below.

Public key of HFE. The public key of the HFE encryption scheme includes:

- The field k and its structure.
- The trapdoor function $P(x_1, \dots, x_n) = (p_1(x_1, \dots, x_n), \dots, p_n(x_1, \dots, x_n))$.

Private key of HFE. The private key of the HFE encryption scheme includes:

- The core polynomial F .

- The two invertible affine transformations S and T .

Encryption in HFE. To encrypt a plaintext $(x_1, \dots, x_n) \in k^n$ we simply evaluate the public key $P = T \circ \varphi \circ F \circ \varphi^{-1} \circ S$ in the plaintext to obtain the ciphertext $(y_1, \dots, y_n) = P(x_1, \dots, x_n) \in k^n$.

Decryption in HFE. To recover the plaintext from the ciphertext we invert each part of P as follows.

- We first compute $(w_1, \dots, w_n) = T^{-1}(y_1, \dots, y_n)$.
- Next we find $Y = \varphi^{-1}(w_1, \dots, w_n)$.
- At this step we must invert F , i.e., we must solve the equation $F(X) = Y$. This equation can have multiple solutions because F is not injective in general. Let \mathcal{Z} be the set

$$\mathcal{Z} = \{X \in K / F(X) = Y\}.$$

This is the main step of the decryption with respect to the complexity. To perform this step we can use Berlekamp's algorithm which has complexity $\mathcal{O}(nD^2 \log_q D + D^3)$. Therefore the degree D of F cannot be too large. If the characteristic of k is odd we can use Cantor-Zassenhaus' algorithm which is slightly faster.

- For each element $X \in \mathcal{Z}$ we find the vector $\varphi(X)$.
- Finally, we apply the transformation S^{-1} to each vector found in the previous step and these are the candidates to be the plaintext. To determine which of these is the original plaintext, some redundant information must be added to the plaintext.

4.2 Toy example

We present here an example that shows step by step how the HFE scheme works. Let $q = 4$ and $n = 4$. Consider the field with four elements $k = GF(2)(a)$, where a is an element of k that satisfies $a^2 + a + 1 = 0$. We select the irreducible polynomial $g(y) = y^4 + y^3 + a^2y^2 + a^2y + a^2 \in k[y]$. A degree n extension field of k is $K = k[y]/(g(y))$. We choose a generator $b \in K$ of the multiplicative group of K such that $g(b) = 0$. We use the following HFE polynomial with degree $D = 8$:

$$\begin{aligned} F(X) &= b^2X^{4+4} + b^{138}X^{4+1} + b^{146}X^{1+1} + b^{95}X^4 + b^{94}X + b^{112} \\ &= b^2X^8 + b^{138}X^5 + b^{95}X^4 + b^{146}X^2 + b^{94}X + b^{112}. \end{aligned}$$

In matrix form F looks like

$$F(X) = \begin{pmatrix} X & X^q & X^{q^2} & X^{q^3} \end{pmatrix} \begin{pmatrix} b^{146} & b^{138} & 0 & 0 \\ 0 & b^2 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} X \\ X^q \\ X^{q^2} \\ X^{q^3} \end{pmatrix} + \begin{pmatrix} b^{94} & b^{95} & 0 & 0 \end{pmatrix} \begin{pmatrix} X \\ X^q \\ X^{q^2} \\ X^{q^3} \end{pmatrix} + b^{112}.$$

Notice that the low degree of F leads to a low rank of the matrix associated to the quadratic part of F , which is one of the main weaknesses of HFE.

We select the invertible affine transformations

$$S(x_1, x_2, x_3, x_4) = \begin{pmatrix} 1 & a & a & 1 \\ a & a & a^2 & a \\ a^2 & 0 & 0 & a \\ 0 & a^2 & a & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \\ a \\ a \end{pmatrix}$$

and

$$T(x_1, x_2, x_3, x_4) = \begin{pmatrix} a^2 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & a & a & a \\ a^2 & a & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} + \begin{pmatrix} 1 \\ a \\ 0 \\ a \end{pmatrix}.$$

The composition $P = T \circ \varphi \circ F \circ \varphi^{-1} \circ S$ yields the public key polynomials

$$p_1 = ax_1^2 + a^2x_1x_2 + x_1x_3 + x_1x_4 + ax_2^2 + x_2x_3 + a^2x_2x_4 + x_3^2 + ax_3x_4 + ax_3 + ax_4^2 + x_4 + a^2,$$

$$p_2 = x_1^2 + a^2x_1x_4 + ax_1 + ax_2^2 + x_2x_3 + a^2x_2x_4 + ax_2 + x_3^2 + x_3x_4 + ax_3 + x_4^2 + a,$$

$$p_3 = x_1x_2 + x_1x_3 + a^2x_1x_4 + a^2x_1 + a^2x_2x_3 + a^2x_2x_4 + x_2 + a^2x_3^2 + a^2x_3x_4 + x_3 + x_4^2 + a,$$

$$p_4 = a^2x_1^2 + a^2x_1x_2 + x_1x_3 + a^2x_1x_4 + ax_1 + x_2x_3 + ax_2x_4 + x_2 + ax_3^2 + a^2x_3 + x_4^2 + ax_4 + a.$$

In order to illustrate the encryption and decryption processes, we choose as plaintext $(x_1, x_2, x_3, x_4) = (0, 1, a, 0)$. After evaluating the public key at the plaintext we get the ciphertext $(y_1, y_2, y_3, y_4) = (a^2, 0, a, 1)$. If we want to recover the plaintext from the ciphertext we first compute

$$T^{-1}(a^2, 0, a, 1) = (a^2, 0, 0, 1).$$

We then find $\varphi^{-1}(a^2, 0, 0, 1) = b^{62}$. Now we need to solve the equation $F(X) = b^{62}$, i.e.,

$$b^2X^8 + b^{138}X^5 + b^{95}X^4 + b^{146}X^2 + b^{94}X + b^{112} = b^{62}.$$

The set of solutions of this equation is

$$\mathcal{Z} = \{b^9, b^{166}\}.$$

We next apply the isomorphism φ to each element of \mathcal{Z} and we get $\varphi(b^9) = (0, a, a^2, a)$ and $\varphi(b^{166}) = (1, a, a, a)$. The candidates to be the plaintext are obtained by applying the transformation S^{-1} , and they are

$$S^{-1}(0, a, a^2, a) = (a^2, a^2, 0, a) \text{ and } S^{-1}(1, a, a, a) = (0, 1, a, 0).$$

We note that one of these elements is the original plaintext $(0, 1, a, 0)$.

4.3 Algebraic attack

Suppose that an attacker intercepts a ciphertext (y_1, \dots, y_n) . Since she has access to the public key $P = (p_1, \dots, p_n)$, she can form the equations

$$\begin{aligned} p_1(x_1, \dots, x_n) - y_1 &= 0, \\ p_2(x_1, \dots, x_n) - y_2 &= 0, \\ &\vdots \\ p_n(x_1, \dots, x_n) - y_n &= 0. \end{aligned}$$

If the attacker finds all the solutions of this system of equations she can determine the plaintext. Solving this system directly is known as the *algebraic attack*. There exist several algorithms to perform this attack, among which we can mention the XL algorithm [8], the Mutant XL algorithms [14, 36, 37] and the F_4 algorithm [24]. In 2003, by means of a special version of the F_4 algorithm, called the F_5 algorithm, Faugère and Joux [25] broke the first HFE challenge ($q = 2, n = 80, D = 96$) proposed by Patarin in [39]. In the F_5 algorithm the field equations $x_i^2 - x_i = 0$ play an important role, allowing to keep low the degrees of the polynomials that appear during the computations. Moreover, they observed that for a system of quadratic equations coming from the public key of an HFE scheme, the algorithm terminates at a lesser degree than for a random system. Also, the degree at which it ends does not depend on the number of variables n , it only depends on the degree D of the secret polynomial. Therefore, for a fixed D and $q = 2$ they claimed that the attack is polynomial in n .

A key concept in the analysis of the complexity of the algebraic attack is the degree of regularity introduced in [2]. We present here a definition of this concept given in [22].

Definition 4.2. Let $P = \{p_1, \dots, p_m\}$ be a set of quadratic polynomials in $k[x_1, \dots, x_n]$ and let I be the ideal generated by the elements of P . We can write each element $g \in I$ in the form

$$g = \sum_{i=1}^n g_i p_i,$$

where $g_i \in k[x_1, \dots, x_n]$. For each integer $d \geq 1$, let V_d be the set of elements of I which are sums of degree d multiples of every p_1, \dots, p_m , i.e.,

$$V_d = \left\{ \sum_{i=1}^n g_i p_i \in I \mid \deg(g_i p_i) = d \right\}.$$

Notice that an element of V_d could have degree smaller than d . We say that a *degree fall* occurs when there exists an element $g \in V_d$ with degree smaller than d . Such an element is called a *mutant*. Naturally, there exist trivial mutants of the form $p_j p_i - p_i p_j$ or $(p_i^{q-1} - 1) p_i$. The *degree of regularity* of the set $P = \{p_1, \dots, p_m\}$ is the smallest d at which a degree fall occurs.

Most of the algorithms that perform the algebraic attack over a system $\{p_1 = 0, \dots, p_n = 0\}$ search at each step of the computations for elements of the ideal $I = \langle p_1, \dots, p_m \rangle$. There exists experimental evidence that such algorithms will terminate at or shortly after the degree of regularity appears. Moreover, if the system comes from the public key of an HFE scheme, the experiments show that the degree of regularity is close to $\log_q D$. However, the only known explicit bound comes from the work of Ding and Hodges [17]. They showed that the degree of regularity of an HFE scheme is bounded by

$$\frac{(q-1) \lceil \log_q D \rceil}{2} + 2.$$

Notice that this bound does not depend on the number of variables n . Therefore, they conclude that

1. if q and D are fixed, the algebraic attack is polynomial in n ;

2. if q is fixed and $D = \mathcal{O}(n^\alpha)$, for some $\alpha \geq 1$, the algebraic attack is quasi-polynomial (order $\mathcal{O}(n^{\log_q D})$).

Notice that this confirms the experimental results from Faugère and Joux [25] for $q = 2$. On the other hand, if q is of size $\mathcal{O}(n)$ the possibility that the algebraic attack is exponential in n remains open. However, the complexity of the decryption process involves the parameter q and then a too large q cannot be used.

In characteristic 2, say $q = 2^l$, it is always possible to see a set of m quadratic equations in n variables over F_q as a set of ml quadratic equations in nl variables over F_2 , via the standard isomorphism between F_q and F_2^l . Therefore the algebraic attack is still effective in this case. For this reason, some authors have suggested to use a high odd characteristic, making the field equations useless and thus the algebraic attack less effective.

4.4 Kipnis-Shamir MinRank attack (KS attack)

The bound D on the degree of the core polynomial F of the HFE scheme implies that the matrix associated to the quadratic part of F has rank at most $r = \lceil \log_q D \rceil$, i.e., this matrix has the form

$$\begin{pmatrix} B & 0 \\ 0 & 0 \end{pmatrix},$$

where B is an $r \times r$ matrix.

In 1999 Kipnis and Shamir [32] proposed a key-recovery attack against HFE that takes advantage of the low rank of this matrix. The KS attack exploits the structure behind the construction of HFE and it links the cryptanalysis of HFE with a linear algebra problem known as the MinRank Problem.

The MinRank Problem. Let k be a finite field and consider m matrices M_1, \dots, M_m over k of size $n \times n$. Given an integer $r \leq n$, the problem is to find, if they exist, scalars

Kipnis-Shamir Modeling, but the equations have degree $r + 1$. In practice, it seems that this approach is equivalent to the Kipnis-Shamir Modeling.

Now we discuss the KS attack. Let k be a finite field of size q and let K be a degree n extension field of k . Consider the standard k -linear isomorphism $\varphi : K \rightarrow k^n$, and let $S, T : k^n \rightarrow k^n$ be two invertible affine transformations. We recall that the HFE public key is

$$P = T \circ \varphi \circ F \circ \varphi^{-1} \circ S,$$

where F is an HFE polynomial with bound D .

We only consider the case q odd. If q is even the attack is slightly different. For simplicity in the exposition, we suppose that S and T are linear transformations. The idea of the KS attack is to lift the public key back to the big field K . More precisely, let $P^* : K \rightarrow K$ be the map

$$P^* = \varphi^{-1} \circ P \circ \varphi = (\varphi^{-1} \circ T \circ \varphi) \circ F \circ (\varphi^{-1} \circ S \circ \varphi).$$

Let $S^* = \varphi^{-1} \circ S \circ \varphi$ and $T^* = \varphi^{-1} \circ T \circ \varphi$. We then have that

$$P^* = T^* \circ F \circ S^*,$$

and thus

$$(4.1) \quad T^{*-1} \circ P^* = F \circ S^*.$$

Equation (4.1) is the starting point for finding T^{*-1} and S^* , and hence T and S . Since S is a linear transformation, by Theorem 2.10, there exist $s_0, \dots, s_{n-1} \in K$ such that

$$S^*(X) = \sum_{i=0}^{n-1} s_i X^{q^i}.$$

Let A be the matrix associated to the quadratic part of F and let A^* be the matrix associated to the quadratic part of P^* . Theorem 2.10 provides a formula for finding A^* . From Equation (4.1), we will get a key relation between the matrices A and A^* . First, we give the explicit form of the matrices associated to $T^{*-1} \circ P^*$ and $F \circ S^*$.

Lemma 4.3. ([15], Sect. 2.4) *The matrix associated to $F \circ S^*$ is*

$$\Lambda A \Lambda^t,$$

where

$$\Lambda = \begin{pmatrix} s_0 & s_{n-1}^q & \cdots & s_1^{q^{n-1}} \\ s_1 & s_0^q & \cdots & s_2^{q^{n-1}} \\ \vdots & \vdots & \ddots & \vdots \\ s_{n-1} & s_{n-2}^q & \cdots & s_0^{q^{n-1}} \end{pmatrix}.$$

Since $\text{Rank}(\Lambda A \Lambda^t) = \text{Rank}(A) \leq r$, the matrix $\Lambda A \Lambda^t$ has low rank as well. This is one of the keys of the attack.

Now we look at the matrix associated to $T^{*-1} \circ P^*$. Notice that $T^{*-1} = (T^{-1})^*$, and so, by Theorem 2.10, there exist scalars $t_0, \dots, t_{n-1} \in K$ such that

$$T^{*-1}(X) = \sum_{i=0}^{n-1} t_i X^{q^i}.$$

Hence, we have that

$$T^{*-1} \circ P^*(X) = \sum_{i=0}^{n-1} t_i P^*(X)^{q^i}.$$

The matrix associated to $P^*(X)^{q^i}$ is closely related to the matrix A^* associated to $P^*(X)$.

The relation is given by the map Γ in Definition 4.4.

Definition 4.4. For $l = 0, \dots, n-1$ and $B = (b_{i,j})_{i,j=0}^{n-1} \in \mathcal{M}_{n \times n}(K)$, let $\Gamma_l(B) \in \mathcal{M}_{n \times n}(K)$

be the matrix defined by

$$\Gamma_l(B)_{i,j} = b_{i-l,j-l}^{q^l},$$

where $i-l$ and $j-l$ are taken modulo n .

Lemma 4.5. ([15], Sect. 2.4) *The matrix associated to $P^*(X)^{q^i}$ is $\Gamma_l(A^*)$.*

Corollary 4.6. *The matrix associated to $T^{*-1} \circ P^*$ is*

$$\sum_{l=0}^{n-1} t_l \Gamma_l(A^*).$$

Equation (4.1) gives us the relation

$$(4.2) \quad \Lambda A \Lambda^t = \sum_{l=0}^{n-1} t_l \Gamma_l(A^*).$$

Since $\text{Rank}(\Lambda A \Lambda^t) \leq r$, we derive the MinRank problem

$$(**) \quad \text{Rank} \left(\sum_{l=0}^{n-1} t_l \Gamma_l(A^*) \right) \leq r.$$

An attacker can compute the matrices $\Gamma_l(A^*)$ since they come from the public key. Let us suppose the attacker finds a solution for the MinRank problem (**), say t'_0, \dots, t'_{n-1} . Even if these are not the original coefficients of T^{*-1} , she can construct an equivalent private key (equivalent keys are studied in [47, 45, 46]). Therefore, without loss of generality, we suppose that she finds the original coefficients t_0, \dots, t_{n-1} of T^{*-1} . Thus, she can compute the map T^{*-1} and then T .

We now discuss how to recover the transformation S . Let $\tilde{A} = \sum_{l=0}^{n-1} t_l \Gamma_l(A^*)$ be the matrix found to solve the MinRank Problem (**). From (4.2) we have that

$$\tilde{A} = \Lambda A \Lambda^t,$$

and thus

$$\Lambda^{-1} \tilde{A} = A \Lambda^t.$$

We assume, without loss of generality, that the rank of A is exactly r . Since the last $n - r$ rows of A are zero, then the last $n - r$ rows of $A \Lambda^t$ are also zero. This gives an attacker $(n - r)n$ q -weight one equations on the n coefficients s_i of S^* (each entry of Λ^{-1} is a Frobenius power of some s_i). Since the matrix \tilde{A} also has rank r , only $(n - r)r$ of these equations are not redundant. Each q -weight one equation leads to n linear equations on new n variables

over the small field k . Hence, she has $n(n-r)r$ linear equations in n^2 variables to find S^* and then S .

Even if the attacker does not find the original private key, she finds transformations S' and T' , and a polynomial F' such that the public key can also be obtained as $P = T' \circ \varphi \circ F' \circ \varphi^{-1} \circ S'$. Therefore she can use this equivalent private key to find pre-images of a given ciphertext. This concludes the KS attack.

The main part of the KS attack, with respect to complexity, is solving the MinRank problem. This leads to a highly overdetermined system (either Kipnis-Shamir or Minors Modeling). Based on this, the authors of the KS attack conjectured that their attack was polynomial in the number of variables and thus effective. However, they did not perform experiments. Later on Ding et al. [30] reviewed the KS attack and they concluded from their experiments that this attack is not as effective as originally claimed. Since then, the attack was considered theoretical. However, recently Faugère et al. [4] improved the KS attack and were able to break HFE and its generalization Multi-HFE. They restated the attack with the matrices coming from the public key whose coefficients are in the small field k , in contrast to the original attack which uses matrices with coefficients in the big field K . This makes the improved KS attack significantly faster than the original KS attack.

In practice, the only difference between the original KS attack and the improved KS attack from [4], is that the improved one uses the public key matrices. In our experiments in Chapter 7 we use the improved KS attack to test our new scheme.

4.5 Some variants of HFE

There have been many attempts to build secure and efficient variants of HFE for both signature and encryption schemes. Here we discuss some of these encryption variants and their security.

