An algebraic approach for the resolution of algorithmic problems raised by cryptography and coding theory

Vlad Florin Dragoi

Normandie Université

THÈSE

Pour obtenir le diplôme de doctorat

Spécialité Informatique

Préparée au sein de l'Université de Rouen Normandie

Approche algébrique pour l'étude et la résolution de problèmes algorithmiques issus de la cryptographie et de la théorie des codes

Présentée et soutenue par
Vlad Florin DRAGOI

Thèse soutenue publiquement le 06 juillet 2017 devant le jury composé de		
M. Daniel AUGOT	Directeur de recherche, INRIA Saclay, Laboratoire d'Informatique de l'Ecole Polytechnique (LIX)	Examinateur
Mme Magali BARDET	Maître de conférences, Université de Rouen Normandie, LITIS	Examinatrice
M. Thierry BERGER	Professeur émérite, Université de Limoges, XLIM	Examinateur
M. Philippe GABORIT	Professeur, Université de Limoges, XLIM	Rapporteur
M. Ayoub OTMANI	Professeur, Université de Rouen Normandie, LITIS	Examinateur
M. Nicolas SENDRIER	Directeur de recherche, INRIA Paris	Rapporteur
M. Jean-Pierre TILLICH	Directeur de recherche, INRIA Paris	Membre Invité

Thèse dirigée par Ayoub OTMANI (LITIS) et co-encadrée par Magali BARDET (LITIS)

UNIVERSITÉ DE ROUEN NORMANDIE

MIIS

litis

Résumé

Tout d'abord, mon sujet de recherche porte sur le cryptographie à clé publique, plus précisément la cryptographie basée sur la théorie des codes correcteurs d'erreurs. L'objectif principal de cette thèse est d'analyser la sécurité des systèmes de chiffrement. Pour cela j'étudie les propriétés structurelles des différentes familles de codes linéaires utilisées dans la pratique. Mon travail de recherche s'est orienté de maniéré naturelle, vers l'étude des deux dernières propositions de cryptosystèmes, plus exactement le schéma de McEliece à base des codes MDPC [MTSB13](moderate parity check codes) et des codes Polaires [SK14].

Dans le cas des codes MDPC on a mis en évidence une faiblesse importante au niveau des clés utilisées par les utilisateurs du système. En effet, on a proposé un algorithme très efficace qui permet de retrouver une clé privé à partir d'une clé publique. Ensuite on a compté le nombre des clés faibles et on a utilisé le problème d'équivalence de codes pour élargir le nombre de clés faibles. On a publié notre travail de recherche dans une conférence internationale en cryptographie [BDLO16].

Ensuite on a étudié les codes Polaires et leur application à la cryptographie à clé publique. Depuis leur découverte par E. Arikan [Arı09], les codes Polaires font partie des famille de codes les plus étudié du point de vue de le théorie de l'information. Ce sont des codes très efficaces en terme de performance car ils atteignent la capacité des canaux binaires symétriques et ils admettent des algorithmes d'encodage et décodage très rapides. Néanmoins, peu des choses sont connu sur leur propriétés structurelles. Dans ce cadre la, on a introduit un formalisme algébrique qui nous a permit de révéler une grande partie de la structure de ces codes. En effet, on a réussi à répondre à des questions fondamentales concernant les codes Polaires comme : le dual ou la distance minimale d'un code Polaire, le groupe des permutations ou le nombre des mots de poids faible d'un code Polaire. On a publié nos résultats dans une conférence internationale en théorie de l'information [BDOT16].

Par la suite on a réussi à faire une cryptanalyse complète du schéma de McEliece à base des codes Polaires. Ce résultat a été une application directe des propriétés découvertes sur les codes Polaires et il a été publié dans une conférence internationale en cryptographie post-quantique [BCD+16].

Abstract

First of all, during my phd I focused on the public key cryptography, more exactly on the code-based cryptography. The main motivation is to study the security of the latest encryption schemes. For that, I analyzed in detail the structural properties of the main code families. Thus, my research was naturally directed to the study of the McEliece based encryption schemes, among which the latest MDCP based variant [MTSB13] and Polar codes variant [SK14].

In the case of the MDPC based variant, we manage to reveal an important weakness regarding the key pairs that are used in the protocol. Indeed, we proposed an efficient algorithm that retrieves the private key given the public key of the scheme. Next we counted the proportion of weak keys and we used the code equivalence problem to extend

the number of weak keys. We published our results in an international conference in cryptography [BDLO16].

Next we studied the Polar codes and their application to public key cryptography. Since they were discovered by Arikan [Arı09], Polar codes are part of the most studied from an information theory point of view, family of codes. In terms of performance they are really efficient since they are capacity achieving over the Binary Discrete Memoryless Channels and they allow extremely fast encoding and decoding algorithms. Nonetheless, few facts are known about their structure. In this context, we have introduced an algebraic formalism which allowed us to reveal a big part of the structure of Polar codes. Indeed, we have managed to answer fundamental questions regarding Polar codes such as the dual, the minimum distance, the permutation group and the number of minimum weight codewords of a Polar code. Our results were published in an international conference in information theory [BDOT16].

We also managed to completely cryptanalyze the McEliece variant using Polar codes. The attack was a direct application of the aforementioned results on the structural properties of Polar codes and it was published in an international conference in post-quantum cryptography [BCD+16].

Acknowledgments

First of all, I am most grateful to my supervisors Ayoub Otmani and Magali Bardet. I would like to thank them for giving me the opportunity to make part of the crypto team at the University of Rouen Normandy. The words are not enough to express my gratitude for their patience and support as well as their suggestions on the relevant papers that I read and studied during the thesis. I would also like to thank them for all the efforts that were done in order for me to built a real project around my thesis and by that I mean the opportunity to assist to many conferences and workshops and also to meet many researchers in the field.

I express my gratitude to the jury members: Daniel Augot, Thierry Berger, Philippe Gaborit, Nicolas Sendrier and Jean-Pierre Tillich. They took the time to read my manuscript and also to have interesting discussion on other potential perspectives of my results. I would like to thank Jean-Pierre Tillich for the valuable comments and his guidance and support.

The collaboration with my co-authors Ayoub Otmani, Magali Bardet, Jean-Gabriel Luque, Jean-Pierre Tillich, Pierre-Louis Cayrel, Julia Chaulet, Tania Richmond and Brice Colombier was very fruitful, for that I thank them.

I also want to thank Hervé Talé Khalachi for our collaboration and the projects that we are working on. He is one of the colleagues that supported me in the hard moments and I appreciate his friendship. I want to express my gratitude to Ali Chouria and Clément Miklarz, maybe the best two colleagues that I could ever imagine. Also I bog thanks to Amazigh Amrane which makes the live much more easier in our office. A big thank you to Mohamed Saheed Taha for the valuable discussions during the coffee breaks.

My thesis was conducted within the computer science department of the University of Rouen Normandy, more exactly in the Combinatorics and Algorithms team which is part of the LITIS Laboratory. Therefore I wand to that all the members of the team for their support and comments. I have a special thought for all the good advice about life in general that Pascal Caron gave me during these years.

I would like to thank all the members from the research group on cryptography and coding theory from INRIA which I meet in our monthly meetings and especially the organizer of this collaboration group, Jean-Pierre Tillich.

Finally, I would like to thank my wife Alina. She had the patience and the courage to

follow me into this adventure. Many of my valuable ideas came to me when I was next to her and maybe the arrival of our son Vladimir was one of the most beautiful thing that ever happened during these years. I also like to thank my parents and my sister Alexandra even though far from Rouen they always encouraged and supported me.

à Vladimir et Alina

Contents

1 Introduction **xi**
- 1.1 Motivations . . . xi
- 1.2 Our contribution . . . xiii
  - 1.2.1 Decreasing Monomial Codes . . . xiii
  - 1.2.2 Cryptanalysis of the McEliece scheme based on Polar codes . . . xiv
  - 1.2.3 Weak keys for the QC-MDPC McEliece . . . xiv

2 Code-based Cryptography **1**
- 2.1 Introduction . . . 1
- 2.2 Coding Theory . . . 2
  - 2.2.1 Channel coding . . . 2
  - 2.2.2 Linear Codes . . . 4
  - 2.2.3 Code families . . . 10
- 2.3 The McEliece Public Key Encryption Scheme . . . 12
  - 2.3.1 Textbook McEliece . . . 12
  - 2.3.2 Security of the scheme . . . 13
  - 2.3.3 McEliece PKC variants . . . 15
- 2.4 Conclusion . . . 17

3 Decreasing Monomial Codes **19**
- 3.1 Introduction . . . 19
- 3.2 Monomial Codes . . . 20
  - 3.2.1 Definitions and Properties . . . 20
  - 3.2.2 Construction of Monomial codes . . . 22
  - 3.2.3 Polar codes are Monomial codes . . . 23
- 3.3 Decreasing and Weakly Decreasing Monomial Codes . . . 27
  - 3.3.1 Definitions and Properties . . . 27
  - 3.3.2 Weakly Decreasing Monomial Codes . . . 28
  - 3.3.3 Decreasing Monomial Codes . . . 28
  - 3.3.4 Polar codes are Decreasing Monomial codes . . . 32
- 3.4 Duality properties . . . 41

3.4.1 Dual of Monomial Codes . . . 41
3.4.2 Dual of Weakly Decreasing Monomial Codes . . . 43
3.5 Minimum Distance . . . 45
3.6 Permutation Group . . . 46
3.6.1 Definitions and Properties . . . 46
3.6.2 Permutation group of Weakly Decreasing Monomial codes. . . . 47
3.6.3 Permutation group of Decreasing Monomial codes . . . 48
3.7 Minimum weight codewords . . . 48
3.7.1 Orbits under the action of LTA$(m, 2)$ . . . 48
3.7.2 Computing the cardinality of orbits . . . 50
3.7.3 The minimum weight codewords of a decreasing monomial code. . . 53
3.8 Perspectives . . . 55

4 Cryptanalysis of the McEliece scheme based on Polar Codes 59
4.1 Introduction . . . 59
4.2 The McEliece PKC variant using Polar codes . . . 59
4.2.1 Introduction . . . 59
4.2.2 Parameters for the scheme . . . 60
4.2.3 Security arguments . . . 60
4.3 Solving the code equivalence problem for Polar codes . . . 61
4.3.1 Definitions . . . 61
4.3.2 Preliminaries . . . 62
4.3.3 Attack algorithm. . . . 62
4.4 Cryptanalyze of the McEliece variant based on Polar codes . . . 63
4.4.1 Step 1 – Minimum weight codewords searching. . . . 63
4.4.2 Step 2 – Signature of orbits in $\mathrm{W}_{\mathrm{min}}$ . . . 63
4.4.3 Step 4 – Identification of affine spaces . . . 67
4.4.4 Step 5 – Equivalence problem for a short decreasing monomial code 67
4.4.5 Step 6 – Induction step . . . 68
4.5 Implementation . . . 68
4.6 Perspectives . . . 69

5 Weak keys in the QC-MDPC McEliece 71
5.1 Introduction . . . 71
5.2 Preliminaries on QC-MDPC Codes . . . 73
5.2.1 Cyclic and Quasi-Cyclic codes . . . 73
5.2.2 QC-MDPC codes . . . 75
5.3 QC-MDPC McEliece . . . 76
5.3.1 Description . . . 76
5.3.2 The choice of parameters for the scheme . . . 77
5.4 Weak keys for the QC-MDPC scheme . . . 78
5.4.1 The key recovery attack . . . 78
5.4.2 The Rational Reconstruction Problem . . . 79
5.4.3 Weak keys . . . 80
5.5 Equivalence of codes . . . 84
5.5.1 Equivalence of cyclic codes . . . 84

5.5.2 Equivalence of quasi-cyclic codes. . . . 86
5.6 Attacking equivalent public keys . . . 87
5.6.1 The modified EEA for the attack . . . 87
5.6.2 Orbits of the QC-MDPC private keys . . . 88
5.7 Weak orbits and Extended weak orbits . . . 90
5.7.1 Orbits under the action of $(\mathbb{F}_p)^2$ . . . 90
5.7.2 Proportion of weak orbits . . . 91
5.7.3 Orbits under the action of $\mathbb{F}_p{}^*$ . . . 92
5.7.4 Proportion of extended weak orbits . . . 92
5.8 Computing the proportion of weak orbits . . . 93
5.8.1 Redefining the problem . . . 93
5.8.2 Lyndon words . . . 94
5.8.3 Longest run of Lyndon words with fixed weight . . . 96
5.8.4 Probability of weak orbits . . . 101
5.9 Extended weak orbits . . . 105
5.9.1 General Properties . . . 105
5.9.2 Proportion of extended weak orbits . . . 109
5.10 Numerical Results . . . 110
5.11 Complexity and Experimental Timings . . . 111
5.11.1 Preliminaries . . . 111
5.11.2 Complexity analysis . . . 112
5.11.3 Numerical results . . . 114
5.11.4 Secure QC-MDPC . . . 115
5.12 Perspectives . . . 116

6 Conclusion **117**

Appendices **119**
A Permutation group of linear codes . . . 121
B Decoding Polar codes . . . 127
C Factorization of $x^p - 1$ . . . 131
D Binomial Coefficient - Asymptotics . . . 133

List of Figures **141**

Bibliography **143**

1 Introduction

1.1 Motivations

Public-key cryptography is one of the newest subfield in cryptology and maybe one of the most challenging one. Essentially based on the evolution of the Internet of things and the highly increasing number of connected devices, the public key cryptography became one of the most spread solution for Internet security issues. It emerged in the late seventies motivated by a practical issue: how to securely exchange keys over a non reliable communication channel. The first key exchange protocol based on the idea of using a public key/private key pair was proposed by Diffie-Hellman [DH76], followed two years after by the public-key encryption scheme of Rivest, Shamir and Adleman [RSA78]. Miller [Mil85] proposed for the first time in 1985 to use elliptic curve in public key cryptography, field that is widely used nowadays. Three of the main technologies on which information technologies and Internet are based, namely TLS, PGP and SSH, they all implement Elliptic Curve Cryptography.

Meanwhile the scientific community begin to study the mathematical problems on which public-key schemes base their security. There are two number theory problems for the aforementioned protocols, the integer factorization and the discrete logarithm. If at the beginning, the general belief was that these two problems are hard enough for a cryptographic purpose, nowadays the future of number theory based cryptosystems is rather uncertain. There are two reasons for that:

- The latest classic algorithms for discrete logarithm in small characteristic finite fields are quasi-polynomial in the size of the group [BGJT14] and therefore they became a real threat from a theoretical point of view as well as a practical point of view.

- The quantum algorithms for factoring integers over $\mathbb{Z}$ and for computing logarithms in the multiplicative group ${\mathbb{F}_p}^*$ have a theoretical polynomial time complexity [Sho94]. Even though in practice we are still far from factoring a 2048-bit RSA module, the danger coming from the quantum computers has been raised.

The question weather it is a classical algorithm or a quantum algorithm that will break for the first time a 2048-RSA public key in practice in a reasonable time, is a challenging debate. Lately another question is making a lot of noise, that is the problem of finding a proper quantum resistant encryption scheme. National security agencies (NSA) as well as

institutions that establish standards in information technologies (NIST in the USA, ETSI in Europe) and international projects (PQCrypto EU) related to Quantum Resistant Cryptography seem to accelerate the process. From the private sector there are many signals that seem to enforce the necessity of a Quantum Resistant Cryptography (see Intel, Microsoft etc.).

There are two fields on which the scientific community focus their attention: quantum cryptography based on quantum theory and post-quantum cryptography based on classic theory. They are both suppose to deal with quantum computer threats and thereby might replace the actual number theory based cryptography. We deal here with the classical vision and study one of the solutions proposed by the post-quantum cryptography.

There are several hard problems on which the post-quantum cryptography base their security and we recall here one of them. Let n and k be two integers such that $1 \leqslant k \leqslant n-1$ and $\boldsymbol{H}$ be a random $(n-k) \times n$ matrix over a field $\mathbb{F}$. Then we have the following problem

Instance: A $(n-k) \times n$ matrix $\boldsymbol{H}$ over $\mathbb{F}$, a vector $\boldsymbol{s} \in \mathbb{F}^{n-k}$ and a small integer $\omega > 0$.

Question: Is there a vector $\boldsymbol{x} \in \mathbb{F}^n$ of *weight* $\leqslant \omega$, such that $\boldsymbol{H}\boldsymbol{x}^T = \boldsymbol{s}$?

We emphasize that without the weight condition this problem is easy to solve using linear algebra. It is exactly the condition on the weight that makes the later problem difficult and there are three different well-known distances which are used for cryptographic purpose:

- when $\mathbb{F}$ is a finite field, $\mathbb{F} = \mathbb{F}_q$ we can consider the Hamming distance. Then the problem is known as the Syndrome Decoding Problem and it was proved NP-complete by Berlekamp, McEliece and van Tilborg in [BMvT78]. Based on this problem McEliece proposed a public key encryption scheme [McE78]. This subfield is known under the name of code-based cryptography field.

- when $\mathbb{F}$ is a finite field such that $\mathbb{F} = \mathbb{F}_{p^m}$ we can consider the Rank distance. Then the problem is known as the Rank Syndrome Decoding Problem and it was proved NP-complete by Gaborit and Zémor in [GZ14]. This subfield is known as the rank-based cryptography, from which we recall the LRPC encryption scheme [GMRZ13].

- when $\mathbb{F}$ is the residue integer ring $\mathbb{F} = \mathbb{Z}_q$ we consider the Euclidean distance and the problem is known as the decisional Closest Vector Problem. Aijtaj begun the study of this type of problems and initiated based on the hardness of the CVP the lattice based cryptography [Ajt96, AD97], which is one of the most promising post-quantum solutions. We recall some well known encryption schemes coming from this area, namely the Ring-LWE [LPR10], the GGH scheme [GGH97] or the NTRU scheme [HPS98].

The resemblance between these three fields is not only in the problem on which their security stands on, it is also in the choice of the encryption schemes. The most promising encryption protocols, and here we refer to efficiency, key size and security arguments, are the QC-MDPC McEliece, the QC-LRPC and the NTRU-cryptosystem. They all have a similar description, since the private key can be defined as a pair of polynomials $(h_1(x), h_2(x)) \in (\mathbb{F}[x]/(x^n-1))^2$ with weight $\|h_1\| + \|h_2\| \leqslant \omega$ and the public key is the polynomial $f(x) = h_1(x)h_2(x)^{-1} \pmod{x^n-1}$.

In this particular challenging context we focus on the first candidate, the code-based cryptography and study the security of the latest encryption schemes, namely the QC-MDPC McEliece and the Polar codes based McEliece.

1.2 Our contribution

1.2.1 Decreasing Monomial Codes

The first contribution is to introduce a class of algebraic error correcting codes, that we call *Decreasing Monomial Codes*. As their name indicates this family of codes possess an algebraic structure since it is defined as evaluation of multivariate monomials. In addition the monomials forming a basis for a Decreasing Monomial code are selected with respect to an ordering, the "$\preceq$" (see Definition 3.3.1).

The initial motivation was rather a cryptographic one, since we first propose the use of such a code family when we cryptanalyzed the McEliece variant using Polar codes. The intriguing part is that Polar codes were introduces as part of the probabilistic codes and the scientific community was searching for an algebraic formalism related to Polar codes. Indeed in the initial article [Arı09], introducing Polar codes, Arikan pointed out that constructing Polar codes is a challenging task that maybe requires more knowledge on the underlying structure of the Polar codes. He also pointed out the strong relation between Polar codes and Reed-Muller codes, fact that gives the first intuition on the algebraic structure of Polar codes. Nevertheless Arikan's article was a major contribution in the field of coding theory since he proved for the first time that Polar codes achieve the capacity of many communication channels. We recall that many interesting properties were known about Polar codes but there is no complete study or knowledge about several issues like: dual of Polar codes, permutation group or minimum weight codewords of a Polar code. Therefore the study of Polar codes from a coding perspective is a highly motivating task and understanding the structure of this codes is necessary.

In the first place we prove that Polar codes designed for the Binary Discrete Memoryless Channels are Decreasing Monomial codes (see Theorem 3.3.31). We also prove that Reed-Muller codes are part of this big family of linear codes (see Proposition 3.3.12). Based on this fact we analyze the structure of the Polar codes, seen as a Decreasing Monomial code.

The main results are the following

- The dual of a Decreasing Monomial code is a Decreasing Monomial code (see Proposition 3.4.12).
- The permutation group of a Decreasing Monomial code contains the Lower Triangular Group, LTA$(m, 2)$ (see Definition 3.6.2), which is a subgroup of the General Afiine group (see Theorem 3.6.6).
- The structure of minimum weight codewords of a Decreasing Monomial code is given by the orbits of the maximum degree monomials (see Proposition 3.7.12).
- We give a counting method for the number of minimum weight codewords of a Decreasing Monomial code based on the Young diagrams associated to the maximum degree monomials (see Theorem 3.7.14).

1.2.2 Cryptanalysis of the McEliece scheme based on Polar codes

A major impact for the code-based community is that Polar codes can not be used in a McEliece type scheme, at least not in the original version of the cryptosystem. Our second major contribution related to Decreasing Monomial codes is that we propose a full cryptanalysis of the McEliece variant based on Polar codes. The attack is based on the fact that we are able to determine exactly the structure of the minimum weight codewords of a Polar code. Therefore we manage to distinguish between the maximum degree monomials for the code and thus solve the Code Equivalence Problem in this case.

1.2.3 Weak keys for the QC-MDPC McEliece

The McEliece variant using QC-MDPC codes [MTSB13] is one of the most promising candidates for a post-quantum cryptosystem using the theory of error correcting codes. There are several reason for that like for example the "random-like" structure of the codes, the efficiency, the key size and many others.

Weak keys approach - differences and advantages from generic model The best attacks against the QC-MDPC scheme are variants of the Information Set Decoding, with exponential complexity in the weight of the error vector [MTSB13]. Our initial motivation was to attack the QC-MDPC's security from another point of view. We investigate particular configurations which are vulnerable against polynomial time attacks. For that we redefine the Key Recovery Problem for the QC-MDPC scheme as a modified version of the well-known Rational Recovery Problem. Hence we use the Extended Euclidean Algorithm, which is one of the possible solutions to the later problem, to recover a private key from the public data.

We emphasize the difference between our method and generic ones. In the general model one considers the work factor of the best algorithm that is able to recover the private key from the public key for any random public key. Hence in this model any private key can be attacked with an exponential algorithm and thus there is no distinction whatsoever between the different private keys. On the other hand in the weak keys model, only a proportion of keys, that has to be determined, might be vulnerable to polynomial time attacks. But since the attack is very efficient, the designers of the scheme must carefully choose the parameters in such manner to avoid weak keys.

Weak keys for the QC-MDPC scheme - results and consequences Our first contribution is to estimate the proportion of weak keys of a $(2p, p, \omega)$ QC-MDPC scheme. The asymptotic analysis shows that the proportion of weak keys for the *smooth* $(2p, p, \omega)$ QC-MDPC schemes, is equivalent when p goes to infinity, to $\omega^{1/2}2^{-H(\alpha)\omega}$ where $H(x)$ is the binary entropy function. We also prove that the proportion of weak keys for all the $(2p, p, \omega)$ QC-MDPC schemes is equivalent to $\omega 2^{-H(\alpha)\omega}$.

The first observation is that the proportion of weak keys is really close to the security level given by the work factor of the best ISD variants. We also stress out that no structural property of the QC-MDPC code was considered in the counting process and thus the aforementioned probabilities might considerably be increased with the use of some specific properties. For this we consider equivalent QC-MDPC codes, fact that we

define in terms of group actions, namely we consider the additive group $(\mathbb{F}_p, +)$ and the multiplicative group $(\mathbb{F}_p{}^*, \cdot)$. With the use of the additive group we increase the proportion of weak keys by a factor equal to ω^2. As for the multiplicative group the proportion of weak keys can be increased in the best case by a factor equal to $p-1$. In the worst case the factor equals $(p-1)/\omega/2$.

We compute the values of the probabilities for all the proposed parameters of the QC-MDPC scheme from [MTSB13] and discover that for several parameters the probability of weak keys is considerably bigger than the announced security level of the scheme. Thus we propose a secured version of the Key Generation algorithm and analyze its time complexity. We also analyze the complexity of the attack and implement a practical attack in MAGMA Software, for which we give the execution timings. We point out that the secured version has two advantages: firstly it eliminated the possible weak keys and secondly it can be implemented in an efficient manner.

2 Code-based Cryptography

2.1 Introduction

Code-based cryptography appeared for the first time in 1978, when McEliece proposed the first public key encryption scheme which is not based on number theory primitives [McE78]. Instead he built a scheme for which the security relies on two problems:

- the hardness of the Syndrome Decoding Problem [BMvT78].
- the difficulty to distinguish between a binary Goppa code and a random linear code [FGO+10].

McEliece's scheme is composed of three algorithms: *key generation, encryption* and *decryption*. The key generation algorithm picks a random $k \times n$ generator matrix $\boldsymbol{G}$ of a binary linear code $\mathscr{C}$ which is itself randomly picked in a family of codes for which t errors can be efficiently corrected. The *secret* key is the decoding algorithm (or the private parameters of the decoding algorithm) associated to $\mathscr{C}$ and the *public* key is $\boldsymbol{G}$. To encrypt $\boldsymbol{m} \in \mathbb{F}_2^k$, the sender chooses a random vector $\boldsymbol{e}$ in $\mathbb{F}_2^n$ of Hamming weight less than or equal to t and computes the ciphertext $\boldsymbol{c} = \boldsymbol{mG} + \boldsymbol{e}$. The receiver then recovers the plaintext by applying a decoding algorithm on $\boldsymbol{c}$.

The scheme disposes of various advantages like for example

- the complexity of the encryption and decryption algorithms are equivalent to those of symmetric schemes and thus are very efficient compared to other public key schemes. [OS09]
- the best attacks for solving the Syndrome Decoding Problem are exponential in the code length, which makes code-based schemes of high potential.[TS16]

However code-based cryptography came with a big disadvantage: the size of the public keys was about five hundred thousands bits, for a 2^{60} security level, which was unacceptable at that time. Nevertheless the scientific community made a huge progress in reducing the key size of the McEliece PKC by proposing different structures like quasi-cyclic or quasi-dyadic codes. Nowadays the key size is no longer an issue and several practical implementations of the McEliece prove the efficiency and potential of the scheme [BS08, Str10b, CHP12, BCS13, HvMG13, MOG15].

Outline. The chapter is divided into two major parts Section 2.2 and Section 2.3. In the first section we recall the main facts concerning the theory of Error-Correcting codes. We begin with a subsection (2.2.1) on the general model of a channel of communication and then give the necessary background on linear codes (2.2.2). We emphasize in this part the main problems related to the cryptographic aspects such as: permutation group, syndrome decoding problem, minimum distance problem and the most frequent code constructions that are used in cryptography like shortened and punctured codes, Plotkin sum codes as well as star product codes. We also define and detail the main code families that we studied during the thesis (2.2.3).