The Minus and Plus methods discussed in Section 3.3 can be applied to HFE in a similar way to MI. Initially it was thought that the Minus method prevents the KS attack. However, just like HFE, this variant is vulnerable to the improved KS attack discussed in the Section 4.4 with a small modification to make it suitable for this variant [4].

One of the latest variants of HFE, namely Multi-HFE [29], proposes to use as core map a system of multivariate polynomials over an extension field instead of a single polynomial as in HFE. In what follows we describe a Multi-HFE scheme with an embedding.

Let k be a finite field of odd characteristic and size q . Select a degree n irreducible polynomial $g(y) \in k[y]$ and let $K = k[y]/(g(y))$ be a degree n extension field of k . Consider the standard k -linear isomorphism $\varphi : K \rightarrow k^n$. Fix an integer $N > 1$ and choose a map $\vec{F} : K^N \rightarrow K^N$ of the form

$$\vec{F}(X_1, \dots, X_N) = (F_1(X_1, \dots, X_N), \dots, F_N(X_1, \dots, X_N)),$$

where each F_l is a randomly chosen quadratic polynomial in $K[X_1, \dots, X_N]$ with the shape

$$F_l(X_1, \dots, X_N) = \sum_{1 \leq i \leq j \leq N} a_{l,i,j} X_i X_j + \sum_{1 \leq i \leq N} b_{l,i} X_i + c_l.$$

Now randomly choose two invertible affine transformations S and T over k^{nN} and for a small positive integer $r < n$ consider the affine embedding $\pi : k^{nN-r} \rightarrow k^{nN}$. The public key of Multi-HFE is then the map $P : k^{nN-r} \rightarrow k^{nN}$ given by

$$P = T \circ (\varphi \times \dots \times \varphi) \circ \vec{F} \circ (\varphi^{-1} \times \dots \times \varphi^{-1}) \circ S \circ \pi.$$

Notice that $N = 1$ and $r = 0$ correspond to the basic HFE scheme. The parameter N must be small because the decryption requires to invert the map \vec{F} . The authors recommended to use $N = 3$. They proposed to use a base field of high characteristic in order to prevent the algebraic attack. Also, they used an embedding with the intention of destroying the hidden field structure which is exploited by the KS attack. However, in [4] the authors were able to

generalize the KS attack and then break a more general version of Multi-HFE. This version uses more general core polynomials F_l with the shape

$$F_l(X_1, \dots, X_N) = \sum_{1 \leq i \leq j \leq N} \sum_{\substack{0 \leq u, v < n \\ q^u + q^v \leq d}} a_{l,i,u,j,v} X_i^{q^u} X_j^{q^v} + \sum_{1 \leq i \leq N} \sum_{\substack{0 \leq u < n \\ q^u \leq d}} b_{l,i,u} X_i^{q^u} + c_l,$$

where $a_{l,i,u,j,v}, b_{l,i,u}, c_l \in K$, and d is a fixed positive integer. The bound d cannot be too large because the inversion of the map \vec{F} would be very slow. Therefore, the total degree of each polynomial F_l must be small.

4.6 The history of HFE

In this section we carry out a brief historical review of the HFE scheme. After breaking the MI scheme in [38], Patarin proposed the generalization HFE [39]. In the extended version of [39] Patarin left two challenges. Challenge 1 is an instance of an HFE encryption scheme with parameters $q = 2$, $n = 80$ and $D = 96$. Challenge 2 is an instance of an HFE signature scheme with parameters $q = 16$, $n = 36$ and $D = 4352$, where 4 polynomials are not given public. Since then, several variants and attacks related to HFE have emerged.

In 1999 Kipnis and Shamir [32] proposed a key-recovery attack over HFE. They linked the cryptanalysis of HFE with a known linear algebra problem called the MinRank Problem, and they introduced an algebraic method to solve the derived MinRank Problem. The last step of this method consists of solving a highly overdetermined system of quadratic equations over a finite field. With this argument, they claimed that their attack was polynomial in the number of variables and thus effective. However, they did not perform any experiments.

In 2001 Nicolas Courtois [9] proposed some new attacks on HFE and he was able to break the HFE challenge 1 with one of these attacks in about 2^{62} operations. Also, he proposed another way (minors modeling) to solve the MinRank Problem derived from the KS attack.

In 2002 Christopher Wolf [44] presented a review of HFE. He surveyed the main variants,

attacks and applications of HFE at that moment. Moreover, he proposed some new HFE variants.

The HFE challenge 1 was broken in 2003 by Faugère and Joux by means of the Gröbner basis algorithm F_5 [25]. Based on their experiments, they concluded that the cryptanalysis of HFE can be performed in polynomial time in the case $q = 2$.

In 2004 Courtois [10] introduced three algebraic attacks over $GF(2^k)$ using modifications of the XL algorithm. He claimed that the best attack by means of these algorithms over the HFE challenge 2 takes about 2^{63} operations. On the other hand, in [2] the authors introduced a key concept for the study of the complexity of the algebraic attack on HFE: the degree of regularity (see Definition 4.2).

In 2005 Wolf and Preneel investigated the existence of equivalent keys in multivariate quadratic schemes like HFE, MI and some variations [47, 45]. The existence of equivalent keys was first reported by Kipnis and Shamir [32] as isomorphic keys. Two private keys are equivalent if they compose into the same public key. The authors in [45] proved that there exist $nq^{2n}(q^n - 1)^2$ equivalent keys for an HFE scheme and they showed that equivalent keys can be used to reduce the implementations memory using a special equivalent key. Moreover, they stated that equivalent keys could be applied to cryptanalysis since they allow us to concentrate on special forms of the private key. In particular they showed that using affine transformations rather than linear does not strengthen the security of an HFE scheme. Later on the same authors did an additional study on equivalent keys in [46].

In 2006 Granboulan, Joux and Stern [28] derived heuristic asymptotic bounds on the degree of regularity in the case $q = 2$. Based on these bounds they concluded that inverting HFE is quasi-polynomial in the case $q = 2$. Their approach was to bind the degree of regularity of an HFE scheme with the degree of regularity of a lifted system over an extension field. In 2010 Dubois and Gama [22] gave a rigorous mathematical foundation for the ideas

in [28], and they derived an algorithm to compute a bound on the degree of regularity of an HFE scheme for a general q .

The original HFE was proposed by Patarin over $GF(2)$. However, in 2008 Ding, Schmidt and Werner proposed an odd characteristic HFE scheme with an embedding modifier [18]. They noticed that for high odd characteristic, the field equations are not useful to reduce the degree of the polynomials used in the Gröbner basis computations and therefore the algebraic attack does not work in this case. The embedding modifier seeks to avoid the KS MinRank attack. They proposed an example for a practical application over $GF(11)$ and gave a challenge problem over $GF(7)$.

In 2008 Ding et al. [30] reviewed the KS attack and concluded from their experiments that the attack is not as effective as originally claimed.

Also in 2008 the authors in [29] proposed a multivariate version of HFE (Multi-HFE). Instead of using a single polynomial as core map, they proposed to use a system of multivariate polynomials over an extension field. In addition, they used odd characteristic and an embedding modifier for this proposal. They suggested a practical implementation where the core map is a system of 3 polynomials in 3 variables over $GF(31)$.

In 2010 Charles Boulliguet et al. exhibited a family of weak keys of HFE [5]. In particular, if the coefficients of the core polynomial map are chosen in the base field, they showed a key-recovery attack over the associated HFE scheme, and they emphasized that this attack is effective even if the core polynomial has high degree.

In 2011 Ding and Hodges [17] derived the first explicit bound on the degree of regularity and concluded that, if q and D are fixed, the algebraic attack is polynomial in the number of variables n .

In 2012 the authors in [4] improved and generalized the KS MinRank attack and were able to break the challenges from [18] and from the generalization Multi-HFE [29].

Chapter 5

The Zhuang-Zi algorithm (ZZ algorithm)

The problem of solving a system of multivariate polynomial equations over a finite field is a central problem in cryptanalysis. The security of an MKP scheme is based on the difficulty of solving a system of quadratic equations over a finite field. Moreover, there exist algebraic attacks against some symmetric schemes like AES [11]. The main tools for solving a multivariate system over a finite field come from the family of XL algorithms [8, 14, 36, 37] and Gröbner basis algorithms [6, 24].

In 2006 Ding et al. introduced a new algorithm for solving a system of multivariate polynomial equations over a finite field which they called the Zhuang-Zi algorithm [16]. To construct the new multivariate trapdoor function in the next chapter, we introduce a reduction method inspired by the way in which ZZ works.

The idea of ZZ is to lift the multivariate system to a univariate polynomial F over an extension field of the original field, and then try to derive a low degree polynomial Ψ by means of a special reduction method that involves the Frobenius powers of F . Due to the special form of the reduction process, the roots of F are also roots of the low degree polynomial Ψ . The roots of Ψ can be found efficiently by means of Berlekamp's algorithm.

We begin by describing ZZ and showing a step-by-step trivial example. Then we show nontrivial examples where ZZ works but the best Gröbner basis algorithms do not.

5.1 Description of ZZ

Let k be a finite field of size q and let K be a degree n extension of k . Consider the standard k -linear isomorphism $\varphi : K \rightarrow k^n$. Suppose that we want to solve the system of multivariate equations

$$f_1(x_1, \dots, x_n) = 0, \dots, f_n(x_1, \dots, x_n) = 0,$$

where $f_i(x_1, \dots, x_n) \in k[x_1, \dots, x_n]$.

Let $f : k^n \rightarrow k^n$ be the map defined by $f = (f_1, \dots, f_n)$. Consider the lifting $F : K \rightarrow K$ defined by

$$F = \varphi^{-1} \circ f \circ \varphi.$$

The proof of Theorem 2.10 provides a very efficient method for writing F as a polynomial in $K[X]/(X^{q^n} - X)$. The ZZ algorithm constantly uses the following procedure with name Reduce-by-degree.

Definition 5.1. (Reduce-by-degree). Let $S = \{S_0, \dots, S_{l-1}\}$ be a set of polynomials in $K[X]/(X^{q^n} - X)$. Create a sequence M with the monomials of all the elements in S and sort M in decreasing order with respect to the degree. Let $|M|$ be the size of M . For each element of S , extract its coefficients with respect to M and create a row vector in $K^{|M|}$ with them in the order given by M . Then construct the *Macaulay matrix* associated to S , i.e., the $n \times |M|$ matrix whose rows are the vectors we just described. Use Gaussian elimination to reduce the Macaulay matrix. With the nonzero rows of the matrix in echelon form produce and return a new set of basis polynomials, which we will also call S . Abusing the notation, write $S = \{S_0, \dots, S_{t-1}\}$, with $t \leq l$ and sort S such that S_{t-1} is the polynomial of lowest degree in S .

Before stating the ZZ algorithm, we illustrate how the procedure Reduce-by-degree works. Let $k = GF(3)$, and for the irreducible polynomial $g(y) = y^2 + 2y + 2 \in k[y]$ consider the

degree n extension field $K = k[y]/(g(y))$. We choose a generator $b \in K$ of the multiplicative group of K such that $g(b) = 0$. Let S be the set in $K[X]$ consisting of the polynomials

$$\begin{aligned} S_1(X) &= bX^7 + b^5X^5 + b^2X^4 + b^3X + b, \\ S_2(X) &= b^3X^7 + bX^5 + X^3 + b^5X^2 + b^2, \\ S_3(X) &= X^7 + b^2X^4 + X^3 + b^5X^2 + b^3X + b^3. \end{aligned}$$

The sequence with the monomials of all the elements in S in decreasing order with respect to the degree is $M = (X^7, X^5, X^4, X^3, X^2, X^1, X^0)$. Thus, the Macaulay matrix associated to S is

$$\begin{array}{ccccccc} X^7 & X^5 & X^4 & X^3 & X^2 & X^1 & X^0 \\ \left(\begin{array}{ccccccc} b & b^5 & b^2 & 0 & 0 & b^3 & b \\ b^3 & b & 0 & 1 & b^5 & 0 & b^2 \\ 1 & 0 & b^2 & 1 & b^5 & b^3 & b^3 \end{array} \right). \end{array}$$

The echelon form of this matrix is

$$\begin{array}{ccccccc} X^7 & X^5 & X^4 & X^3 & X^2 & X^1 & X^0 \\ \left(\begin{array}{ccccccc} 1 & 0 & b^2 & 1 & b^5 & b^3 & b^3 \\ 0 & 1 & 1 & 1 & b^5 & b & b^5 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array} \right). \end{array}$$

Therefore, $\text{Reduce-by-degree}(S)$ is the set with the two polynomials

$$\begin{aligned} S_1(X) &= X^7 + b^2X^4 + X^3 + b^5X^2 + b^3X + b^3, \\ S_2(X) &= X^5 + X^4 + X^3 + b^5X^2 + bX + b^5. \end{aligned}$$

We now continue with the exposition of the ZZ algorithm. The ZZ algorithm has as input the polynomials $f_1, \dots, f_n \in k[x_1, \dots, x_n]$ and a positive integer D_0 such that every univariate

polynomial equation with degree less than or equal to D_0 can be solved efficiently using a chosen algorithm, such as Berlekamp's algorithm. The output of the algorithm is the set of the solutions in k^n for the system $\{f_1 = 0, \dots, f_n = 0\}$. The algorithm performs the following steps.

1. Set $f = (f_1, \dots, f_n)$ and $F = \varphi^{-1} \circ f \circ \varphi$. If $\deg(F) \leq D_0$, go to Step 4, otherwise let F_0, F_1, \dots, F_{n-1} be the Frobenius powers of F . If $\deg(F_i) \leq D_0$ for some $i \in \{0, 1, \dots, n-1\}$, go to Step 4, otherwise go to the next step.
2. Let S be the set consisting of the Frobenius powers of F . Compute $\text{Reduce-by-degree}(S)$ to produce a new set $S = \{S_0, \dots, S_{t-1}\}$. If $\deg(S_{t-1}) \leq D_0$, go to the Step 4, otherwise go to the next step.
3. For $i = 0, 1, \dots, t-1$ and $j = 0, 1, \dots, n-1$ compute

$$X^{q^j} S_i(X) \bmod (X^{q^n} - X).$$

Add these polynomials to S . Compute $\text{Reduce-by-degree}(S)$, and abusing the notation again, use t to denote the size of this new S . If $\deg(S_{t-1}) \leq D_0$, go to the Step 4, otherwise repeat this step.

4. At this step there exists a polynomial Ψ of degree less than or equal to D_0 . Apply the chosen algorithm to find the set \mathcal{Z} of roots of this polynomial Ψ . The set \mathcal{Z} contains the roots of F . The solutions of the system $\{f_1 = 0, \dots, f_n = 0\}$ correspond to the roots of F by means of the standard isomorphism $\varphi : K \rightarrow k^n$.

5.2 Toy example

We consider the field with four elements $k = GF(2)(a)$, where $a^2 + a + 1 = 0$. We select the irreducible polynomial $g(y) = y^2 + y + a^2 \in k[y]$ to generate the extension field of degree

two $K = k[y]/(g(y))$. We choose a generator $b \in K$ of the multiplicative group of K such that $g(b) = 0$. The system to be solved consists of the following two quadratic equations in two variables with coefficients in k :

$$\begin{aligned} f_1(x_1, x_2) &= ax_1^2 + a^2x_1x_2 + a = 0, \\ f_2(x_1, x_2) &= ax_1^2 + a^2x_2^2 + ax_2 + a^2 = 0. \end{aligned}$$

We show how to lift the map $f = (f_1, f_2)$ to the map $F = \varphi^{-1} \circ f \circ \varphi$ by means of the method described in the proof of Theorem 2.10. We need to write f_1 and f_2 in matrix form:

$$\begin{aligned} f_1(x_1, x_2) &= \begin{pmatrix} x_1 & x_2 \end{pmatrix} \begin{pmatrix} a & 0 \\ a^2 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 0 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + a \\ &= \vec{x}A_1\vec{x}^t + B_1\vec{x}^t + c_1, \\ f_2(x_1, x_2) &= \begin{pmatrix} x_1 & x_2 \end{pmatrix} \begin{pmatrix} a & a \\ a & a^2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 0 & a \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + a^2 \\ &= \vec{x}A_2\vec{x}^t + B_2\vec{x}^t + c_2. \end{aligned}$$

One basis for K over k is $(\theta_1, \theta_2) = (1, b)$, and thus the isomorphism φ and its inverse can be represented using the matrix

$$\Delta = \begin{pmatrix} \theta_1 & \theta_1^q \\ \theta_2 & \theta_2^q \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ b & b^4 \end{pmatrix}.$$

According to Theorem 2.10, we have that $F = \vec{X}A\vec{X}^t + B\vec{X}^t + c$, where

$$\begin{aligned} A &= \theta_1\Delta^{-1}A_1(\Delta^{-1})^t + \theta_2\Delta^{-1}A_2(\Delta^{-1})^t, \\ B &= \theta_1B_1(\Delta^{-1})^t + \theta_2B_2(\Delta^{-1})^t, \text{ and} \\ c &= (\theta_1, \theta_2)(c_1, c_2)^t. \end{aligned}$$

So we have that

$$\begin{aligned}
F &= \vec{X}A\vec{X}^t + B\vec{X}^t + c \\
&= \begin{pmatrix} X & X^4 \end{pmatrix} \begin{pmatrix} b^{13} & b^{13} \\ b^7 & b \end{pmatrix} \begin{pmatrix} X \\ X^4 \end{pmatrix} + \begin{pmatrix} b^{11} & b^{11} \end{pmatrix} \begin{pmatrix} X \\ X^4 \end{pmatrix} + b^7 \\
&= bX^8 + b^5X^5 + b^{11}X^4 + b^{13}X^2 + b^{11}X + b^7.
\end{aligned}$$

Since this is a toy example, the polynomial F can be factorized directly to get

$$F(X) = (X + b^9)^2(X^6 + b^3X^4 + b^4X^3 + b^7X^2 + b^7X + b^3),$$

and so we observe that the unique root of F is $X = b^9$, since it is easy to verify that the second factor is irreducible over K using Magma [34]. In order to show how the ZZ algorithm works, we set $D_0 = 2$, i.e., we will search for a degree two polynomial Ψ . Notice that here $q = 2$ and $n = 4$ and therefore the Frobenius powers of F are

$$\begin{aligned}
F_0(X) &= F(X) = bX^8 + b^5X^5 + b^{11}X^4 + b^{13}X^2 + b^{11}X + b^7, \\
F_1(X) &= F^4(X) = b^7X^8 + b^5X^5 + b^{14}X^4 + b^4X^2 + b^{14}X + b^{13}.
\end{aligned}$$

Let $S = \{F_0, F_1\}$ and apply $\text{Reduce-by-degree}(S)$ to get the polynomials

$$\begin{aligned}
S_0(X) &= X^8 + b^{11}X^4 + b^{12}X^2 + b^{11}X + b^6, \\
S_1(X) &= X^5 + b^{10}X^4 + b^{10}X.
\end{aligned}$$

Because we have not yet reached a degree two polynomial, we multiply both $S_0(X)$ and $S_1(X)$ by X and X^4 to obtain four polynomials that we add to S . After applying Reduce-

by-degree(S) we obtain the new set S with polynomials

$$\begin{aligned}
S_0(X) &= X^{12} + b^4 X^3 + b^{12} X^2 + X + 1, \\
S_1(X) &= X^9 + bX^3 + b^2 X^2 + b^{10} X + b^2, \\
S_2(X) &= X^8 + b^3 X^3 + b^5 X^2 + b^{12} X + b^{10}, \\
S_3(X) &= X^6 + b^{12} X^3 + bX^2 + b^6 X + b, \\
S_4(X) &= X^5 + b^2 X^3 + b^{13} X^2 + b^{11} X + b^6, \\
S_5(X) &= X^4 + b^7 X^3 + b^3 X^2 + b^4 X + b^{11}.
\end{aligned}$$

Again, we still do not have in S a degree two polynomial and thus we multiply each $S_i(X)$ by X and X^4 . The derived set of polynomials are added to S to obtain a set with eighteen polynomials. Reduce-by-degree(S) produces the set of eleven polynomials

$$\begin{aligned}
S_0(X) &= X^{13} + b^{12} X + b^4, \\
S_1(X) &= X^{12} + b^{11} X + b^{11}, \\
S_2(X) &= X^{10} + b^{11} X + b^{10}, \\
S_3(X) &= X^9 + b^9 X + b^2, \\
S_4(X) &= X^8 + b^8 X + b^7, \\
S_5(X) &= X^7 + X + b, \\
S_6(X) &= X^6 + b^3 X + b^8, \\
S_7(X) &= X^5 + b^6 X, \\
S_8(X) &= X^4 + b^{12} X, \\
S_9(X) &= X^3 + b^{11} X + b^{14}, \\
S_{10}(X) &= X^2 + b^{14} X + b^{13}.
\end{aligned}$$

The algorithm terminates since it has found the degree two polynomial $\Psi = X^2 + b^{14} X + b^{13}$.

The set of roots of Ψ is $\{b^4, b^9\}$. However, only b^9 is root of F . Since $b^9 = a^2 + a^2b$, the solution of the system $\{f_1 = 0, f_2 = 0\}$ is $\varphi(b^9) = (a^2, a^2)$.

5.3 Nontrivial examples

The authors of ZZ showed that there exist cases of multivariate systems where their algorithm works and the best Gröbner basis algorithms, like F_4 algorithm, do not. Such nontrivial examples can be generated because the complexity for finding a Gröbner basis for n random quadratic polynomials in n variables is exponential in n . For example, if $k = GF(2^3)$, and K is a degree n extension of k , consider the low degree polynomial

$$F(X) = X^{72} + a_1X^{65} + a_2X^{64} + a_3X^{16} + a_4X^9 + a_5X^8 + a_6X^2 + a_7X + a_8,$$

where the coefficients a_i are randomly chosen from the small field k . The polynomial F has q -weight two. Therefore Theorem 2.10 ensures that the associated multivariate set derived from the composition $\varphi \circ F \circ \varphi^{-1}(x_1, \dots, x_n)$ is a set of quadratic polynomials. The low degree allow us to easily factor the polynomial F over K using Berlekamp's algorithm, regardless of the value of n . Because the complexity of the Gröbner basis algorithms is exponential in n , for large values of n these algorithms fail.

The previous example only uses the first step of ZZ and Berlekamp's algorithm. The authors of ZZ showed another nontrivial example where the Magma implementation of F_4 does not succeed with their PC (1.73 GHz, 1 GB of RAM) and ZZ needs to use at least once the reduction procedure Reduce-by-degree to succeed. Take $k = GF(2)(a)$, where $a^2 + a + 1 = 0$, and define $K = k[y]/(g(y))$, where $g(y) \in k[y]$ is the irreducible polynomial

$$g(y) = y^{12} + y^{11} + ay^{10} + ay^9 + y^8 + y^7 + y^5 + a^2y^4 + ay^3 + a^2y^2 + ay + a.$$

Let $F(X) \in K[X]$ be the polynomial

$$\begin{aligned} F(X) &= a^2 X^{17664} + X^{5440} + aX^{5376} + X^{4416} + aX^{4096} + aX^{1360} \\ &\quad + X^{1344} + X^{1280} + a^2 X^{1024} + a^2 X^{336} + aX^{320} + a^2 X^{276} \\ &\quad + X^{85} + aX^{84} + aX^{64} + aX^{21} + X^{20} + a. \end{aligned}$$

The high degree of F prevents us to solve the equation $F(X) = 0$ directly by Berlekamp's algorithm. On the other hand, the ZZ algorithm produces the low degree polynomial

$$\Psi = X^{276} + aX^{85} + a^2 X^{84} + a^2 X^{64} + a^2 X^{21} + aX^{20} + a.$$

After factoring the polynomial Ψ , using Berlekamp's algorithm, we obtain the solutions $X = 1$ and $X = a$ of the equation $F(X) = 0$.

The ZZ algorithm can be used to perform the algebraic attack over a multivariate public key scheme like HFE. Unfortunately, the experiments show that ZZ is not very effective in this task. The last example was constructed “artificially”, using the Step 3 of ZZ in reverse order. However, ZZ can be combined with other algorithms that perform the algebraic attack, like F_4 or XL, to enhance them.

Finally, in [20] Ding and Schmidt introduced a variant of ZZ called Mutant Zhuang-Zi algorithm which is based on Ding's mutant concept (see Definition 4.2). This variant improves ZZ in a lot of cases.

Chapter 6

New candidates for multivariate trapdoor functions

The weakness of the HFE cryptosystem lies on the use of a low degree core polynomial F . This polynomial is used for both encryption and decryption. The process of decryption involves inverting the map F (search of pre-images). Therefore, if we take a polynomial of high degree the decryption could be impossible, and if otherwise we take a polynomial of low degree the attacks mentioned in Chapter 4 would work.

To overcome this weakness we developed a reduction method for building pairs of HFE polynomials of very high degree, and such that the map constructed with such a pair is easy to invert, using a low degree polynomial derived from a special reduction via q -weight three polynomials. This low degree polynomial is easy to invert by means of Berlekamp's algorithm. In this way, we are able to use two HFE polynomials of high degree to construct a new candidate for a trapdoor function, and a polynomial of small degree as the trapdoor used to invert such trapdoor function.

6.1 The Reduction method

Let us briefly review the main ideas of the ZZ algorithm (Section 5.1) for solving a system of polynomials equations $\{f_1 = 0, \dots, f_n = 0\}$ in n variables over a finite field k of size q . Let K be a degree n extension field of k . Let \mathcal{A} be an algorithm that solves univariate

polynomial equations over the field K , such as Berlekamp's algorithm. Let D_0 be a positive integer such that every univariate polynomial equation with degree at most D_0 can be solved efficiently using the algorithm \mathcal{A} . The first step of ZZ is lifting the system of equations to a polynomial F over K . The goal of ZZ is to obtain a univariate polynomial Ψ with degree at most D_0 and whose roots contain the roots of F . If such a polynomial Ψ can be found, then its roots are obtained using the algorithm \mathcal{A} and so ZZ terminates. If $\deg(F) \leq D_0$, then $\Psi = F$ and the algorithm terminates. Otherwise ZZ computes the set of Frobenius powers of F , say $S = \{F_0, F_1, \dots, F_{n-1}\}$. If there exists in S a polynomial Ψ of degree at most D_0 the algorithm terminates, otherwise ZZ applies the procedure Reduce-by-degree to the set S to produce a new set $S = \{S_0, \dots, S_{t-1}\}$ and looks for a polynomial Ψ of degree at most D_0 in this new set S . If there exists such polynomial Ψ the algorithm terminates, otherwise ZZ multiplies each element S_i by all the monomials X^{q^j} , $j = 0, \dots, n-1$, and appends the polynomials $X^{q^j} S_i$ to the set S , to obtain a new set which we also call S . Next, ZZ applies the procedure Reduce-by-degree to S and looks for a polynomial Ψ of degree less than or equal to D_0 in the new set S . If there exists such polynomial Ψ the algorithm terminates, otherwise the process is repeated (multiplying by the monomials X^{q^j}) with the set of polynomials S derived from Reduce-by-degree until finding a polynomial Ψ of degree at most D_0 . Clearly, the roots of F are also roots of Ψ . The solutions of the system $\{f_1 = 0, \dots, f_n = 0\}$ correspond to the roots of F by means of the standard isomorphism $\varphi : K \rightarrow k^n$.

In this work we developed a method to reduce pairs of high degree HFE polynomials with some similar ideas to those of ZZ. Unlike ZZ, here we start with two HFE polynomials F and \tilde{F} with unknown coefficients. As it occurs in ZZ, we look for a low degree polynomial Ψ that involves the Frobenius powers of these polynomials. One difference is that here the coefficients of these polynomials are to be determined. Moreover, we only multiply by the

monomials X and X^q and the reduction process is not done by means of the procedure Reduce-by-degree. More precisely, let $F : K \rightarrow K$ and $\tilde{F} : K \rightarrow K$ be two high degree HFE polynomials with the shape

$$F(X) = \sum_{0 \leq j \leq i}^{n-1} a_{ij} X^{q^i + q^j} + \sum_{i=0}^{n-1} b_i X^{q^i} + c,$$

$$\tilde{F}(X) = \sum_{0 \leq j \leq i}^{n-1} \tilde{a}_{ij} X^{q^i + q^j} + \sum_{i=0}^{n-1} \tilde{b}_i X^{q^i} + \tilde{c},$$

where the coefficients $a_{ij}, b_i, c, \tilde{a}_{ij}, \tilde{b}_i, \tilde{c} \in K$ are to be determined. Next, let F_0, F_1, \dots, F_{n-1} be the Frobenius powers of F and let $\tilde{F}_0, \tilde{F}_1, \dots, \tilde{F}_{n-1}$ be the Frobenius powers of \tilde{F} , i.e.,

$$F_i(X) = [F(X)]^{q^i} \quad \text{and} \quad \tilde{F}_i(X) = [\tilde{F}(X)]^{q^i}, \quad \text{for } i = 0, 1, \dots, n-1.$$

As in ZZ, let \mathcal{A} be an algorithm that solves univariate polynomial equations over the field K . Let D_0 be a positive integer such that every univariate polynomial equation with degree at most D_0 can be solved efficiently using the algorithm \mathcal{A} .

The key part of this method is to construct a polynomial Ψ of the form

$$\Psi = X \left(\alpha_1 F_0 + \dots + \alpha_n F_{n-1} + \beta_1 \tilde{F}_0 + \dots + \beta_n \tilde{F}_{n-1} \right) +$$

$$X^q \left(\alpha_{n+1} F_0 + \dots + \alpha_{2n} F_{n-1} + \beta_{n+1} \tilde{F}_0 + \dots + \beta_{2n} \tilde{F}_{n-1} \right),$$

such that $\deg(\Psi) \leq D_0$. Notice that Ψ is a q -weight three polynomial.

To accomplish this, we need to determine the coefficients of F and \tilde{F} , also the scalars α_i and β_i , in such a way that the coefficients of the terms in Ψ of degree greater than D_0 are equal to zero. We derive a system of equations from these vanishing coefficients in Ψ of degree higher than D_0 . This yields a system of equations of the form

$$g_1(z_1, z_2, \dots, z_N) = 0, \dots, g_t(z_1, z_2, \dots, z_N) = 0,$$

where the variables z_1, z_2, \dots, z_N are the coefficients of F and \tilde{F} , together with the scalars α_i and β_i . Notice that $N = 2 \left(\frac{n(n+1)}{2} + n + 1 \right) + 4n = n^2 + 7n + 2$ if $q \neq 2$ and $N = 2 \left(\frac{n(n-1)}{2} + n + 1 \right) + 4n = n^2 + 5n + 2$ if $q = 2$.

The number t of equations of this system depends on how small we want the degree bound D_0 to be. More precisely, t is the number of different terms in Ψ with degree higher than D_0 . To invert the trapdoor function, which we will describe in Section 6.4, using the polynomial Ψ , we require that the polynomial Ψ has degree at most D_0 .

If we write each variable z_j in terms of the basis $\{1, y, \dots, y^{n-1}\}$, we obtain a system of quadratic equations. More precisely, each variable z_j in this system can be written in the form

$$(6.1) \quad z_j = u_{1j} + u_{2j}y + \dots + u_{nj}y^{n-1},$$

where u_{1j}, \dots, u_{nj} are n new variables over k . Next, by the linearity of the Frobenius powers, we get

$$(6.2) \quad z_j^{q^i} = u_{1j} + u_{2j}y^{q^i} + \dots + u_{nj}y^{(n-1)q^i}.$$

After writing each power y^m as a linear combination of the elements of the basis $1, y, \dots, y^{n-1}$ with coefficients in k , and group like terms, we get that

$$(6.3) \quad z_j^{q^i} = h_{1j}(u_{1j}, \dots, u_{nj}) + h_{2j}(u_{1j}, \dots, u_{nj})y^2 + \dots + h_{nj}(u_{1j}, \dots, u_{nj})y^{n-1},$$

where each h_{ij} is a linear function with coefficients in k .

We now write each variable of the system $\{g_1 = 0, \dots, g_t = 0\}$ in the form (6.1), and proceed like in (6.2) and (6.3). Comparing the coefficients of the elements of the basis $\{1, y, y^2, \dots, y^{n-1}\}$ we obtain a system of nt quadratic equations in nN variables over k . These equations are in fact bilinear, i.e., each term of these equations has the product of a variable that comes from the coefficients and a variable that comes from the scalars. Thus,

if we randomly fix the variables associated to the scalars we obtain a sparse linear system coming only from the coefficients of F and \tilde{F} . Due to its construction, this linear system has more variables than equations, i.e., $nt < nN$, and hence we can always get nontrivial solutions for it. We then randomly choose one of those solutions to build the high degree polynomials F and \tilde{F} and the reduced polynomial Ψ of degree less than or equal to D_0 , as explained above.

One could be tempted to try something different and randomly choose the variables coming from the coefficients of F and \tilde{F} , and then try to solve the linear system for the variables coming from the scalars, with the purpose of having generic core polynomials F and \tilde{F} . However, this approach leads to a linear system with more equations than variables, and thus, in general this system has no nontrivial solutions.

Remark 6.1. *Our first attempt was to consider a single polynomial F and look for a low degree polynomial Ψ of the form*

$$\begin{aligned} \Psi = & X(\alpha_1 F_0 + \cdots + \alpha_n F_{n-1}) \\ & + X^q(\alpha_{n+1} F_0 + \cdots + \alpha_{2n} F_{n-1}) \\ & + X^{q^2}(\alpha_{2n+1} F_0 + \cdots + \alpha_{3n} F_{n-1}). \end{aligned}$$

The problem with this attempt was that we got a linear system with more equations than variables and then we had difficulties finding solutions for it. For some instances we were able to find solutions for this linear system, but these solutions produced a function F with a low degree K -linear combination of its Frobenius powers. Therefore, F cannot be used to build a multivariate trapdoor function because the KS attack would work against it.

6.2 Complexity of the reduction method and dimension of the solution space

If we take the coefficients of the core polynomials F and \tilde{F} in the small field k , the reduction method is very fast. However, as we pointed out in Section 4.6, this would lead to weak

keys of the associated encryption scheme we will explain in the next chapter. This problem occurs even if the core map polynomials have high degree. Therefore we always consider the coefficients of F and \tilde{F} in the extension field K .

Even with the coefficients in the big field K , the described method leads to a sparse linear system over the small field k with more variables than equations. This system has about n^3 variables and thus the complexity of the reduction method is polynomial: $\mathcal{O}((n^3)^\omega)$, where $2 \leq \omega \leq 3$ is a constant that depends on the elimination algorithm used to solve the sparse linear system.

On the other hand, after we choose the $4n$ scalars α_i and β_i in the system of equations $\{g_i(z_1, z_2, \dots, z_N) = 0 : i = 1, \dots, t\}$, we get a new system over the big field K with t equations and $N - 4n$ variables (the coefficients of F and \tilde{F}). Therefore, in this new system the number of variables exceeds the number of equations by $(N - 4n) - t$. Hence the final linear system over the small field k has at least $n((N - 4n) - t)$ free variables. Then, we have at least $q^{n((N - 4n) - t)} > q^n$ possible choices for the coefficients of the polynomials F and \tilde{F} . Thus, if we choose large parameters q and n , and if we randomly choose a solution from the solution space, it is infeasible for anyone to guess correctly the polynomials we will use. The large dimension of the solution space also ensures that there are sufficiently many choices for the core map.

6.3 How to build the trapdoor function

For building a new candidate for multivariate trapdoor function, we make use of a map of the form $G = (F, \tilde{F}) : K \rightarrow K \times K$, in which F and \tilde{F} have been constructed by the method described in Section 6.1. We select two invertible affine transformations $S : k^n \rightarrow k^n$ and $T : k^{2n} \rightarrow k^{2n}$. Similar to HFE, the multivariate trapdoor function will be the composition from k^n to k^{2n} given by $P = T \circ (\varphi \times \varphi) \circ G \circ \varphi^{-1} \circ S$ (see Figure 6.1).

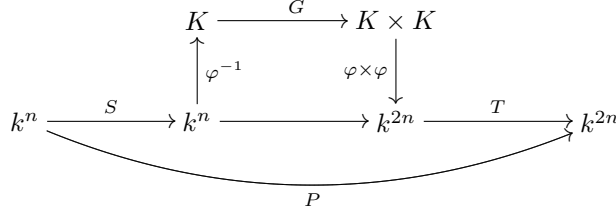


Figure 6.1: New candidate for multivariate trapdoor function.

In order to consider the new and improved version of the MinRank KS attack from [4] against the new trapdoor function P , we would like to point out that P can be viewed as the public key of a Multi-HFE scheme with $N = 2$, and without the affine embedding π (see Section 4.5 for notation). To see this, suppose without loss of generality that $S = (s_1(x_1, \dots, x_n), \dots, s_n(x_1, \dots, x_n))$ is an invertible linear transformation. Consider the invertible linear transformation $S' : k^{2n} \rightarrow k^{2n}$ given by

$$S'(x_1, \dots, x_{2n}) = (s_1(x_1, \dots, x_n), \dots, s_n(x_1, \dots, x_n), x_{n+1}, \dots, x_{2n}).$$

Define $F_1, F_2 : K \times K \rightarrow K \times K$ by $F_1(X_1, X_2) = F(X_1)$ and $F_2(X_1, X_2) = \tilde{F}(X_1)$. The components of the trapdoor function P are the same components of the Multi-HFE instance

$$P'(x_1, \dots, x_{2n}) = T \circ (\varphi \times \varphi) \circ (F_1, F_2) \circ (\varphi^{-1} \times \varphi^{-1}) \circ S'(x_1, \dots, x_{2n}).$$

The main difference is that here the polynomials F_1 and F_2 have high total degree, while for the original Multi-HFE scheme it is not possible to use high total degree polynomials because the decryption process would be inefficient, if not impossible.