In Section 2.3 we explain how the code-based cryptography evolved from its beginnings with the original McEliece scheme up to nowadays. In parallel we give a survey of the existing variants of the McEliece PKC (2.3.3) and the potential security flaws as well as the existing attacks for some of the variants. We also discuss from a general point of view which are the main threats for this scheme (2.3.2) and point out which are the remaining candidates.

2.2 Coding Theory

Coding theory is a fundamental field in communication and was introduce in 1948 by Claude E. Shannon in [Sha48]. The main purpose of coding theory is to propose schemes that can be used to efficiently and reliably transmit information over a noisy channel. From a practical point of view the main challenge is to design coding schemes that could approach channel capacity, also known as "the Shannon limit". Shannon's objective required extraordinary efforts and finally proved to be possible with the help of probabilistic codes.

2.2.1 Channel coding

Figure 2.1 – Channel coding

Channel coding can be defined as the process of adding enough *redundancy* to a message that has to be sent over a communication channel in order to recover the message after errors have been added by the channel.

Definition 2.2.1 (Discrete channel)**.** *Let k and m be two strictly positive integers. Then a discrete channel W is defined by*

- *A finite input alphabet $\mathcal{X} = \{x_1, \ldots, x_k\}$.*

- *A finite output alphabet* $\mathcal{Y} = \{y_1, \ldots, y_m\}$.
- *The transition probability* $k \times m$ *matrix* $\boldsymbol{P} = (p_{i,j})_{1 \leqslant i \leqslant k\,,\, 1 \leqslant j \leqslant m}$ *with* $p_{i,j} = W(y_j|x_i)$ *is the probability that* y_i *is received knowing that* x_i *was sent over the channel.*

The communication process can now be defined using a probability model for the source, the channel and the receiver.

- The source is defined through an input probability space given by $(\mathcal{X}, \mathsf{prob}_{\mathcal{X}})$, where $\mathsf{prob}_{\mathcal{X}}$ is a discrete probability distribution over $\mathcal{X}$. We denote the input probability vector by $\boldsymbol{p}_{\mathcal{X}} = (\mathsf{prob}_{\mathcal{X}}(x_1), \ldots, \mathsf{prob}_{\mathcal{X}}(x_k))$.
- The channel is defined through $\boldsymbol{P} = (W(y|x))_{(x,y) \in \mathcal{X} \times \mathcal{Y}}$.
- The receiver is defined through the output probability space given by $(\mathcal{Y}, \mathsf{prob}_{\mathcal{Y}})$ where the output probability vector $\boldsymbol{p}_{\mathcal{Y}} = (\mathsf{prob}_{\mathcal{Y}}(y_1), \ldots, \mathsf{prob}_{\mathcal{Y}}(y_m))$ is defined by the relation
$$\boldsymbol{p}_{\mathcal{Y}} = \boldsymbol{p}_{\mathcal{X}} \boldsymbol{P}.$$

We extend the conditional probability from symbols (letters) to finite length words in a natural manner: we define the transition probability $W(\boldsymbol{y}|\boldsymbol{x})$ as the probability of receiving the word $\boldsymbol{y} = y^{(1)} \ldots y^{(n)} \in \mathcal{Y}^n$ knowing that $\boldsymbol{x} = x^{(1)} \ldots x^{(n)} \in \mathcal{X}^n$ was sent over the channel. For "memoryless" channels we have

$$W(\boldsymbol{y}|\boldsymbol{x}) = \prod_{i=1}^{n} W(y^{(i)}|x^{(i)})$$

Figure 2.2 shows two well known memoryless channels, the Binary Symmetric (BSC) and Binary Erasure (BEC) channels.

- The BSC(p) is defined for $0 \leqslant p \leqslant 1/2$ by
$$\mathcal{X} = \mathcal{Y} = \{0,1\} \quad and \quad W(y_j|x_i) = \begin{cases} 1-p & if \quad y_j = x_i \\ p & if \quad y_j \neq x_i \end{cases}$$
- The BEC(p) is defined for $0 \leqslant p \leqslant 1$ by
$$\mathcal{X} = \{0,1\}, \mathcal{Y} = \mathcal{X} \cup \{?\} \quad and \quad W(y_j|x_i) = \begin{cases} 1-p & if \quad y_j = x_i \\ p & if \quad y_j =? \\ 0 & elsewhere \end{cases}$$

The "memoryless" property comes from the fact that both channels take a bit as input and flip (BSC) or erase (BEC) the bit independently of the past or the future. They are symmetric since the probabilities are the same regardless the value of the input. They are part of the big family of Binary Discrete Memoryless Channels (B-DMC). The parameter p is the crossover probability and represents the probability that a bit is not sent correctly, in particular it can be deleted in the case of a BEC or flipped in the case of a BSC.

Furthermore we define the notion of concatenated channel and degraded channel.

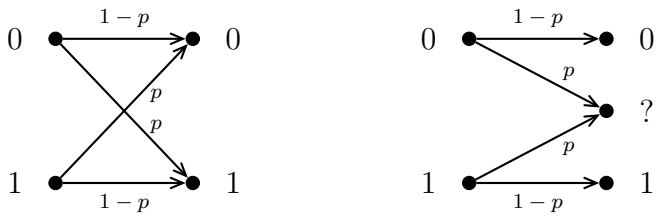


Figure 2.2 – (left) BSC(p) and (right) BEC(p)

Definition 2.2.2 (Concatenated channel)**.** *Let $W : \mathcal{X} \to \mathcal{Y}$ and $W' : \mathcal{Y} \to \mathcal{Z}$ be two memoryless channels such that the input alphabet of W' is equal to the output alphabet of W. The concatenation of W with W' is denoted by $W' \circ W$ and is a memoryless channel $W" : \mathcal{X} \to \mathcal{Z}$ with transition probabilities specified by*

$$W"(z|x) = \sum_{y \in \mathcal{Y}} W'(z|y)W(y|x).$$

Definition 2.2.3 (Degraded channel)**.** *Let $W : \mathcal{X} \to \mathcal{Z}$ and $W' : \mathcal{X} \to \mathcal{Y}$ be two memoryless channels, both with input alphabet $\mathcal{X}$ and respective output alphabets $\mathcal{Z}$ and $\mathcal{Y}$. We say that W is a channel degradation of W' if and only if there exists a memoryless channel $W" : \mathcal{Y} \to \mathcal{Z}$ such that $W = W" \circ W'$, that is*

$$W(z|x) = \sum_{y \in \mathcal{Y}} W"(z|y)W'(y|x).$$

We write

$$W \preceq_{\mathrm{d}} W'$$

to denote that W is degraded with respect to W'.

Lemma 2.2.4. *$\preceq_{\mathrm{d}}$ is a transitive relation*

$$\left.\begin{array}{lll} W & \preceq_{\mathrm{d}} & W' \\ W' & \preceq_{\mathrm{d}} & W" \end{array}\right\} \Rightarrow W \preceq_{\mathrm{d}} W". \tag{2.1}$$

Proof. Let $W : \mathcal{X} \to \mathcal{Z}$, $W' : \mathcal{X} \to \mathcal{Y}$ and $W'' : \mathcal{X} \to \mathcal{U}$. By definition of a degraded channel we have that $W = W_1 \circ W'$ with $W_1 : \mathcal{Y} \to \mathcal{Z}$ and $W' = W_2 \circ W''$ with $W_2 : \mathcal{U} \to \mathcal{Y}$. Then $W = W_3 \circ W''$ with $W_3 : \mathcal{U} \to \mathcal{Z}$ is such that $W_3 = W_1 \circ W_2$, which ends the proof. □

An example of degradation from [RU08] is the family of binary erasure channels $\{\mathrm{BEC}(p)\}_{0 \leqslant p \leqslant 1}$. For any pair of parameters (ε, δ) with $0 \leqslant \varepsilon < \delta \leqslant 1$ we have that $\mathrm{BEC}(\delta) \preceq_{\mathrm{d}} \mathrm{BEC}(\varepsilon)$ (see Figure 2.3).

2.2.2 Linear Codes

The main facts that we remind in this part are well known results coming from classical books like [MS86] or [Rot06].

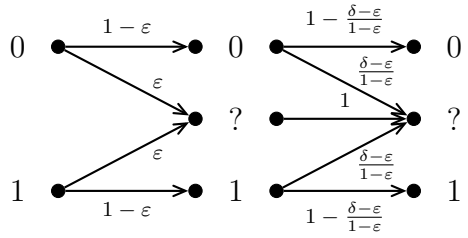


Figure 2.3 – BEC(δ) seen as a degradation of BEC(ε) for $0 \leqslant \varepsilon < \delta < 1$.

Preliminaries. All the objects that we recall in this part are defined over $\mathbb{F}_2$ since we concentrate our study on binary linear codes.

Definition 2.2.5 (Binary linear code)**.** *An $[n, k]$ linear code $\mathscr{C}$ over $\mathbb{F}_2$ is a linear subspace of dimension k of the vector space $\mathbb{F}_2^n$.*

Any element in $\mathscr{C}$ is called a *codeword*. The *redundancy* of $\mathscr{C}$ is the difference $n - k$ and the *rate* of a code is denoted by $r = k/n$.

Generator and Parity check matrix. A *generator matrix* for a $[n, k]$ linear code is a $k \times n$ binary matrix (often denoted by $\boldsymbol{G}$) whose rows form a basis for the code, as a vector space. A generator matrix is in *systematic form*, if $\boldsymbol{G} = (\boldsymbol{I}_k \mid \boldsymbol{R})$ where $\boldsymbol{I}_k$ is the identity matrix and $\boldsymbol{R}$ is a $k \times (n - k)$ binary matrix.

Definition 2.2.6 (Information set)**.** *Any subset $\mathcal{I} \subset \{0, 1, \ldots, n\}$ of the positions of a $[n, k]$ linear code is called an information set if for any generator matrix $\boldsymbol{G}$ the restriction of $\boldsymbol{G}$ to $\mathcal{I}$, denoted $\boldsymbol{G}_{\mathcal{I}}$, is invertible.*

A binary $(n - k) \times n$ matrix $\boldsymbol{H}$, is called a *parity-check matrix* of a linear code $\mathscr{C} = [n, k]$, if we have

$$\boldsymbol{c} \in \mathscr{C} \Leftrightarrow \boldsymbol{H}\boldsymbol{c}^T = \boldsymbol{0}_{\boldsymbol{n-k}}.$$

Since this equation is equivalent to $\boldsymbol{H}\boldsymbol{G}^T = \boldsymbol{0}_{\boldsymbol{(n-k)\times k}}$ and $\boldsymbol{H}$ has rank $(n-k)$, we can compute the parity-check matrix of a code given the generator matrix by finding a basis for the kernel of $\boldsymbol{G}$. This can be done easily when $\boldsymbol{G}$ is into systematic form since in this case we have $\boldsymbol{H} = \left(-\boldsymbol{R}^T \mid \boldsymbol{I}_{n-k}\right)$.

The *dual* of a linear code, denoted by $\mathscr{C}^\perp$ is the binary linear code which consists of all vectors $\boldsymbol{u} \in \mathbb{F}_2^n$ such that $\forall\, \boldsymbol{c} \in \mathscr{C}$, $\boldsymbol{u} \cdot \boldsymbol{c}^T = 0$. The generator matrix of the dual code is the parity-check matrix of the code and we have $\left(\mathscr{C}^\perp\right)^\perp = \mathscr{C}$.

Minimum distance of a linear code. The Hamming distance $\mathsf{d_H}$ between two codewords is the number of coordinates on which they differ. The Hamming distance endows the set of binary n length vectors with a metric and the pair $(\mathbb{F}_2^n, \mathsf{d_H})$ forms a metric space called the Hamming space of dimension n.

The Hamming weight of a codeword is the number of its coordinates different from zero. The support of a codeword $\mathsf{supp}(\boldsymbol{c})$ is the set of its coordinates different from zero.

Definition 2.2.7 (Minimum distance)**.** *The minimum distance of a linear code is:*

$$\mathsf{d}_{\min}(\mathscr{C}) \;=\; \min_{\substack{(\boldsymbol{c},\boldsymbol{c}^*)\in\mathscr{C}\times\mathscr{C}\\ \boldsymbol{c}\neq\boldsymbol{c}^*}} \mathsf{d}_{\mathsf{H}}(\boldsymbol{c},\boldsymbol{c}^*)$$

Since $\mathscr{C}$ is a binary linear code we have that $\mathsf{d}_{\mathsf{H}}(\boldsymbol{c},\boldsymbol{c}^*) = \mathsf{wt}(\boldsymbol{c}-\boldsymbol{c}^*)$. So $\mathsf{d}_{\min}(\mathscr{C}) = \min_{\boldsymbol{c}\in\mathscr{C},\boldsymbol{c}\neq 0} \mathsf{wt}(\boldsymbol{c})$ which is also equal to $\mathsf{d}_{\min}(\mathscr{C}) = \min_{\boldsymbol{c}\in\mathscr{C},\boldsymbol{c}\neq 0} |\mathsf{supp}(\boldsymbol{c})|$. In order to estimate $\mathsf{wt}(\boldsymbol{x}-\boldsymbol{y})$ for any pair of vectors $(\boldsymbol{x},\boldsymbol{y})$ in the Hamming space we have to define the intersection of $\boldsymbol{x}$ and $\boldsymbol{y}$

Definition 2.2.8 (Component-wise product)**.** *Let $\boldsymbol{x}$ and $\boldsymbol{y} \in \mathbb{F}_2^n$, then the component-wise product of $\boldsymbol{y}$ and $\boldsymbol{x}$ is*

$$\boldsymbol{x}\star\boldsymbol{y} \stackrel{def}{=} (x_1y_1,\ldots,x_ny_n) \;\in\; \mathbb{F}_2^n.$$

Proposition 2.2.9. *Let $\boldsymbol{x}$ and $\boldsymbol{y} \in \mathbb{F}_2^n$, then we have*

$$\mathsf{wt}(\boldsymbol{x}+\boldsymbol{y}) = \mathsf{wt}(\boldsymbol{x}) + \mathsf{wt}(\boldsymbol{y}) - 2\mathsf{wt}(\boldsymbol{x}\star\boldsymbol{y}). \tag{2.2}$$

Notation 2.2.10. *Furthermore for a linear code of length n, dimension k and minimum distance d we use the $[n,k,d]$ notation.*

There are code families for which the minimum distance is known, most of them are algebraic codes. In the general case estimating the minimum distance of a linear code is a hard problem.

Remark 2.2.11 (Decisional minimum distance problem)**.** *The minimum distance decision problem is NP-complete [Var97] and can be formalized as follow*

Instance: *A binary $(n-k)\times n$ matrix $\boldsymbol{H}$ and an integer $\omega > 0$.*
Question: *Is there a nonzero vector $\boldsymbol{x} \in \mathbb{F}_2^n$ of weight $\leqslant \omega$, s.t. $\boldsymbol{H}\boldsymbol{x}^T = \boldsymbol{0}_{n-k}$?*

Decoding linear codes.

Definition 2.2.12. *Let $\mathscr{C}$ be a $[n,k,d]$ linear code over $\mathbb{F}_2$ and W be a discrete channel defined by $(\mathbb{F}_2,\mathcal{Y},\boldsymbol{P})$. A decoder for $\mathscr{C}$ with respect to W is a function*

$$\mathcal{D} : \mathcal{Y}^n \to \mathscr{C}.$$

The probability that a codeword $\boldsymbol{c}$ is decoded erroneously, given that $\boldsymbol{c}$ was transmitted over the channel W is

$$\mathrm{P}_{\mathrm{err}}(\boldsymbol{c}) \stackrel{def}{=} \sum_{\substack{\boldsymbol{y}\in\mathcal{Y}^n\\ \mathcal{D}(\boldsymbol{y})\neq\boldsymbol{c}}} W(\boldsymbol{y}\mid\boldsymbol{c}).$$

The error probability *of $\mathcal{D}$ is*

$$\mathrm{P}_{\mathrm{err}} = \max_{\boldsymbol{c}\in\mathscr{C}} \mathrm{P}_{\mathrm{err}}(\boldsymbol{c}).$$

The error probability is a measure of the efficiency of a decoder and the goal is to have decoders with $\mathrm{P}_{\mathrm{err}}$ as small as possible. A particular decoding strategy is the maximum-likelihood decoder.

Definition 2.2.13 (Maximum-Likelihood Decoder)**.** *Given a $[n, k, d]$ linear code $\mathscr{C}$ over $\mathbb{F}_2$ and a channel $W = (\mathbb{F}_2, \mathcal{Y}, \boldsymbol{P})$ a* maximum-likelihood decoder *(MLD) for $\mathscr{C}$ with respect to W is the function $\mathcal{D}_{\text{MLD}} : \mathcal{Y}^n \to \mathscr{C}$ defined as:*

$$\text{for every } \boldsymbol{y} \in \mathcal{Y}^n \text{ , } \mathcal{D}(\boldsymbol{y}) \stackrel{def}{=} \underset{\boldsymbol{c} \in \mathscr{C}}{\operatorname{argmax}} W(\boldsymbol{y} \mid \boldsymbol{c}).$$

Take for example the case of a BSC(p) with crossover probability $0 < p < 1/2$. Then we have

$$\begin{aligned} W(\boldsymbol{y} \mid \boldsymbol{c}) &= \prod_{i=1}^{n} W(y_i \mid c_i) \\ &= p^{\mathsf{d_H}(\boldsymbol{y},\boldsymbol{c})}(1-p)^{n-\mathsf{d_H}(\boldsymbol{y},\boldsymbol{c})} \\ &= (1-p)^n \left(\frac{p}{1-p}\right)^{\mathsf{d_H}(\boldsymbol{y},\boldsymbol{c})}. \end{aligned}$$

Since $p/(1-p) < 1$ when $p < 1/2$ we have that $\mathcal{D}_{\text{MLD}}(\boldsymbol{y})$ is the codeword $\boldsymbol{c}$ which minimize the Hamming distance $\mathsf{d_H}(\boldsymbol{y}, \boldsymbol{c})$. In other words $\boldsymbol{c}$ is the closest codeword of $\mathscr{C}$ to $\boldsymbol{y}$.

Definition 2.2.14 (Nearest Codeword Decoder)**.** *The nearest codeword decoder for a $[n, k, d]$ linear code $\mathscr{C}$ over $\mathbb{F}_2$ is the function $\mathcal{D} : \mathbb{F}_2^n \to \mathscr{C}$ defined as:*

$$\forall \boldsymbol{y} \in \mathbb{F}_2^n \text{ , } \mathcal{D}(\boldsymbol{y}) \stackrel{def}{=} \underset{\boldsymbol{c} \in \mathscr{C}}{\operatorname{argmin}} \mathsf{d_H}(\boldsymbol{y}, \boldsymbol{c}).$$

Remark 2.2.15. *We notice that the closest codeword to $\boldsymbol{y}$ might not be unique. Indeed, the unicity property can be guaranteed when $\mathsf{d_H}(\boldsymbol{y}, \boldsymbol{c}) \leqslant \lfloor (d-1)/2 \rfloor$.*

Proof. In order to prove this fact suppose that $\mathscr{C}$ is a $[n, k, d]$ binary linear code and that $\boldsymbol{c}$ is the transmitted codeword and $\boldsymbol{y}$ the received vector with $\mathsf{d_H}(\boldsymbol{y}, \boldsymbol{c}) \leqslant (d-1)/2$. Suppose that $\mathcal{D}(\boldsymbol{y}) = \boldsymbol{c}' \neq \boldsymbol{c}$. By the definition of the decoder we have $\mathsf{d_H}(\boldsymbol{y}, \boldsymbol{c}') \leqslant \mathsf{d_H}(\boldsymbol{y}, \boldsymbol{c})$ and hence by the triangular inequality we obtain

$$d \leqslant \mathsf{d_H}(\boldsymbol{c}, \boldsymbol{c}') \leqslant \mathsf{d_H}(\boldsymbol{y}, \boldsymbol{c}) + \mathsf{d_H}(\boldsymbol{y}, \boldsymbol{c}') \leqslant d - 1.$$

□

Error correction. We consider here the BSC(p) with crossover probability $0 < p < 1/2$. Given a $[n, k, d]$ linear code $\mathscr{C}$ over $\mathbb{F}_2$, let $\boldsymbol{c} \in \mathscr{C}$ be the transmitted codeword and $\boldsymbol{y} \in \mathbb{F}_2^n$ the received word. During the communication process the channel modified the codeword which means that some bits of $\boldsymbol{c}$ were flipped. The error vector $\boldsymbol{e} \in \mathbb{F}_2^n$ is such that $\boldsymbol{c} = \boldsymbol{y} - \boldsymbol{e}$. In this case the nearest codeword problem can be reformulated as:

Definition 2.2.16 (Nearest Codeword Problem for BSC)**.**

Given: *$[n, k, d]$ linear code $\mathscr{C}$ over $\mathbb{F}_2$ and a word $\boldsymbol{y} \in \mathbb{F}_2^n$.*
Find: *$\boldsymbol{e} \in \mathbb{F}_2^n$ of minimum Hamming weight such that $\boldsymbol{y} - \boldsymbol{e} \in \mathscr{C}$.*

There are several methods for the nearest codeword decoder but we remind here only one, namely the syndrome decoding. First we define the syndrome of a vector with respect to a parity-check matrix

Definition 2.2.17. *Let $\mathscr{C}$ be a $[n,k,d]$ linear code over $\mathbb{F}_2$ and $\boldsymbol{H}$ be a $(n-k)\times n$ parity-check matrix for $\mathscr{C}$. Then the syndrome of a word $\boldsymbol{y}\in\mathbb{F}_2^n$ with respect to $\boldsymbol{H}$ is the the vector $\boldsymbol{s}\in\mathbb{F}_2^{n-k}$ defined as:*

$$\boldsymbol{s}\stackrel{def}{=}\boldsymbol{H}\boldsymbol{y}^T.$$

Since for every vector $\boldsymbol{c}\in\mathbb{F}_2^n$ we have that $\boldsymbol{c}\in\mathscr{C}\Leftrightarrow\boldsymbol{H}\boldsymbol{c}^T=\boldsymbol{0}_{\boldsymbol{n-k}}$ we can reformulate the nearest codeword decoder in an equivalent manner, more exactly

Given: a $[n,k,d]$ linear code over $\mathbb{F}_2$ with a $(n-k)\times n$ parity-check matrix $\boldsymbol{H}$ and a received word $\boldsymbol{y}\in\mathbb{F}_2^n$

Find: a word $\boldsymbol{e}\in\mathbb{F}_2^n$ of minimum Hamming weight such that $\boldsymbol{s}=\boldsymbol{H}\boldsymbol{e}^T$, where $\boldsymbol{s}$ is the syndrome of $\boldsymbol{y}$ with respect to $\boldsymbol{H}$.

Remark 2.2.18. *The decision problem it is known in the literature as the Syndrome Decoding Problem (SDP) and it is NP-complete [BMvT78]. The SDP can be formalized as follow*

Instance: *A binary $(n-k)\times n$ matrix $\boldsymbol{H}$, a vector $\boldsymbol{s}\in\mathbb{F}_2^{n-k}$ and an integer $\omega>0$.*

Question: *Is there a vector $\boldsymbol{x}\in\mathbb{F}_2^n$ of weight $\leqslant\omega$, such that $\boldsymbol{H}\boldsymbol{x}^T=\boldsymbol{s}$?*

Operations on codes

- *Basic constructions*

  The intersection between a code $\mathscr{C}$ and its dual is called the *Hull* of $\mathscr{C}$

  $$\mathcal{H}(\mathscr{C})=\mathscr{C}\cap\mathscr{C}^{\perp}.$$

  A code $\mathscr{C}$ with dimension $k\leqslant n/2$ is called *weakly self-dual* if $\mathcal{H}(\mathscr{C})=\mathscr{C}\subset\mathscr{C}^{\perp}$ and *self-dual* if $\mathcal{H}(\mathscr{C})=\mathscr{C}=\mathscr{C}^{\perp}$ (in this case $k=n/2$).

- *Shortened and Punctured codes*

  For a given code $\mathscr{C}$ and a subset $\mathcal{J}\subseteq\{0,\ldots,n-1\}$ the *punctured* code $\mathcal{P}_{\mathcal{J}}(\mathscr{C})$ and *shortened* code $\mathcal{S}_{\mathcal{J}}(\mathscr{C})$ are defined as:

  $$\mathcal{P}_{\mathcal{J}}(\mathscr{C})=\left\{(c_i)_{i\notin\mathcal{J}}\mid\boldsymbol{c}\in\mathscr{C}\right\};$$
  $$\mathcal{S}_{\mathcal{J}}(\mathscr{C})=\left\{(c_i)_{i\notin\mathcal{J}}\mid\exists\boldsymbol{c}=(c_i)_i\in\mathscr{C}\text{ such that }\forall i\in\mathcal{J},\ c_i=0\right\}.$$

  Instead of writing $\mathcal{P}_{\{j\}}(\mathscr{C})$ and $\mathcal{S}_{\{j\}}(\mathscr{C})$ when $\mathcal{J}=\{j\}$ we rather use the notation $\mathcal{P}_j(\mathscr{C})$ and $\mathcal{S}_j(\mathscr{C})$.