6.4 How to invert the trapdoor function

The crucial part to invert the trapdoor function P is the inversion of the core map $G = (F, \tilde{F})$, since the transformations S and T and the isomorphism φ are easy to invert. In what follows we explain how to invert G . Let F_0, \dots, F_{n-1} be the Frobenius powers of F

and let $\tilde{F}_0, \dots, \tilde{F}_{n-1}$ be the Frobenius powers of \tilde{F} . By the construction of F and \tilde{F} , there exist scalars $\alpha_1, \dots, \alpha_{2n}, \beta_1, \dots, \beta_{2n}$ such that the polynomial

$$\Psi = \sum_{j=1}^2 X^{q^{j-1}} \sum_{i=1}^n \alpha_{i+n(j-1)} F_{i-1} + \beta_{i+n(j-1)} \tilde{F}_{i-1}$$

has degree at most D_0 .

Proposition 6.2. *Let (Y_1, Y_2) be an element in $\text{Im } G \subseteq K \times K$. Then the set of preimages of (Y_1, Y_2) under the map $G = (F, \tilde{F})$ is a subset of the roots of the low degree polynomial*

$$\Psi' = \Psi - \sum_{j=1}^2 X^{q^{j-1}} \sum_{i=1}^n \alpha_{i+n(j-1)} Y_1^{q^{i-1}} + \beta_{i+n(j-1)} Y_2^{q^{i-1}}.$$

Proof. Let $X_0 \in K$ such that $G(X_0) = (Y_1, Y_2)$. Define $F' = F - Y_1$, $\tilde{F}' = \tilde{F} - Y_2$ and

$$g = \sum_{j=1}^2 X^{q^{j-1}} \sum_{i=1}^n \alpha_{i+n(j-1)} F'_{i-1} + \beta_{i+n(j-1)} \tilde{F}'_{i-1}.$$

First, notice that $F'(X_0) = 0$ and $\tilde{F}'(X_0) = 0$. Therefore $g(X_0) = 0$. Secondly, since the Frobenius powers are linear transformations, we have that $F'_i = (F - Y_1)^{q^i} = F^{q^i} - Y_1^{q^i} = F_i - Y_1^{q^i}$ and $\tilde{F}'_i = (\tilde{F} - Y_2)^{q^i} = \tilde{F}^{q^i} - Y_2^{q^i} = \tilde{F}_i - Y_2^{q^i}$. Thus

$$\begin{aligned} g &= \sum_{j=1}^2 X^{q^{j-1}} \sum_{i=1}^n \alpha_{i+n(j-1)} F'_{i-1} + \beta_{i+n(j-1)} \tilde{F}'_{i-1} \\ &= \sum_{j=1}^2 X^{q^{j-1}} \sum_{i=1}^n \alpha_{i+n(j-1)} \left(F_{i-1} - Y_1^{q^{i-1}} \right) + \beta_{i+n(j-1)} \left(\tilde{F}_{i-1} - Y_2^{q^{i-1}} \right) \\ &= \sum_{j=1}^2 X^{q^{j-1}} \sum_{i=1}^n \alpha_{i+n(j-1)} F_{i-1} + \beta_{i+n(j-1)} \tilde{F}_{i-1} \\ &\quad - \sum_{j=1}^2 X^{q^{j-1}} \sum_{i=1}^n \alpha_{i+n(j-1)} Y_1^{q^{i-1}} + \beta_{i+n(j-1)} Y_2^{q^{i-1}} \\ &= \Psi - \sum_{j=1}^2 X^{q^{j-1}} \sum_{i=1}^n \alpha_{i+n(j-1)} Y_1^{q^{i-1}} + \beta_{i+n(j-1)} Y_2^{q^{i-1}}. \end{aligned}$$

Therefore $g = \Psi'$ and thus we have that $\Psi'(X_0) = 0$. □

Notice that $\deg(\Psi') \leq \max(\deg(\Psi), q)$. Therefore the polynomial Ψ' , just like Ψ , has degree less than or equal to D_0 if we take $D_0 \geq q$. Thus we can efficiently find the roots of Ψ' using the algorithm \mathcal{A} .

Remark 6.3. *The formula for computing Ψ' in Proposition 6.2 does not involve the polynomials F and \tilde{F} , only the low degree polynomial Ψ . This is important because we want to create an encryption scheme where, for size reasons, F and \tilde{F} are not part of the private key.*

We now discuss the complexity of the trapdoor function inversion. The isomorphism φ and its inverse φ^{-1} can be represented in matrix form (Corollary 2.9). Thus, except for the inversion of the core map G , the computational cost of each step of the algorithm to invert the trapdoor function P is the cost of a matrix multiplication. The degrees of the polynomials F and \tilde{F} , which are the components of the map G , are extremely high (usually close to q^{n-1}), which makes impossible to invert G directly for practical values of n . However, as noted above, the inversion of the map G can be reduced to finding the roots of the low degree polynomial Ψ' . This can be done efficiently using the algorithm \mathcal{A} (usually Berlekamp's algorithm). Therefore, inverting the trapdoor function is a very efficient process. In the next chapter we will show experimental data to confirm this fact.

6.5 Toy example

We present here a toy example to explain step by step the reduction method described in Section 6.1.

Example 6.4.

For this toy example we take $q = 3$ and $n = 2$. We take $k = GF(3)$, and we select the irreducible polynomial $g(y) = y^2 + 2y + 2 \in k[y]$. A degree n extension field of k is

$K = k[y]/(g(y))$. We choose a generator $b \in K$ of the multiplicative group of K such that $g(b) = 0$. Two HFE polynomials in the ring $K[X]/(X^{3^2} - X)$ are of the form

$$\begin{aligned} F(X) &= a_{11}X^6 + a_{01}X^4 + a_{00}X^2 + b_1X^3 + b_0X + c, \\ \tilde{F}(X) &= \tilde{a}_{11}X^6 + \tilde{a}_{01}X^4 + \tilde{a}_{00}X^2 + \tilde{b}_1X^3 + \tilde{b}_0X + \tilde{c}, \end{aligned}$$

where $a_{11}, a_{01}, a_{00}, b_1, b_0, c, \tilde{a}_{11}, \tilde{a}_{01}, \tilde{a}_{00}, \tilde{b}_1, \tilde{b}_0, \tilde{c} \in K$.

The Frobenius powers of F and \tilde{F} , in that order, are:

$$\begin{aligned} F_0 &= a_{11}X^6 + a_{01}X^4 + a_{00}X^2 + b_1X^3 + b_0X + c, \\ F_1 &= a_{11}^3X^2 + a_{01}^3X^4 + a_{00}^3X^6 + b_1^3X + b_0^3X^3 + c^3, \end{aligned}$$

and

$$\begin{aligned} \tilde{F}_0 &= \tilde{a}_{11}X^6 + \tilde{a}_{01}X^4 + \tilde{a}_{00}X^2 + \tilde{b}_1X^3 + \tilde{b}_0X + \tilde{c}, \\ \tilde{F}_1 &= \tilde{a}_{11}^3X^2 + \tilde{a}_{01}^3X^4 + \tilde{a}_{00}^3X^6 + \tilde{b}_1^3X + \tilde{b}_0^3X^3 + \tilde{c}^3. \end{aligned}$$

We now multiply the Frobenius powers by X and X^3 and we obtain

$$\begin{aligned} XF_0 &= a_{11}X^7 + a_{01}X^5 + a_{00}X^3 + b_1X^4 + b_0X^2 + cX, \\ XF_1 &= a_{11}^3X^3 + a_{01}^3X^5 + a_{00}^3X^7 + b_1^3X^2 + b_0^3X^4 + c^3X, \\ X^3F_0 &= a_{11}X + a_{01}X^7 + a_{00}X^5 + b_1X^6 + b_0X^4 + cX^3, \\ X^3F_1 &= a_{11}^3X^5 + a_{01}^3X^7 + a_{00}^3X + b_1^3X^4 + b_0^3X^6 + c^3X^3, \end{aligned}$$

and

$$\begin{aligned} X\tilde{F}_0 &= \tilde{a}_{11}X^7 + \tilde{a}_{01}X^5 + \tilde{a}_{00}X^3 + \tilde{b}_1X^4 + \tilde{b}_0X^2 + \tilde{c}X, \\ X\tilde{F}_1 &= \tilde{a}_{11}^3X^3 + \tilde{a}_{01}^3X^5 + \tilde{a}_{00}^3X^7 + \tilde{b}_1^3X^2 + \tilde{b}_0^3X^4 + \tilde{c}^3X, \\ X^3\tilde{F}_0 &= \tilde{a}_{11}X + \tilde{a}_{01}X^7 + \tilde{a}_{00}X^5 + \tilde{b}_1X^6 + \tilde{b}_0X^4 + \tilde{c}X^3, \\ X^3\tilde{F}_1 &= \tilde{a}_{11}^3X^5 + \tilde{a}_{01}^3X^7 + \tilde{a}_{00}^3X + \tilde{b}_1^3X^4 + \tilde{b}_0^3X^6 + \tilde{c}^3X^3. \end{aligned}$$

Then we form the polynomial

$$\Psi = X \left(\alpha_1 F_0 + \alpha_2 F_1 + \beta_1 \tilde{F}_0 + \beta_2 \tilde{F}_1 \right) + X^3 \left(\alpha_3 F_0 + \alpha_4 F_1 + \beta_3 \tilde{F}_0 + \beta_4 \tilde{F}_1 \right).$$

In this example we are taking $D_0 = 3$, i.e, we want to determine the coefficients $a_{ij}, b_i, c, \tilde{a}_{ij}, \tilde{b}_i$ and \tilde{c} and also the scalars α_i, β_i such that the terms of degree > 3 in Ψ vanish. In order to do that, we have to solve the following four equations

$$\alpha_1 a_{11} + \alpha_2 a_{00}^3 + \alpha_3 a_{01} + \alpha_4 a_{01}^3 + \beta_1 \tilde{a}_{11} + \beta_2 \tilde{a}_{00}^3 + \beta_3 \tilde{a}_{01} + \beta_4 \tilde{a}_{01}^3 = 0,$$

$$\alpha_3 b_1 + \alpha_4 b_0^3 + \beta_3 \tilde{b}_1 + \beta_4 \tilde{b}_0^3 = 0,$$

$$\alpha_1 a_{01} + \alpha_2 a_{01}^3 + \alpha_3 a_{00} + \alpha_4 a_{11}^3 + \beta_1 \tilde{a}_{01} + \beta_2 \tilde{a}_{01}^3 + \beta_3 \tilde{a}_{00} + \beta_4 \tilde{a}_{11}^3 = 0,$$

$$\alpha_1 b_1 + \alpha_2 b_0^3 + \alpha_3 b_0 + \alpha_4 b_1^3 + \beta_1 \tilde{b}_1 + \beta_2 \tilde{b}_0^3 + \beta_3 \tilde{b}_0 + \beta_4 \tilde{b}_1^3 = 0.$$

Notice that the coefficients c and \tilde{c} do not appear in these equations. We randomly choose the scalars $(\alpha_1, \dots, \alpha_4) = (b, b^3, 2, b^5)$ and $(\beta_1, \dots, \beta_4) = (b^2, b^2, b^5, 2)$. Then we write the variables $a_{00}, a_{01}, a_{11}, b_0, b_1, \tilde{a}_{00}, \tilde{a}_{01}, \tilde{a}_{11}, \tilde{b}_0, \tilde{b}_1$ in terms of the basis $\{1, y, \dots, y^{n-1}\}$, as follows:

$$a_{00} = u_1 + u_2 y, \dots, \tilde{b}_1 = u_{19} + u_{20} y.$$

Proceeding as explained in Section 6.1, we get the linear equations

$$2u_8 + 2u_{10} + u_{17} + 2u_{18} = 0,$$

$$2u_7 + u_8 + u_{10} + u_{18} + u_{19} = 0,$$

$$2u_1 + u_3 + u_6 + 2u_{12} + 2u_{13} + u_{14} + 2u_{15} + 2u_{16} = 0,$$

$$2u_2 + 2u_5 + 2u_{11} + 2u_{12} + 2u_{13} + u_{14} + u_{16} = 0,$$

$$u_8 + 2u_9 + 2u_{17} + 2u_{18} + 2u_{20} = 0,$$

$$2u_7 + 2u_{10} + u_{18} + 2u_{19} + 2u_{20} = 0,$$

$$u_1 + 2u_2 + 2u_3 + u_4 + u_6 + u_{11} + 2u_{13} + u_{14} + u_{15} + u_{16} = 0,$$

$$2u_1 + 2u_2 + 2u_3 + 2u_4 + u_5 + u_6 + u_{11} + 2u_{12} + 2u_{13} + u_{15} + 2u_{16} = 0.$$

One solution of this system is

$$(u_1, \dots, u_{20}) = (0, 1, 1, 1, 1, 2, 2, 0, 0, 0, 2, 0, 0, 0, 2, 1, 1, 1, 1, 1).$$

This solution leads to the coefficients

$$(a_{00}, a_{01}, a_{11}, b_0, b_1, \tilde{a}_{00}, \tilde{a}_{01}, \tilde{a}_{11}, \tilde{b}_0, \tilde{b}_1) = (b, b^2, b^3, 2, 2, b^7, b^2, b^2).$$

With these coefficients we get the polynomials

$$F = b^3 X^6 + b^2 X^4 + b X^2 + 2X \quad \text{and} \quad \tilde{F} = b^7 X^6 + b^2 X^3 + 2X^2 + b^2 X.$$

We now use these polynomials, together the scalars α_i and β_i , to form the reduced polynomial $\Psi = b^5 X^3 + b^5 X^2 + b^2 X$.

6.6 Big examples

In this section we present two large scale examples of the reduction method described in Section 6.1. In Example 6.5, we also show the way to invert the map $G = (F, \tilde{F})$ as it was described in Section 6.4.

Example 6.5.

Let $q = 5$ and $n = 8$. We take $k = GF(5)$ and we select the irreducible polynomial $y^8 + y^4 + 3y^2 + 4y + 2 \in k[y]$. A degree n extension field of k is $K = k[y]/(g(y))$. We choose a generator $b \in K$ of the multiplicative group of K such that $g(b) = 0$. In this example we are taking $D_0 = 40$, i.e, we require that the terms of degree > 40 in Ψ vanish.

Following the same procedure explained in Example 6.4, we obtain the polynomials

$$\begin{aligned}
F(X) = & b^{193588} X^{156250} + b^{95686} X^{93750} + b^{284623} X^{81250} + b^{160172} X^{78750} + b^{313712} X^{78250} \\
& + b^{235602} X^{78150} + b^{221834} X^{78130} + b^{376167} X^{78126} + b^{275347} X^{78125} + b^{368321} X^{31250} \\
& + b^{102411} X^{18750} + b^{156633} X^{16250} + b^{388585} X^{15750} + b^{116832} X^{15650} + b^{371543} X^{15630} \\
& + b^{146359} X^{15626} + b^{286664} X^{15625} + b^{388903} X^{6250} + b^{77757} X^{3750} + b^{138809} X^{3250} \\
& + b^{169231} X^{3150} + b^{365819} X^{3130} + b^{281776} X^{3126} + b^{278761} X^{3125} + b^{125804} X^{1250} \\
& + b^{175902} X^{750} + b^{127779} X^{650} + b^{161150} X^{630} + b^{144579} X^{626} + b^{254441} X^{625} \\
& + b^{199075} X^{250} + b^{219766} X^{150} + b^{64647} X^{130} + b^{277102} X^{126} + b^{152760} X^{125} \\
& + b^{311781} X^{50} + b^{206113} X^{30} + b^{172953} X^{26} + b^{102994} X^{25} + b^{37496} X^{10} + b^{1608} X^6 \\
& + b^{250302} X^5 + b^{25721} X^2 + b^{259503} X,
\end{aligned}$$

$$\begin{aligned}
\tilde{F}(X) = & b^{127857} X^{156250} + b^{189925} X^{93750} + b^{290048} X^{81250} + b^{19423} X^{78750} + b^{132151} X^{78250} \\
& + b^{375166} X^{78150} + b^{100474} X^{78130} + b^{53348} X^{78126} + b^{30604} X^{78125} + b^{334818} X^{31250} \\
& + b^{70571} X^{18750} + b^{205616} X^{16250} + b^{355953} X^{15750} + b^{159663} X^{15650} + b^{374321} X^{15630} \\
& + b^{276328} X^{15626} + b^{374862} X^{15625} + b^{15376} X^{6250} + b^{351616} X^{3750} + b^{63371} X^{3250} \\
& + b^{99546} X^{3150} + b^{115611} X^{3130} + b^{259941} X^{3126} + b^{233678} X^{3125} + b^{28695} X^{1250} \\
& + b^{281966} X^{750} + b^{214174} X^{650} + b^{362722} X^{630} + b^{264275} X^{626} + b^{278500} X^{625} \\
& + b^{217548} X^{250} + b^{235648} X^{150} + b^{47958} X^{130} + b^{127955} X^{126} + b^{113479} X^{125} \\
& + b^{206907} X^{50} + b^{89670} X^{30} + b^{355802} X^{26} + b^{223357} X^{25} + b^{148589} X^{10} + b^{250488} X^6 \\
& + b^{89872} X^5 + b^{62097} X^2 + b^{252872} X.
\end{aligned}$$

The previous polynomials were constructed using the randomly chosen scalars

$$(\alpha_1, \dots, \alpha_{16}) = (b^{18997}, b^{269732}, b^{62323}, b^{292807}, b^{75948}, b^{205887}, b^{68718}, b^{277548}, \\ b^{242166}, b^{277138}, b^{124005}, b^{173370}, b^{335026}, b^{34310}, b^{231372}, b^{57464}),$$

and

$$(\beta_1, \dots, \beta_{16}) = (b^{305839}, b^{381782}, b^{97713}, b^{386643}, b^{217587}, b^{186365}, b^{375250}, b^{125793}, \\ b^{301982}, b^{228633}, b^{175403}, b^{153354}, b^{222133}, b^{101489}, b^{192773}, b^{281414}).$$

The scalars α_i and β_i , together with the polynomials F and \tilde{F} , lead to the reduced polynomial

$$\Psi = b^{171243} X^{35} + b^{268011} X^{31} + b^{15747} X^{30} + b^{277951} X^{27} + b^{92638} X^{26} + b^{60251} X^{15} + b^{215473} X^{11} \\ + b^{294576} X^{10} + b^{53530} X^7 + b^{183008} X^6 + b^{247124} X^3 + b^{218973} X^2.$$

The high degrees of F and \tilde{F} prevent us to invert $G = (F, \tilde{F})$ directly, but we can invert G using the low degree polynomial Ψ as explained in Section 6.4. To show how to invert G , we randomly choose $X_0 = b^{254095} \in K$. Then we calculate $(Y_1, Y_2) = G(X_0) = (b^{374629}, b^{234588})$. We now show how to recover X_0 from (Y_1, Y_2) . According to Proposition 6.2, X_0 is a root of the polynomial

$$\Psi' = \Psi - \sum_{j=1}^2 X^{q^j-1} \sum_{i=1}^n \alpha_{i+n(j-1)} Y_1^{q^i-1} + \beta_{i+n(j-1)} Y_2^{q^i-1} \\ = b^{171243} X^{35} + b^{268011} X^{31} + b^{15747} X^{30} + b^{277951} X^{27} + b^{92638} X^{26} + b^{60251} X^{15} + b^{215473} X^{11} \\ + b^{294576} X^{10} + b^{53530} X^7 + b^{183008} X^6 + b^{51964} X^5 + b^{247124} X^3 + b^{218973} X^2 + b^{126167} X.$$

The set of roots of the polynomial Ψ' , found very quickly by Berlekamp's algorithm, is $\{0, b^{254095}\}$. Notice that X_0 is one of these roots.

Example 6.6.

With $q = 11$ and $n = 25$, and taking $D_0 = 3000$, we found a pair of polynomials F and \tilde{F} with the same degree

$$D = 19699465351615222189423682 = 2q^{n-1},$$

which is the highest possible degree for these polynomials.

The reduced polynomial Ψ that we found has degree 2673. Hence, inverting the core map $G = (F, \tilde{F})$ is a simple task using Berlekamp's algorithm to find the roots of the polynomial Ψ' , which also has degree 2673. These polynomials are too big to be displayed here. In this example we needed to solve a sparse linear system of size 16275×17500 .

Chapter 7

New multivariate public key encryption schemes

In this chapter we propose an MPK cryptosystem based on the new trapdoor function constructed in Chapter 6. We give theoretical and experimental arguments to show that, for this cryptosystem, the encryption/decryption processes are very efficient. After performing the main known attacks that can threaten the security of these kind of schemes –the direct algebraic and the MinRank attacks–, we propose parameters for this new cryptosystem. We show the values of the main features of this cryptosystem for the suggested parameters.

One drawback of our MPK encryption scheme is the generation time of the private key. We have to deal with huge matrices to reach large values of n . On the plus side we have that these matrices are sparse, which is an advantage in terms of efficiency. Some data about the sparsity of these matrices and time generation of the private key appears in Appendix A. The expensive process of computing the private key is compensated with the high level of security obtained for the use of high degree core polynomials in this cryptosystem.

7.1 The encryption scheme

Let k be a finite field of size q . Fix a positive integer n and choose a degree n irreducible polynomial $g(y) \in k[y]$. Consider the field extension $K = k[y]/(g(y))$ and the isomorphism $\varphi: K \rightarrow k^n$ defined by $\varphi(u_1 + u_2y + \dots + u_ny^{n-1}) = (u_1, u_2, \dots, u_n)$. Let F , \tilde{F} and Ψ be three polynomials in $K[X]/(X^{q^n} - X)$ constructed using the method described in Section

6.1, i.e., F and \tilde{F} are two high degree HFE polynomials and Ψ is a low degree q -weight three polynomial which allows us to invert the map $G = (F, \tilde{F})$. Then we select two invertible affine transformations $S : k^n \rightarrow k^n$ and $T : k^{2n} \rightarrow k^{2n}$. The public key of the new encryption scheme is the multivariate trapdoor function

$$P(x_1, \dots, x_n) = T \circ (\varphi \times \varphi) \circ G \circ \varphi^{-1} \circ S(x_1, \dots, x_n).$$

Notice that P is a map from k^n to k^{2n} (see Figure 6.1). The private key consists of the low degree polynomial Ψ , the transformations S and T , and the scalars $\alpha_1, \dots, \alpha_{2n}, \beta_1, \dots, \beta_{2n}$. In summary we have for the new encryption scheme:

Public key. The public key of the new encryption scheme includes:

- The field k and its structure.
- The trapdoor function $P(x_1, \dots, x_n) = T \circ (\varphi \times \varphi) \circ G \circ \varphi^{-1} \circ S(x_1, \dots, x_n)$.

Private key. The private key of the new encryption scheme includes:

- The low degree polynomial Ψ .
- The two invertible affine transformations S and T .
- The scalars $\alpha_1, \dots, \alpha_{2n}, \beta_1, \dots, \beta_{2n}$.

Public key generation. The public key is generated by means of the reduction method from Section 6.1. In Section 6.2 we saw that this reduction method has complexity $\mathcal{O}((n^3)^\omega)$, where $2 \leq \omega \leq 3$.

Private key generation. The reduction method from Section 6.1 also gives us the low degree polynomial Ψ and the scalars $\alpha_1, \dots, \alpha_{2n}, \beta_1, \dots, \beta_{2n}$, which are part of the private key. Now we discuss the generation of the transformations S and T . The probability of

randomly choosing an invertible matrix from $\mathcal{M}_{n \times n}(k)$ is

$$\prod_{i=1}^n (1 - q^{i-1-n}),$$

which is big for large n . For example, if $q = 7$ and $n = 55$ this probability is approximately 0.83. Therefore we need very few attempts to generate the transformations S and T .

Encryption. To encrypt a plaintext $(x_1, \dots, x_n) \in k^n$ we simply plug this plaintext into the public key $P = T \circ (\varphi \times \varphi) \circ G \circ \varphi^{-1} \circ S$ to obtain the ciphertext

$$(y_1, \dots, y_{2n}) = P(x_1, \dots, x_n) \in k^{2n}.$$

Decryption. To recover the plaintext from the ciphertext we must invert each part of P . Similar to the decryption process in HFE, we perform the following steps:

- We first compute $(w_1, \dots, w_{2n}) = T^{-1}(y_1, \dots, y_{2n})$.
- Next we calculate $(Y_1, Y_2) = (\varphi^{-1}(w_1, \dots, w_n), \varphi^{-1}(w_{n+1}, \dots, w_{2n}))$.
- At this step we must invert the map $G = (F, \tilde{F})$, i.e., we have to solve the equation $G(X) = (Y_1, Y_2)$. The solutions of this equation are part of the roots of the low degree polynomial Ψ' , obtained from Ψ and (Y_1, Y_2) as in Proposition 6.2. Let \mathcal{Z} be the set

$$\mathcal{Z} = \{X \in K / \Psi'(X) = 0\}.$$

We must now determine which elements of \mathcal{Z} are solutions of the polynomial equation $G(X) = (Y_1, Y_2)$. In our extensive experiments we always got that only one element of \mathcal{Z} was a solution for this equation.

- For each solution $X \in \mathcal{Z}$ of the equation $G(X) = (Y_1, Y_2)$ we compute the vector $\varphi(X) \in k^n$.

- Finally, we apply the transformation S^{-1} to each vector found in the previous step and these vectors are the candidates to be the plaintext. To determine which of these is the original plaintext, some redundant information must be added to the plaintext¹.

7.2 Toy example

This example shows how the new encryption scheme works. Set $q = 3$ and $n = 3$, and consider the field with three elements $k = GF(3)$. We select the irreducible polynomial $g(y) = y^3 + 2y + 1 \in k[y]$. A degree n extension field of k is $K = k[y]/(g(y))$. We can choose a generator $b \in K$ of the multiplicative group of K such that $g(b) = 0$, and we use this element to write the elements of K as powers of it. Let us take $D_0 = 4$. We now randomly choose the scalars $(\alpha_1, \dots, \alpha_6) = (b^{14}, b^{23}, b^{20}, b^{20}, b^{22}, b^{14})$ and $(\beta_1, \dots, \beta_6) = (b^9, b^{16}, 2, b^3, b^6, b^{20})$. Then, as explained in Section 6.1, we construct the polynomials

$$F(X) = b^{24}X^{18} + b^9X^{12} + bX^{10} + b^3X^9 + b^{16}X^6 + b^7X^4 + b^{10}X^3 + b^{12}X^2 + b^{10}X,$$

$$\tilde{F}(X) = b^9X^{12} + b^{25}X^{10} + b^{17}X^9 + b^{22}X^6 + b^7X^4 + b^{20}X^3 + 2X^2 + b^{17}X,$$

$$\Psi(X) = b^8X^4 + b^6X^3 + b^4X^2.$$

We also select the invertible affine transformations

$$S(x_1, x_2, x_3) = \begin{pmatrix} 2 & 2 & 2 \\ 1 & 2 & 1 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} + \begin{pmatrix} 0 \\ 2 \\ 2 \end{pmatrix}$$

¹In all our extensive experiments, for each ciphertext there was only one candidate to be the plaintext.

and

$$T(x_1, x_2, x_3, x_4, x_5, x_6) = \begin{pmatrix} 1 & 2 & 0 & 1 & 2 & 2 \\ 2 & 2 & 1 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 2 & 1 \\ 0 & 2 & 0 & 0 & 2 & 2 \\ 1 & 0 & 2 & 2 & 0 & 0 \\ 2 & 1 & 0 & 1 & 2 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{pmatrix} + \begin{pmatrix} 2 \\ 1 \\ 1 \\ 1 \\ 2 \\ 1 \end{pmatrix}.$$

The core map is $G(X) = (F(X), \tilde{F}(X))$. The composition $P(x_1, x_2, x_3) = T \circ (\varphi \times \varphi) \circ G \circ \varphi^{-1} \circ S(x_1, x_2, x_3)$ yields the public key polynomials

$$p_1(x_1, x_2, x_3) = 2x_1^2 + x_1x_2 + x_1x_3 + 2x_2^2 + x_2x_3 + 2x_3^2 + x_3 + 1,$$

$$p_2(x_1, x_2, x_3) = x_1^2 + x_1x_2 + x_1 + 2x_2^2 + x_2x_3 + 2x_2 + 2x_3^2 + x_3 + 2,$$

$$p_3(x_1, x_2, x_3) = x_1x_3 + x_1 + x_2x_3 + x_2 + x_3^2 + x_3 + 1,$$

$$p_4(x_1, x_2, x_3) = 2x_1^2 + x_1x_2 + 2x_1 + 2x_2^2 + x_2x_3 + 2x_2 + x_3^2 + 2,$$

$$p_5(x_1, x_2, x_3) = 2x_1^2 + 2x_1 + 2x_2^2 + x_2x_3 + 1,$$

$$p_6(x_1, x_2, x_3) = 2x_1^2 + x_1x_2 + 2x_1x_3 + 2x_1 + x_2^2 + 2x_2 + 2x_3^2 + 2x_3 + 2.$$

We now illustrate the encryption/decryption processes. Let $(x_1, x_2, x_3) = (1, 1, 2)$ be a plaintext. After plugging this plaintext into the public key, we obtain the ciphertext

$$(y_1, y_2, y_3, y_4, y_5, y_6) = (2, 0, 1, 2, 0, 2).$$

In order to recover the plaintext from the ciphertext we first compute

$$(w_1, \dots, w_6) = T^{-1}(2, 0, 1, 2, 0, 2) = (0, 1, 0, 2, 2, 2).$$

We then calculate

$$\begin{aligned}
(Y_1, Y_2) &= (\varphi^{-1}(w_1, w_2, w_3), \varphi^{-1}(w_4, w_5, w_6)) \\
&= (\varphi^{-1}(0, 1, 0), \varphi^{-1}(2, 2, 2)) \\
&= (b, b^{19}).
\end{aligned}$$

As explained in Proposition 6.2, we now create the low degree polynomial Ψ' using the low degree polynomial Ψ , the scalars $\alpha_1, \dots, \alpha_6, \beta_1, \dots, \beta_6$ and the vector $(Y_1, Y_2) = (b, b^{19})$:

$$\Psi' = b^8 X^4 + b^{10} X^3 + b^4 X^2 + b^7 X.$$

The set of roots of Ψ' is²

$$\mathcal{Z} = \{0, b^8, b^{11}, b^{19}\}.$$

The only element of \mathcal{Z} which is solution of the equation $G(X) = (Y_1, Y_2) = (b, b^{19})$ is $X = b^{11}$. If we apply the isomorphism φ we get $\varphi(b^{11}) = (2, 1, 1)$. We next apply the transformation S^{-1} and then we recover the plaintext $S^{-1}(2, 1, 1) = (1, 1, 2)$.

The main part of the decryption process is the inversion of the map Ψ' . For this task we use Berlekamp's algorithm, which has complexity $\mathcal{O}(nD^2 \log_q D + D^3)$, where D is the degree of the univariate polynomial. Given the complexity of this algorithm it is expected that the degree of Ψ' , which is determined by the parameter D_0 , has the greatest impact on the decryption time. This fact was confirmed by our experiments. Table 7.1 shows some average encryption and decryption times for several choices of the parameters (q, n, D_0) . For each parameter choice we encrypted and decrypted 100 messages. To perform the experiments we used the software Magma V2.20-2 on an Intel Core i5-3210M CPU 2.50 GHz \times 4 with 12 GB of memory installed.

To finalize the description of the new cryptosystem, we suggest values for the parameters (q, n, D_0) for a realistic application of this encryption scheme. We base our choices on the

²These roots are found using the Magma implementation of Berlekamp's algorithm.

q	n	D_0	Average encryption time [s]	Maximum encryption time [s]	Average decryption time [s]	Maximum decryption time [s]
7	15	57	0.000	0.000	0.012	0.020
7	15	105	0.000	0.000	0.022	0.030
7	15	693	0.000	0.000	0.428	0.450
7	25	57	0.002	0.010	0.025	0.030
7	25	105	0.002	0.010	0.063	0.070
7	25	693	0.002	0.010	1.151	1.230
7	35	57	0.006	0.010	0.089	0.090
7	35	105	0.005	0.010	0.272	0.290
7	35	693	0.004	0.010	3.996	4.200
7	55	105	0.024	0.030	0.427	0.440
11	15	33	0.000	0.000	0.003	0.010
11	15	253	0.000	0.000	0.127	0.140
11	15	1463	0.000	0.000	1.954	2.120
11	25	33	0.002	0.010	0.016	0.020
11	25	253	0.001	0.010	0.337	0.360
11	25	1463	0.001	0.010	5.193	5.390
11	35	33	0.003	0.010	0.043	0.050
11	35	253	0.005	0.010	0.760	0.790
11	35	1463	0.004	0.010	12.366	13.180
17	15	51	0.000	0.000	0.019	0.020
17	15	323	0.000	0.000	0.267	0.280
17	15	595	0.001	0.010	0.711	0.760
17	25	51	0.001	0.010	0.106	0.120
17	25	323	0.002	0.010	0.984	1.070
17	25	595	0.000	0.000	2.070	2.100
17	35	51	0.009	0.010	0.223	0.240
17	35	323	0.006	0.010	2.039	2.110
17	35	595	0.008	0.010	4.367	4.420
17	55	51	0.028	0.030	0.763	0.810
17	55	595	0.028	0.030	12.742	12.880

Table 7.1: Encryption and decryption time for the new encryption scheme, 100 messages were tested per key.

data collected with the extensive experiments of this section and with the security analysis that we perform in Sections 7.3 and 7.4. We propose two sets of parameters:

$$q = 7$$

$$n = 55$$

$$D_0 = 105$$

Length of a message: 19.4 Bytes

Average encryption time: 0.024 seconds

Average decryption time: 0.427 seconds

Public key size: 65 KB

$$q = 17$$

$$n = 55$$

$$D_0 = 595$$

Length of a message: 28.2 Bytes

Average encryption time: 0.028 seconds

Average decryption time: 12.742 seconds

Public key size: 109 KB

7.3 Algebraic attack

Let us briefly review the algebraic attack (Section 4.3). Suppose that someone, who does not know the private trapdoor information, wants to invert the public key $P: k^n \rightarrow k^{2n}$ of the new encryption scheme ($P = (p_1, \dots, p_{2n})$). She wants to find the pre-images of an element $(y_1, \dots, y_{2n}) \in \text{Im } P \subseteq k^{2n}$. This person only has access to the public key P . In

order to accomplish this, she tries to solve the system of quadratic equations

$$\begin{aligned}
 & p_1(x_1, \dots, x_n) - y_1 = 0 \\
 & p_2(x_1, \dots, x_n) - y_2 = 0 \\
 & \quad \quad \quad \vdots \\
 & p_{2n}(x_1, \dots, x_n) - y_{2n} = 0.
 \end{aligned}
 \tag{7.1}$$

Solving the system 7.1 directly is known as the *direct algebraic attack*. One way to solve this system is finding a Gröbner basis for the ideal

$$I = (p_1 - y_1, \dots, p_{2n} - y_{2n}) \leq k[x_1, \dots, x_n].$$

The F_4 function of MAGMA, [34], is the most efficient implementation of the Gröbner basis F_4 algorithm that is currently available. We ran extensive experiments using the F_4 algorithm of MAGMA to perform the direct algebraic attack for several choices of the parameters (q, n, D_0) . For each choice of the parameters we used 10 different sets of quadratic equations to run the experiments.

As we commented in Section 4.3, for high characteristic the field equations do not help the process when computing a Gröbner basis. So, except for the case $q = 2$, in our experiments we did not append the field equations to the set of quadratic polynomials for which we wanted to find a Gröbner basis.

Our first experiments were performed with $q = 2$. In Table 7.2 and Figure 7.1 we can observe that the time needed to solve the equations coming from the public key of the new encryption scheme has an exponential growth in n . We can also see this behaviour with the memory used by the F_4 algorithm³. This situation is different from the one observed by Faugere and Joux in [25]. The difference lies on the fact that in [25] the quadratic equations are produced using a polynomial of fixed low degree as core map in the HFE

³All the computations for $q = 2$ were run using Magma V2.20-2 on a Sun X4440 server, with four Quad-Core AMD Opteron™ Processor 8356 CPUs and 128 GB of main memory (each CPU is running at 2.3 GHz).

n	Average time [s]	Minimum time [s]	Maximum time [s]	$\lceil \log_q D \rceil$
18	0.100	0.100	0.100	17
20	0.205	0.200	0.210	19
22	0.434	0.420	0.440	21
24	0.849	0.840	0.860	23
26	7.981	7.950	8.020	25
28	32.046	31.550	32.690	27
30	90.770	76.430	110.250	29
32	225.557	221.310	230.720	31

Table 7.2: Algebraic attack against the new cryptosystem for $q = 2$ and $D_0 = 386$.