  Theorem 2.2.19. *Let $\mathscr{C}$ be a $[n,k,d]$ binary linear code and $\mathcal{J}\subseteq\{0,\ldots,n-1\}$ a non empty set of coordinates. Then we have*

  $$\mathcal{S}_{\mathcal{J}}\left(\mathscr{C}^{\perp}\right)=\left(\mathcal{P}_{\mathcal{J}}(\mathscr{C})\right)^{\perp}$$

  A direct consequence of Theorem 2.2.19 is the following equality

  $$\left(\mathcal{S}_{\mathcal{J}}(\mathscr{C})\right)^{\perp}=\mathcal{P}_{\mathcal{J}}\left(\mathscr{C}^{\perp}\right).$$

Example 2.2.20. *Let $\mathscr{C}$ be a $[6,3,2]$ binary linear code defined by the generator matrix* $\begin{pmatrix}1&0&1&1&0&0\\0&1&0&1&0&1\\0&0&0&0&1&0\end{pmatrix}$ *and $\mathcal{I}=\{1,2\}$ be a set of positions. Then we have*

- $\mathcal{S}_{\mathcal{I}}(\mathscr{C})$ *is a $[4,1,1]$ binary linear code with generator matrix* $\begin{pmatrix}0&0&1&0\end{pmatrix}$
- $\mathcal{P}_{\mathcal{I}}(\mathscr{C})$ *is a $[4,3,1]$ binary linear code with generator matrix* $\begin{pmatrix}1&0&0&1\\0&1&0&1\\0&0&1&0\end{pmatrix}$

We also have that $\mathscr{C}^{\perp}$ is a $[6,3,2]$ binary linear code with generator matrix $\begin{pmatrix}1&0&0&1&0&1\\0&1&0&0&0&1\\0&0&1&1&0&1\end{pmatrix}$. *Then we obtain*

- $\mathcal{S}_{\mathcal{I}}\left(\mathscr{C}^{\perp}\right)$ *is a $[4,1,3]$ binary linear code with generator matrix* $\begin{pmatrix}1&1&0&1\end{pmatrix}$
- $\mathcal{P}_{\mathcal{I}}\left(\mathscr{C}^{\perp}\right)$ *is a $[4,3,1]$ binary linear code with generator matrix* $\begin{pmatrix}1&0&0&0\\0&1&0&0\\0&0&0&1\end{pmatrix}$

- *Plotkin sum*

Definition 2.2.21. *For $i\in\{1,2\}$ let $\mathscr{C}_i$ be a $[n,k_i,d_i]$ binary linear code. Then the Plotkin sum of $\mathscr{C}_1$ and $\mathscr{C}_2$ is the binary linear code $[2n,k_1+k_2,\min(2d_1,d_2)]$ denoted $\mathscr{C}$*

$$\mathscr{C}\stackrel{def}{=}\{(\boldsymbol{c}_1|\boldsymbol{c}_1+\boldsymbol{c}_2)\mid \boldsymbol{c}_i\in\mathscr{C}_i\quad i\in\{1,2\}\}.$$

If $\mathscr{C}_i$ has generator matrix $\boldsymbol{G}_i$ and parity check-matrix $\boldsymbol{H}_i$ for $i\in\{1,2\}$ then a generator matrix and a parity check matrix for $\mathscr{C}$ are the two following block matrices

$$\begin{pmatrix}\boldsymbol{G}_1&\boldsymbol{G}_1\\\boldsymbol{0}&\boldsymbol{G}_2\end{pmatrix}\text{ and }\begin{pmatrix}\boldsymbol{H}_1&\boldsymbol{0}\\-\boldsymbol{H}_2&\boldsymbol{H}_2\end{pmatrix}.$$

- *Star product of codes.*

Using the component-wise product of binary vectors in $\mathbb{F}_2^n$ (see Definition 2.2.8) we define the star product of two codes

Definition 2.2.22 (Star product code)**.** *For $i\in\{1,2\}$ let $\mathscr{C}_i$ be a $[n,k_i,d_i]$ binary linear code. Then the star product code of $\mathscr{C}_1$ and $\mathscr{C}_2$ is the binary linear code denoted $\mathscr{C}_1\star\mathscr{C}_2$ defined as*

$$\mathscr{C}_1\star\mathscr{C}_2\stackrel{def}{=}\mathrm{Span}_{\mathbb{F}_2}\{\boldsymbol{c}_1\star\boldsymbol{c}_2\mid \boldsymbol{c}_1\in\mathscr{C}_1\text{ and }\boldsymbol{c}_2\in\mathscr{C}_2\}.$$

Proposition 2.2.23. *Let $\mathscr{C}_1$ be a $[n,k_1,d_1]$ binary linear code and $\mathscr{C}_2$ be a $[n,k_2,d_2]$ binary linear code. Then we have*

$$\mathsf{d}_{\min}(\mathscr{C}_1\star\mathscr{C}_2)\leqslant\min\{d_1,d_2\}.$$

$$\dim(\mathscr{C}_1 \star \mathscr{C}_2) \leqslant \min\left\{n, k_1k_2 - \binom{\dim(\mathscr{C}_1 \cap \mathscr{C}_2)}{2}\right\}.$$

When the two codes are equal we will rather denote $\mathscr{C} \star \mathscr{C}$ by $\mathscr{C}^2$ and call it the square code of $\mathscr{C}$. If $\mathscr{C}$ is a $[n, k, d]$ binary linear code then $\mathscr{C}^2$ is a binary linear code of length n, minimum distance $\mathsf{d}_{\min}(\mathscr{C}^2) \leqslant d$ and dimension $\dim(\mathscr{C}^2) \leqslant \min\left\{n, \binom{k+1}{2}\right\}$.

Example 2.2.24. *Let $\mathscr{C}$ be a $[6, 2, 2]$ binary linear code defined by the generator matrix $\begin{pmatrix} 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \end{pmatrix}$. Then the square code $\mathscr{C}^2$ is a $[6, 3, 1]$ binary linear code defined by the generator matrix $\begin{pmatrix} 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$. We remark that here the dimension of $\mathscr{C}^2$ equals $\binom{2+1}{2} = 3$.*

Automorphism group of binary linear codes

Definition 2.2.25. *Any mapping $\tau : \mathbb{F}_2^n \to \mathbb{F}_2^n$ that preserves the Hamming distance is called an* isometry. *An* automorphism *of a linear code $\mathscr{C}$ is an isometry which maps $\mathscr{C}$ onto itself.*

There are two types of isometries known in the literature, for linear codes, namely linear and semi-linear isometries [Huf98]. In the case of binary fields we have the following result: the group of linear isometries of the Hamming space of dimension n over $\mathbb{F}_2$ is isomorphic to the symmetric group $\mathfrak{S}_n$ [SS13]. So we can define the permutation group of a binary linear code as the group of permutations of positions in the support of the code, that leave the code globally invariant:

Definition 2.2.26 (Permutation group of a code)**.** *Let $\mathscr{C}$ be a $[n, k, d]$ binary linear code and $\pi \in \mathfrak{S}_n$. We denote by $\boldsymbol{x}^\pi = (x_{\pi^{-1}(i)})_{1 \leqslant i \leqslant n}$ the vector $\boldsymbol{x}$ permuted by π. $\mathscr{C}^\pi = \{\boldsymbol{c}^\pi \mid \boldsymbol{c} \in \mathscr{C}\}$ denotes the permuted code of $\mathscr{C}$. Then the permutation group of a code is*

$$\mathsf{Perm}(\mathscr{C}) \stackrel{def}{=} \{\pi \in \mathfrak{S}_n \mid \mathscr{C}^\pi = \mathscr{C}\}$$

When $\mathsf{Perm}(\mathscr{C})$ is reduced to only one element then we speak of "trivial" group. Despite the fact that we know the permutation group of some code families, in general it is not an easy task to compute it. In Appendix A we give some relations between the permutation group of a code and its dual or its Hull.

2.2.3 Code families

We propose here to recall and give the main properties of some of the code families that are used in public key cryptography. We emphasize that during the thesis other code families were studied but we prefer to detail each one at adequate moment.

Reed-Muller codes. The Reed-Muller codes were introduced by David Muller [Mul54] and rediscovered shortly after with an efficient decoding algorithm by Irving Reed [Ree54].[1]

[1] Although it seems that these codes were firstly discovered by Mitani in 1951 [Mit51], they became popular only after the article of Muller and Reed.

The scientific community was highly interested in this family of codes and therefore discovered many structural properties of Reed-Muller codes citeMS86.

Definition 2.2.27 (Reed-Muller codes)**.** *Let m and r be two integers such that $1 \leqslant r \leqslant m$ and let $n \stackrel{def}{=} 2^m$. Then the r^{th} order Reed-Muller code $\mathscr{R}(r,m)$ is the binary linear code defined as the set of all vectors $(g(v_1,\ldots,v_m))_{(v_1,\ldots,v_m)\in\mathbb{F}_2^m} \in \mathbb{F}_2^n$, where $g \in \mathbb{F}_2[x_1,\ldots,x_m]$ ranges over the set of polynomials over $\mathbb{F}_2$ in m variables with degree at most r.*

$$\mathscr{R}(r,m) \stackrel{def}{=} \Big\{ (g(v_1,\ldots,v_m))_{(v_1,\ldots,v_m)\in\mathbb{F}_2^m} \;\mid\; \deg g \leqslant r \Big\}.$$

Properties 2.2.28 (Chapter 13, [MS86])**.** *The $\mathscr{R}(r,m)$ code has the following properties:*

1. *The $\mathscr{R}(r,m)$ code is a $\left[n, \sum_{i=0}^{r}\binom{m}{r}, 2^{m-r}\right]$ binary linear code.*
2. *The dual of a Reed-Muller code is a Reed-Muller code, namely*

$$\mathscr{R}(r,m)^{\perp} = \mathscr{R}(m-r-1,m).$$

3. *The number of minimum weight codewords equals [KT70]*

$$\mathrm{W}_{\min} = 2^r \begin{bmatrix} m \\ r \end{bmatrix}_2,$$

where $\begin{bmatrix} m \\ r \end{bmatrix}_2 \stackrel{def}{=} \prod_{i=0}^{m-r-1} \frac{2^{m-i}-1}{2^{m-r-i}-1}$ is the 2-analog of the Binomial coefficient also known as the Gaussian coefficient.

Definition 2.2.29. *The set of affine transformations over $\mathbb{F}_2^m$ of the form $\boldsymbol{x} \mapsto \boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}$ where $\boldsymbol{A} \in \mathbb{F}_2^{m\times m}$ is an invertible binary matrix and $\boldsymbol{b} \in \mathbb{F}_2^m$ forms a group called the* general affine group $\mathrm{GA}(m)$.

Proposition 2.2.30. *The permutation group of binary Reed-Muller codes for $1 \leqslant r \leqslant m-2$ is the general affine group*

$$\mathsf{Perm}\left(\mathscr{R}(r,m)\right) = \mathrm{GA}(m).$$

Generalized Reed-Solomon and Goppa codes. Generalized Reed-Solomon codes, or shortly GRS codes, were introduced by Reed and Solomon in [ISR60] and represent a powerful family of codes with many applications. Ten years after, a new class of codes, binary Goppa codes, was introduced by Valery Goppa [Gop70]. The main reason we detail Goppa codes in the same paragraph with GRS codes is because Goppa codes can be defined as subfield subcodes of GRS codes.

Definition 2.2.31 (Generalized Reed-Solomon codes)**.** *Let k and n be two integers such that $1 \leqslant k < n \leqslant q$ where $q = p^m$ is a power of a prime number p. Let $(\boldsymbol{x},\boldsymbol{y}) \in \mathbb{F}_q^n \times \mathbb{F}_q^n$ be a pair such that $\boldsymbol{x}$ is an n-tuple of distinct elements of $\mathbb{F}_q$ and the elements y_i are nonzero elements in $\mathbb{F}_q$. Then the Generalized Reed-Solomon code is*

$$\mathbf{GRS}_k(\boldsymbol{x},\boldsymbol{y}) \stackrel{def}{=} \{(y_1 f(x_1),\ldots,y_n f(x_n) \mid f \in \mathbb{F}_q[x]\ ,\ \deg(f) < k\}\ .$$

The vector $\boldsymbol{x}$ is called the support of the code and $\boldsymbol{y}$ the multiplier vector. GRS codes are MDS since the minimum distance is $d = n - k - 1$.

Generalized Reed-Solomon codes can be defined using the generator matrix, more exactly

$$\boldsymbol{G}_{\mathbf{GRS}_k(\boldsymbol{x},\boldsymbol{y})} = \begin{pmatrix} 1 & 1 & \dots & 1 \\ x_1 & x_2 & \dots & x_n \\ x_1^2 & x_2^2 & \dots & x_n^2 \\ \vdots & \vdots & \vdots & \vdots \\ x_1^{k-1} & x_2^{k-1} & \dots & x_n^{k-1} \end{pmatrix} \begin{pmatrix} y_1 & & & \\ & y_2 & & 0 \\ & & \ddots & \\ 0 & & & y_n \end{pmatrix}.$$

Proposition 2.2.32.

$$\mathbf{GRS}_k(\boldsymbol{x}, \boldsymbol{y})^{\perp} = \mathbf{GRS}_{n-k}(\boldsymbol{x}, \boldsymbol{z}),$$

where $\boldsymbol{z}$ is a non zero codeword of a $[n, 1, n]$ GRS code with the same code locators and column multipliers as $\mathbf{GRS}_k(\boldsymbol{x}, \boldsymbol{y})$, i.e. $\boldsymbol{H}_{\mathbf{GRS}_{n-1}(\boldsymbol{x},\boldsymbol{y})}\boldsymbol{z}^T = 0$.

We notice that the vector $\boldsymbol{z}$ with $\forall\ 1 \leqslant i \leqslant n\ , z_i \neq 0$ exists since any non zero codeword of a $[n, 1, n]$ GRS code has a Hamming weight equal to n.

Definition 2.2.33 (Alternant codes)**.** *Let $\mathscr{C} \subset \mathbb{F}_{p^m}^n$ be a linear code over $\mathbb{F}_{p^m}$. Then an alternant code is a linear code over $\mathbb{F}_p^n$ of length n dimension $k \geqslant n - mr$ and minimum distance $d \geqslant r + 1$ is defined by*

$$\mathbf{Alt}_r(\boldsymbol{x}, \boldsymbol{y}) \stackrel{def}{=} \mathbf{GRS}_r(\boldsymbol{x}, \boldsymbol{y})^{\perp} \cap \mathbb{F}_p^n.$$

Using Proposition 2.2.32 we obtain that $\mathbf{Alt}_r(\boldsymbol{x}, \boldsymbol{y})$ consists of all vectors $\boldsymbol{c}$ over $\mathbb{F}_p$ such that $\boldsymbol{H}_{\mathbf{GRS}_{n-r}(\boldsymbol{x},\boldsymbol{y})}\boldsymbol{c}^T = 0$.

Definition 2.2.34 (Binary Goppa codes)**.** *Let $\boldsymbol{x} \in \mathbb{F}_{2^m}^n$ be a $n-$tuple of distinct elements and $g \in \mathbb{F}_{2^m}[x]$ be a polynomial of degree t such that $\forall\ i, g(x_i) \neq 0$. Let $\boldsymbol{y} \stackrel{def}{=} (1/g(x_i), \dots, 1/g(x_n))$ then the binary Goppa code is defined by*

$$\mathbf{\Gamma}(\boldsymbol{x}, g) \stackrel{def}{=} \mathbf{Alt}_t(\boldsymbol{x}, \boldsymbol{y}).$$

2.3 The McEliece Public Key Encryption Scheme

2.3.1 Textbook McEliece

The McEliece public key encryption scheme [McE78] is composed of three algorithms: *key generation* (KeyGen), *encryption* (Encrypt) and *decryption* (Decrypt).

The first step is the key generation algorithm that takes as input the integers n, k, t such that $k < n$ and $t < n$ and outputs the public key/private key pair $(\mathsf{pk}, \mathsf{sk})$. In order to encrypt a message $\boldsymbol{m} \in \mathbb{F}_2^k$ one applies the $\mathsf{Encrypt}(,)$ function. The last step is the decryption function that takes as input a ciphertext $\boldsymbol{z}$ and the private key sk and outputs the corresponding message $\boldsymbol{m}$.

1. Pick a generator matrix $\boldsymbol{G}$ of a $[n, k]$ binary Goppa code $\boldsymbol{\Gamma}(\boldsymbol{x}, g)$ that can corrects t errors.
2. Pick at random a $k \times k$ invertible matrix $\boldsymbol{S}$ and a $n \times n$ permutation matrix $\boldsymbol{P}$.
3. Compute $\boldsymbol{G}_{\text{pub}} \stackrel{\text{def}}{=} \boldsymbol{SGP}$.
4. Return
$$\mathsf{pk} = (\boldsymbol{G}_{\text{pub}}, t) \text{ and } \mathsf{sk} = (\boldsymbol{S}, \boldsymbol{P}).$$

Figure 2.4 – The Key Generation function of the original McEliece scheme - $\mathsf{KeyGen}(n, k, t) = (\mathsf{pk}, \mathsf{sk})$

1. Generate a random error-vector $\boldsymbol{e} \in \mathbb{F}_2^n$ with $\|\boldsymbol{e}\| \leqslant t$
2. Return $\boldsymbol{z} = \boldsymbol{m}\boldsymbol{G}_{\text{pub}} \oplus \boldsymbol{e}$

Figure 2.5 – The Encryption function of the original McEliece scheme - $\mathsf{Encrypt}(\boldsymbol{m}, \mathsf{pk}) = \boldsymbol{z}$

1. Compute $\boldsymbol{z}^* = \boldsymbol{z}\boldsymbol{P}^{-1}$ and $\boldsymbol{m}^* = \mathcal{D}ecode(\boldsymbol{z}^*, \boldsymbol{H})$
2. Return $\boldsymbol{m}^*\boldsymbol{S}^{-1}$.

Figure 2.6 – The Decryption function of the original McEliece scheme - $\mathsf{Decrypt}(\boldsymbol{z}, \mathsf{sk}) = \boldsymbol{m}$

Here $\mathcal{D}ecode(.,.)$ is an efficient decoding algorithm for $\boldsymbol{\Gamma}(\boldsymbol{x}, g)$. Notice that multiplying the error vector by a permutation does not change the weight of the vector. One can easily verify the correctness of the scheme by checking

$$\mathsf{Decrypt}(\mathsf{Encrypt}(\boldsymbol{m}, \mathsf{pk}), \mathsf{sk}) = \boldsymbol{m}.$$

2.3.2 Security of the scheme

In the last part of his article McEliece brought up a discussion on the security of the scheme. The author proposed two types of attacks: firstly a key recovery attack in which he imagine an adversary that might recover the private generator matrix of the Goppa code and then use the Paterson algorithm to decode. Secondly he proposed a message recovery attack in which the codeword is retrieved from the ciphertext without the knowledge of the private key. In a sense McEliece established the major security threats the one has to consider when designing a code based cryptosystem, namely the Message Recovery Attack (MRA) and the Key Recovery Attack (KRA). He proposed as parameters for the scheme $n = 2^{10}$ and $t = 50$ as for the dimension of the code he picked $k = 524$. Nowadays, these parameters correspond to a 60 bits security level.

- **Message Recovery Attack**

  The message recovery attack aims to retrieve the message that was sent using only the ciphertext and the public key. In this case an adversary has to solve the Syndrome Decoding Problem (see 2.2.18).

The most efficient algorithm to solve the Syndrome Decoding Problem, for the moment, is the Information Set Decoding (ISD). The ISD algorithm searches for an information set such that the error positions are all out of the information set. Details about the different variants of ISD are given in [TS16]. Almost all McEliece variants base their security on the assumption that the public code is indistinguishable from a random linear code. Therefore the security level against the MRA attacks is based on the complexity of the ISD, namely $O(e^{-\omega \log(1-R)(1+o(1))})$ when $\omega = o(n)$ [TS16].

- **Key Recovery Attack**

  The key recovery adversary aims to retrieve the private key of a McEliece type cryptosystem given the public key. If the attacker manages to efficiently recover the private key, then he can also decode and find all the messages that have been encrypted with that key. Therefore it is considered as the most powerful possible attack. In the KRA scenario the adversary can be reduced to solve the code equivalence problem.

Code Equivalence Problem. Since we consider only binary linear codes we will define the *Permutation Equivalence problem* (see [SS13]).

Definition 2.3.1 (Permutation Code Equivalence Problem)**.** *Let $\boldsymbol{G}$ and $\boldsymbol{G}^*$ be the generating matrices for two $[n, k]$ binary linear codes. Given $\boldsymbol{G}$ and $\boldsymbol{G}^*$ does there exist a $k \times k$ binary invertible matrix $\boldsymbol{S}$ and $n \times n$ permutation matrix $\boldsymbol{P}$ such that $\boldsymbol{G}^* = \boldsymbol{SGP}$?*

The computational problem was studied by Petrank and Roth over the binary field [PR97]. The authors proved that this problem is harder than the *Graph Isomorphism problem* but that the Code Equivalence problem is not NP-complete. One solution to this problem, that is employed by the MAGMA software, is to use Leon's algorithm that searches for minimum weight codewords. Since the complexity of Leon's algorithm is exponential in the weight of the codewords that are searched, it is not efficient in the case of random codes. So another solution to this problem is the Support Splitting Algorithm (SSA) [Sen00]. This algorithm computes the weight enumerator polynomial of the hull. It has *heuristic* time complexity for random $[n, k]$ linear codes equal to $O(n^3 + 2^h n^2 \log n)$, where h is the dimension of the hull [SS13]. This algorithm is very efficient in general but cannot be used in the case of codes with large Hulls or codes with large Permutation group.

Distinguisher Attack The distinguisher problem is a recent issue that had a major impact on the code-based cryptography. The question that we raise here is:

Can a linear code be distinguished from a random linear code using an efficient deterministic algorithm ?

The article [FGO+10] was a breakthrough in this field since it proposed a deterministic polynomial-time distinguisher for high rate Alternant codes, that is the dimension of the square code [2]. It allowed to distinguish many algebraic codes from random codes, like

[2]For the square code we refer here to the square star product, for more details see Definition 2.2.22

Reed-Solomon codes, Goppa codes, Reed-Muller codes etc. Ever since the original article on the distinguisher was published, many successful cryptanalysis used the star product, attacks that we recall them in Section 2.3.3.

2.3.3 McEliece PKC variants

Binary irreducible Goppa codes were proposed in the original paper of McEliece [McE78]. Even though the original parameters were broken by Bernstein, Lange and Peters in [BLP08], they proposed a new set of parameters (see Figure 2.7). Despite their well known structure for the moment there are no efficient key recovery or decoding attacks against binary irreducible Goppa codes. A **distinguisher** exists in the case of high rate Goppa codes [FGO+10]. But despite of this potential vulnerability there is no efficient algorithm for the moment exploiting the knowledge and the properties of the distinguisher.

The existence of **weak keys** for Goppa codes was raised by Sendrier and Loidreau in [LS01]. They managed to distinguish the Goppa codes generated by binary irreducible polynomials. The number of such codes is exponentially smaller than the number of all Goppa codes, more exactly the probability of choosing a weak key is approximately $2^{-(m-1)t}$.

Security level(-bit)	$[n, k]$	t	Public Key size (bits)
80	$[1632, 1269]$	33	460647
128	$[2960, 2288]$	56	1537536
256	$[6624, 5129]$	115	7667855

Figure 2.7 – Parameters for McEliece with Goppa codes from [BLP08]

We also mention the existence of a compact variant of the McEliece scheme based on quasi-dyadic Goppa codes due to Misoczki and Barreto [MB09], variant that is not yet broken in the binary case.

Generalized Reed-Solomon codes were proposed for the first time by Niederreiter in [Nie86] but turned out to be an insecure solution. Indeed, six years after the article was published, Sidelnikov and Shestakov proposed a polynomial time attack against this variant [SS92]. Nevertheless the idea of using GRS codes was reconsidered more than ten years after by Berger and Loidreau when they proposed to consider subcodes of GRS codes [BL05]. Unfortunately this technique was also attacked in two steps by Wieschebrink [Wie06a, Wie09], using the **square code structure**.

Other attempts to repair the Niederreiter variant were proposed by Wieschebrink [Wie06b] who's idea was to add random column to the generator matrix. Baldi, Bianchi, Chiaralice, Rosenthal and Schipani [BBC+16] proposed to use Wieschebrink's idea but in addition they suggest to change the permutation matrix. But many of these variants turned out to be extremely unsecured against **square code type attacks** or **filtration type attacks** [CGG+14].

Reed-Muller codes were proposed by Sidelnikov's in [Sid94]. This variant was firstly attacked by Minder and Shokrollahi [MS07]. In the case of Reed-Muller codes the Key

Recovery Attack is reduced to solving the code equivalence problem since there is only one $\mathscr{R}(r,m)$. Minder and Shokrollahi managed to solve this problem using a filtration type attack based on the structure properties of the minimum weight codewords. The complexity of their algorithm was dominated by the minimum weight codewords searching algorithm.

Chizhov and Borodin proposed another attack. Their algorithm could solve the code equivalence problem, for some of the parameters of the Reed-Muller codes, in polynomial time [CB13, CB14].

A modified version [GM13] using the masking technique introduced by Wieschebrink was recently broken by Otmani and Kalachi using a **square code** type attack [OK15].

Algebraic-geometry codes was suggested by Janwa and Moreno in 1996 [JM96]. Several articles discussed the potential vulnerabilities of this variant and proposed algorithms that could be deployed to attack in some particular cases [FM08, SS92]. Nevertheless they can not be generalized and suffer in terms of efficiency. In [CMCP14] Couvreur, Marquez-Corbella and Pellikaan proposed a polynomial type algorithm that works on codes from curves of arbitrary genus. They managed to recover an error-correcting pair from the square code using a **filtration type attack** and the structure of the **square code**.

Concatenated codes were the first family of probabilistic codes analyzed from a cryptographic point of view. Sendrier detailed in [Sen94, Sen98] the main vulnerabilities of ordinary concatenated codes.

LDPC codes presented a disputed class of codes in cryptography. In the book of Baldi [Bal14] all the details about the thrilling combats defeating and attacking the LDPC codes are given. Monico, Rosenthal and Shokrollahi were the first ones to propose and analyze a McEliece variant using low density parity check codes in [MRAS00]. Partially inspired by the idea of Gaborit to consider quasi-cyclic codes [Gab05] [3] the new QC-LDPC cryptosystem was presented by Baldi and Chiaraluce in [BC07]. Both BCH codes and LDPC codes with the quasi-cyclic structure were successfully cryptanalyzed by Otmani, Tillich and Dallot [OTD08]. In order to prevent the scheme to the last attack a modification that aims to increase the weight of the codewords in the dual of the LDPC code was proposed in [BBC08]. This modification seems to be working for the moment since no other structural attacks were discovered.

Wild Goppa codes was a natural extension from binary Goppa codes to non binary fields. It was proposed by Bernstein, Lange and Peters in [BLP10] and [BLP11]. Many of the proposed parameters, namely those for which the extension degree was equal to 2, were broken by Couvreur, Otmani and Tillich using **filtration type techniques** [COT14a, COT14b].

Srivastava codes were proposed in [Per12]. The author is using Quasi-Dyadic Srivastava codes and gives another application of these types of codes, namely for signature schemes. Even though the parameters for the signature were broken in [FOP+14], the parameters for the encryption scheme are still valid.

[3] In [Gab05] the author proposes BCH codes with the quasi-cyclic structure. The idea of adding the quasi cyclic structure became one of the main techniques for reducing the key size in the McEliece scheme.

Convolutional codes represented among the shortest term solutions since between the proposal article by Londahl and Johansson [LJ12] and an efficient attack by Landais and Tillich [LT13] only one year passed.

2.4 Conclusion

Many of the McEliece variants were successfully cryptanalyzed, mainly because of their algebraic structure. Therefore probabilistic codes were proposed as possible substitutions for algebraic codes. We propose to analyze the security of two of the latest variants, namely the MDPC and Polar code variants.

Polar codes were firstly proposed by Shrestha and Kim [SK14]. A second proposal, using subcodes of Polar codes was given in [HSEA14]. In Chapter 4 we analyze the security of the first variant and propose an attack based on the structure of the minimum weight codewords.

MDPC codes are probably the most suitable codes for the moment in a McEliece type scheme [MTSB13]. Many cryptographic arguments are in favor of using this family of codes like efficiency, small key size when used with a quasi-cyclic structure and the most important to our opinion the lack of algebraic structure. We analyze the security of the QC-MDPC McEliece in Chapter 5.