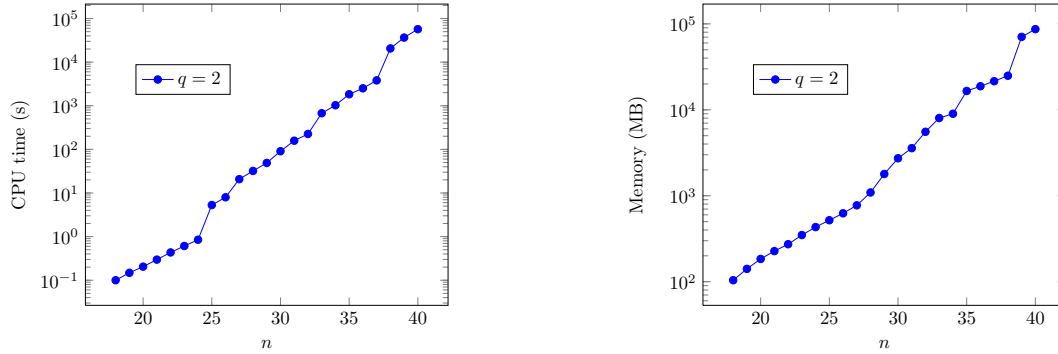


Figure 7.1: Algebraic attack against the new encryption scheme for $q = 2$ and $D_0 = 386$.

cryptosystem, and in our new cryptosystem the quadratic equations are generated via two high degree polynomials. In our experiments, in general, these two high degree polynomials have the same degree D and this degree increases as n increases (see Table 7.2). This is the fundamental security improvement of our new method.

Another evidence that the complexity of the algebraic attack against the new encryption scheme is exponential, is that the degree of regularity of the trapdoor function increases as n increases. This behaviour can be observed in Figure 7.2.

In order to compare with the MQ-problem, we chose systems of random quadratic equations of the same dimensions ($k^n \rightarrow k^{2n}$) and performed the algebraic attack against these systems too. For each system of random equations, we found that the time needed to solve such equations using Gröbner bases is essentially the same that the one needed to solve the quadratic equations from the public key of the new encryption scheme. These data are

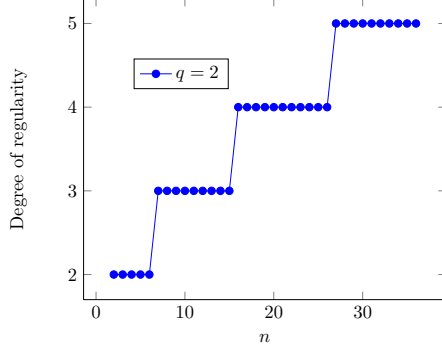


Figure 7.2: Algebraic attack for the new encryption scheme for $q = 2$ and $D_0 = 386$.

(a) New cryptosystem

(b) Random equations

n	Average time [s]	Memory [MB]	Degree of regularity	n	Average time [s]	Memory [MB]	Degree of regularity
14	0.019	3	3	14	0.040	12	3
16	0.142	5	4	16	0.060	13	4
18	0.100	8	4	18	0.100	16	4
20	0.205	13	4	20	0.200	21	4
22	0.434	20	4	22	0.440	31	4
24	0.849	33	4	24	0.830	46	4
26	7.981	118	4	26	7.800	105	4
28	32.046	1121	5	28	34.700	1087	5
30	90.770	2769	5	30	87.810	2725	5
32	225.557	5610	5	32	239.260	5549	5

Table 7.3: Algebraic attack comparison between the new encryption scheme and a system of random equations for $q = 2$ and $D_0 = 386$.

shown in Table 7.3 and Figure 7.3. Notice that the degree of regularity is the same in both cases.

According to all our experiments, it seems that for $q = 2$ the algebraic attack is no more efficient solving the equations from the new encryption scheme that solving a system of random quadratic equations of the same dimensions. However, if we wanted to use $q = 2$ for our new cryptosystem, we would have to consider values of n of at least 80 to avoid the exhaustive search. The estimated time needed to construct an example with $q = 2$ and $n = 80$ is about 30 days and the estimated memory needed is 230 GB. Table A.1 and Figure A.1, in Appendix A, show the time and memory needed to construct the private key of the new encryption scheme up to $n = 35$ for $q = 2$. Taking this issue into account, we decided

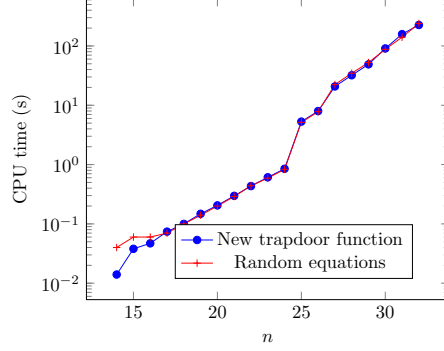


Figure 7.3: Algebraic attack comparison between the new encryption scheme and a system of random equations for $q = 2$ and $D_0 = 386$.

to abandon the case $q = 2$ and switch to odd characteristic.

For fixed n and D_0 , the size of the matrices needed in the reduction method for constructing the new encryption scheme do not depend on the parameter q (see Section 6.1 for the reduction method). For example, the matrices for constructing an encryption scheme with parameters $(q, n, D_0) = (2, 60, 1000)$ and $(q, n, D_0) = (17, 60, 1000)$ have the same size. The benefit that we get working with a high characteristic field is that we can achieve a higher level of security with a smaller value of n .

n	Average time [s]	Minimum time [s]	Maximum time [s]	Memory [MB]	$\lceil \log_q D \rceil$
12	0.071	0.06	0.09	32	11
13	0.136	0.13	0.15	32	12
14	0.289	0.28	0.31	32	13
15	0.785	0.76	0.85	32	14
16	5.564	5.5	5.64	64	15
17	13.658	13.26	15.03	96	16
18	31.392	31.01	32.19	128	17
19	70.301	69.61	71.14	192	18
20	148.208	143.69	160.73	288	19
21	307.118	303.22	316.41	448	20
22	942.269	663.62	988.45	681	21
23	2058.543	2050.51	2064	1107	22
24	18114.05	18099.43	18128.67	8334	23

Table 7.4: Algebraic attack against the new encryption scheme for $q = 7$ and $D_0 = 105$.

For odd characteristic we carry out similar experiments to those performed in the case

$q = 2$. Table 7.4⁴ and Figure 7.4⁴ contain these results for $q = 7$ and $D_0 = 105$, and several values of n . We observe an exponential growth in both time and memory as n increases. But more importantly, Figure 7.5⁵ shows that the degree of regularity increases as n increases, which confirms the exponential growth of the time. The values of D_0 that we use are chosen in such a way that the low degree polynomial Ψ does not have too few terms. The reason to do this is that we do not want to have a very simple function Ψ that might introduce somehow a security weakness to the cryptosystem. For $(q, D_0) = (7, 105)$ and $(q, D_0) = (17, 595)$ the polynomial Ψ has 14 terms.

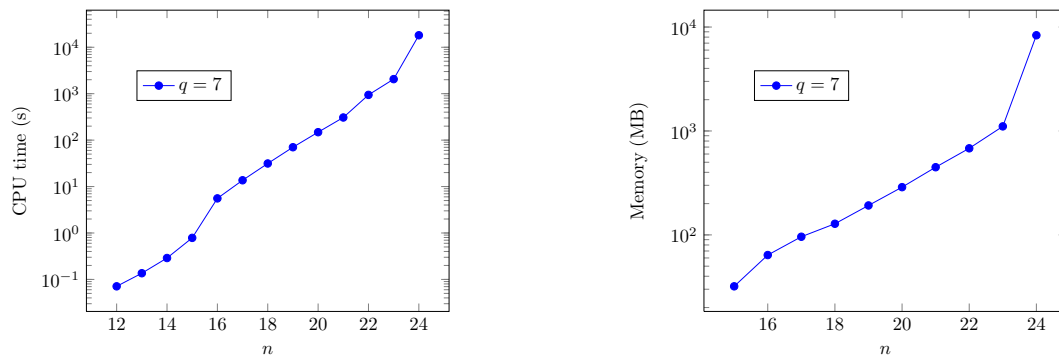


Figure 7.4: Algebraic attack against the new encryption scheme for $q = 7$ and $D_0 = 105$.

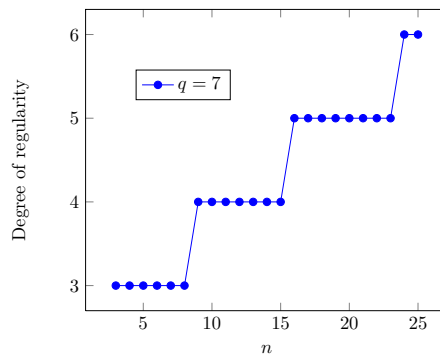


Figure 7.5: Algebraic attack for the new encryption scheme for $q = 7$ and $D_0 = 105$.

In order to compare with the MQ-problem, we also chose systems of random quadratic

⁴For these computations we used the software Magma V2.20-2 on an Intel Core i5-3210M CPU 2.50 GHz \times 4 with 12 GB of memory installed.

⁵For these computations we used the software Magma V2.20-2 on a Sun X4440 server, with four Quad-Core AMD Opteron™ Processor 8356 CPUs and 128 GB of main memory (each CPU is running at 2.3 GHz).

(a) New encryption scheme				(b) Random equations			
n	Average time [s]	Memory [MB]	Degree of regularity	n	Average time [s]	Memory [MB]	Degree of regularity
12	0.071	32	4	12	0.07	32	4
13	0.136	32	4	13	0.13	32	4
14	0.289	32	4	14	0.28	32	4
15	0.785	32	4	15	0.79	32	4
16	5.564	64	5	16	5.6	64	5
17	13.658	96	5	17	13.74	96	5
18	31.392	128	5	18	32.19	128	5
19	70.301	192	5	19	70.94	192	5
20	148.208	288	5	20	144.09	288	5
21	307.118	448	5	21	310.52	448	5
22	942.269	681	5	22	991.72	681	5
23	2058.543	1107	5	23	1923.99	1107	5
24	18114.05	8334	6	24	18012.19	8334	6

Table 7.5: Algebraic attack comparison between the new encryption scheme and a system of random equations for $q = 7$ and $D_0 = 105$.

equations of the same dimensions ($k^n \rightarrow k^{2n}$) and performed the algebraic attack against them. For each system we found that the time needed to solve those quadratic equations is essentially the same that the one needed to solve the quadratic equations from the new encryption scheme. Table 7.5⁶ and Figure 7.6⁶ show this comparison between the public key of the new encryption scheme and a system of random equations for $q = 7$ and different values of n . We can observe that the time required by the algebraic attack, in both cases, grows exponentially as n increases. More importantly, we can see that the degree of regularity is the same for both systems, and that it increases as n increases. Again, as it was pointed out for $q = 2$, we can see that for $q = 7$ our new cryptosystem behaves as if it were a random system of quadratic equations with respect to the direct algebraic attack.

We repeated for $q = 17$ all the experiments performed in this section for $q = 2$ and $q = 7$. Figure 7.7⁷ and Table 7.6⁸ show an exponential growth of the time and memory required by the F_4 algorithm to perform the algebraic attack. The degree of regularity for $q = 17$ is

⁶For these computations we used the software Magma V2.20-2 on an Intel Core i5-3210M CPU 2.50 GHz \times 4 with 12 GB of memory installed.

⁷For these computations we used the software Magma V2.20-2 on a Sun X4440 server, with four Quad-Core AMD OpteronTM Processor 8356 CPUs and 128 GB of main memory (each CPU is running at 2.3 GHz).

⁸For these computations we used the software Magma V2.20-2 on an Intel Core i5-3210M CPU 2.50 GHz \times 4 with 12 GB of memory installed.

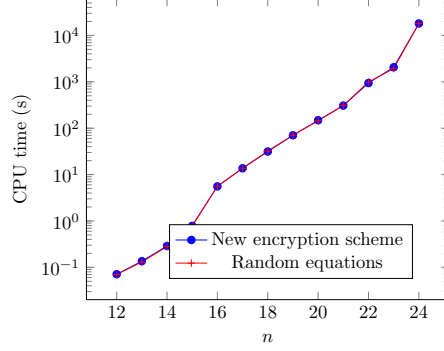


Figure 7.6: Algebraic attack comparison between the new encryption scheme and a system of random equations for $q = 7$ and $D_0 = 105$.

n	Average time [s]	Minimum time [s]	Maximum time [s]	Memory [MB]	$\lceil \log_q D \rceil$
12	0.067	0.06	0.07	12	11
13	0.13	0.12	0.14	12	12
14	0.281	0.27	0.29	14	13
15	0.763	0.76	0.77	17	14
16	5.65	5.56	5.77	42	15
17	13.726	13.52	13.96	68	16
18	31.874	31.75	32.06	111	17
19	71.751	71.51	72.27	179	18
20	148.29	146.45	152.55	286	19
21	317.488	313.83	321.39	460	20
22	982.01	686.95	1018.91	772	21
23	1975.074	1964.34	2000.97	1197	22
24	18793.792	18689.94	18956.88	8627	23

Table 7.6: Algebraic attack against the new encryption scheme for $q = 17$ and $D_0 = 595$.

shown in Figure 7.8⁸. We notice that these degrees of regularity are exactly the same as the ones observed for $q = 7$ (see Figure 7.5). Once more, we can see for $q = 17$ how the new cryptosystem behaves almost the same as a system of random quadratic equations, with respect to the algebraic attack. This fact can be observed in Table 7.7⁸ and Figure 7.9⁸.

We also studied the impact of the parameter D_0 in our experiments and we observed that reducing the values of this parameter do not affect the security of the new encryption scheme with respect to the algebraic attack. So, it seems that we can take D_0 as small as the reduction method allows us. This behaviour can be observed in Table 7.8⁹.

⁹For these computations we used the software Magma V2.20-2 on an Intel Core i5-3210M CPU 2.50 GHz \times 4 with 12 GB of memory installed.

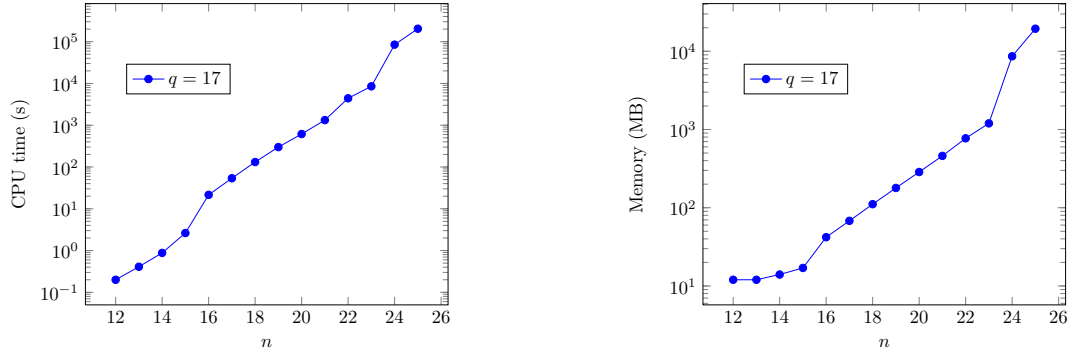


Figure 7.7: Algebraic attack against the new encryption scheme for $q = 17$ and $D_0 = 595$.

(a) New encryption scheme

n	Average time [s]	Memory [MB]	Degree of regularity
12	0.067	12	4
13	0.13	12	4
14	0.281	14	4
15	0.763	17	4
16	5.65	42	5
17	13.726	68	5
18	31.874	111	5
19	71.751	179	5
20	148.29	286	5
21	317.488	460	5
22	982.01	772	5
23	1975.074	1197	5
24	18793.792	8627	6

(b) Random equations

n	Average time [s]	Memory [MB]	Degree of regularity
12	0.069	12	4
13	0.129	13	4
14	0.291	14	4
15	0.811	17	4
16	5.73	42	5
17	13.615	68	5
18	32.046	111	5
19	73.084	179	5
20	147.597	286	5
21	317.329	460	5
22	1011.964	774	5
23	2036.291	1204	5
24	18914.614	8679	6

Table 7.7: Algebraic attack comparison between the new encryption scheme and a system of random quadratic equations for $q = 17$ and $D_0 = 595$.

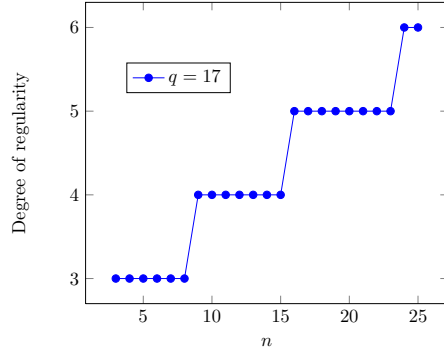


Figure 7.8: Algebraic attack for the new encryption scheme for $q = 17$ and $D_0 = 595$.

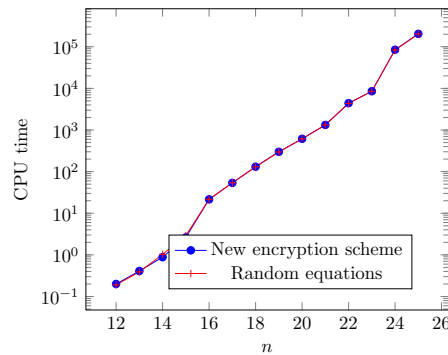


Figure 7.9: Algebraic attack comparison between the new encryption scheme and a system of random quadratic equations for $q = 17$ and $D_0 = 595$.

Based on all the information that we collected with our experiments, it seems that the algebraic attack is no more efficient in solving the equations coming from the public key of the new encryption scheme than a set of random quadratic equations of the same dimensions. In other words, with respect to the direct algebraic attack, the public key of the new encryption scheme behaves as if it were a system of random quadratic equations.

Finally, in order to suggest practical values for the parameters (q, n, D_0) , we want to establish a relationship between the time needed by the F_4 algorithm and the parameter n . To accomplish this, we use the time consumption that we reported in Table 7.4 and Table 7.6, for $q = 7$ and $q = 17$, respectively. We extrapolate these data using the least squares algorithm, and using the speed of our computer processor as a conversion factor, we estimate the number of operations that the algebraic attack will require for a large n . In both cases,

q	D_0	n	Average time [s]	Minimum time [s]	Maximum time [s]	Memory [MB]
7	21	12	0.070	0.070	0.070	64
7	21	14	0.280	0.280	0.280	96
7	21	16	5.620	5.620	5.620	128
7	21	18	31.580	31.580	31.580	224
7	21	20	144.350	144.350	144.350	352
7	21	22	656.540	656.540	656.540	681
7	105	12	0.071	0.06	0.09	64
7	105	14	0.289	0.28	0.31	96
7	105	16	5.564	5.5	5.64	128
7	105	18	31.392	31.01	32.19	224
7	105	20	148.208	143.69	160.73	352
7	105	22	942.269	663.62	988.45	672
17	51	12	0.07	0.07	0.07	64
17	51	14	0.28	0.28	0.28	96
17	51	16	5.65	5.65	5.65	160
17	51	18	32.08	32.08	32.08	256
17	51	20	147.98	147.98	147.98	416
17	51	22	1058.86	1058.86	1058.86	777
17	595	12	0.067	0.06	0.07	64
17	595	14	0.281	0.27	0.29	96
17	595	16	5.65	5.56	5.77	160
17	595	18	31.874	31.75	32.06	256
17	595	20	148.29	146.45	152.55	416
17	595	22	982.01	686.95	1018.91	777
23	69	12	0.069	0.06	0.08	64
23	69	14	0.276	0.27	0.28	96
23	69	16	5.573	5.55	5.63	160
23	69	18	31.969	31.87	32.23	256
23	69	20	148.192	147.02	153.18	384
23	69	22	989.413	682.47	1057.46	809

Table 7.8: Algebraic attack for the new encryption scheme for several choices of (q, n, D_0) .