3 Decreasing Monomial Codes

3.1 Introduction

Decreasing Monomial codes are a family of algebraic codes that contains two well known class of codes: Polar and Reed-Muller codes. We introduced for the first time this family of codes during the cryptanalysis of a public key encryption scheme à la McEliece using Polar codes [SK14, HSEA14]. At that moment few things were known about the structure of Polar codes, even though they were studied since 2007, when they were introduced by Arıkan [Arı09]. The main results concerning Polar codes were about the performance, construction and decoding capacity.

Contributions Our first major contribution is to propose a partial order $\preceq$, on the set of monomials in the polynomial ring $\mathbb{F}_2[x_0, x_1, \ldots, x_{m-1}]/(x_0^2 - x_0, \ldots, x_{m-1}^2 - x_{m-1})$, which is used to redefine Polar codes as vector spaces spanned by the evaluation of monomials that belong to a decreasing monomial subset I.

We define three families of codes ordered by inclusion and analyze the structure of these codes: Decreasing Monomial codes ($\preceq$) are included in the class of Weakly Decreasing Monomial codes ($\preceq_{\mathrm{w}}$), which are included in the class of Monomial codes. We focus our interest on the class of Decreasing Monomial codes and try to answer several fundamental questions like: the minimum distance, the permutation group, the duality properties and many more.

Our first major contribution is to prove that Reed-Muller codes and Polar codes are Decreasing Monomial codes. This result represent a new step into classifying "universal" Polar codes for the Binary Discrete Memoryless Channels, which was one of the most challenging problems proposed in [Arı09].

Among the most relevant properties of Decreasing Monomial codes that we reveal in this chapter we enumerate: the dual of a Decreasing Monomial code is still a Decreasing Monomial code, the permutation group of Decreasing Monomial codes contains the *Lower Triangular Affine Group*, which is a subgroup of the General Affine group. We also raise another non-trivial question regarding Decreasing Monomial codes, that is the number of minimum weight codewords. For that we give a counting technique for the number of minimum weight codewords based on Young diagrams and hence propose a closed formula. Several opened questions remain about Decreasing Monomial codes, fact that we point

out at the end of the chapter. Some of the results in this Chapter were published in [BDOT16].

3.2 Monomial Codes

3.2.1 Definitions and Properties

Throughout this chapter we will use the following conventions and notations
Notation 3.2.1.

- *The ambient space is the polynomial ring*

$$\mathbb{R}_m = \mathbb{F}_2[x_0, x_1, \ldots, x_{m-1}]/(x_0^2 - x_0, \ldots, x_{m-1}^2 - x_{m-1}).$$

- *We set* $n = 2^m$.
- *Any binary vector* $\boldsymbol{u} \in \mathbb{F}_2^m$ *is denoted by* $\boldsymbol{u} = (u_0, \ldots, u_{m-1})$, *where* u_{m-1} *is the most significant bit.*

Furthermore we will define an order on the elements of $\mathbb{F}_2^m$. A natural order is to associate to any element $\boldsymbol{u} \in \mathbb{F}_2^m$ the integer $u \in \mathbb{Z}$ defined by $u = \sum_{i=0}^{m-1} u_i 2^i$, and then to use the natural order on the integers. Notice that that value u is computed regardless of the fact that $u_i \in \mathbb{F}_2$.

This order it is known in the literature as the index order and it is already used by Arıkan in [Arı09]. We remark that the index order is equivalent to the lexicographic order induced by $0 < 1$, over the sequences $u_{m-1} \ldots u_0$. In other words, given $\boldsymbol{u}$ and $\boldsymbol{v}$ we say that $\boldsymbol{u}$ is smaller than or equal to $\boldsymbol{v}$ if $u_{m-1} \ldots u_0 \preceq_{\text{lex}} v_{m-1} \ldots v_0$. The following example illustrates our proposal

Example 3.2.2. *Let* $\boldsymbol{v} = (0, 1, 0, 1)$ *and* $\boldsymbol{u} = (1, 0, 0, 1)$. *We have that* $\boldsymbol{u} \preceq_{\text{lex}} \boldsymbol{v}$ *since* $1001 \preceq_{\text{lex}} 1010$.

On the other hand $v = 2 + 8 = 10$ *and* $u = 1 + 8 = 9$ *and by the index order we obtain* $\boldsymbol{u} < \boldsymbol{v}$.

We choose here to write the elements in $\mathbb{F}_2^m$ in decreasing index order, which means that we begin with the greatest element $(1, \ldots, 1)$ and end up with the smallest element $(0, \ldots, 0)$.

Example 3.2.3. *For* $m = 2$ *we have*

$$\mathbb{F}_2^2 = [(1, 1), (0, 1), (1, 0), (0, 0)].$$

Next we will define the evaluation function that associate to a polynomial $g \in \mathbb{R}_m$ the binary vector denoted by $\mathsf{ev}(g)$ in $\mathbb{F}_2^n$.

Definition 3.2.4. *Let* $g \in \mathbb{R}_m$ *and order the elements in* $\mathbb{F}_2^m$ *with respect to the decreasing index order. Define the evaluation function*

$$\begin{array}{ccc} \mathbb{R}_m & \to & \mathbb{F}_2^n \\ g & \mapsto & \mathsf{ev}(g) = \Big(g(\boldsymbol{u})\Big)_{\boldsymbol{u} \in \mathbb{F}_2^m} \end{array}$$

Lemma 3.2.5. *[Car10a] The function* ev *is a bijection.*

Corollary 3.2.6. *The function* ev *defines a vector space isomorphism between the vector space* $(\mathbb{R}_m, +, \cdot)$ *and* $(\mathbb{F}_2^n, +, \cdot)$.

Example 3.2.7. *For* $m = 3$ *and* $g = x_0x_1 + x_0$ *we have*

		111	*011*	*101*	*001*	*110*	*010*	*100*	*000*
$\mathsf{ev}(x_0x_1)$	=	*1*	*0*	*0*	*0*	*1*	*0*	*0*	*0*
$\mathsf{ev}(x_0)$	=	*1*	*0*	*1*	*0*	*1*	*0*	*1*	*0*
$\mathsf{ev}(g)$	=	*0*	*0*	*1*	*0*	*0*	*0*	*1*	*0*

Remark 3.2.8. *The polynomials in* $\mathbb{R}_m$ *are also known in the literature as a particular representation of Boolean function, that is the Algebraic Normal Form, or shortly ANF (see [Car10a, Car10b]). In [Car10a, Section 2.1] several properties are given, including those regarding the* ev *function. Since* ev *is bijective, an efficient algorithm to compute its inverse is given. This algorithm is called the Fast Mobius Transform and it is running in* $m2^m$ *bit operations.*

An important part in the formalism will be played by the monomials.

Notation 3.2.9. *We denote by* $\boldsymbol{x^i}$ *the monomial* $x_0^{i_0} \cdots x_{m-1}^{i_{m-1}}$, *where* $\boldsymbol{i} \in \mathbb{F}_2^m$. *We also denote the set of monomials*

$$\mathcal{M}_m \stackrel{def}{=} \left\{ \boldsymbol{x^i} \mid \boldsymbol{i} = (i_0, \ldots, i_{m-1}) \in \mathbb{F}_2^m \right\}.$$

For any monomial $g \in \mathcal{M}_m$ *of degree* $1 \leqslant s \leqslant m$ *we use the notation* $g = x_{l_1} \ldots x_{l_s}$ *where* $0 \leqslant l_1 < l_2 \cdots < l_s \leqslant m-1$. *We also denote the support of a monomial by* $\mathrm{ind}(g) = \{l_1 \ldots, l_s\}$.

Definition 3.2.10 (Polynomial and Monomial code)**.** *Let* $I \subseteq \mathbb{R}_m$ *be a finite set of polynomials in* m *variables. The linear code defined by* I *is the vector subspace* $\mathscr{C}(I) \subseteq \mathbb{F}_2^n$ *generated by* $\{\mathsf{ev}(f) \mid f \in I\}$.

- *When* $I \subseteq \mathbb{R}_m$ *we say that* $\mathscr{C}(I)$ *is a* polynomial code.
- *When* $I \subseteq \mathcal{M}_m$ *we say that* $\mathscr{C}(I)$ *is a* monomial code.

Proposition 3.2.11. *For all* $I \subseteq \mathcal{M}_m$ *the dimension of the monomial code* $\mathscr{C}(I)$ *is equal to* $|I|$.

Proof. This comes first of all from the linear independence of the monomials in $\mathbb{R}_m$ and second of all from the fact that ev is bijective (see Lemma 3.2.5). □

Remark 3.2.12. $\mathscr{R}(r, m)$ *is a monomial code with dimension* $k = \sum_{i=0}^{r} \binom{m}{i}$. *In order to demonstrate this fact we recall the definition of* $\mathscr{R}(r, m)$ *from 2.2.27, more exactly* $\mathscr{R}(r, m) \stackrel{def}{=} \left\{ \mathsf{ev}(g) \mid g \in \mathbb{R}_m, \deg g \leqslant r \right\}$.

The later remark stress out the importance of the degree of polynomials involved in a Reed-Muller code. We will see in Section 3.5 that the notion of degree it is also used to determine the minimum distance of a Monomial code. Therefore we introduce the following notation

Notation 3.2.13. *Let* $I \subseteq \mathcal{M}_m$. *We denote the subset of monomials of degree* r *in* I *by* $I_r = \{g \in I \mid \deg(g) = r\}$.

3.2.2 Construction of Monomial codes

In order to construct monomial codes we use Arikan's definition [Arı08b]. It is mainly based on the recursive Plotkin construction. Therefore we use this last construction to define the generator matrix of a monomial code. We will call this matrix the monomial evaluation basis. Let's begin by recalling the Kronecker product of two matrices.

Definition 3.2.14 (Kronecker product)**.** *Let* $\boldsymbol{A} = (a_{ij})_{\substack{1 \leqslant i \leqslant r_a \\ 1 \leqslant j \leqslant c_a}}$ *and* $\boldsymbol{B} = (b_{i,j})_{\substack{1 \leqslant i \leqslant r_b \\ 1 \leqslant j \leqslant c_b}}$ *be two matrices defined over the same field. Then the Kronecker product of* $\boldsymbol{A}$ *and* $\boldsymbol{B}$ *is the* $r_a r_b \times c_a c_b$ *matrix defined by:*

$$\boldsymbol{A} \otimes \boldsymbol{B} \stackrel{def}{=} \begin{pmatrix} a_{1,1}\boldsymbol{B} & \cdots & a_{1,c_a}\boldsymbol{B} \\ \vdots & b_{i,j}\boldsymbol{B} & \vdots \\ a_{r_a,1}\boldsymbol{B} & \cdots & a_{r_a,c_a}\boldsymbol{B} \end{pmatrix}.$$

Using the Kronecker product we define the following matrix with coefficients in $\mathbb{F}_2$

$$\boldsymbol{G}_m \stackrel{\text{def}}{=} \underbrace{\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} \otimes \cdots \otimes \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}}_{m \text{ times}}.$$

Example 3.2.15. *When* $m = 2$ *we have*

$$\boldsymbol{G}_2 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{pmatrix}.$$

Remark 3.2.16. *We notice that* $\boldsymbol{G}_m$ *is a basis for the vector space* $(\mathbb{F}_2^n, +, \times)$*. Indeed,* $\boldsymbol{G}_m$ *is a matrix with full rank, more exactly* $\operatorname{rank}(\boldsymbol{G}_m) = 2^m$*.*

In the next paragraph we will prove that each row of $\boldsymbol{G}_m$ is the evaluation of a monomial in $\mathcal{M}_m$ on all the elements $\boldsymbol{u}$ of $\mathbb{F}_2^m$, in decreasing index order. Since any monomial is defined as $\boldsymbol{x}^{\boldsymbol{i}}$ with $\boldsymbol{i} \in \mathbb{F}_2^m$ we choose the same order on the exponents $\boldsymbol{i}$ as for the evaluation elements $\boldsymbol{u}$. For example when $m = 2$ we evaluate the monomials in the in the following order: $[x_0x_1, x_1, x_0, 1]$.

Example 3.2.17. *For* $m = 2$ *we have*

		11	01	10	00
11	$\mathsf{ev}(x_0x_1)$	*1*	*0*	*0*	*0*
01	$\mathsf{ev}(x_1)$	*1*	*1*	*0*	*0*
10	$\mathsf{ev}(x_0)$	*1*	*0*	*1*	*0*
00	$\mathsf{ev}(1)$	*1*	*1*	*1*	*1*

Remark 3.2.18. *The function* ev *defines a one-to-one mapping between the rows of the monomial evaluation matrix (in the previous example* $\boldsymbol{G}_2$*, which is a generating matrix for the vector space* $(\mathbb{F}_2^4, +, \times)$*) and the monomials in* $\mathcal{M}_m$ *(in the example* $\mathcal{M}_2 = \{1, x_0, x_1, x_1x_0\}$ *which is a generating basis for* $(\mathbb{R}_2, +, \times)$*).*

Lemma 3.2.19. *$\boldsymbol{G}_m$ is the monomial evaluation basis for the vector space $\mathbb{F}_2^n$ and*

$$\boldsymbol{G}_m[i] = \mathsf{ev}(x_0^{i_0} \dots x_{m-1}^{i_{m-1}}).$$

Proof. We use induction on m. For $m = 1$ we have that

$$\begin{aligned}\mathsf{ev}(x_0) &= (1,0)\\ \mathsf{ev}(1) &= (1,1)\end{aligned}$$

which equals $\boldsymbol{G}_1$.

We suppose that our statement is true up to an integer m and prove it for $m+1$. By definition we have $\boldsymbol{G}_{m+1} = \begin{pmatrix} \boldsymbol{G}_m & \boldsymbol{0_{2^m}} \\ \boldsymbol{G}_m & \boldsymbol{G}_m \end{pmatrix}$.

From the order on the evaluation points we have that the first half of the columns is represented by the 2^m elements having their last position equal to 1, namely $(u_0, \dots, u_{m-1}, 1)$, and the second half by the elements $(u_0, \dots, u_{m-1}, 0)$ where $u_i \in \mathbb{F}_2$.

Using the same argument we have that the lower half of the rows is represented by the monomials involving only the first m variables, more exactly $x_0 \dots x_{m-1}, \dots, x_0, 1_m$. As for the upper half, the monomials here admit x_m as variable, namely the set of monomials is $\{x_0 \dots x_m, \dots, x_0 x_m, x_m\}$. Therefore the evaluation of the first rows block on the first columns block equals $\boldsymbol{G}_m$ since x_m equals one on this block. In the same sense on the second column block x_m equals zero so we obtain $\boldsymbol{0_{2^m}}$. As for the lower part the variable x_m is not involved in the evaluation, thus we obtain that the two blocks are equal to $\boldsymbol{G}_m$.

In conclusion $\boldsymbol{G}_{m+1}$ is the evaluation of all the monomials in $m+1$ variables over all the elements $(u_0, \dots, u_m) \in \mathbb{F}_2^{m+1}$. □

Straightforward we can define Monomial codes in an equivalent manner:

Proposition 3.2.20. *Let $\boldsymbol{G}_m$ be the generator matrix for the vector space $\mathbb{F}_2^n$. Then any row sub-matrix of $\boldsymbol{G}_m$ defines a linear code $\mathscr{C} \subseteq \mathbb{F}_2^n$, that we call monomial code.*

Notation 3.2.21. *Any row sub-matrix of $\boldsymbol{G}_m$ indexed by the monomials in the set I will be furthermore denoted by $\boldsymbol{G}_I$.*

3.2.3 Polar codes are Monomial codes

In order to prove that Polar codes are monomial codes we could just recall the definition given by Arıkan in [Arı09]. He pointed out that the generator matrices of both Reed-Muller and Polar codes are obtained by a suitable choice of the rows of $\boldsymbol{G}_m$.

Nevertheless, in order to have a better understanding on how the selecting rule works for Polar codes, we recall the construction technique that was originally used for this family of codes.

Polar codes

The Polar codes construction is based on the channel combining (Figure 3.1) and splitting technique (Figure 3.3). Arıkan used this two techniques in [Arı09], in order to construct

the circuit that we illustrate in Figure 3.2. This circuit will be used both for encoding and decoding of Polar codes over a specific channel. For this purpose we denote by W the memoryless channel for which the polar code is devised. Its input alphabet is binary and its output alphabet is denoted by $\mathcal{Y}$ and it is also assumed to be discrete. We assume that the channel is symmetric meaning that there exists a permutation π of $\mathcal{Y}$ which is also an involution ($\pi^{-1} = \pi$) and $W(y|1) = W(\pi(y)|0)$ for all $y \in \mathcal{Y}$.

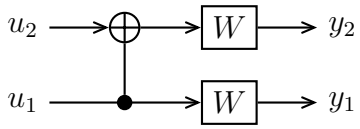


Figure 3.1 – The channel combining for $m = 1$

Duplicating the construction m times results in a complex circuit, denoted W_m, that we plot in Figure 3.2.

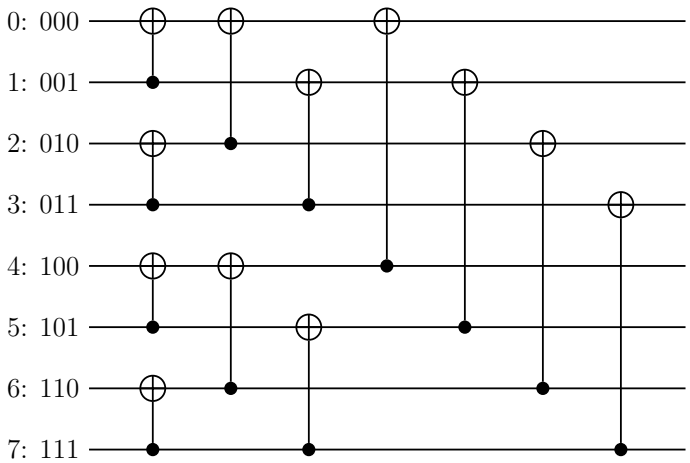


Figure 3.2 – The combined circuit for $m = 3$ and the binary expansion of each row. The index order is used for the row numbers (Arıkan's notation)

The resulting channel W_m can be defined as a vector channel, for example W_1 is defined by $W(y_1, y_2 \mid u_1, u_2) = W(y_1 \mid u_1)W(y_2 \mid u_1 \oplus u_2)$. Using the second technique the circuit will be split into 2^m synthetic channels. We define the Arıkan channel transforms W^+ and W^- of W which are both binary-input memoryless symmetric channel with transitions probabilities specified by

Definition 3.2.22 (Synthetic channels)**.**

$$W^-(y_1, y_2|u_2) \overset{def}{=} \frac{1}{2} \sum_{u_1 \in \mathbb{F}_2} W(y_1|u_1)W(y_2|u_1 \oplus u_2)$$
$$W^+(y_1, y_2, u_2|u_1) \overset{def}{=} \frac{1}{2} W(y_1|u_1)W(y_2|u_1 \oplus u_2)$$

Here the output alphabet of W^- is $\mathcal{Y} \times \mathcal{Y}$ whereas the output alphabet of W^+ is $\mathcal{Y} \times \mathcal{Y} \times \mathbb{F}_2$.

Remark 3.2.23. *Notice that the formula for* $W^+(y_1, y_2, u_2 \mid u_1)$ *can be deduced directly from the definition on* W_1 *using Bayes formula. Indeed,*

$$W^+(y_1, y_2, u_2 \mid u_1) = W(y_1, y_2 \mid u_1, u_2)\mathsf{prob}(u_2 \mid u_1)$$

which implies $W^+(y_1, y_2, u_2 \mid u_1) = \frac{1}{2}W(y_1, y_2 \mid u_1, u_2)$. *As for the second synthetic channel we have*

$$\begin{aligned} W^-(y_1, y_2|u_2) &= \sum_{u_1 \in \mathbb{F}_2} W(y_1, y_2, u_1 \mid u_2) \\ &= \sum_{u_1 \in \mathbb{F}_2} W(y_1, y_2 \mid u_1, u_2)\mathsf{prob}(u_1 \mid u_2) \\ &= \frac{1}{2} \sum_{u_1 \in \mathbb{F}_2} W(y_1|u_1)W(y_2|u_1 \oplus u_2). \end{aligned}$$

Figure 3.3 – The two synthetic channels
(left) $W^- : \mathbb{F}_2 \to \mathcal{Y} \times \mathcal{Y}$, (right) $W^+ : \mathbb{F}_2 \to \mathcal{Y} \times \mathcal{Y} \times \mathbb{F}_2$

In order to have a better understanding of the synthetic channels we detail in Appendix B how the successive cancellation decoder works. Notice that encoding (u_2, u_1) into $(u_2 \oplus u_1, u_1)$ can also be described as vector matrix multiplication

$$(u_2, u_1) \times \boldsymbol{G}_1 = (u_2 + u_1, u_1).$$

Therefore we can identify the channel W^- to W^{x_0} and $W^+ = W^1$. Straightforward we define the synthetic channel corresponding to a monomial

Definition 3.2.24. *Let* $g \in \mathcal{M}_m$. *Then the synthetic channel corresponding to* g *is*

$$W^g = W^{v_{m-1}\cdots v_0},$$

where $v_i = -$ *if* $x_i|g$ *and* $v_i = +$ *otherwise.*

Example 3.2.25. *For instance when* $m = 5$ *we have* $W^{x_0x_1x_3} = W^{+-+--}$. *Figure 3.4 plots the synthetic channels corresponding to each monomial for* $m = 3$.

We also need to define the Bhattacharyya parameter $\mathcal{B}(W)$ of a binary-input symmetric channel W

Definition 3.2.26. *Let* W *be a B-DMC with output alphabet* $\mathcal{Y}$. *Then the Bhattacharyya parameter of the channel* W *is*

$$\mathcal{B}(W) = \sum_{y \in \mathcal{Y}} \sqrt{W(y|0)W(y|1)} \tag{3.1}$$

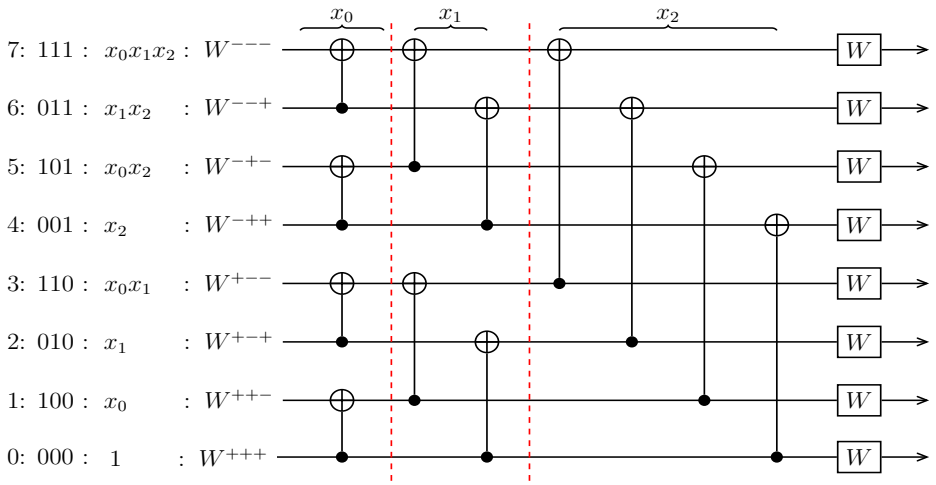


Figure 3.4 – The combined circuit for $m = 3$ with the monomial representation and the corresponding synthetic channels. The rows are in decreasing index order.

Example 3.2.27. *In the case of the Binary Erasure Channels and Binary Symmetric Channels we have*

- $$\mathcal{B}\left(\mathrm{BEC}(p)\right) = p.$$
- $$\mathcal{B}\left(\mathrm{BSC}(p)\right) = 2\sqrt{p(1-p)}.$$

Remark 3.2.28. *Let W be a* BEC(p) *then we have that $\mathcal{B}(W^-) = 2p - p^2$ and $\mathcal{B}(W^+) = p^2$. This fact can be proved directly either from the model of the synthetic channels in the case of the* BEC *family (see Exemple B.0.3) or by computing the Bhattacharyya parameter on W^- and W^+ using Defintion 3.2.26 and 3.2.22.*

With these definitions we can construct a polar code of length $n = 2^m$ and dimension k devised for a binary-input symmetric channel W.

Definition 3.2.29. *The polar code of length $n = 2^m$ and dimension k devised for the channel W is the monomial code $\mathscr{C}(I)$ where I is the set of k monomials in $\mathcal{M}_m$ which take the k smallest values $\mathcal{B}(W^g)$ among all g in $\mathcal{M}_m$.*

Note that the output alphabet size of the channels W^g is exponential in m which makes this ranking rather delicate. However there are various methods for computing these k "best" channels. The first method was proposed by Arıkan and uses the Monte Carlo simulation in order to estimate the Bhattacharyya parameters [Ari08a]. No more that one year after, he proposed to ignore the actual channel type that is used and compute the best channels in the case of a BEC [Ari09]. Another technique based of the factor graph is used by Mori and Tanaka in [MT09a]. Tal and Vardy proposed in [TV13] a *sandwich* method where each channels is caught between a degraded and an upgraded channel that are apparently very tight in terms of capacity.

3.3 Decreasing and Weakly Decreasing Monomial Codes

3.3.1 Definitions and Properties

Polar codes and Reed-Muller codes are both monomial codes. But the family of monomial codes is too large to explain the intriguing algebraic properties of polar codes (for instance their very large automorphism group). Therefore we will introduce a partial order over the set of monomials

Definition 3.3.1. *Let f and g be two monomials in $\mathcal{M}_m$.*

- *The $\preceq_{\mathrm{w}}$ order between f and g is defined as*

$$f \preceq_{\mathrm{w}} g \quad \textit{iff} \quad f|g.$$

- *The $\preceq$ order between f and g is defined as*
  - *when $\deg(f) = \deg(g) = s$ and $f = x_{i_1} \dots x_{i_s}$, $g = x_{j_1} \dots x_{j_s}$ we have*

$$f \preceq g \quad \textit{iff} \quad \forall\, 1 \leqslant \ell \leqslant s \quad i_\ell \leqslant j_\ell.$$

  - *when $\deg(f) < \deg(g)$ we have*

$$f \preceq g \quad \textit{iff} \quad \exists g^* \in \mathcal{M}_m \textit{ s.t. } f \preceq g^* \preceq_{\mathrm{w}} g.$$

Remark 3.3.2.

- *The two relations $\preceq_{\mathrm{w}}$ and $\preceq$ are well defined partial orders since they are reflexive, antisymmetric, and transitive.*
- *The order of divisibility was already used in the case of Polar codes but in a completely different context by Mori and Tanaka in [MT09b, Section VI]. In their case the purpose was to tighten the bounds of the error block probability of a Polar code designed for the* BEC *family.*
- *The order $\preceq_{\mathrm{w}}$ is weaker than $\preceq$ in the sense that any pair of monomials f, g that satisfy the relation $f \preceq_{\mathrm{w}} g$, also satisfy the relation $f \preceq g$, by definition. The inverse is not always true, take for example $f = x_0x_2$ and $g = x_1x_2$. By definition $f \preceq g$ but $f \not\preceq_{\mathrm{w}} g$.*
- *The constant polynomial 1 is the smallest element for both $\preceq$ and $\preceq_{\mathrm{w}}$. We also have that for $\preceq$ the variables are totally ordered*

$$x_0 \preceq x_1 \preceq \cdots \preceq x_{m-1}.$$

3.3.2 Weakly Decreasing Monomial Codes

Definition 3.3.3. *Let f and g be two monomials in $\mathcal{M}_m$ such that $f \preceq_{\mathrm{w}} g$ and $I \subset \mathcal{M}_m$.*

- *We define the closed interval $[f,g]_{\preceq_{\mathrm{w}}}$ with respect to the partial order $\preceq_{\mathrm{w}}$ as the set of monomials $h \in \mathcal{M}_m$ such that $f \preceq_{\mathrm{w}} h \preceq_{\mathrm{w}} g$.*
- *The set I is called a weakly decreasing set if and only if ($f \in I$ and $g \preceq_{\mathrm{w}} f$) implies $g \in I$.*
- *Let I weakly decreasing set. We define the subset of maximum monomials of I*

$$I_{\max_{\preceq_{\mathrm{w}}}} = \{f \in I \mid \nexists g \in I, g \neq f \ \ s.t. \ f \preceq_{\mathrm{w}} g\}.$$

We notice that an interval $[f,g]_{\preceq_{\mathrm{w}}}$ is composed of at least two elements f and g.

Remark 3.3.4. *Any weakly decreasing set is uniquely defined as a union of weakly decreasing intervals :*

$$I = \bigcup_{g \in I_{\max_{\preceq_{\mathrm{w}}}}} [1,g]_{\preceq_{\mathrm{w}}}.$$

We stress out that in general for a weakly monomial set I the set of maximum monomials with respect to $\preceq_{\mathrm{w}}$ order is different from the set of maximum degree monomials. Nevertheless the two sets might be equal and this is the case when $\forall g \in I_{\max_{\preceq_{\mathrm{w}}}}, \deg(g) = r$, where $0 \leqslant r \leqslant m$.

Definition 3.3.5. *Let $I \subset \mathcal{M}_m$ be a weakly decreasing set. Then the linear code $\mathscr{C}(I)$ is called weakly decreasing monomial code.*

Proposition 3.3.6. *Reed-Muller codes are weakly decreasing monomial codes*

$$\mathscr{R}(r,m) = \mathscr{C}\left(\bigcup_{\deg(g)=r} [1,g]_{\preceq_{\mathrm{w}}}\right).$$

If I represents the set of monomial defining the $\mathscr{R}(r,m)$ code, then we have equality between the subset of maximum degree monomials and the subset of maximum monomials for the order $\preceq_{\mathrm{w}}$: $I_{\max_{\preceq_{\mathrm{w}}}} = I_r$.

Proof. The result is a direct consequence of the definition of Reed-Muller codes 2.2.27 and the Definition 3.3.1 of $\preceq_{\mathrm{w}}$. □

3.3.3 Decreasing Monomial Codes

Definition 3.3.7. *Let f and g be two monomials in $\mathcal{M}_m$ such that $f \preceq g$ and $I \subset \mathcal{M}_m$.*

- *We define the closed interval $[f,g]_{\preceq}$ with respect to the partial order $\preceq$ as the set of monomials $h \in \mathcal{M}_m$ such that $f \preceq h \preceq g$.*
- *The set I is called a decreasing set if and only if ($f \in I$ and $g \preceq f$) implies $g \in I$.*

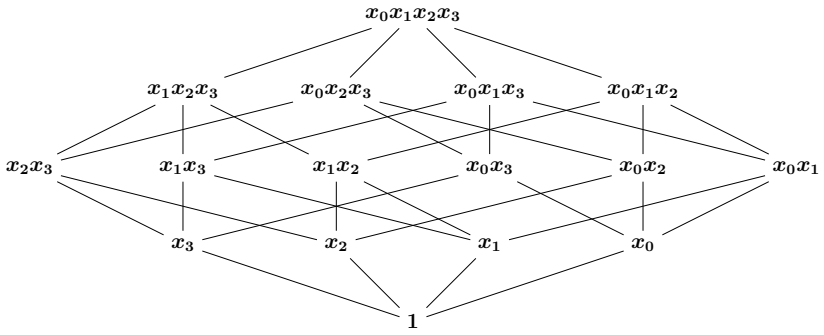


Figure 3.5 – The Hasse diagram for the weakly decreasing monomial order when $m = 4$

- *Let I decreasing set. We define the subset of maximum monomials of I*

$$I_{\max_{\preceq}} = \{f \in I \mid \nexists g \in I, g \neq f \ \ s.t. \ f \preceq g\}.$$

Remark 3.3.8. *Let $I \subseteq \mathcal{M}_m$ be a decreasing set.*

- *Then I is also a weakly decreasing set and we have the following equalities*

$$I = \bigcup_{g \in I_{\max_{\preceq}}} [1, g]_{\preceq}.$$

and

$$I = \bigcup_{f \in I_{\max_{\preceq_{\mathrm{w}}}}} [1, f]_{\preceq_{\mathrm{w}}}.$$

- *In general the intervals are different since $[1, f]_{\preceq_{\mathrm{w}}} \subseteq [1, f]_{\preceq}$. And this is a direct implication of the fact that $I_{\max_{\preceq}} \subseteq I_{\max_{\preceq_{\mathrm{w}}}}$.*

  Nonetheless there are particular cases when the two intervals are equal, namely for any $f = x_0 \dots x_t$ with $0 \leqslant t \leqslant m - 1$.

- *We also stress out that as for the weakly decreasing sets, the set of maximum degree monomials is, in general, different from the set of maximum monomials with respect to the $\preceq$ order.*

  But there is a particular case when the two sets are equals, for example when the set of maximum monomials is reduced to one element $I_{\max_{\preceq}} = \{x_0 \dots x_t\}$. Then we have $I_{\max_{\preceq}} = I_{\max_{\preceq_{\mathrm{w}}}} = I_t$.

- *We notice that that for any monomial set I the subset of maximum degree monomials I_t is independent to any order property that I might satisfy. For this reason in general I_t is different from the subset of maximum monomials with respect to $\preceq_{\mathrm{w}}$ or $\preceq$.*

Definition 3.3.9 (Decreasing monomial code)**.** *Let $I \subset \mathcal{M}_m$ be a decreasing set. Then the linear code $\mathscr{C}(I)$ is called decreasing monomial code.*

Proposition 3.3.10. *Let f, g in $\mathcal{M}_m$ such that $f \preceq g$ and let us set f^*, g^* as the monomials in $\mathcal{M}_m$ such that $f = f^* \gcd(f,g)$ and $g = g^* \gcd(f,g)$, then we have:*

$$f^* \preceq g^*.$$

Furthermore, for any h in $\mathcal{M}_m$ such that $\gcd(g,h) = 1$, we have:

$$fh \preceq gh.$$

Proof. If $\gcd(f,g) = 1$ we directly deduce the relation $f^* \preceq g^*$. So suppose that $\gcd(f,g) \neq 1$.

Firstly consider the case $\deg(f) = \deg(g) = s$. Let $\mathrm{ind}(f) = \{i_1, \ldots, i_s\}$ and $\mathrm{ind}(g) = \{j_1, \ldots, j_s\}$ where $1 \leqslant s \leqslant m$. Now let $\mathrm{ind}(\gcd(f,g)) = \{i_{l_1}, \ldots, i_{l_r}\} = \{j_{k_1}, \ldots, j_{k_r}\} = \mathrm{ind}(f) \cap \mathrm{ind}(g)$ with $1 \leqslant r \leqslant s$. By definition of the support of a monomial and of the $\preceq$ order we have that $i_{l_1} = j_{k_1}$ with $k_1 \leqslant l_1$, and $\ldots$, and $i_{l_r} = j_{k_r}$ with $k_r \leqslant l_r$.

Now consider any element in $\mathrm{ind}(\gcd(f,g))$, for example i_{l_1}. Then we have $i_{l_1} = j_{k_1}$ with $k_1 \leqslant j_1$. Since $i_{k_1} < j_{m_1+1}$ and $\ldots$ $i_{l_1-1} < j_{l_1}$ we obtain

$$f/x_{i_{l_1}} \preceq g/x_{j_{m_1}}.$$

Continue down to $f/(x_{i_{l_1}} \ldots x_{i_{l_r}}) \preceq g/(x_{k_1} \ldots x_{k_r})$.

Secondly consider that $\deg(f) < \deg(g)$. Then by definition of the $\preceq$ order we have $g = g_1 g_2$ with $\deg(g_1) = \deg(f)$ and $f \preceq g_1$. This implies that we can write $f = f^* \gcd(f,g_1)\gcd(f,g_2)$ and $g = g_1^* \gcd(f,g_1) g_2^* \gcd(f,g_2)$ with $g_i = g_i^* \gcd(f,g_i)$ for $1 \leqslant i \leqslant 2$. By the first step of our proof we deduce $f^* \gcd(f,g_2) \preceq g_1^*$, which implies $f^* \gcd(f,g_2) \preceq g_1^* g_2^* \gcd(f,g_2)$. We directly deduce that $f^* \preceq g_1^* g_2^* = g^*$. □

Remark 3.3.11. *The condition on h to be prime with f and g is essential. Indeed, for instance $x_0 \preceq x_1$ but $x_0 x_1 \not\preceq x_1^2$ since $x_1^2 = x_1$ in $\mathbb{R}_m$.*

Proposition 3.3.12. *Reed-Muller codes are Decreasing Monomial codes given by*

$$\mathscr{R}(r,m) = \mathscr{C}\left([1, x_{m-r} \cdots x_{m-1}]_{\preceq}\right). \tag{3.2}$$

Proof. Let I be the set of monomials in $\mathbb{R}_m$ of degree at most r. We have $\mathscr{R}(r,m) = \mathscr{C}(I)$. Note now that $x_{m-r} \ldots x_{m-1}$ belongs to I and that all monomials f of degree at most r are smaller than or equal to this monomial

$$f \preceq x_{m-r} \cdots x_{m-1}.$$

This implies

$$I \subseteq [1, x_{m-r} \cdots x_{m-1}]_{\preceq}.$$

Moreover no monomial of degree greater than r can be smaller than $x_{m-r} \ldots x_{m-1}$. Therefore we have

$$I = [1, x_{m-r} \cdots x_{m-1}]_{\preceq}.$$

□

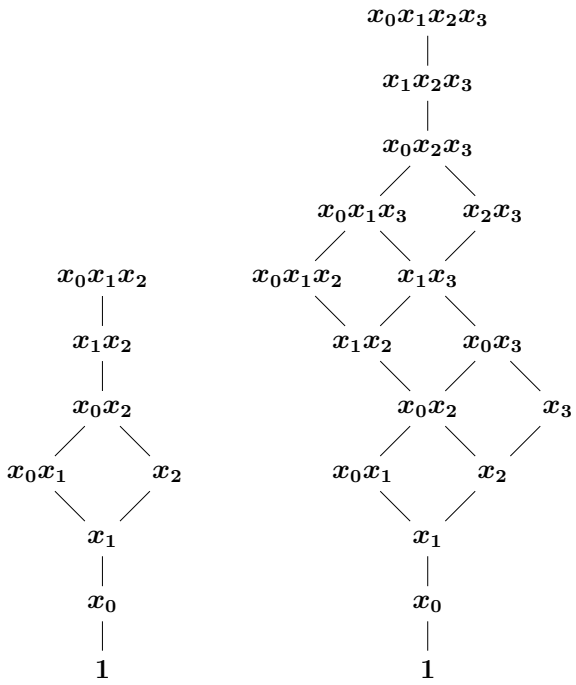


Figure 3.6 – The Hasse diagram for the decreasing monomial order (left) $m = 3$ and (right) $m = 4$

Remark 3.3.13.

Let $1 \leqslant k \leqslant n$ and $\{\mathscr{C}(I)\}_{I\subseteq\mathcal{M}_m,|I|=k}$ be the class of Monomial codes of length n and dimension k. Then any subclass of $\{\mathscr{C}(I)\}_{I\subseteq\mathcal{M}_m,|I|=k}$ inherits all the properties of the class of Monomial codes. Therefore any property of the Monomial codes is transmitted to Weakly Decreasing Monomial codes and Decreasing Monomial codes.

In the same manner the class of Decreasing Monomial codes inherits the properties of the class of Weakly Decreasing Monomial codes.

Example 3.3.14 (Weakly decreasing and decreasing sets)**.**

- *Let $m = 4$ and I be a monomial set defined by*

$$I = \{1, x_0, x_1, x_2, x_3, x_0x_1, x_0x_2, x_0x_3, x_1x_2\}.$$

Then we have that I is a decreasing set with $I = [1, x_0x_3]_{\preceq} \cup [1, x_1x_2]_{\preceq}$. We also have $I_{\max_{\preceq}} = \{x_0x_3, x_1x_2\}$ and $I_2 = \{x_0x_1, x_0x_2, x_0x_3, x_1x_2\}$.

On the other hand I is a weakly decreasing set $I = [1, x_0x_1]_{\preceq_{\mathrm{w}}} \cup [1, x_0x_2]_{\preceq_{\mathrm{w}}} \cup [1, x_1x_2]_{\preceq_{\mathrm{w}}} \cup [1, x_0x_3]_{\preceq_{\mathrm{w}}}$. Here we have that $I_{\max_{\preceq_{\mathrm{w}}}} = I_2$.

- *Let $= 4$ and J be a monomial set defined by*

$$J = \{1, x_0, x_1, x_2, x_3, x_0x_3, x_1x_2\}.$$

Then we have that J is a weakly decreasing monomial set, more exactly $J = [1, x_0x_3]_{\preceq_{\mathrm{w}}} \cup [1, x_1x_2]_{\preceq_{\mathrm{w}}}$. We also have that $J_{\max_{\preceq_{\mathrm{w}}}} = J_2 = \{x_0x_3, x_1x_2\}$.

But J is not a decreasing monomial set since $x_0x_2 \notin J$.

3.3.4 Polar codes are Decreasing Monomial codes

In this part we will focus on proving that Polar codes are Decreasing Monomial codes. We will begin by considering the Binary Erasure Channel and then the general case of Binary Discrete Memoryless Channels. We point out that an equivalent demonstration of the fact that Polar codes over the B-DMC are Decreasing Monomial codes was given in parallel by Christian Schürch in [Sch16].

Polar codes over the BEC

The Binary Erasure Channel was a deeply studied case and also the first channel for which we proved that Polar codes satisfy the decreasing property.

Proposition 3.3.15. *The Bhattacharyya parameter for the synthetic channels in the case of a* BEC(p) *is:*

$$\mathcal{B}\left(W^{\boldsymbol{x^u}}\right)(p) = f_{u_0} \circ \cdots \circ f_{u_{m-1}}(p),$$

where $f_0 : \begin{array}{ccc} [0,1] & \to & [0,1] \\ p & \mapsto & p^2 \end{array}$ *and* $f_1 : \begin{array}{ccc} [0,1] & \to & [0,1] \\ p & \mapsto & 1-(1-p)^2 \end{array}$.

This fact comes directly from the Definition 3.2.26 of the Bhattacharyya parameter and Example B.0.3 on the BEC(p).

Remark 3.3.16. *We remark that the two functions* f_0, f_1 *are increasing functions and thus* $\mathcal{B}\left(W^{\boldsymbol{x^u}}\right)$ *is an increasing function over* $[0,1]$ *as the composition of increasing functions.*

Proposition 3.3.17 (Bhattacharyya function of a monomial over the BEC)**.**
Let $g = x_{g_1} \dots x_{g_s}$ *with* $0 \leqslant g_1 < \cdots < g_s \leqslant m-1$. *Then the Bhattacharyya function for the* BEC(p) *associated to* g *is*

$$\mathcal{B}\left(W^g\right)(p) = f_0^{g_1} \circ f_1 \circ f_0^{g_2-g_1-1} \circ f_1 \cdots \circ f_0^{g_s-g_{s-1}-1} \circ f_1 \circ f_0^{m-g_s-1}(p).$$

Example 3.3.18. *For* $m = 2$ *we have*

$$\begin{aligned}
\mathcal{B}\left(W^{x_0x_1}\right) &= \mathcal{B}\left(W^{--}\right) = f_1 \circ f_1 = p(2-p)\left(2-p(2-p)\right) \\
\mathcal{B}\left(W^{x_1}\right) &= \mathcal{B}\left(W^{-+}\right) = f_0 \circ f_1 = p^2(2-p)^2 \\
\mathcal{B}\left(W^{x_0}\right) &= \mathcal{B}\left(W^{+-}\right) = f_1 \circ f_0 = p^2(2-p^2) \\
\mathcal{B}\left(W^{1}\right) &= \mathcal{B}\left(W^{++}\right) = f_0 \circ f_0 = p^4.
\end{aligned}$$

We notice that for a given binary erasure channel BEC(p) the order induced by the Bhattacharyya parameter on the synthetic channels is a total order. We raise here a more general question, that is what happens when the crossover probability of the channel it's unknown to the designer? In this case we define the Bhattacharyya order over the family of $\{\text{BEC}(p)\}_{0 \leqslant p \leqslant 1}$.

Definition 3.3.19 (Bhattacharyya order over the family of BEC)**.** *Let* W *and* W' *be two synthetic channels. Then we say that* $\mathcal{B}(W) \leqslant \mathcal{B}(W')$ *if and only if for any erasure probability* $p \in [0,1]$ *we have that*

$$\mathcal{B}(W)(p) \leqslant \mathcal{B}(W')(p).$$

We say that two channels W and W' are incomparable, with respect to the Bhattacharyya order, if there exist a pair of erasure probabilities $(p, p') \in [0,1] \times [0,1]$ such that

$$\mathcal{B}(W)(p) < \mathcal{B}(W')(p) \quad \text{and} \quad \mathcal{B}(W)(p') > \mathcal{B}(W')(p').$$

In Figure 3.7 and 3.8 we plot the Bhattacharyya order over the family of BEC as a function of the channel probability for all the synthetic channels. We show that up to $m = 4$ the Bhattacharyya order is total. Starting from $m = 5$ the Bhattacharyya order over the family of BEC becomes a partial order. In this particular case the monomials that are no longer comparable are x_4 and x_1x_0. Moreover they generate, with respect to the divisibility order, two other non comparable pair of monomials x_4x_3 with $x_3x_1x_0$ and x_4x_2 with $x_2x_1x_0$. But this partial order it is stronger that the "$\preceq$" and thus for this family of channels Polar codes might admit a better ordering than the $\preceq$, fact that opens new perspectives in this topic.

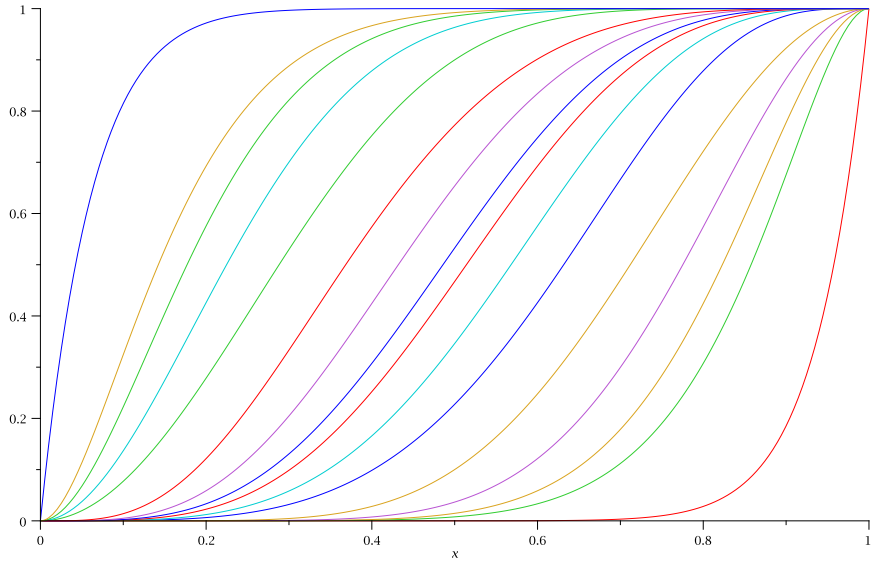


Figure 3.7 – The Bhattacharyya parameter for all the synthetic channels when $m = 4$

In the next paragraph we will detail some necessary properties for proving that Polar codes are Decreasing Monomial codes over the BEC.

Lemma 3.3.20. *For $i \in \{1, \dots, s\}$, let l_i, l_i^* be increasing functions from $[0,1] \to [0,1]$ such that $\forall x \in [0,1], \quad l_i^*(x) \leqslant l_i(x)$. Let $Z = l_1 \circ \cdots \circ l_s$ and $Z^* = l_1^* \circ \cdots \circ l_s^*$, then*

$$\forall x \in [0,1], \quad Z^*(x) \leqslant Z(x).$$

Proof. The proof is by induction on s. For $s = 1$, the result comes from the assumption. For $s \geqslant 2$, let $Z_{s-1} = l_2 \circ \cdots \circ l_s$ and $Z_{s-1}^* = l_2^* \circ \cdots \circ l_s^*$, then for any $x \in [0,1]$ by induction we have $Z_{s-1}^*(x) \leqslant Z_{s-1}(x)$, hence using the fact that l_1^* is increasing and that $l_1^* \leqslant l_1$ we get $Z^*(x) = l_1^*(Z_{s-1}^*(x)) \leqslant l_1^*(Z_{s-1}(x)) \leqslant l_1(Z_{s-1}(x)) = Z(x)$. □

Lemma 3.3.21 (The bit position)**.** *Let $g = x_{g_1} \dots x_{g_{s-1}} x_{g_s}$ with $0 \leqslant g_1 < \cdots < g_s \leqslant m-1$ and $g^{(*i)} = x_{g_1} \dots x_{g_i} x_{g_{i+1}^*} x_{g_{i+2}} \dots x_{g_s}$ with $g_i \leqslant g_{i+1}^* \leqslant g_{i+1}$. Then $g^{(*i)} \preceq g$ and*

$$\mathcal{B}\left(W^{g^*}\right) \leqslant \mathcal{B}(W^g).$$

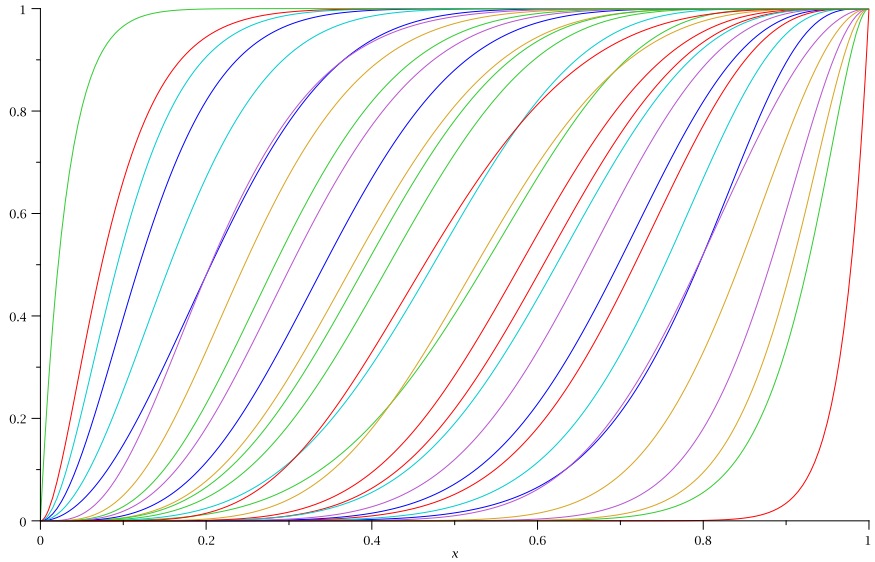


Figure 3.8 – The Bhattacharyya parameter for all the synthetic channels when $m = 5$

Proof. From the definition of the Bhattacharyya parameter we have $\mathcal{B}(W^g) = l_1 \circ f_0^{g_{i+1}-g^*_{i+1}} \circ f_1 \circ l_3$ and $\mathcal{B}\left(W^{g^*}\right) = l_1 \circ f_1 \circ f_0^{g_{i+1}-g^*_{i+1}} \circ l_3$ where $l_1 = f_0^{g_1} \circ f_1 \circ \cdots \circ f_0^{g_i-g_{i-1}-1} \circ f_1 \circ f_0^{g^*_{i+1}-g_i-1}$ and $l_3 = f_0^{g_{i+2}-g_{i+1}-1} \circ f_1 \circ \cdots \circ f_0^{m-g_s-1}$. Let $l_2^* = f_1 \circ f_0^{g_{i+1}-g^*_{i+1}}$ and $l_2 = f_0^{g_{i+1}-g^*_{i+1}} \circ f_1$. In order to use the previous lemma, it remains to prove that $, \forall y \in [0,1] \quad l_2^*(y) \leqslant l_2(y)$.