$q = 7$ and $q = 17$, we conclude that for $n = 55$ the algebraic attack will need more than 2^{80} operations to be successful. This analysis yields the parameters that we suggested in Section 7.1.

7.4 Kipnis-Shamir MinRank attack

The KS MinRank attack or KS attack was discussed in Section 4.4. Although we are using high degree and high rank polynomials as core maps, this attack could work if there was a low rank linear combination of their Frobenius powers. Because of this, we have to carefully consider this attack for our new cryptosystem.

We now test our new encryption scheme against the KS attack, by performing extensive computer experiments for the case of odd characteristic. For characteristic 2 the attack is slightly different, and for the special case of $q = 2$ we give a theoretical argument (without applying the attack directly) to demonstrate why the KS MinRank attack does not work against the new encryption scheme. All the computations of this section were run using Magma V2.20-2 on a Sun X4440 server, with four Quad-Core AMD Opteron™ Processor 8356 CPUs and 128 GB of main memory (each CPU is running at 2.3 GHz).

The main part of the KS attack, with respect to the complexity, is to solve the MinRank problem. The original version of the KS attack was not as efficient as its authors claimed, because the derived MinRank problem worked with matrices with entries in the big field K . Recently, Faugère et al. [4] improved and generalized the KS attack, and were able to break HFE and its generalization Multi-HFE for all practical choices of their parameters. Their main improvement was to restate the MinRank problem with the matrices associated to the public key, whose entries are in the small field k . This makes the improved KS attack significantly faster than the original version.

In Section 6.3 we noted that the new trapdoor function P , which is the public key of the our new encryption scheme, can be seen as a particular case of an *unbounded* Multi-HFE cryptosystem, with $N = 2$ (for unbounded we mean that the core polynomials have no restrictions for their degrees). Because of this, in this section we perform the KS attack as it was done in [4] for a Multi-HFE scheme. For given parameters q , n and D_0 , we generate the $2n$ public key polynomials p_1, \dots, p_{2n} of the new encryption scheme ($P = (p_1, \dots, p_{2n})$). Then, we compute the symmetric matrix M_i associated to the quadratic part of each public key polynomial p_i , $i = 1, \dots, 2n$. Let $\text{Q-Rank}(P)$ be the minimal rank of elements in the K -linear space generated by the matrices M_1, \dots, M_{2n} . In [4] they showed that $\text{Q-Rank}(P)$ coincides with the minimal quadratic rank of elements in the K -linear space generated by

the Frobenius powers of the core polynomials F and \tilde{F} . The KS attack is successful against Multi-HFE when $\text{Q-Rank}(P)$ is low (see [4]). The main purpose of this section is to show that $\text{Q-Rank}(P)$ increases as n increases for the new encryption scheme, and therefore the KS attack will not work against this new cryptosystem.

MinRank problem: Fix a positive integer $r < n$ (start with $r = 1$) and try to find scalars $\lambda_1, \dots, \lambda_{2n} \in K$, not all zero, such that

$$\text{Rank} \left(\sum_{i=1}^{2n} \lambda_i M_i \right) \leq r.$$

If there is no solution, set $r = r + 1$ and repeat this step until a solution is found.

As it was proved in [4] for a Multi-HFE scheme, in order to accelerate the MinRank problem, we can randomly fix $N = 2$ of the scalars $\lambda_1, \dots, \lambda_{2n} \in K$, not all to zero. In our experiments we fixed $\lambda_{n-1} = 0$ and $\lambda_n = 1$, and we used the Kipnis-Shamir modelling for solving this MinRank problem. The reason to choose this modelling is that the minors modelling uses considerably more memory than the KS option.

Before we continue with this attack, let us illustrate a way to solve the MinRank problem with a toy example.

Example 7.1.

In this toy example we chose the parameters $(q, n, D_0) = (7, 4, 110)$. Let $k = GF(q)$, and consider the irreducible polynomial $y^4 + 5y^2 + 4y + 3 \in k[y]$. A degree n extension field of k is $K = k[y]/(g(y))$. After generating the core polynomials F and \tilde{F} , using the method

described in Section 6.1, we get the public key polynomials

$$p_1(x_1, \dots, x_4) = x_1^2 + 4x_1x_2 + 2x_1x_3 + 3x_1x_4 + 2x_2^2 + x_2x_3 + 6x_2x_4 + 4x_3^2 + 5x_3x_4 + x_4^2,$$

$$p_2(x_1, \dots, x_4) = 3x_1^2 + 3x_1x_3 + 4x_2^2 + 2x_2x_3 + 4x_2x_4 + 2x_3^2 + 4x_3x_4 + 4x_4^2,$$

$$p_3(x_1, \dots, x_4) = 5x_1x_2 + 4x_1x_3 + 2x_1x_4 + x_2^2 + 6x_2x_3 + 2x_2x_4 + 5x_3^2 + 2x_3x_4 + x_4^2,$$

$$p_4(x_1, \dots, x_4) = 5x_1x_2 + 5x_1x_3 + 4x_1x_4 + x_2^2 + 6x_2x_3 + 6x_2x_4 + 6x_3^2 + 2x_3x_4 + 3x_4^2,$$

$$p_5(x_1, \dots, x_4) = 6x_1x_2 + 2x_1x_3 + 2x_1x_4 + x_2^2 + 3x_2x_3 + 6x_2x_4 + 2x_3^2 + x_3x_4 + 2x_4^2,$$

$$p_6(x_1, \dots, x_4) = 3x_1x_3 + 5x_1x_4 + 6x_2^2 + 3x_2x_3 + x_3^2 + 5x_3x_4 + 5x_4^2,$$

$$p_7(x_1, \dots, x_4) = 3x_1^2 + 3x_1x_2 + 6x_1x_3 + x_1x_4 + x_2^2 + 4x_2x_3 + 3x_2x_4 + x_3^2 + 3x_3x_4,$$

$$p_8(x_1, \dots, x_4) = 5x_1^2 + 2x_1x_3 + 5x_2^2 + 2x_2x_3 + 5x_2x_4 + 4x_3^2 + 6x_3x_4 + 3x_4^2.$$

The symmetric matrices M_1, \dots, M_8 associated to the quadratic parts of the polynomials

p_1, \dots, p_8 , respectively, are:

$$\begin{pmatrix} 1 & 2 & 1 & 5 \\ 2 & 2 & 4 & 3 \\ 1 & 4 & 4 & 6 \\ 5 & 3 & 6 & 1 \end{pmatrix}, \begin{pmatrix} 3 & 0 & 5 & 0 \\ 0 & 4 & 1 & 2 \\ 5 & 1 & 2 & 2 \\ 0 & 2 & 2 & 4 \end{pmatrix}, \begin{pmatrix} 0 & 6 & 2 & 1 \\ 6 & 1 & 3 & 1 \\ 2 & 3 & 5 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}, \begin{pmatrix} 0 & 6 & 6 & 2 \\ 6 & 1 & 3 & 3 \\ 6 & 3 & 6 & 1 \\ 2 & 3 & 1 & 3 \end{pmatrix}, \\ \begin{pmatrix} 0 & 3 & 1 & 1 \\ 3 & 1 & 5 & 3 \\ 1 & 5 & 2 & 4 \\ 1 & 3 & 4 & 2 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 5 & 6 \\ 0 & 6 & 5 & 0 \\ 5 & 5 & 1 & 6 \\ 6 & 0 & 6 & 5 \end{pmatrix}, \begin{pmatrix} 3 & 5 & 3 & 4 \\ 5 & 1 & 2 & 5 \\ 3 & 2 & 1 & 5 \\ 4 & 5 & 5 & 0 \end{pmatrix}, \begin{pmatrix} 5 & 0 & 1 & 0 \\ 0 & 5 & 1 & 6 \\ 1 & 1 & 4 & 3 \\ 0 & 6 & 3 & 3 \end{pmatrix}.$$

In this example we want to find scalars $\lambda_1, \dots, \lambda_8$ such that the matrix $M = \lambda_1 M_1 + \dots + \lambda_8 M_8$ has rank one, i.e, we are taking $r = 1$. We now fix $\lambda_7 = 0$ and $\lambda_8 = 1$, and then apply the Kipnis-Shamir modelling to get the solution $(\lambda_1, \dots, \lambda_6) = (3, 1, 0, 3, 4, 6)$ for this

MinRank problem. The matrix M and its reduced echelon form U are

$$M = \begin{pmatrix} 4 & 1 & 5 & 5 \\ 1 & 2 & 3 & 3 \\ 5 & 3 & 1 & 1 \\ 5 & 3 & 1 & 1 \end{pmatrix}, \quad U = \begin{pmatrix} 1 & 2 & 3 & 3 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$

In the previous toy example we were able to solve the MinRank problem coming from the public key of the new encryption scheme, and determined that $\text{Q-Rank}(P) = 1$. In this toy case we could use the remaining steps of the KS attack to recover the private key (this is possible because $\text{Q-Rank}(P) = 1$ is too small). However, for larger choices of n the situation is very different, as we will see in the rest of this section.

To continue with the KS attack, we now use the MinRank problem to determine the $\text{Q-Rank}(P)$ for different combinations of the parameters (q, n, D_0) . For each n we start by taking $r = 1$ and then use the KS modelling. We utilize the Magma implementation of the F_4 algorithm to solve the equations produced by this modelling. Table 7.9 shows the results obtained for $q = 7$ and $D_0 = 105^{10}$. If for $r = 1$ the solution set of the MinRank problem is empty, then we set $r = r + 1$ and repeat this process until a solution is found. For example, in Table 7.9 the expression “ > 3 ” means that for $r \in \{1, 2, 3\}$ the solution set obtained for the MinRank problem was empty, so $\text{Q-Rank}(P) > 3$ for that case.

n	$\text{Q-Rank}(P)$	Average time	Maximum memory
2	1	0.010 s	32
4	1	0.010 s	32
6	2	1.340 s	32
8	> 3	> 10 days	> 50 GB ¹⁰
10	> 3	> 10 days	> 50 GB ¹⁰

Table 7.9: KS attack against the new encryption scheme, for $q = 7$ and $D_0 = 105$.

¹⁰The instances $n = 8$ and $n = 10$ for $r = 4$ did not terminate since the process had a 50 GB memory limitation. After reaching this limit the process automatically stopped after more than 10 days of running time.

In Table 7.10 we show the time and memory needed to find the solution set for the MinRank problem for $(q, n, D_0) = (7, 8, 110)$ and different values of r . The same situation is observed for other combinations of the parameters. We can see how fast those values increase as r increases. The results in Tables 7.9 and 7.10 lead us to think that the larger $\text{Q-Rank}(P)$ is the less feasible the MinRank problem is.

r	Average Time	Maximum memory
1	0.040 s	32
2	0.510 s	32 MB
3	297.410 s	462 MB
4	> 10 days	> 50 GB ¹⁰

Table 7.10: Time and memory needed to find the solution set for the KS attack against the new encryption scheme, for $q = 7$, $n = 8$ and $D_0 = 105$.

Now, for a fixed pair (q, n) we randomly choose a set of $2n$ quadratic equations in n variables, and perform the same process that we just used with the new encryption scheme, in order to compare with the results that we obtained for such a cryptosystem. The results are summarized in Table 7.11. We notice that we get exactly the same results for both cases. We also see that, for the new cryptosystem, the value of $\text{Q-Rank}(P)$ is independent of the value of D_0 . According to our experiments and the fact that we are using high rank core polynomials to construct the public key, we believe that our new cryptosystem behaves as if it were a set of random equations with respect to the KS attack.

(a) New encryption scheme					(b) Random equations	
n	$D_0 = 105$	$D_0 = 399$	$D_0 = 2751$	$D_0 = 4809$	n	$\text{Q-Rank}(P)$
2	1	1	1	1	2	1
4	1	1	1	1	4	1
6	2	2	2	2	6	2
8	> 3	> 3	> 3	> 3	8	> 3
10	> 3	> 3	> 3	> 3	10	> 3

Table 7.11: $\text{Q-Rank}(P)$ comparison between the new trapdoor function and random equations for $q = 7$.

Another interesting experiment is to compare the effect of the KS attack against the new cryptosystem with the effect of that attack against a system built in a similar way, but with

low rank core polynomials F and \tilde{F} , i.e., a standard (bounded) Multi-HFE scheme. Table 7.12 shows these results for $q = 7$ and several values of n . We can observe that for the standard Multi-HFE the KS MinRank attack succeeds, while for the new encryption scheme (Table 7.9) it does not. According to Tables 7.9, 7.11 and 7.12, we think that the quadratic rank $\text{Q-Rank}(P)$ grows as n grows.

n	$\text{Q-Rank}(P)$	Average time [s]	Maximum Memory [MB]
2	1	0.050	32
4	1	0.100	32
6	2	1.135	32
8	2	1.190	32
10	2	6.090	32
12	2	23.080	64
14	2	67.500	138
16	2	192.850	211
18	2	479.150	363
20	2	885.720	711

Table 7.12: KS attack against a bounded Multi-HFE scheme for $q = 7$ and $\lfloor \log_q D \rfloor = 2$.

Using the same procedure, we can construct similar tables for other values of q . In the Appendix we can also find the case $q = 17$. Based on all the information gathered from our extensive experiments, we believe that the KS MinRank attack does not work against the new encryption scheme in the case of odd characteristic.

For the case $q = 2$ we give here theoretical arguments to show why the KS attack does not work against the new encryption scheme. We would like to recall the recent result on the degree of regularity of Ding and Hodges [17]. We know that for an HFE system P the degree of regularity is bounded by

$$\frac{(q-1) \text{Q-Rank}(P)}{2} + 2,$$

where $\text{Q-Rank}(P)$ is the quadratic rank for the quadratic operator P . So for $q = 2$ we have that this degree of regularity is bounded by

$$\frac{\text{Q-Rank}(P)}{2} + 2.$$

Since the corresponding quadratic rank used in the Kipnis-Shamir MinRank attack is also given by $\text{Q-Rank}(P)$, we can see that if an HFE system has a high degree of regularity when $q = 2$, this HFE system must have a high quadratic rank for the Kipnis-Shamir attack as well. From this we conclude that, for $q = 2$, it suffices to show that our new encryption scheme has high degree of regularity, in order to demonstrate that the KS MinRank attack will not work against this new cryptosystem. In Section 7.3 we gave evidence that the algebraic attack does not work against the new encryption scheme, even in the case $q = 2$. Thus, for $q = 2$ we can conclude that our new cryptosystem is also resistant to the KS MinRank attack.

Chapter 8

Conclusions and Future Work

In this thesis we have created a procedure to build new candidates for multivariate trapdoor functions using pairs of HFE polynomials of high degree. The way to invert these trapdoor functions is through a low degree polynomial of Hamming weight three.

Using this trapdoor function we constructed a new multivariate public key encryption scheme. Until now, no one had proposed any idea of how to use high degree polynomials for the core map in HFE or any of its variants, since there always was the problem of the inversion of such core polynomials. Our novel idea has allowed us to invert a map built with two high degree polynomials by means of a third polynomial with low degree.

We showed that the encryption/decryption processes for this cryptosystem are very efficient. Moreover, we showed that the attacks that have threatened the security of HFE, the direct algebraic and the Kipnis-Shamir MinRank attacks, do not work against our new encryption scheme.

We performed numerous computer experiments to test the security and measure the encryption/decryption times for several sets of parameters of our new encryption scheme. The data we collected guided our choices for the parameters (q , n and D_0) for plausible schemes.

In the future we want to study ways of speeding up the reduction method to construct the

trapdoor functions. Speeding up the reduction method will allow us to reach larger values of n and therefore we will be able to implement plausible schemes with smaller values of q , for example $q = 2$. We also want to study the effect that the matrix sparsity has on the complexity of the algorithm used to construct the private key of the new encryption scheme.

We would also like to study the effect of the parameter D_0 on the security of the new encryption scheme. D_0 cannot be too large because that would affect the decryption speed. On the other hand, we think that D_0 cannot be too low either, since that would produce a polynomial Ψ with very few terms and that might be exploited by an attacker. Further study of this point must be done to prevent possible attacks.

We do not think that the one presented here is the unique way to reduce high degree HFE polynomials with the aim of creating an encryption scheme. Our new cryptosystem is only a first step in this direction. In fact, the method described here was not our first attempt to reduce high degree HFE polynomials along the same line. Among failed attempts, we considered using a single polynomial F , but the linear systems we needed to solve had more equations than variables and then we could not guarantee nontrivial solutions for them. So, we would also like to study new ways to reduce high degree HFE polynomials that could be more efficient.

Bibliography

Bibliography

- [1] Gwéno le Ars, Jean-Charles Faug re, Hideki Imai, Mitsuru Kawazoe, and Makoto Sugita. Comparison between xl and Gr bner basis algorithms. In PilJoong Lee, editor, *Advances in Cryptology - ASIACRYPT 2004*, volume 3329 of *Lecture Notes in Computer Science*, pages 338–353. Springer Berlin Heidelberg, 2004.
- [2] Magali Bardet, Jean charles Faug re, and Bruno Salvy. On the complexity of Gr bner basis computation of semi-regular overdetermined algebraic equations, 2004.
- [3] Daniel J. Bernstein, Johannes Buchmann, and Erik Dahmen. *Post quantum cryptography*. Springer, 2009.
- [4] Luk Bettale, Jean-Charles Faug re, and Ludovic Perret. Cryptanalysis of hfe, multi-hfe and variants for odd and even characteristic. *Designs, Codes and Cryptography*, 69(1):1–52, 2013.
- [5] Charles Bouillaguet, Pierre-Alain Fouque, Antoine Joux, and Joana Treger. A family of weak keys in hfe (and the corresponding practical key-recovery). Cryptology ePrint Archive, Report 2009/619, 2009.
- [6] B. Buchberger. Ein algorithmus zum auffinden der basiselemente des restklassenringes nach einem nulldimensionalen polynomideal. Master’s thesis, PhD thesis, University of Innsbruck, Austria, 1965.
- [7] JonathanF. Buss, GudmundS. Frandsen, and JeffreyO. Shallit. The computational complexity of some problems of linear algebra. In Rdiger Reischuk and Michel Morvan, editors, *STACS 97*, volume 1200 of *Lecture Notes in Computer Science*, pages 451–462. Springer Berlin Heidelberg, 1997.
- [8] Nicolas Courtois, Alexander Klimov, Jacques Patarin, and Adi Shamir. Efficient algorithms for solving overdefined systems of multivariate polynomial equations. In *Advances in Cryptology EUROCRYPT 2000*, volume 1807 of *Lecture Notes in Computer Science*, pages 392–407. Springer Berlin Heidelberg, 2000.
- [9] NicolasT. Courtois. The security of hidden field equations (hfe). In David Naccache, editor, *Topics in Cryptology CT-RSA 2001*, volume 2020 of *Lecture Notes in Computer Science*, pages 266–281. Springer Berlin Heidelberg, 2001.
- [10] NicolasT. Courtois. Algebraic attacks over $gf(2^k)$, application to hfe challenge 2 and sflash-v2. In Feng Bao, Robert Deng, and Jianying Zhou, editors, *Public Key Cryptography PKC 2004*, volume 2947 of *Lecture Notes in Computer Science*, pages 201–217. Springer Berlin Heidelberg, 2004.
- [11] NicolasT. Courtois and Josef Pieprzyk. Cryptanalysis of block ciphers with overdefined systems of equations. In Yuliang Zheng, editor, *Advances in Cryptology ASIACRYPT 2002*, volume 2501 of *Lecture Notes in Computer Science*, pages 267–287. Springer Berlin Heidelberg, 2002.
- [12] D.A. Cox, J. Little, and D. O’Shea. *Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra*. Undergraduate Texts in Mathematics. Springer, 2010.
- [13] W. Diffie and M. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654, 1976.