Since $l_2^*(y) = y^{2^{g_{i+1}-g^*_{i+1}}}(2 - y^{2^{g_{i+1}-g^*_{i+1}}})$ and $l_2(y) = (y(2-y))^{2^{g_{i+1}-g^*_{i+1}}}$ we obtain

$$l_2(y) - l_2^*(y) = y^{2^{g_{i+1}-g^*_{i+1}}}((2-y)^{2^{g_{i+1}-g^*_{i+1}}} - 2 + y^{2^{g_{i+1}-g^*_{i+1}}}). \tag{3.3}$$

Let L_k be the function defined by $L_k(y) = (2-y)^k - 2 + y^k$ for $y \in [0,1]$ and $k \in \mathbb{N}^*$. As $L_{k+1}(y) - L_k(y) = (1-y)((2-y)^k - y^k) \geqslant (1-y)(2-y-y) = 2(1-y)^2 \geqslant 0$ for $y \in [0,1]$, then $L_{k+1}(y) \geqslant L_k(y) \geqslant L_0(y) = 0$ by induction. Hence, $l_2(y) - l_2^*(y) = y^{2^{g_{i+1}-g^*_{i+1}}} L_{2^{g_{i+1}-g^*_{i+1}}}(y) \geqslant 0$. □

Proposition 3.3.22. *Let $g = x_{g_1} \dots x_{g_{s-1}} x_{g_s}$ and $h = x_{h_1} \dots x_{h_{s-1}} x_{h_s}$ be two monomials such that $g \preceq h$. Then*

$$\mathcal{B}(W^g) \leqslant \mathcal{B}\left(W^h\right).$$

Proof. For $i = 0, \dots, s$ let $g^{(*i)} = x_{g_1} \dots x_{g_i} x_{h_{i+1}} \dots x_{h_s}$. We have $g^{(*0)} = h$, $g^{(*s)} = g$, and $g^{(*(i+1))} \preceq g^{(*i)}$ verify the hypotheses of the previous lemma. Applying the previous lemma s times, we get $\mathcal{B}(W^g) \leqslant \mathcal{B}\left(W^h\right)$. □

Proposition 3.3.23. *Let $0 \leqslant t \leqslant s \leqslant m-1$ and $g = x_{g_1} \dots x_{g_{t-1}} x_{g_t}$ and $h = x_{h_1} \dots x_{h_{s-1}} x_{h_s}$ be two monomials such that $g \preceq h$. Then*

$$\mathcal{B}(W^g) \leqslant \mathcal{B}\left(W^h\right).$$

Proof. From the definition of the order $\preceq$, we have $g \preceq x_{h_{s-t+1}} \dots x_{h_s}$ and $\mathcal{B}(W^g) \leqslant \mathcal{B}(W^{x_{h_{s-t+1}} \dots x_{h_s}})$. But we can write

$$\mathcal{B}(W^{x_{h_{s-t+1}} \dots x_{h_s}}) = f_0^{h_{s-t+1}} \circ l_2,$$

with $l_2 = f_1 f_0^{h_{s-t+2}-h_{s-t+1}} \circ \cdots \circ f_0^{m-1-h_s}$ and $\mathcal{B}\left(W^h\right) = l_1 \circ l_2$ where l_1 is the composition of h_{s-t+1} functions f_0 or f_1. As $f_0 \leqslant f_1$, applying Lemma 3.3.20 we get the result. □

Finally we can state our result, that is

Theorem 3.3.24. *Polar codes designed for the Binary Erasure Channel are Decreasing Monomial codes.*

Proof. Let $\mathscr{C}(I)$ be a Polar code of length $N = 2^m$ over the BEC family and $I \subset \mathcal{M}_m$. We prove that I is decreasing, i.e. for any monomial $g \in I$, for any monomial $h \in \mathcal{M}_m$, if $h \preceq g$ then $h \in I$. This will be the case if the Bhattacharyya parameter of g and h verify $\mathcal{B}\left(W^g\right) \leqslant \mathcal{B}\left(W^h\right)$,which is the case from Proposition 3.3.23. □

Polar codes over the B-DMC

In this part we will prove a more general statement that is *Polar codes are Decreasing Monomial codes*. We will start by proving a weaker statement whose ingredients and lemmas used in the proof will be essential for the proof of Theorem 3.3.31. But first we begin by recalling some useful facts about concatenated and degraded channel.

Lemma 3.3.25 ([RU08, p207])**.** *Let W be a binary input channel. If $W \preceq_{\mathrm{d}} W'$ then $\mathcal{B}\left(W'\right) \leqslant \mathcal{B}\left(W\right)$.*

Lemma 3.3.26 ([Kob09, Lemma 4.7],[TV13, Lemma 5])**.** *Let W be a binary input channel and let W' be a degradation of this channel ($W' \preceq_{\mathrm{d}} W$). Then*

$$\begin{aligned} W'^{-} &\preceq_{\mathrm{d}} W^{-} \\ W'^{+} &\preceq_{\mathrm{d}} W^{+}. \end{aligned}$$

From this lemma we obtain that

Corollary 3.3.27. *Let W be a binary input channel and let W' be a degradation of this channel: $W' \preceq_{\mathrm{d}} W$. For any monomial f in $\mathcal{M}_m$ we have*

$$W'^{f}_{m} \preceq_{\mathrm{d}} W^{f}_{m}.$$

Lemma 3.3.28. *For any binary-input symmetric channel $W : \{0,1\} \to \mathcal{Y}$ we have*

$$W^{-} \preceq_{\mathrm{d}} W \preceq_{\mathrm{d}} W^{+}.$$

Proof. First of all let us recall why we have $W \preceq_{\mathrm{d}} W^{+}$. Consider the output (y_1, y_2, u_2) of the W^{+} channel when a bit u_1 has been sent through it. By definition y_1 is the result of sending u_1 through the W-channel. Therefore if we define W' as the channel $W' : \mathcal{Y} \times \mathcal{Y} \times \{0,1\} \to \mathcal{Y}$ which takes (y_1, y_2, u_1), erases y_2 and u_1 to send just y_1, we clearly have that $W = W' \circ W^{+}$.

Let us prove now that $W^{-} \preceq_{\mathrm{d}} W$ by constructing a channel W'' such that $W'' \circ W = W^{-}$. The channel W'' is defined as follows. It takes as input y_1. At that point a bit u_1 is drawn uniformly at random and sent through channel W to get some value y_2. Then W'' outputs the pair (y_1, y_2) when $u_1 = 0$ and $(\pi(y_1), y_2)$ when $u_1 = 1$. Here π is the involution acting on $\mathcal{Y}$ such that $W(y|1) = W(\pi(y)|0)$.

Now sending b through channel $W'' \circ W$ and receiving $y_1 y_2$ happens in two cases

- when $u_1 = 0$, this happens when y_1 has been received after sending u_2 through W and y_1, y_2 has been received after sending y_1 through W''. This means that for W'', y_2 has been received with the second use of W when u_1 has been sent through it;
- when $u_1 = 1$, this happens when $\pi(y_1)$ has been received after sending u_2 through W (so that W'' changes $\pi(y_1)$ into $\pi(\pi(y_1)) = y_1$) and y_2 has been received with the second use of W when u_1 has been sent through it.

This implies that

$$\begin{aligned}
(W'' \circ W)(y_1, y_2|u_2) &= \mathsf{prob}(x = 0)\mathsf{prob}(\text{receiving } (y_1, y_2)|u_2 \text{ was sent}, u_1 = 0) + \\
&\quad \mathsf{prob}(u_1 = 1)\mathsf{prob}(\text{receiving } (y_1, y_2)|u_2 \text{ was sent}, u_1 = 1) \\
&= \mathsf{prob}(u_1 = 0)W(y_1|u_2)W(y_2|u_1 = 0) \\
&+ \mathsf{prob}(u_1 = 1)W(\pi(y_1)|u_2)W(y_2|u_1 = 1) \\
&= \frac{1}{2}\{W(y_1|u_2)W(y_2|u_1 = 0) + W(\pi(y_1)|u_2)W(y_2|u_1 = 1)\} \\
&= \frac{1}{2}\{W(y_1|u_2)W(y_2|u_1 = 0) + W(y_1|1 \oplus u_2)W(y_2|u_1 = 1)\} \\
&= \frac{1}{2}\sum_{u \in \mathbb{F}_2} W(y_1|u)W(y_2|u \oplus u_2) \\
&= W^-(y_1, y_2|u_2)
\end{aligned}$$

This computation shows that $W'' \circ W$ is precisely the channel W^-. □

Proposition 3.3.29. *Let W be a binary input symmetric channel. Let f and g be two monomials of $\mathcal{M}_m$. If $f \preceq_{\mathrm{w}} g$ then*

$$W_m^g \preceq_{\mathrm{d}} W_m^f.$$

Proof. This follows by induction on m. When $m = 1$ we just have to prove that

$$W_1^{x_0} \preceq_{\mathrm{d}} W_1^1. \tag{3.4}$$

Recall that

$$\begin{aligned}
W_1^{x_0} &= W^- \\
W_1^1 &= W^+
\end{aligned}$$

The inequality (3.4) follows directly from Lemma 3.3.28. Assume now that Proposition 3.3.29 holds for some positive integer m and let f and g be in $\mathcal{M}_{m+1}$ such that $f \preceq_{\mathrm{w}} g$.

Let us define the following monomials:

$$\begin{aligned}
f_{0\ldots(m-1)} &= \gcd\left(f, \prod_{i=0}^{m-1} x_i\right) \\
g_{0\ldots(m-1)} &= \gcd\left(g, \prod_{i=0}^{m-1} x_i\right)
\end{aligned}$$

$$f_m = \begin{cases} x_0 & \text{if } x_m \text{ divides } f, \\ 1 & \text{otherwise.} \end{cases}$$
$$g_m = \begin{cases} x_0 & \text{if } x_m \text{ divides } g, \\ 1 & \text{otherwise.} \end{cases}$$

Note now that

$$W_{m+1}^f = \left(W_1^{f_m}\right)_m^{f_{0\dots m-1}}$$
$$W_{m+1}^g = (W_1^{g_m})_m^{g_{0\dots m-1}}$$

Since $f_{0\dots m-1}$ divides $g_{0\dots m-1}$ we have by the induction hypothesis

$$\left(W_1^{f_m}\right)_m^{g_{0\dots m-1}} \preceq_{\mathrm{d}} \left(W_1^{f_m}\right)_m^{f_{0\dots m-1}} \tag{3.5}$$

Since $f_m \preceq_{\mathrm{w}} g_m$ we have

$$W_1^{g_m} \preceq_{\mathrm{d}} W_1^{f_m}.$$

From Corollary 3.3.27 we deduce that

$$(W_1^{g_m})_m^{g_{0\dots m-1}} \preceq_{\mathrm{d}} \left(W_1^{f_m}\right)_m^{g_{0\dots m-1}} \tag{3.6}$$

From (3.5) and (3.6) we deduce that

$$(W_1^{g_m})_m^{g_{0\dots m-1}} \preceq_{\mathrm{d}} \left(W_1^{f_m}\right)_m^{f_{0\dots m-1}}$$

which proves the statement of the proposition for $m+1$. □

We are ready now to prove that Polar codes are Weakly Decreasing Monomial codes

Proposition 3.3.30. *Let $\mathscr{C}(I)$ be a polar code devised for a binary discrete memoryless channel W. Then $\mathscr{C}(I)$ is a Weakly Decreasing Monomial code.*

Proof. Let $\mathscr{C}(I)$ be a polar code generated by the set of monomials I devised for a channel W. Let f and g be two monomials such that g is in I and $f \preceq_{\mathrm{w}} g$. From Proposition 3.3.29 we know that

$$W_m^g \preceq_{\mathrm{d}} W_m^f.$$

By applying Lemma 3.3.25 we deduce that

$$\mathcal{B}\left(W_m^f\right) \leqslant \mathcal{B}\left(W_m^g\right).$$

This implies that f also belongs to I. □

Theorem 3.3.31. *Polar codes are Decreasing Monomial codes.*

In order to prove Therorem 3.3.31 we will need to have a finer understanding of the W_m^f's and for that we give the following lemma:

Lemma 3.3.32. *Let $x \in \mathbb{F}_2$ be the input of a synthetic channel W_m^f. Then the channel W_m^f can be described as follows:*

1. *choose binary word $\boldsymbol{a} = (a_g)_{g\in\mathcal{M}_m}$ of length 2^m indexed by the monomials of $\mathcal{M}_m$ using the decreasing index order from Section 3.2.*
2. *let $\boldsymbol{a}' = (a'_g)_{g\in\mathcal{M}_m}$ be the binary word of length 2^m such that for all $g \in \mathcal{M}_m$ $a'_g = a_g$ with the exception of a'_f where $a'_f = x$.*
3. *compute $\boldsymbol{b} \stackrel{def}{=} \boldsymbol{a}'\boldsymbol{G}_m$.*
4. *send the bits of $\boldsymbol{b}$ through channel W to obtain a vector $\boldsymbol{y} \in \mathcal{Y}^{2^m}$.*
5. *the output of the channel W_m^f is then $\boldsymbol{y}, (a_g)_{g:g>f}$.*

Proof. This lemma is a direct consequence of the Definition 3.2.24 of W_m^f and Definition 3.2.22 of the synthetic channels. □

Theorem 3.3.31 heavily relies on Proposition 3.3.29 on one hand and the following lemma on the other hand.

Lemma 3.3.33. *Let W be a B-DMC. In such a case for any positive integer m we have*

$$W_m^{x_1x_2\dots x_{m-2}x_{m-1}} \preceq_{\mathrm{d}} W_m^{x_0x_1\dots x_{m-2}}.$$

Proof. We use Lemma 3.3.32 and consider $W_m^{x_0x_1\dots x_{m-2}}$. As explained in this lemma, the channel takes as input a bit c and outputs $\boldsymbol{y}$ and $(a_g)_{g:g>x_0x_1\dots x_{m-2}}$, where $\boldsymbol{y} = (y_g)_{g\in\mathcal{M}_m} \in \mathcal{Y}^{2^m}$ is the result of sending $\boldsymbol{a}'\boldsymbol{G}_m$ through the channel W.

Let $\tau : \mathcal{M}_m \longrightarrow \mathcal{M}_m$ be a permutation on the set of monomials in $\mathcal{M}_m$ defined by

$$\tau(x_{i_1}\cdots x_{i_t}) \;=\; x_{(i_1-1) \pmod m}\cdots x_{(i_t-1) \pmod m}.$$

Let W' be a B-DMC which consists in reordering $\boldsymbol{y}$ as $\boldsymbol{y}^\tau \stackrel{\text{def}}{=} (y_{\tau(g)})_{g\in\boldsymbol{M}}$ and deleting all the entries of $(a_g)_{g:g>x_0x_1\dots x_{m-2}}$ with the exception of $a_{x_0\dots x_{m-1}}$. To finish the proof we check that

$$W_m^{x_0x_1\dots x_{m-2}} \circ W' = W_m^{x_1x_2\dots x_{m-2}x_{m-1}}.$$

□

Before giving a slightly more general statement, let us introduce some notation which will be very helpful.

Notation 3.3.34. *Let $f = x_{i_1}\dots x_{i_s}$ be a monomial in $\mathcal{M}_m$. We denote by $f_{[a..b]}$ the monomial $\Pi_{i_j:a\leqslant i_j\leqslant b}x_{i_j}$ and by f^{-t}, with $t \leqslant i_1$ the monomial $x_{i_1-t}\dots x_{i_s-t}$.*

Example 3.3.35. *Let $m = 7$ and $f = x_0x_2x_3x_4x_6$. Then $f_{[1..3]} = x_2x_3$ and $f^{-2}_{[2..4]} = x_0x_1x_2$.*

Lemma 3.3.36. *Let f and g be two monomials of $\mathcal{M}_m$ of the same degree such that*

(i) $f \preceq g$,

(ii) $f_{[0..i-1]} = g_{[0..i-1]}$,

(iii) $f_{[i+t+1..m-1]} = g_{[i+t+1..m-1]}$,

(iv) $f_{[i..i+t]} = x_i x_{i+1} \dots x_{i+t-1}$,

(v) $g_{[i..i+t]} = x_{i+1} x_{i+2} \dots x_{i+t}$.

Then

$$W_m^g \preceq_{\mathrm{d}} W_m^f$$

More generally in the case of two monomials f and g satisfying (ii) and (iii) and

$$W_{t+1}^{g_{[i..i+t]}^{-i}} \preceq_{\mathrm{d}} W_{t+1}^{f_{[i..i+t]}^{-i}} \tag{3.7}$$

then

$$W_m^g \preceq_{\mathrm{d}} W_m^f.$$

Proof. We can write in the first case

$$\begin{aligned} W_m^f &= \left(\left(W_{m-i-t-1}^{f_{[i+t+1..m-1]}}\right)_{t+1}^{x_0 x_1 \dots x_{t-1}}\right)_i^{f_{[0..i-1]}} \\ W_m^g &= \left(\left(W_{m-i-t-1}^{f_{[i+t+1..m-1]}}\right)_{t+1}^{x_1 x_2 \dots x_t}\right)_i^{f_{[0..i-1]}}. \end{aligned}$$

We can apply Lemma 3.3.33 to $W_{m-i-t-1}^{f_{[i+t+1..m-1]}}$ to deduce that

$$\left(\left(W_{m-i-t-1}^{f_{[i+t+1..m-1]}}\right)_{t+1}^{x_1 x_2 \dots x_t}\right) \preceq_{\mathrm{d}} \left(\left(W_{m-i-t-1}^{f_{[i+t+1..m-1]}}\right)_{t+1}^{x_0 x_1 \dots x_{t-1}}\right).$$

By applying Corollary 3.3.27 we obtain

$$\left(\left(W_{m-i-t-1}^{f_{[i+t+1..m-1]}}\right)_{t+1}^{x_1 x_2 \dots x_t}\right)_i^{f_{[0..i-1]}} \preceq_{\mathrm{d}} \left(\left(W_{m-i-t-1}^{f_{[i+t+1..m-1]}}\right)_{t+1}^{x_0 x_1 \dots x_{t-1}}\right)_i^{f_{[0..i-1]}}.$$

The second statement follows by a similar reasoning but uses now (3.7) directly instead of using Lemma 3.3.33. □

This lemma can now be used to prove by induction on m that

Lemma 3.3.37. *Let W be a symmetric binary input channel. Let m be a positive integer and let f and g be two monomials of $\mathcal{M}_m$ that are of same degree and such that $f \preceq g$. Then*

$$W_m^g \preceq_{\mathrm{d}} W_m^f.$$

Proof. When $m = 1$ the two monomials are necessarily equal and we are done. Assume now that the property we want to prove holds for all positive integers up to some positive integer m. Consider now two monomials f and g of $\mathcal{M}_{m+1}$ with the same degree t and such that $f \preceq g$. We can write

$$\begin{aligned} f &= x_{i_1} \dots x_{i_t} \\ g &= x_{j_1} \dots x_{j_t} \end{aligned}$$

with $i_1 < i_2 < \cdots < i_t$ and $j_1 < j_2 < \ldots j_t$. Let i_l be the smallest index among $\{i_1, \ldots, i_t\}$ such that $i_l < j_l$. If such an index does not exist we have $f = g$ and therefore $W^g_{m+1} \preceq_{\mathrm{d}} W^f_{m+1}$ which is what we want to prove. Let i_s be the largest index greater than or equal to i_l such that

$$i_j = i_l + (j - l)$$

for all j in $\{l, l+1, \ldots, s\}$. In other words in such a case

$$x_{i_l} \ldots x_{i_s} = x_{i_l} x_{i_l+1} \ldots x_{i_l+(s-l)}.$$

Observe that we can write f as

$$f = f_{[0..i_l-1]} x_{i_l} x_{i_l+1} \ldots x_{i_l+(s-l)} f_{[i_l+(s-l)+2..m]}.$$

We can apply the previous lemma and obtain

$$W^{f_{[0..i_l-1]} x_{i_l+1} x_{i_l+2} \ldots x_{i_l+(s-l)+1} f_{[i_l+(s-l)+2..m]}}_{m+1} \preceq_{\mathrm{d}} W^f_{m+1}. \tag{3.8}$$

Observe now that g is such that

$$x_{i_l+1} x_{i_l+2} \ldots x_{i_l+(s-l)+1} f_{[i_l+(s-l)+2..m]} \preceq g_{[i_l..m]}.$$

This comes from the fact that all the j_u's for $u \in \{l, l+1, \ldots, s\}$ necessarily satisfy $j_u \geqslant i_u + 1$, since this is true for $u = l$ and can be shown for values that are larger by noting that $j_u \geqslant j_l + (l - u) \geqslant i_l + 1 + (l - u) = i_u + 1$. We can apply the induction hypothesis to the pair $g^{-i_l}_{[i_l..m]}$ and $x_1 \ldots x_{(s-l)+1} f^{-i_l}_{[i_l+(s-l)+2..m]}$ since

$$x_1 \ldots x_{(s-l)+1} f^{-i_l}_{[i_l+(s-l)+2..m]} \preceq g^{-i_l}_{[i_l..m]}$$

and therefore

$$W^{g^{-i_l}_{[i_l..m]}}_{m-i_l+1} \preceq_{\mathrm{d}} W^{x_1 \ldots x_{(s-l)+1} f^{-i_l}_{[i_l+(s-l)+2..m]}}_{m-i_l+1}.$$

By applying Corollary 3.3.27 we deduce that

$$W^g_{m+1} \preceq_{\mathrm{d}} W^{g_{[0..i_l-1]} x_{i_l+1} \ldots x_{i_l+(s-l)+1} f_{[i_l+(s-l)+2..m]}}_{m+1}. \tag{3.9}$$

Using the fact that by definition $g_{[0..i_l-1]} = f_{[0..i_l-1]}$ and putting (3.8) and (3.9) together by using the transitivity of $\preceq_{\mathrm{d}}$ we get

$$W^g_{m+1} \preceq_{\mathrm{d}} W^f_{m+1}.$$

□

We are ready now for the proof of Theorem 3.3.31.

Proof of Theorem 3.3.31. Let $\mathscr{C}(I)$ be a polar code generated by the set of monomials I devised for a channel W. Let f and g be two monomials such that g is in I and $f \preceq g$. Assume first that f and g have the same degree. In such a case we can apply Lemma 3.3.37 and deduce that

$$W^g_m \preceq_{\mathrm{d}} W^f_m.$$

When f and g are not of the same degree, we know that there exists a divisor g^* of g such that g^* and f have the same degree and $f \preceq g^*$. By applying Lemma 3.3.37 to the pair (f, g^*) we deduce that

$$W_m^{g^*} \preceq_{\mathrm{d}} W_m^f.$$

Since g^* divides g we know from Proposition 3.3.30 that

$$W_m^g \preceq_{\mathrm{d}} W_m^{g^*}.$$

By transitivity of $\preceq_{\mathrm{d}}$ we deduce again that

$$W_m^g \preceq_{\mathrm{d}} W_m^f.$$

Therefore in all cases we can apply Lemma 3.3.25 and obtain that

$$\mathcal{B}\left(W_m^f\right) \leqslant \mathcal{B}\left(W_m^g\right).$$

This implies that f also belongs to I. □

3.4 Duality properties

We begin here by studying the problem of duality in the case of monomial codes. We will see that the dual of a monomial code is a polynomial code, but not necessarily a monomial code. We will prove that up to a permutation of the support of the code, the dual of a monomial code is still a monomial code, fact that is already known from Vardy and Mahdavifar [MV15]. Moreover the dual of a weakly decreasing monomial code turns out to be a weakly decreasing monomial code.

3.4.1 Dual of Monomial Codes

Definition 3.4.1. *The* multiplicative complement *of a monomial $g \in \mathcal{M}_m$ is defined as:*

$$\check{g} \stackrel{def}{=} \prod_{i \in \{0,\dots,m-1\} \setminus \mathrm{ind}(g)} x_i = \frac{x_0 \dots x_{m-1}}{g}.$$

By extension for any subset $I \subseteq \mathcal{M}_m$, the set $\check{I} \subseteq \mathcal{M}_m$ denotes $\check{I} = \{\check{f} \ : \ f \in I\}$.

Definition 3.4.2. *Let m be a positive integer and $s < m$. We define the application ψ that associates to any monomial g in $\mathcal{M}_m$ the polynomial $\psi(g)$ in $\mathbb{R}_m$*

$$\begin{array}{rcl} \psi : \mathcal{M}_m & \longrightarrow & \mathbb{R}_m \\ g = x_{i_1} \dots x_{i_s} & \longmapsto & \psi(g) = (x_{i_1} + 1) \dots (x_{i_s} + 1). \end{array}$$

Lemma 3.4.3. *Let π be the permutation that swaps the positions i with $2^m - i + 1$ for all $0 \leqslant i \leqslant 2^m - 1$.*

$$\pi \stackrel{def}{=} \begin{pmatrix} 0 & \dots & i & \dots & 2^m - 1 \\ 2^m - 1 & \dots & 2^m - i & \dots & 0 \end{pmatrix}.$$

Then

$$\mathsf{ev}(\psi(g))^\pi = \mathsf{ev}(g).$$

Proof. Since for any element $\boldsymbol{u} = (u_0, \ldots, u_{m-1}) \in \mathbb{F}_2^m$ we have that $\pi(\boldsymbol{u}) = (u_0 + 1, \ldots, u_{m-1} + 1)$ it is straightforward to remark that for any monomial $g \in \mathcal{M}_m$ the evaluation of $\psi(g)$ on $\pi(\boldsymbol{u})$ equals the evaluation of g on $\boldsymbol{u}$.

□

Example 3.4.4. *Let $m = 2$ then we have*

	11	01	10	00
$\mathsf{ev}(x_0x_1)$	1	0	0	0
$\mathsf{ev}(x_1)$	1	1	0	0
$\mathsf{ev}(x_0)$	1	0	1	0
$\mathsf{ev}(1)$	1	1	1	1

$= \boldsymbol{G}_2.$

If we apply ψ to all the monomials we obtain

	11	01	10	00
$\mathsf{ev}((x_0+1)(x_1+1))$	0	0	0	1
$\mathsf{ev}(x_1+1)$	0	0	1	1
$\mathsf{ev}(x_0+1)$	0	1	0	1
$\mathsf{ev}(1)$	1	1	1	1

$= \boldsymbol{G}_2^{\pi}.$

Lemma 3.4.5. *Let $\mathscr{C}(I)$ be a monomial code. Then*

$$\dim(\mathscr{C}(I)) = \dim(\mathscr{C}(\psi(I))).$$

Proof. We denote as usual $\boldsymbol{G}$ the generator matrix of $\mathscr{C}(I)$ and $\boldsymbol{G}^*$ the generator matrix of $\mathscr{C}(\psi(I))$. First we recall that for any permutation $\pi \in \mathfrak{S}_n$ and any row submatrix $\boldsymbol{G}_I$ of $\boldsymbol{G}_m$, indexed by a subset of monomials we have

$$\operatorname{rank}(\boldsymbol{G}_I) = \operatorname{rank}(\boldsymbol{G}_I^{\pi}) = |I|. \tag{3.10}$$

Then we use Lemma 3.4.3, more exactly the fact that $\boldsymbol{G}^*_{\psi(I)} = \boldsymbol{G}_I^{\pi}$, where π is the permutation that swaps the positions i with $2^m - i + 1$ for all $0 \leqslant i \leqslant 2^m - 1$. Hence $\operatorname{rank}(\boldsymbol{G}_I) = \operatorname{rank}(\boldsymbol{G}^*_{\psi(I)})$. □

Proposition 3.4.6. *Let $\mathscr{C}(I)$ be a monomial code, then its dual is a polynomial code given by*

$$\mathscr{C}(I)^{\perp} = \mathscr{C}\left(\psi(\mathcal{M}_m \setminus \check{I})\right).$$

Proof. First of all we apply Lemma 3.4.5 to obtain $\dim(\mathscr{C}(\psi(\mathcal{M}_m \setminus \check{I}))) = n - |I| = \dim(\mathscr{C}(I)^{\perp})$, since $\mathcal{M}_m \setminus \check{I}$ is a monomial set and $|\check{I}| = |I|$. Hence we have to prove only one inclusion.

Let $(f, g) \in \left(I, \psi(\mathcal{M}_m \setminus \check{I})\right)$ be a pair of elements such that $\deg(fg) \leqslant m - 1$. Since for any polynomial $P \in \mathbb{R}_m$ with $\deg(P) \leqslant m - 1$ we have $\mathsf{wt}(\mathsf{ev}(P)) = 0 \mod 2$, then we have that $\mathsf{ev}(f) \cdot \mathsf{ev}(g) = 0$. In other words the vectors corresponding to the evaluation of f and g are orthogonal.

Now suppose that $\deg(fg) = m$, which means that $fg = x_0 \ldots x_{m-1} + h(\boldsymbol{x})$ where $h \in \mathbb{R}_m$ with $\deg(h) < m$. Since $g \in \psi(\mathcal{M}_m \setminus \check{I})$ we have by definition that $g = (x_{i_1} + 1) \ldots (x_{i_s} + 1)$ with $x_{i_1} \ldots x_{i_s} \in \mathcal{M}_m \setminus \check{I}$ and $s < m$. This implies that $f \cdot (x_{i_1} + 1) \ldots (x_{i_s} + 1) = x_0 \ldots x_{m-1} + h(\boldsymbol{x})$, which means that $\check{f} | x_{i_1} \ldots x_{i_s}$.

There are two possible cases:

1. if $\check{f} = x_{i_1} \dots x_{i_s}$ it's impossible since $\check{f} \in \check{I}$ and $x_{i_1} \dots x_{i_s} \in \mathcal{M}_m \setminus \check{I}$.

2. if $\check{f} | x_{i_1} \dots x_{i_s}$ and $\deg(\check{f}) < s$ we have that it exists at least one variable x_{i_l} with $l \in \{1..s\}$ so that $\gcd(x_{i_l}, \check{f}) = 1$. But this implies that $x_{i_l} | f$ and furthermore

$$f \cdot (x_{i_1} + 1) \dots (x_{i_s} + 1) = 0$$

fact that is impossible since $\deg(h) < m$.

□

3.4.2 Dual of Weakly Decreasing Monomial Codes

Proposition 3.4.7. *Let I be a weakly decreasing monomial set, with respect to $\preceq_{\mathrm{w}}$. Then*

$$\mathscr{C}(I) = \mathscr{C}\left(\psi(I)\right).$$

Proof. Since for any $g \in \mathcal{M}_m$ we have $\psi(g) = \prod_{i \in \mathrm{ind}(g)} (x_i + 1) = \sum_{\substack{f | g \\ f \in \mathcal{M}_m}} f$, we deduce that $\mathscr{C}([1, g]_{\preceq_{\mathrm{w}}}) = \mathscr{C}(\psi([1, g]_{\preceq_{\mathrm{w}}}))$. Now using Definition 3.3.3 and 3.3.5 we have that $\mathscr{C}(I) = \mathscr{C}(\bigcup_{g_i \in I_{\max_{\preceq_{\mathrm{w}}}}} [1, g_i]_{\preceq_{\mathrm{w}}})$ and since ψ and $\cup$ commute, namely $\bigcup_{g_i \in I_{\max_{\preceq_{\mathrm{w}}}}} \psi([1, g_i]_{\preceq_{\mathrm{w}}} = \psi(\bigcup_{g_i \in I_{\max}} [1, g_i]_{\preceq_{\mathrm{w}}})$ we obtain the wanted result. □

Lemma 3.4.8. *For all f and g in $\mathcal{M}_m$, $f \preceq_{\mathrm{w}} g$ if and only if $\check{g} \preceq_{\mathrm{w}} \check{f}$.*

Proof. Let $f \preceq_{\mathrm{w}} g$. Then we have $g = f \gcd(f, g)$ and by definition

$$\check{g} = \frac{x_0 \dots x_{m-1}}{f \gcd(f, g)} = \frac{\check{f}}{\gcd(f, g)} \preceq_{\mathrm{w}} \check{f}.$$

For the second implication use the first result applied to $\check{f}$ and $\check{g}$ and the fact that $\check{\check{f}} = f$ for any monomial f. □

Example 3.4.9. *Take $m = 5$ and $f = x_0 x_1$ and $g = x_0 x_1 x_4$. Then we have $f \preceq_{\mathrm{w}} g$ and*

$$\check{g} = x_2 x_3 \preceq_{\mathrm{w}} \check{f} = x_2 x_3 x_4.$$

Corollary 3.4.10. *Let $I \subseteq \mathcal{M}_m$ be a weakly decreasing set. Then $\mathcal{M}_m \setminus \check{I}$ is a weakly decreasing set.*

Proof. Let h be a monomial that belongs to $\mathcal{M}_m \setminus \check{I}$, and let $g \in \mathcal{M}_m$ be a monomial such that $g \preceq_{\mathrm{w}} h$. Assume by contradiction that $g \notin \mathcal{M}_m \setminus \check{I}$, i.e. $g \in \check{I}$. Then there exists $f \in I$ such that $g = \check{f} \preceq_{\mathrm{w}} h$, which implies that $\check{h} \preceq_{\mathrm{w}} f$ by Lemma 3.4.8. Since I is a weakly decreasing set, $\check{h} \in I$, that is to say, $\check{\check{h}} = h \in \check{I}$ which contradicts the assumption. Therefore $\mathcal{M}_m \setminus \check{I}$ is a weakly decreasing set □

Proposition 3.4.11. *Let $\mathscr{C}(I)$ be a weakly decreasing monomial code. Then its dual is a weakly decreasing monomial code given by*

$$\mathscr{C}(I)^{\perp} = \mathscr{C}(\mathcal{M}_m \setminus \check{I}).$$

Proof. As $|\check{I}| = |I|$, we have $\dim \mathscr{C}(\mathcal{M}_m \setminus \check{I}) = |\mathcal{M}_m \setminus \check{I}| = N - \dim \mathscr{C}(I) = \dim \mathscr{C}(I)^\perp$, so we need to prove only one inclusion.

Let $f \in \mathcal{M}_m \setminus \check{I}$ and consider $g \in I$. If $\deg(fg) < m$ we have that $\mathsf{wt}(\mathsf{ev}(fg)) \equiv 0 \mod 2$, therefore the vectors $\mathsf{ev}(f)$ and $\mathsf{ev}(g)$ are orthogonal.

Now assume that $fg = x_0 \cdots x_{m-1}$. This means there exists $h \in \mathcal{M}_m$ such that $f = h\check{g}$, or equivalently $\check{g} \preceq_{\mathrm{w}} f$, that is to say $f \in \check{I}$ because I is a weakly decreasing set (and thanks to Lemma 3.4.8). Hence the inclusion $\mathscr{C}(\mathcal{M}_m \setminus \check{I}) \subseteq \mathscr{C}(I)^\perp$ is proved. □

Dual of Decreasing Monomial Code

The later results on the dual of a weakly decreasing monomial code are also valid for any decreasing monomial code.

Proposition 3.4.12. *Let $\mathscr{C}(I)$ be a decreasing monomial code. Then its dual is a decreasing monomial code given by*

$$\mathscr{C}(I)^\perp = \mathscr{C}(\mathcal{M}_m \setminus \check{I}).$$

Proof. Since the class of Decreasing Monomial codes inherits the properties of Weakly Decreasing Monomial codes (see Remark 3.3.13) we obtain the wanted result. □

Corollary 3.4.13. *Using the result on the dual of Decreasing Monomial codes we deduce a well known fact on the dual of Reed-Muller codes*

$$\begin{aligned} \mathscr{R}(r,m)^\perp &= \mathscr{C}(\mathcal{M}_m \setminus [x_0 \dots x_{m-r-1}; x_0 \dots x_{m-1}]_\preceq) \\ &= \mathscr{C}\left([1; x_{r+1} \dots x_{m-1}]_\preceq\right) = \mathscr{R}(m-r-1, m), \end{aligned}$$

A straightforward consequence of Proposition 3.4.12 is that under some conditions, any decreasing monomial code is weakly self-dual.

Corollary 3.4.14. *Let $\mathscr{C}(I)$ be a decreasing monomial code with $|I| \leqslant n/2$. Then $\mathscr{C}(I) \subseteq \mathscr{C}(I)^\perp$ if and only if for any $f \in I$, $\check{f} \notin I$, or in other words $I \subseteq \mathcal{M}_m \setminus \check{I}$.*

There are Decreasing Monomial codes for which the later condition in not satisfied. Consider for example $I = [1, x_0 \cdots x_{m-3}]_\preceq \cup [1, x_{m-2}x_{m-1}]_\preceq$. The dimension of the code $\mathscr{C}(I)$ is $\dim(\mathscr{C}(I)) = 1 + m(m+1)/2 + 2^{m-2}$. Then for the values of m that satisfy the inequality $1 + m(m+1)/2 + 2^{m-2} \leqslant 2^{m-1}$ we have that $\mathscr{C}(I)$ is a decreasing monomial code with rate smaller that 0.5, which is not weakly self-dual. This comes from the fact that $x_{m-2}x_{m-1}$ and $x_0 \cdots x_{m-3}$ belong to the set I but they do not satisfy the condition in Corollary 3.4.14.

Example 3.4.15. *Let $m = 3$ and*

$$I = \{1, x_0, x_1, x_0x_2\} \quad , \quad J = \{1, x_0, x_2, x_0x_2\} \quad , \quad L = \{1, x_0, x_1, x_0x_1\}.$$

We have that I is a monomial set, J is a weakly decreasing monomial set and L is a decreasing monomial set. We compute the multiplicative complement of the three sets and obtain

$$\begin{aligned} \check{I} &= \{x_0x_1x_2, x_1x_2, x_0x_2, x_1\} \\ \check{J} &= \{x_0x_1x_2, x_1x_2, x_0x_1, x_1\} \\ \check{L} &= \{x_0x_1x_2, x_1x_2, x_0x_2, x_2\} \end{aligned}$$

from which we obtain

$$\mathcal{M}_m \setminus \check{I} = \{1, x_0, x_2, x_0x_1\}$$
$$\mathcal{M}_m \setminus \check{J} = \{1, x_0, x_2, x_0x_2\}$$
$$\mathcal{M}_m \setminus \check{L} = \{1, x_0, x_1, x_0x_1\}$$

We obtain that $\mathscr{C}(J)^\perp = \mathscr{C}(J)$ *and* $\mathscr{C}(L)^\perp = \mathscr{C}(L)$. *As for the monomial code we have*

$$\mathscr{C}(I)^\perp = \mathscr{C}\left(\{1, x_0, x_2, x_0x_1 + x_1\}\right).$$

3.5 Minimum Distance

The minimum distance of Polar codes was already studied in the literature by Korada [Kor09]. The estimation of the minimum distance of a monomial code is quite similar and needs the following notion.

Definition 3.5.1. *Let* $\mathscr{C}(I)$ *be a monomial code over* m *variables. We let*

$$r_-(\mathscr{C}(I)) \stackrel{def}{=} \max\left\{r \mid \mathscr{R}(r,m) \subseteq \mathscr{C}(I)\right\}$$
$$r_+(\mathscr{C}(I)) \stackrel{def}{=} \min\left\{r \mid \mathscr{C}(I) \subseteq \mathscr{R}(r,m)\right\}$$

We notice that another way of defining these quantities is that r_- is the largest r for which the monomial $x_{m-r}\cdots x_{m-1}$ is in I. On the other hand r_+ is the largest integer r for which $x_0\cdots x_{r-1}$ is in I. These quantities are related to the minimum distance of a decreasing monomial code and its dual through the following result

Proposition 3.5.2. *Let* $\mathscr{C}(I)$ *be a weakly decreasing monomial code over* m *variables. We have the following properties:*

1. *The minimum distance of* $\mathscr{C}(I)$ *is equal to* $2^{m-r_+(\mathscr{C}(I))}$.
2. $r_-(\mathscr{C}(I)^\perp)$ *and* $r_+(\mathscr{C}(I)^\perp)$ *satisfy the equalities:*

$$r_-(\mathscr{C}(I)^\perp) = m - 1 - r_+(\mathscr{C}(I)) \tag{3.11}$$
$$r_+(\mathscr{C}(I)^\perp) = m - 1 - r_-(\mathscr{C}(I)) \tag{3.12}$$

3. *The minimum distance of* $\mathscr{C}(I)^\perp$ *is equal to* $2^{r_-(\mathscr{C}(I))+1}$

Proof. 1. We notice that r_+ is the largest degree of a monomial in I. If we consider the evaluation of any monomial in I_{r_+} we obtain a codeword of weight $2^{m-r_+(\mathscr{C}(I))}$. This implies that the minimum distance of $\mathscr{C}(I)$ is smaller than or equal to this quantity. On the other hand, the minimum distance of $\mathscr{C}(I)$ is larger than or equal to the minimum distance of $\mathscr{R}(r_+, m)$ which is equal to $2^{m-r_+(\mathscr{C}(I))}$, which implies our claim.

2. Equation (3.11) and (3.12) follow immediately from Proposition 3.4.11: $\mathscr{C}(I)^\perp = \mathscr{C}(\mathcal{M}_m \setminus \check{I})$ and the alternative definitions of $r_-(\mathscr{C}(I)^\perp)$ and of $r_+(\mathscr{C}(I)^\perp)$ which are respectively the largest degree r such that all monomials of degree r are monomials in $\mathcal{M}_m \setminus \check{I}$ and the largest degree of a monomial that belongs to $\mathcal{M}_m \setminus \check{I}$.

3. Using the two previous points we have

$$\begin{aligned}\mathsf{d}_{\min}(\mathscr{C}(I)^{\perp}) &= 2^{m-r_{+}(\mathscr{C}(I)^{\perp})} \\ &= 2^{m-(m-1-r_{-}(\mathscr{C}(I)))} \\ &= 2^{1+r_{-}(\mathscr{C}(I))}\end{aligned}$$

□

Corollary 3.5.3. *The same properties hold for any Decreasing Monomial codes.*

Proof. Since the class of Decreasing Monomial codes inherits the properties of Weakly Decreasing Monomial codes (see Remark 3.3.13) we obtain the wanted result. □

3.6 Permutation Group

3.6.1 Definitions and Properties

Applying an affine permutation to a monomial code yields a polynomial code but not necessarily a monomial code. Furthermore, polynomial codes and monomial codes may have a trivial permutation group. However by considering the subclass of Decreasing Monomial codes we obtain codes with a very large permutation group which is the *lower triangular affine group*. Before giving its precise definition, we introduce some notation.

Notation 3.6.1. *Binary square matrices with m rows (and m columns) are denoted by $\mathbb{F}_2^{m\times m}$.*

Let us recall that a bijective affine transformation over $\mathbb{F}_2^m$ can be represented by a pair $(\boldsymbol{A}, \boldsymbol{b})$ where $\boldsymbol{A}$ lies in the general linear group $\mathrm{GL}(m, 2)$ and $\boldsymbol{b}$ in $\mathbb{F}_2^m$. The action of $(\boldsymbol{A}, \boldsymbol{b})$ on a monomial g is denoted by $(\boldsymbol{A}, \boldsymbol{b}) \cdot g$. It basically consists in replacing each monomial x_i by a "new" monomial y_i defined by:

$$y_i = \sum_{j=0}^{m-1} a_{ij} x_j + b_i.$$

In the case of Decreasing Monomial codes, we are interested in a subclass of these transformations that are *lower triangular*. We recall that a matrix $\boldsymbol{A} = (a_{i,j})$ is lower triangular if $a_{i,j} = 0$ whenever $j > i$.

Definition 3.6.2. *Let $\boldsymbol{A} \in \mathbb{F}_2^{m\times m}$ be a lower triangular binary matrix with $a_{i,i} = 1$ and $\boldsymbol{b} \in \mathbb{F}_2^m$. Define the following affine transformation*

$$\begin{aligned}\mathcal{M}_m &\longrightarrow \mathbb{R}_m \\ \boldsymbol{x} &\longmapsto \boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}.\end{aligned}$$

We denote the set of such transformations $\mathrm{LTA}(m, 2)$.

Proposition 3.6.3. *The set* $\mathrm{LTA}(m, 2)$ *forms a group that we call* lower triangular affine group.

Proof. LTA$(m,2)$ can be decomposed into the semi-direct product of two groups and therefore it forms a group. The two groups are

- The group of translations $\mathrm{T}(m,2)$

$$\begin{aligned} \mathbb{F}_2^m &\longrightarrow \mathbb{F}_2^m \\ \boldsymbol{x} &\longmapsto \boldsymbol{x}+\boldsymbol{b}, \end{aligned}$$

  where $\boldsymbol{b} \in \mathbb{F}_2^m$.

- The subgroup of $\mathrm{GL}(m,2)$ defined by the set of all lower triangular matrices $\boldsymbol{A} \in \mathbb{F}_2^{m\times m}$ with $a_{i,i}=1$.

Then $(\mathrm{LTA}(m,2),\circ)$ is defined by:

$$(\boldsymbol{A},\boldsymbol{b}),(\boldsymbol{A}',\boldsymbol{b}') \in \mathrm{LTA}(m,2) \text{ we have } (\boldsymbol{A},\boldsymbol{b})\circ(\boldsymbol{A}',\boldsymbol{b}') = (\boldsymbol{A}\boldsymbol{A}', \boldsymbol{A}\boldsymbol{b}'+\boldsymbol{b})$$

□

3.6.2 Permutation group of Weakly Decreasing Monomial codes.

Proposition 3.6.4. *The permutation group of a weakly decreasing monomial code contains the group of translations* $\mathrm{T}(m,2)$.

Proof. Let I be a weakly decreasing set of monomials in m variables and $g = x_{i_1}\dots x_{i_s}$ be a monomial that belongs to I. Then consider en element $\boldsymbol{b} \in \mathrm{T}(m,2)$ which acts on g

$$x_{i_1}\dots x_{i_s} \mapsto (x_{i_1}+b_{i_1})\dots(x_{i_s}+b_{i_s}) = g^*.$$

Consider a subset of $\mathrm{ind}(g)$ that we denote $S_{\boldsymbol{b},g}$, for which $\forall j \in S_{\boldsymbol{b},g}\ , b_j = 0$. This set might be empty and in that case

$$g^* = \prod_{i\in\mathrm{ind}(g)} (x_i+1) = \sum_{f\preceq_{\mathrm{w}} g} f.$$

But since I is a weakly decreasing set all the monomials in the sum belong to I. Hence if $S_{\boldsymbol{b},g} = \emptyset$ we have $\mathsf{ev}(g^*) \in \mathscr{C}(I)$.

If $S_{\boldsymbol{b},g}$ is non empty then we have

$$g^* = \prod_{i\in S_{\boldsymbol{b},g}} x_i \prod_{j\in\mathrm{ind}(g)\setminus S_{\boldsymbol{b},g}} (x_j+1) = \prod_{i\in S_{\boldsymbol{b},g}} x_i \sum_{f\preceq_{\mathrm{w}} g_{\boldsymbol{b}}} f,$$

where $g_{\boldsymbol{b}} = \prod_{i\in\mathrm{ind}(g)\setminus S_{\boldsymbol{b},g}} x_i$. Therefore g^* equals the sum of all monomials $h \preceq_{\mathrm{w}} g$ such that $g/g_{\boldsymbol{b}} \preceq_{\mathrm{w}} h$. Hence $\mathsf{ev}(g^*) \in \mathscr{C}(I)$. □

Remark 3.6.5.

- *Since the order of the group* $\mathrm{T}(m,2)$ *equals* 2^m *we have that for any weakly decreasing monomial code* $|\mathsf{Perm}(\mathscr{C}(I))|/2^m \in \mathbb{N}^*$.

- *For the Reed-Muller codes we have* $|\mathsf{Perm}(\mathscr{R}(r,m))| = |\mathrm{GA}(m)| = 2^m \prod_{i=0}^{m-1}(2^m-2^i)$.

- *Nevertheless, the structure of the hole group of permutations for weakly decreasing monomial codes remains an opened question.*

3.6.3 Permutation group of Decreasing Monomial codes

Theorem 3.6.6. *The permutation group of a decreasing monomial code in m variables contains* $\mathrm{LTA}(m,2)$.

Proof. Let $\mathscr{C}(I)$ be a decreasing monomial code and let $(\boldsymbol{A},\boldsymbol{b})$ be in $\mathrm{LTA}(m,2)$. The action of $(\boldsymbol{A},\boldsymbol{b})$ where $\boldsymbol{A}=(a_{i,j})\in\mathbb{F}_2^{m\times m}$ and $\boldsymbol{b}\in\mathbb{F}_2^m$ can be viewed as a change of variables where x_i is replaced by the variable y_i defined by

$$y_i = x_i + \sum_{j=0}^{i-1} a_{ij}x_j + b_i.$$

Hence if $x_{i_1}\cdots x_{i_s}$ belongs to I with $0\leqslant i_1<\cdots<i_s\leqslant m-1$ then $y_{i_1}\cdots y_{i_s}$ is a linear combination involving only monomials of the form $\prod_{i\in J}x_i$ where J describes the powerset of $\{i_1,\ldots,i_s\}$. In particular, $\prod_{i\in J}x_i$ is in I since I is decreasing and therefore $\mathsf{ev}(y_{i_1}\cdots y_{i_s})$ belongs to $\mathscr{C}(I)$, which terminates the proof.

□

Remark 3.6.7.

- *The order of* $\mathrm{LTA}(m,2)$ *equals* $2^{m+\binom{m}{2}}$. *Therefore for any Decreasing Monomial code* $\mathscr{C}(I)$ *we have* $|\mathsf{Perm}(\mathscr{C}(I))|/2^{m+\binom{m}{2}}\in\mathbb{N}^*$.
- *For the Reed-Muller codes we have*

$$|\mathsf{Perm}(\mathscr{R}(r,m))| = |\mathrm{GA}(m)| = |\mathrm{LTA}(m,2)|\prod_{i=1}^{m}(2^i-1).$$

- *The problem of finding the hole permutation group remains open for Decreasing Monomial codes.*

In a recent work Soro, Lacan, Roca, Savin and Cunche [SLR+16] used a similar permutation in order to propose a recursive algorithm for decoding Reed-Muller over the Binary Erasure Channel. The algorithm exploits the Plotkin construction of Reed-Muller codes and the permutation group, in order to improve the generic decoder. Their idea might also be used in the case of any Decreasing Monomial code but it is not exploited for the moment. Nevertheless we exploit the permutation group of Decreasing Monomial codes to answer another fundamental question: what is the structure of minimum weight codewords of a Decreasing Monomial Code?

3.7 Minimum weight codewords

3.7.1 Orbits under the action of $\mathrm{LTA}(m,2)$

A natural object when dealing with group actions is the orbit of an element. We denote by

$$\mathrm{LTA}(m,2)\cdot g = \{(\boldsymbol{A},\boldsymbol{b})\cdot g \,|\, (\boldsymbol{A},\boldsymbol{b})\in\mathrm{LTA}(m,2)\} \text{ for } g\in\mathcal{M}_m$$

the orbit of a monomial g under the action of $\mathrm{LTA}(m,2)$.When g is equal to the monomial x_i then its orbits is of the form $\left\{x_i + \sum_{j=0}^{i-1} a_j x_j + b \mid a_j \text{ and } b \in \mathbb{F}_2\right\}$. A consequence is that the cardinality of the orbit of x_i equals 2^{i+1}.

When the degree of g is greater than 1 counting the number of elements in the orbit is less obvious. The reason why the task is more complicated comes from the fact that the stabilizer subgroup of $\mathrm{LTA}(m,2)$ with respect to g is not trivial. The following example illustrates this fact.

Example 3.7.1. *Let $g = x_0x_1$ then by definition*

$\mathrm{LTA}(m,2).g = \{(x_0+b_0)(x_1+a_{1,0}x_0+b_1) | b_0, a_{1,0}, b_1 \in \mathbb{F}_2\}$. *We remark that there are two group elements in* $\mathrm{LTA}(m,2)$ *that leave g invariant: $(x_0+b_0)(x_1+a_{1,0}x_0+b_1) = x_0x_1$ if and only if $b_0 = 0$ and $a_{1,0} = b_1$, in other words $x_0x_1 = x_0(x_1+x_0+1)$. Hence there are 4 distinct polynomials in the orbit of x_0x_1 which are x_0x_1, $x_0(x_1+1)$, $(x_0+1)x_1$ and $(x_0+1)(x_1+1)$.*

Definition 3.7.2. *For any g from $\mathcal{M}_m$ we define* $\mathrm{LTA}(m,2)_g$ *as the subgroup of $(\boldsymbol{A}, \boldsymbol{b}) \in$* $\mathrm{LTA}(m,2)$ *such that:*

$$b_i = 0 \text{ and } a_{ij} = 0 \quad \text{if} \quad i \notin \mathrm{ind}(g) \text{ or } j \in \mathrm{ind}(g).$$

Proposition 3.7.3. *For any monomial g in $\mathcal{M}_m$ the orbit of g under the action of* $\mathrm{LTA}(m,2)$ *is equal to the orbit of g under the action of* $\mathrm{LTA}(m,2)_g$*:*

$$\mathrm{LTA}(m,2) \cdot g = \mathrm{LTA}(m,2)_g \cdot g. \tag{3.13}$$

Proof. The inclusion $\mathrm{LTA}(m,2)_g \subseteq \mathrm{LTA}(m,2)$ is clear. We prove the converse inclusion by induction on $\deg g$.

Let $\deg g = 1$. We have that $g = x_i$ and by definition of $\mathrm{LTA}(m,2)$

$$\mathrm{LTA}(m,2) \cdot x_i = \{x_i + \sum_{j<i} \alpha_j x_j + b_i \text{ with } \alpha_j, b_i \in \mathbb{F}_2\}.$$

By definition of $\mathrm{LTA}(m,2)_g$ we have that $\boldsymbol{A} = \begin{pmatrix} 0 & \dots & \dots & \dots & \dots & 0 \\ a_{i0} & \dots & a_{i(i-1)} & 1 & \dots & 0 \\ 0 & \dots & \dots & \dots & \dots & 0 \end{pmatrix}$ and $\boldsymbol{b} = (0, \dots, 0, b_i, 0, \dots, 0)^t$. Therefore $\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b} = b_i + a_{i0}x_0 + \dots a_{i(i-1)}x_{i-1} + x_i$.