- [14] J. Ding, J. Buchmann, M. Mohamed, W. Mohamed, and R. Weinmann. Mutant xl. In *First International Conference on Symbolic Computation and Cryptography (SCC 2008), Beijing*, 2008.
- [15] Jintai Ding, Jason E. Gower, and Dieter S. Schmidt. *Multivariate public key cryptosystems*, volume 25 of *Advances in Information Security*. Springer, New York, 2006.
- [16] Jintai Ding, Jason E. Gower, and Dieter S. Schmidt. Zhuang-zi: A new algorithm for solving multivariate polynomial equations over a finite field, 2006.
- [17] 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.
- [18] Jintai Ding, Dieter Schmidt, and Fabian Werner. *Algebraic Attack on HFE Revisited*. Springer Berlin Heidelberg, 2008.
- [19] Jintai Ding, Dieter Schmidt, and Fabian Werner. Algebraic attack on HFE revisited. In Tzong-Chen Wu, Chin-Laung Lei, Vincent Rijmen, and Der-Tsai Lee, editors, *ISC*, volume 5222 of *Lecture Notes in Computer Science*, pages 215–227. Springer, 2008.
- [20] Jintai Ding and Dieter S. Schmidt. Mutant zhuang-zi algorithm. In Nicolas Sendrier, editor, *Post-Quantum Cryptography*, volume 6061 of *Lecture Notes in Computer Science*, pages 28–40. Springer Berlin Heidelberg, 2010.
- [21] Vivien Dubois, Pierre-Alain Fouque, Adi Shamir, and Jacques Stern. Practical cryptanalysis of sflash. In Alfred Menezes, editor, *Advances in Cryptology - CRYPTO 2007*, volume 4622 of *Lecture Notes in Computer Science*, pages 1–12. Springer Berlin Heidelberg, 2007.
- [22] Vivien Dubois and Nicolas Gama. The degree of regularity of hfe systems. In Masayuki Abe, editor, *Advances in Cryptology - ASIACRYPT 2010*, volume 6477 of *Lecture Notes in Computer Science*, pages 557–576. Springer Berlin Heidelberg, 2010.
- [23] Taher ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. In George Robert Blakley and David Chaum, editors, *Advances in Cryptology*, volume 196 of *Lecture Notes in Computer Science*, pages 10–18. Springer Berlin Heidelberg, 1985.
- [24] 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).
- [25] Jean-Charles Faugère and Antoine Joux. Algebraic cryptanalysis of hidden field equation (HFE) cryptosystems using Gröbner bases. In *Advances in cryptology—CRYPTO 2003*, volume 2729 of *Lecture Notes in Comput. Sci.*, pages 44–60. Springer, Berlin, 2003.
- [26] M.R. Garey, D.S. Johnson, et al. *Computers and Intractability: A Guide to the Theory of NP-completeness*. WH Freeman San Francisco, 1979.
- [27] Shafi Goldwasser and Silvio Micali. Probabilistic encryption & how to play mental poker keeping secret all partial information. In *Proceedings of the Fourteenth Annual ACM Symposium on Theory of Computing*, STOC '82, pages 365–377, New York, NY, USA, 1982. ACM.
- [28] Louis Granboulan, Antoine Joux, and Jacques Stern. Inverting hfe is quasipolynomial. In Cynthia Dwork, editor, *Advances in Cryptology - CRYPTO 2006*, volume 4117 of *Lecture Notes in Computer Science*, pages 345–356. Springer Berlin Heidelberg, 2006.
- [29] Chia hsin Owen Chen, Ming shing Chen, Jintai Ding, Fabian Werner, and Bo yin Yang. B.y.: Odd-char multivariate hidden field equations. cryptology eprint archive. Cryptology ePrint Archive, Report 2008/543, 2008.

- [30] Xin Jiang, Jintai Ding, and Lei Hu. Kipnis-shamir attack on hfe revisited. In Dingyi Pei, Moti Yung, Dongdai Lin, and Chuankun Wu, editors, *Information Security and Cryptology*, volume 4990 of *Lecture Notes in Computer Science*, pages 399–411. Springer Berlin Heidelberg, 2008.
- [31] Jonathan Katz and Yehuda Lindell. *Introduction to Modern Cryptography (Chapman & Hall/Crc Cryptography and Network Security Series)*. Chapman & Hall/CRC, 2007.
- [32] 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 Comput. Sci.*, pages 19–30. Springer, Berlin, 1999.
- [33] Rudolf Lidl and Harald Niederreiter. *Finite fields*, volume 20 of *Encyclopedia of Mathematics and its Applications*. Cambridge University Press, Cambridge, second edition, 1997. With a foreword by P. M. Cohn.
- [34] The MAGMA computational algebra system home page. <http://magma.maths.usyd.edu.au/magma>.
- [35] Tsutomu Matsumoto and Hideki Imai. Public quadratic polynomial-tuples for efficient signature-verification and message-encryption. In *Advances in cryptology—EUROCRYPT '88 (Davos, 1988)*, volume 330 of *Lecture Notes in Comput. Sci.*, pages 419–453. Springer, Berlin, 1988.
- [36] MohamedSaiedEmam Mohamed, Daniel Cabarcas, Jintai Ding, Johannes Buchmann, and Stanislav Bulygin. Mxl3: An efficient algorithm for computing Gröbner bases of zero-dimensional ideals. In *Information, Security and Cryptology ICISC 2009*, volume 5984 of *Lecture Notes in Computer Science*, pages 87–100. Springer Berlin Heidelberg, 2010.
- [37] MohamedSaiedEmam Mohamed, WaelSaidAbdElmageed Mohamed, Jintai Ding, and Johannes Buchmann. Mxl2: Solving polynomial equations over $gf(2)$ using an improved mutant strategy. In *Post-Quantum Cryptography*, volume 5299 of *Lecture Notes in Computer Science*, pages 203–215. Springer Berlin Heidelberg, 2008.
- [38] Jacques Patarin. Cryptoanalysis of the Matsumoto and Imai public key scheme of Eurocrypt'88. In *CRYPTO '95: Proceedings of the 15th Annual International Cryptology Conference on Advances in Cryptology*, pages 248–261, London, UK, 1995. Springer-Verlag.
- [39] Jacques Patarin. Hidden Field Equations (HFE) and Isomorphisms of Polynomials (IP): Two new families of asymmetric algorithms. In Ueli Maurer, editor, *Advances in Cryptology—EUROCRYPT 96*, volume 1070 of *Lecture Notes in Computer Science*, pages 33–48. Springer-Verlag, 1996.
- [40] Jacques Patarin, Nicolas Courtois, and Louis Goubin. Flash, a fast multivariate signature algorithm. In David Naccache, editor, *Topics in Cryptology CT-RSA 2001*, volume 2020 of *Lecture Notes in Computer Science*, pages 298–307. Springer Berlin Heidelberg, 2001.
- [41] Jacques Patarin, Louis Goubin, and Nicolas Courtois. C + * and hm: Variations around two schemes of t. matsumoto and h. imai. In Kazuo Ohta and Dingyi Pei, editors, *Advances in Cryptology ASI-ACRYPT98*, volume 1514 of *Lecture Notes in Computer Science*, pages 35–50. Springer Berlin Heidelberg, 1998.
- [42] RL Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.
- [43] Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Rev.*, 41(2):303–332 (electronic), 1999.
- [44] Christopher Wolf. “hidden field equations” (HFE) - variations and attacks. Master’s thesis, Universität Ulm, 2002.
- [45] Christopher Wolf and Bart Preneel. Equivalent keys in hfe, c^* , and variations. In Ed Dawson and Serge Vaudenay, editors, *Progress in Cryptology Mycrypt 2005*, volume 3715 of *Lecture Notes in Computer Science*, pages 33–49. Springer Berlin Heidelberg, 2005.

- [46] Christopher Wolf and Bart Preneel. Equivalent keys in multivariable quadratic public key systems. *Lecture Notes in Computer Science*, 4(4):375–415, 2005.
- [47] Christopher Wolf and Bart Preneel. Large superfluous keys in multivariate quadratic asymmetric systems. In Serge Vaudenay, editor, *Public Key Cryptography - PKC 2005*, volume 3386 of *Lecture Notes in Computer Science*, pages 275–287. Springer Berlin Heidelberg, 2005.

Appendices

APPENDIX A

Additional key generation data

n	CPU Time [s]	Memory [MB]	$\lfloor \log_q D \rfloor$	n	CPU Time [s]	Memory [MB]	$\lfloor \log_q D \rfloor$
13	7.21	25	12	25	920.15	590	24
15	23.31	46	14	27	1620.43	891	26
17	58.45	85	16	29	2587.3	1272	28
19	124.18	154	18	31	4114.26	1774	30
21	279.61	252	20	33	6359.63	2491	32
23	524.97	391	22	35	9572.66	3349	34

Table A.1: Private key generation for $q = 2$ and $D_0 = 386$.

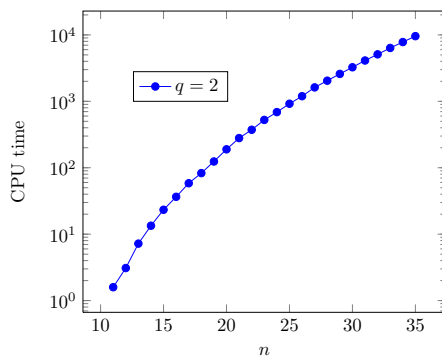


Figure A.1: Private key generation for $q = 2$ and $D_0 = 386$.

n	CPU time [s]	Memory [MB]	Number of rows	Number of columns	Density
11	2.86	64	1408	1694	12.94
12	4.28	64	1836	2160	11.66
13	7.60	64	2340	2704	11.18
14	11.21	96	2926	3332	10.27
15	17.77	128	3600	4050	9.86
16	25.46	160	4368	4864	9.21
17	41.44	192	5236	5780	8.97
18	56.65	256	6210	6804	8.30
19	85.57	320	7296	7942	8.07
20	109.38	416	8500	9200	7.59
21	153.57	512	9828	10584	7.41
22	204.51	608	11286	12100	6.96
23	272.96	778	12880	13754	6.81
24	343.37	940	14616	15552	6.43
25	462.57	1166	16500	17500	6.30
26	560.41	1361	18538	19604	5.98
27	735.24	1685	20736	21870	5.87
28	911.06	1977	23100	24304	5.61
29	1148.27	2333	25636	26912	5.50
30	1402.11	2700	28350	29700	5.25
31	1756.90	3282	31248	32674	5.17
32	2019.73	3609	34336	35840	4.95
33	2523.72	4384	37620	39204	4.88
34	2984.43	5035	41106	42772	4.68
35	3661.46	5813	44800	46550	4.61

Table A.2: Private key generation for $q = 7$ and $D_0 = 105$.

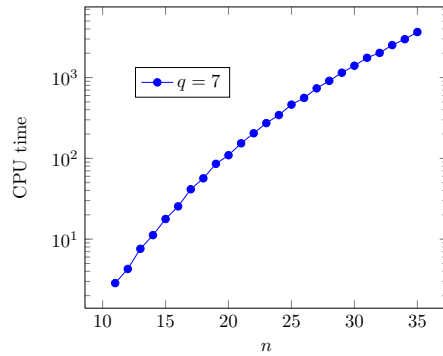


Figure A.2: Private key generation for $q = 7$ and $D_0 = 105$.

APPENDIX B

Additional KS attack data

n	Q-Rank(P)	Average time	Maximum memory
2	1	0.010 s	32
4	1	0.010 s	32
6	2	1.390 s	32
8	> 3	> 10 days	> 50 GB
10	> 3	> 10 days	> 50 GB

Table B.1: KS attack against the new encryption scheme, for $q = 17$ and $D_0 = 595$.

r	Average Time	Maximum memory
1	0.040 s	32
2	0.440 s	32 MB
3	281.360 s	470 MB
4	> 10 days	> 50 GB

Table B.2: Time and memory needed to find the solution set for the KS attack against the new encryption scheme, for $q = 17$, $n = 8$ and $D_0 = 595$.

(a) New encryption scheme					(b) Random equations	
n	$D_0 = 595$	$D_0 = 5219$	$D_0 = 9843$	$D_0 = 88451$	n	Q-Rank(P)
2	1	1	1	1	2	1
4	1	1	1	1	4	1
6	2	2	2	2	6	2
8	> 3	> 3	> 3	> 3	8	> 3
10	> 3	> 3	> 3	> 3	10	> 3

Table B.3: Q-Rank(P) comparison between the new trapdoor function and random equations for $q = 17$.

APPENDIX C

Gröbner Bases

Let k be a field and let n be a positive integer. For each $\alpha = (\alpha_1, \dots, \alpha_n) \in \mathbb{Z}_{\geq 0}^n$ we can construct the monomial $x^\alpha = x_1^{\alpha_1} \cdots x_n^{\alpha_n} \in k[x_1, \dots, x_n]$. This gives a natural bijection between the set of monomials in the ring $k[x_1, \dots, x_n]$ and the set $\mathbb{Z}_{\geq 0}^n$.

Definition C.1. A *monomial order* on $k[x_1, \dots, x_n]$ is a total ordering $>$ on $\mathbb{Z}_{\geq 0}^n$, or equivalently a total ordering on the set of monomials of $k[x_1, \dots, x_n]$, having two additional properties:

1. If $\alpha > \beta$ and γ are in $\mathbb{Z}_{\geq 0}^n$, then $\alpha + \gamma > \beta + \gamma$.
2. The order $>$ is a well-ordering on $\mathbb{Z}_{\geq 0}^n$, i.e., every nonempty subset of $\mathbb{Z}_{\geq 0}^n$ has a smallest element respect to $>$.

One example of a monomial order is the *lexicographic order* ($>_{lex}$). We say $\alpha >_{lex} \beta$ if the leftmost nonzero entry of the vector $\alpha - \beta$ is positive. For example, in $\mathbb{Z}_{\geq 0}^3$, if $\alpha = (2, 1, 0)$ and $\beta = (1, 2, 3)$ we have $\alpha >_{lex} \beta$ since $\alpha - \beta = (1, -1, -3)$. This means that $x^2y >_{lex} xy^2z^3$ on $k[x, y, z]$.

Definition C.2. Let $>$ be a monomial order on $k[x_1, \dots, x_n]$ and let $f = \sum_{\alpha \in A_f} a_\alpha x^\alpha$ be a nonzero polynomial in $k[x_1, \dots, x_n]$, where A_f is a nonempty finite subset of $\mathbb{Z}_{\geq 0}^n$. Define the leading monomial of f to be

$$\text{LM}(f) = \max \{x^\alpha : \alpha \in A_f \text{ and } a_\alpha \neq 0\}.$$

Also define the leading term of f as $\text{LT}(f) = a_\alpha x^\alpha$ and the leading coefficient of f as $\text{LC}(f) = a_\alpha$, where $\text{LM}(f) = x^\alpha$.

For example, if $k = GF(3)$ and $f = x^3yz^2 + 2x^3y^2z + z^4 \in k[x, y, z]$, then $\text{LM}(f) = x^3y^2z$, $\text{LT}(f) = 2x^3y^2z$, and $\text{LC}(f) = 2$. Here we use the lexicographic order.

If $I \subset k[x_1, \dots, x_n]$ is an ideal, $I \neq \{0\}$, we denote by $\text{LT}(I)$ the set of leading terms of elements of I . We also denote by $\langle \text{LT}(I) \rangle$ the ideal generated by the elements of $\text{LT}(I)$. If $I = \langle g_1, \dots, g_t \rangle$ for some $g_1, \dots, g_t \in I$, it is clear that $\langle \text{LT}(g_1), \dots, \text{LT}(g_t) \rangle \subseteq \langle \text{LT}(I) \rangle$. In general these sets are not the same. When these sets are equal, the set $\{g_1, \dots, g_t\}$ is called a Gröbner basis for I .

Definition C.3. Fix a monomial order on $k[x_1, \dots, x_n]$. A *Gröbner basis* of an ideal $I \subset k[x_1, \dots, x_n]$, $I \neq \{0\}$, is a set $G = \{g_1, \dots, g_t\} \subset I$ such that

$$\langle \text{LT}(g_1), \dots, \text{LT}(g_t) \rangle = \langle \text{LT}(I) \rangle.$$

It can be proved that every ideal other than $\{0\}$ has a Gröbner basis (see [12]). The classical algorithm to compute Gröbner bases is due to Buchberger [6]. Recently, two more powerful algorithms to compute Gröbner basis, called F_4 and F_5 , were introduced by Faugère [24].

If $I = \langle f_1, \dots, f_m \rangle$ for some $f_1, \dots, f_m \in I$, then a Gröbner basis of I is useful to determine the common solutions in k^n of the system of equations

$$\begin{aligned} f_1(x_1, \dots, x_n) &= 0 \\ f_2(x_1, \dots, x_n) &= 0 \\ &\vdots \\ f_m(x_1, \dots, x_n) &= 0. \end{aligned} \tag{C.1}$$

When we try to solve the system (C.1) over a finite field k of size q , we can add the field equations. Thus, we create the ideal

$$I = \langle f_1, \dots, f_m, x_1^q - x_1, \dots, x_n^q - x_n \rangle.$$

The field equations are added with the purpose of discarding solutions outside k^n . For small q the field equations allow us to keep down the maximum degree of the polynomials during the computation of a Gröbner basis using the F_4 algorithm. However, in [19] the authors pointed out that for high values of q , adding the field equations is not as useful as it is for small values of q .