Let us assume that is true for any monomial of degree $\leqslant d$ where $d \geqslant 1$. Let g be a monomial in $\mathcal{M}_m$ of degree $d+1$. Let us consider $(\boldsymbol{A}, \boldsymbol{b})$ in $\mathrm{LTA}(m,2)$ and for any i in $\mathrm{ind}(g)$ we define y_i as:

$$y_i = x_i + \sum_{t=0}^{i-1} a_{it}x_t + b_i$$

Hence we have $(\boldsymbol{A}, \boldsymbol{b}) \cdot g = \prod_{j \in \mathrm{ind}(g)} y_j$. Now let i be the *maximum* element of $\mathrm{ind}(g)$. We then have:

$$y_i = x_i + \sum_{t \in \mathrm{ind}(g), t \neq i} a_{it}x_t + \sum_{t=0, t \notin \mathrm{ind}(g)}^{i-1} a_{it}x_t + b_i$$

Using the fact that $f^2 = f$ for any f in $\mathbb{R}_m$, we also have:

$$\begin{aligned}\prod_{j\in\mathrm{ind}(g)} y_j &= y_i \prod_{t\in\mathrm{ind}(g),t\neq i} y'_t \\ &= \left(x_i + \sum_{t\in\mathrm{ind}(g),t\neq i} a_{it}(x_t + 1 + y_t) \right. \\ &\quad \left. + \sum_{t=0,t\notin\mathrm{ind}(g)}^{i-1} a_{it}x_t + b_i\right) \prod_{t\in\mathrm{ind}(g),t\neq i} y_t\end{aligned}$$

Since $\prod_{t\in\mathrm{ind}(g),t\neq i} y_t$ is of degree d then by induction assumption, there exists $(\boldsymbol{A}^*, \boldsymbol{b}^*) \in \mathrm{LTA}(m,2)_g$ such that for any $t \in \mathrm{ind}(g)$ and $t \neq i$, it holds:

$$y_t = x_t + \sum_{s=0,s\notin\mathrm{ind}(g)}^{t-1} a^*_{ts}x_s + b^*_t.$$

This implies in particular that we can write:

$$\sum_{t\in\mathrm{ind}(g),t\neq i} a_{it}(x_t + 1 + y_t) = \sum_{\substack{t\in\mathrm{ind}(g)\\ t\neq i}} \sum_{\substack{s=0\\ s\notin\mathrm{ind}(g)}}^{t-1} a_{it}(a^*_{ts}x_s + b^*_t + 1)$$

This last equation only involves variables x_s with $0 \leqslant s < i$ and $s \notin \mathrm{ind}(g)$. Hence we can find a binary vector $(a'_{i0}, \ldots, a'_{i(i-1)})$ with $a'_{it} = 0$ if $t \in \mathrm{ind}(g)$, and $b'_i \in \mathbb{F}_2$ such that:

$$\prod_{t\in\mathrm{ind}(g)} y_t = \prod_{t\in\mathrm{ind}(g)} \left(x_t + \sum_{s=0,s\notin\mathrm{ind}(g)}^{t-1} a'_{ts}x_s + b'_t\right).$$

This last equality proves $\mathrm{LTA}(m,2)\cdot g \subseteq \mathrm{LTA}(m,2)_g \cdot g$ and concludes the proof. □

Proposition 3.7.4. *For any $g \in \mathcal{M}_m$ we have*

$$|\mathrm{LTA}(m,2)\cdot g| = |\mathrm{LTA}(m,2)_g|\,.$$

Proof. From Proposition 3.7.3 we have that $|\mathrm{LTA}(m,2)\cdot g| \leqslant |\mathrm{LTA}(m,2)_g|$.

Let $g \in \mathcal{M}_m$ and let us consider $(\boldsymbol{A}, \boldsymbol{b})$ and $(\boldsymbol{A}', \boldsymbol{b}')$ in $\mathrm{LTA}(m,2)_g$. We prove that if $(\boldsymbol{A}, \boldsymbol{b})\cdot g = (\boldsymbol{A}', \boldsymbol{b}')\cdot g$ in $\mathbb{R}_m$ then $\boldsymbol{A} = \boldsymbol{A}'$ and $\boldsymbol{b} = \boldsymbol{b}'$.

This comes from the fact that in the polynomial $(\boldsymbol{A}, \boldsymbol{b})\cdot g \in \mathbb{R}_m$, the coefficient of $x_j \prod_{k\in\mathrm{ind}(g),k\neq i} x_k$ when $i \in \mathrm{ind}(g)$ and $j \notin \mathrm{ind}(g)$ is exactly a_{ij} and the coefficient of $\prod_{k\in\mathrm{ind}(g),k\neq i} x_k$ is b_i. This proves that $(\boldsymbol{A}, \boldsymbol{b}) = (\boldsymbol{A}', \boldsymbol{b}')$.

Therefore there is a bijection between the two sets and counting the number of elements in the orbit of g is equivalent to counting the number of pairs $\boldsymbol{A}, \boldsymbol{b} \in \mathrm{LTA}(m,2)_g$. □

3.7.2 Computing the cardinality of orbits

In order to give the cardinality of an orbit we use a well-known combinatorial object called the Ferrers diagram (or Young diagram).

Young diagrams

Definition 3.7.5 ([Com12]). *A* Young diagram *is a finite collection of boxes arranged in left-justified rows, with the rows sizes weakly increasing.*

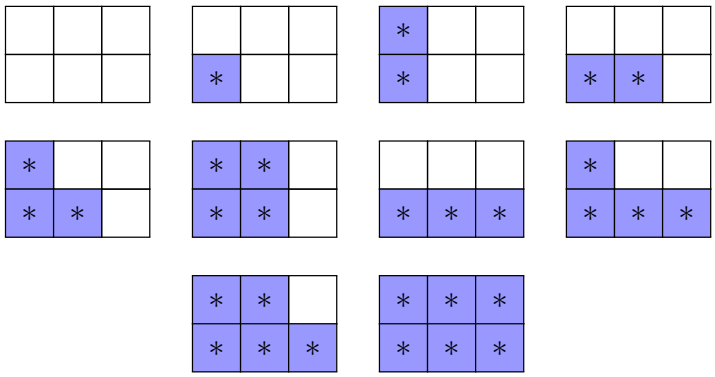


Figure 3.9 – Young diagrams inside a 2×3 grid

The diagram is generally used to represent a partition λ of integers. In the above figure we have the following partitions from left to right:

$$\varepsilon, (1), (1,1), (2), (2,1), (2,2), (3), (3,1), (3,2), (3,3).$$

Notation 3.7.6. *Let m and d be two positive integers such that $d \leqslant m$. We denote by $\lambda \subset d \times (m-d)$ the set of all partitions $\lambda = (\lambda_{d-1}, \ldots, \lambda_0)$ inside the $d \times (m-d)$ grid, where $0 \leqslant \lambda_i \leqslant m-d$ and $\lambda_0 \leqslant \cdots \leqslant \lambda_{d-1}$.*

In the literature the usual convention is to have $1 \leqslant \lambda_i \leqslant m-d$. But here we prefer to write the zero elements since we consider partitions inside a fixed grid.

Example 3.7.7. *For the diagrams in Figure3.9 the partitions, using our convention, are:*

$$(0,0), (1,0), (1,1), (2,0), (2,1), (2,2), (3,0), (3,1), (3,2), (3,3).$$

Bijection between monomials and Young diagrams

We construct a bijection between Young diagrams in grids of size $d \times (m-d)$ and monomials of degree d in m variables. First of all we associate to a monomial g the corresponding matrix $\boldsymbol{A} \in \mathrm{LTA}(m,2)_g$. Then we will use a well known bijection between the Young diagrams and the matrices in row echelon form [Knu71a].

The bijection works as follows: if $(\boldsymbol{A}, \boldsymbol{b}) \in \mathrm{LTA}(m,2)_g$, then by definition of $\boldsymbol{A}$ the rows $i \notin \mathrm{ind}(g)$ and the columns $j \in \mathrm{ind}(g)$ contains only a 1 on the diagonal (and 0 elsewhere). If we remove from $\boldsymbol{A}$ the rows $i \notin \mathrm{ind}(g)$ and the columns $j \in \mathrm{ind}(g)$, we get a $d \times (m-d)$ matrix with possible non-zero coefficients exactly inside the boxes of the associated Ferrers diagram.

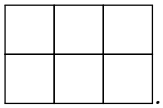
Proposition 3.7.8. *For any integers m, d with $1 \leqslant d \leqslant m$, there is a bijection between monomials in $\mathcal{M}_m$ of degree d and Young diagrams inside the $d \times (m-d)$ grid.*

Proof. If $g = x_{i_0} \cdots x_{i_{d-1}} \in \mathcal{M}_m$ is a monomial of degree d, then the partition associated to g is $\lambda_g = (i_{d-1} - (d-1), i_{d-2} - (d-2), \ldots, i_0 - 0)$ inside the $d \times (m-d)$ grid. It is a partition since $i_k - k \geqslant i_{k-1} - (k-1)$.

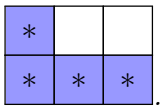
If $\lambda = (\lambda_{d-1}, \ldots, \lambda_0)$ is a partition inside the $d \times (m-d)$ grid with $\lambda_{d-1} \geqslant \lambda_{d-2} \geqslant \cdots \geqslant \lambda_0 \geqslant 0$, then the monomial g associated to it is $g = x_{i_1} \cdots x_{i_d}$ where $i_k = \lambda_k + k > \lambda_{k-1} + k - 1 = i_{k-1}$. □

Example 3.7.9. *We consider in this example that $m = 5$ and the monomial g has* $\deg(g) = 2$.

- *Let λ_g be the empty partition $\lambda_g = \varepsilon$ inside the 2×3 grid. With our convention $\lambda_g = (0, 0)$ and we have $g = x_0 x_1$.*



- *Now let $g = x_1 x_4$ then the partition associated to g is $\lambda_g = (4-1, 1-0) = (3, 1)$ and it's Young diagram in the 2×3 grid is*



We also illustrate the bijection between $\boldsymbol{A}$ and λ_g.

$$\boldsymbol{A} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ a_{10} & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ a_{40} & 0 & a_{42} & a_{43} & 1 \end{pmatrix}.$$

After deleting the rows corresponding to x_0, x_2, x_3 and the columns corresponds to x_1, x_4, we get $\begin{pmatrix} a_{10} & 0 & 0 \\ a_{40} & a_{42} & a_{43} \end{pmatrix}$ which corresponding to the 8^{th} Ferrers diagram from Figure 3.9.

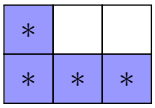
Using the same technique we obtain that the monomials associated to the Young diagrams from Figure 3.9 are (in the same order as in the fugure): x_0x_1, x_0x_2, x_1x_2, x_0x_3, x_1x_3, x_2x_3, x_0x_4, x_1x_4, x_2x_4 and x_3x_4.

We denote by λ_g the Ferrers diagram corresponding to g and $|\lambda_g|$ the size of a diagram, that is to say the number of * in the diagram. Thanks to Proposition 3.7.4 we can state the following.

Proposition 3.7.10. *The cardinality of the orbit of g under the action of* $\mathrm{LTA}(m, 2)$ *is*

$$|\mathrm{LTA}(m, 2) \cdot g| = 2^{\deg(g) + |\lambda_g|}$$

Example 3.7.11. *Let $m = 5$ and $g = x_1x_4$ then the partition associated to g is $\lambda_g = (4-1, 1-0) = (3,1)$ and it's Young diagram in the 2×3 grid is*



Using Proposition 3.7.10 we have that the cardinality of the orbit of x_1x_4 equals $2^{2+4} = 2^6$.

We can also use the matrix representation, which is in this case $\begin{pmatrix} a_{10} & 0 & 0 \\ a_{40} & a_{42} & a_{43} \end{pmatrix}$ We deduce that there are 2^4 different matrices $\boldsymbol{A}$ in $\mathrm{LTA}(m,2)_g$, and 2^2 different vectors $\boldsymbol{b}$ which gives a cardinality equal to 2^6

3.7.3 The minimum weight codewords of a decreasing monomial code.

Characterizing the minimum weight codewords is often quite difficult and there are few families of codes for which the structure of the minimum weight codewords is well known. In the case of Decreasing Monomial codes the subgroup $\mathrm{LTA}(m,2)$ gives enough information to understand the structure of the minimum weight codewords.

Definitions and Properties

We suppose that $\mathscr{C}(I)$ is a decreasing monomial code. From Proposition 3.5.2, the set of minimum weight codewords is

$$\mathrm{W}_{\min} = \{\boldsymbol{c} \in \mathscr{C}(I) \mid |\boldsymbol{c}| = 2^{m-r_+}\},$$

where $r_+ = r_+(\mathscr{C}(I))$.

Proposition 3.7.12.

$$\mathrm{W}_{\min} = \left\{\mathsf{ev}(P) \mid \exists f \in I_{r_+},\ P \in \mathrm{LTA}(m,2) \cdot f\right\}.$$

Proof. The $\supseteq$ inclusion comes from the fact that $\mathrm{LTA}(m,2)$ acts on $\mathscr{C}(I)$ as a permutation, and thus for any $f \in I_{r_+}$, $\mathsf{ev}(f)$ has weight 2^{m-r_+}.

As for the $\subseteq$ inclusion, consider an arbitrary element $\mathsf{ev}(P) \in \mathrm{W}_{\min}$. From [KTA76] we know that an element of weight 2^{m-r_+} is the evaluation of a product of r_+ independent affine linear forms $P = \prod_{j=1}^{r_+} \ell_j$. Each linear form can be written as

$$\ell_j = x_{i_j} + \sum_{0 \leqslant k < i_j} a_{jk}x_k + \varepsilon_j.$$

If all the maximal variables x_{i_j} in the r_+ linear forms are pairwise distinct then $P \in \mathrm{LTA}(m,2) \cdot (x_{i_1} \dots x_{i_{r_+}})$.

Now suppose that this assumption is not true and consider two linear forms ℓ_1, ℓ_2 sharing the same maximum variable and such that $\ell_1\ell_2 \neq 0$. In other words $\ell_1 = x_{i_2} + \sum_{k<i_2} a_{1k}x_k + \varepsilon_1$

and $\ell_2 = x_{i_2} + \sum_{k<i_2} a_{2k}x_k + \varepsilon_2$. Let $\ell_1^* = x_{i_2} + \ell_1$ and $\ell_2^* = x_{i_2} + \ell_2$. It is clear that $\ell_1^* + \ell_2^* \neq 0$ because of the independence condition. We also have that $\ell_1^* + \ell_2^* \neq 1$ because if not we would obtain $\ell_1\ell_2 = 0$. Using the relation $f^2 = f$ for any $f \in \mathbb{R}_m$ we can rewrite

$$\ell_1\ell_2 = (\ell_1 + \ell_2 + 1)\,\ell_2 = (\ell_1^* + \ell_2^* + 1)\,\ell_2 = \ell_1'\ell_2$$

where the maximum variable of ℓ_1' is strictly smaller than x_{i_2} and the two forms are independent.

By induction on the number of linear forms, we can prove that any product of r_+ linear forms can be rewritten in a product of r_+ linear independant forms with distincts maximal variable. □

Furthermore we prove that for any two distinct monomials the intersection of their orbits is the empty set and conclude with the formula counting the number of minimum weight codewords of a decreasing monomial code.

Lemma 3.7.13. *Let f and g be two different monomials in $\mathcal{M}_m$. Then the intersection of their orbits is equal to the empty set.*

$$\mathrm{LTA}(m,2)\cdot f \cap \mathrm{LTA}(m,2)\cdot g = \emptyset.$$

Proof. If the two monomial can be compared with respect to our order then we can consider without loss of generality that $f \preceq g$. Using the definition of $\mathrm{LTA}(m,2)$ we have that any polynomial in the orbit of g contains the monomial g and any polynomial in the orbit of f does not contain the monomial g. So it is impossible to find a polynomial belonging to both orbits.

If the two monomials are incomparable the proof works in the same way. Let f and g be two monomials such that $f = \gcd(f,g)f^*$ and $g = \gcd(f,g)g^*$. Since f and g are incomparable we know that there exists a maximum variable x_i such that $x_i | f$ and $x_i \nmid g$ and $\forall j \in \mathrm{ind}(g)\ , i > j$. Therefore any polynomial in the orbit of f contains the monomial $x_i \gcd(f,g)$. On the other hand since the variable x_i is bigger than all the variables in g^* the monomial $x_i \gcd(f,g)$ in not contaned in any of the polynomials in the orbit of g. □

Computing the number of minimum weight codewords

Theorem 3.7.14. *Let $\mathscr{C}(I)$ be a decreasing monomial code, then the number of minimum weight codewords in $\mathscr{C}(I)$ equals*

$$|\mathrm{W}_{\min}| = 2^{r_+} \sum_{g \in I_{r_+}} 2^{|\lambda_g|}.$$

Proof. Use Proposition 3.7.12 and Lemma 3.7.13 □

Corollary 3.7.15. *The number of minimum weight codewords in $\mathscr{R}(r,m)$ equals*

$$|\mathrm{W}_{\min}(\mathscr{R}(r,m))| = 2^r \begin{bmatrix} m \\ r \end{bmatrix}_2$$

where $\begin{bmatrix} m \\ r \end{bmatrix}_2 = \dfrac{(2^m-1)\dots(2^m-2^{r-1})}{(2^r-1)\dots(2^r-2^{r-1})}$ *is the Gaussian binomial coefficient.*

Proof. Recall that $\begin{bmatrix} m \\ r \end{bmatrix}_2$ represents the number of r-dimensional subspaces of $\mathbb{F}_2^m$. The problem of counting the number of r-dimensional subspaces of $\mathbb{F}_2^m$ is equivalent to counting the number of $r \times m$ matrices of rank r in reduced echelon form. Each matrix gives rise to a Young diagram inside the $r \times (m-r)$ grid and each diagram λ can be obtained from $2^{|\lambda|}$ matrices [Knu71a]. So we have the following combinatorial identity:

$$\begin{bmatrix} m \\ r \end{bmatrix}_2 = \sum_{\lambda \subset r \times (m-r)} 2^{|\lambda|}. \tag{3.14}$$

Moreover we recall that $\mathscr{R}(r,m) = \mathscr{C}\left([1, x_{m-r} \dots x_{m-1}]\right)$, which implies that the Young diagrams corresponding to all the maximum degree monomials of a Reed-Muller Code are all the possible diagrams in the $r \times (m-r)$ grid. Thus we have

$$|\mathrm{W}_{\min}(\mathscr{R}(r,m))| = 2^r \sum_{\lambda \subset r \times (m-r)} 2^{|\lambda|} = 2^r \begin{bmatrix} m \\ r \end{bmatrix}_2$$

□

Remark 3.7.16. *We notice that the number of minimum weight codewords of the Reed-Muller codes represent an upper bound on the number of minimum weight codewords of any Decreasing Monomial code $\mathscr{C}(I)$*

$$|\mathrm{W}_{\min}(\mathscr{C}(I))| \leqslant |\mathrm{W}_{\min}(\mathscr{R}(r_+, m))|.$$

	Decreasing Monomial	Weakly Decreasing Monomial	Monomial
$\mathscr{C}(I)^\perp$	$\mathscr{C}(\mathcal{M}_m \setminus \check{I})$	$\mathscr{C}(\mathcal{M}_m \setminus \check{I})$	$\mathscr{C}\left(\psi(\mathcal{M}_m \setminus \check{I})\right)$
$\mathsf{d}_{\min}(\mathscr{C}(I))$	2^{m-r_+}	2^{m-r_+}	2^{m-r_+}
$\mathsf{Perm}\left(\mathscr{C}(I)\right)$	$\mathrm{LTA}(m,2) \subseteq \mathsf{Perm}\left(\mathscr{C}(I)\right)$	$\mathrm{T}(m,2) \subseteq \mathsf{Perm}\left(\mathscr{C}(I)\right)$	
$\lvert\mathrm{W}_{\min}\rvert$	$2^{r_+} \sum_{g \in I_{r_+}} 2^{\lvert\lambda_g\rvert}$		

Figure 3.10 – Main properties of Monomial, Weakly Decreasing Monomial and Decreasing Monomial codes

3.8 Perspectives

There are many questions related to this topic and we do not pretend to select here the most important ones. We just enumerate some of the subjects on which we tend to focus in the near future.

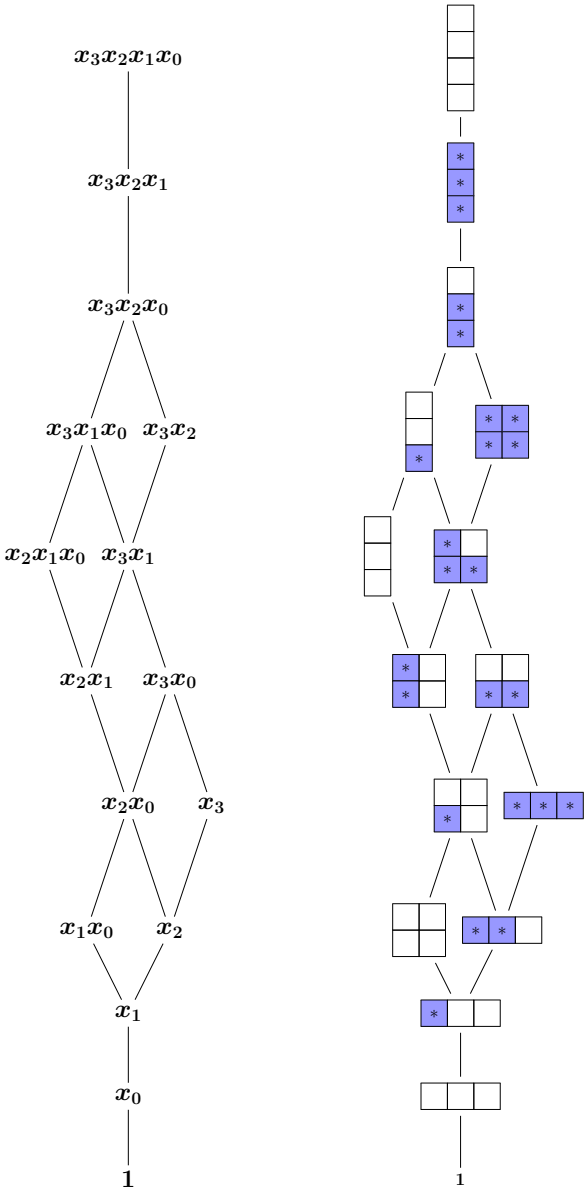


Figure 3.11 – The decreasing order over the Young diagrams corresponding to the monomials when $m = 4$

1. The first topic is a combinatorial question concerning the number of Decreasing Monomial codes of fixed dimension, let's say k. If the answer if obvious for the class of Monomial codes, that is $\binom{n}{k}$, in the case of Weakly Decreasing Monomial codes and Decreasing Monomial codes the question doesn't have a trivial answer. The main direction that we that we tend to follow is a detail study of ideals of posets.

2. Experiments show that the order of the full permutation group for Decreasing Monomial codes is way bigger that the order of the lower triangular group. In many cases it is close up to a constant to the general affine group. Therefore to reveal the full permutation group for the Decreasing Monomial codes is still an opened question that could unlock other important results.

3. Since both Reed-Muller codes and Polar codes achieve the capacity of the Binary Erasure Channel, a natural question is whether any Decreasing Monomial code achieve the capacity of the BEC. For that we believe that the previous question, regarding the permutation group might be extremely useful. Also the results that we obtained on the structure of the minimum weight codewords could be potentially one of the ideas to be used in this sense.

4 Cryptanalysis of the McEliece scheme based on Polar Codes

4.1 Introduction

Since Polar codes benefit of various interesting properties like fast encoding and decoding algorithms and high decoding capacity they seem fitted in cryptographic applications. They were proposed in a public key encryption scheme à la McEliece by [SK14] and [HSEA14]. as well as for a secret key cryptosystem in [HSA13]. Due to their high decoding capacity they might also be suitable for signature schemes [OT12].

Contribution We propose in this Chapter to analyze the security of the McEliece variant based on Polar codes. More exactly we focus on the Key Recovery Problem and show that in the case of Polar codes it can be reduced to the Code Equivalence Problem. Based on the results that we proved in Chapter 3 we explain that generic algorithms, like the SSA [Sen00], for solving this problem are not feasible due to the structural properties of Polar codes.

Our main contribution is to propose a Key Recovery Attack against the McEliece scheme based on Polar codes. The advantage of our algorithm is that it can be applied on any Decreasing Monomial code that admits a specific discriminant that we detail in Section 4.4.2. The signature that we propose for our attack is based on the number of minimum weight codewords in the dual of a shortened Polar code. With this tool we manage to discriminate between several monomials that define the Polar code fact that allows us to solve the code equivalence problem. We detail our algorithm on a toy example, over a small extension of $\mathbb{F}_2$, and also give some details about a practical implementation applicable for cryptographic parameters. The results in this Section were published in [BCD+16].

4.2 The McEliece PKC variant using Polar codes

4.2.1 Introduction

Shrestha and Kim proposed a public key encryption scheme à la McEliece based on Polar codes [SK14]. The idea is the same as in the original scheme, it only replaces

the binary Goppa code with a Polar code designed for a BSC(p). Therefore during the KeyGeneration algorithm, one has to compute the Bhattacharya parameter for all the 2^m synthetic channels and choose the best k channels in order to output the generator matrix $\boldsymbol{G}$ of the Polar code. The masking technique is the usual one, using a permutation and an invertible matrix. Moreover the encryption and decryption algorithms work exactly in the same way as in the original version (see Section 2.3 for details).

4.2.2 Parameters for the scheme

The authors in [SK14] proposed as practical parameters, codes of length 2048 and rate close to 1/3, more exactly a $[2048, 614]$ Polar code. The Shannon limit for the noise on a binary symmetric channel of crossover probability p that a code of rate $\frac{614}{2048}$ is able to sustain is about $p = 0.19$. So in order to built the generator matrix of the Polar code we have to choose the best 614 rows of $\boldsymbol{G}_{11}$ which give the best performance for the successive cancellation decoder. Regarding the performance of the SC decoder we illustrate in Figure 4.1 the decoding error probability in function of the weight of the error vector in the case of a $[2048, 614]$ Polar code designed for a BSC(0.19).

$\mathsf{wt}(\boldsymbol{e})$	143	164	184	205	225	246	266
$\mathrm{P_{err}}(\boldsymbol{e})$	2.10^{-6}	$3,2.10^{-5}$	$2,56.10^{-4}$	$1,02.10^{-3}$	$3,37.10^{-3}$	$1,49.10^{-2}$	$4,76.10^{-2}$

Figure 4.1 – Error probability for a $[2048, 614]$ Polar code in function of the error weight

If we suppose that the scheme has to be deployed in a practical environment where a reasonable error probability would be lower than or equal to 10^{-3}, then we choose to consider error vectors with Hamming weight less than 200.

4.2.3 Security arguments

Shrestha and Kim proposed a security analysis based on two aspects: firstly they analyze the complexity of the brute force attack and secondly they give some arguments against the eventual use of the attack against the Sidelnikov cryptosystem [Sid94]. But there is no reference to any generic attack, therefore we give in Figure 4.2 the security level against the MRA, for a $[2048, 614]$ Polar code in function of the error vector weight, with respect to the complexity of the ISD algorithm.

$\mathsf{wt}(\boldsymbol{e})$	143	164	184	205	225	246	266
security level	78	89	100	111	123	135	146

Figure 4.2 – Security level against the MRA of the $[2048, 614]$ Polar code in function of the error weight

We notice that in this case we could get a 100-bit security if we choose error vectors with a Hamming weight at least 184. In this case the error probability equals $2.56 \cdot 10^{-4}$ which is reasonable for a practical purpose.

For the Key Recovery Attack, we face in the case of Polar codes the Code Equivalence Problem. There is in essence a single Polar code of a given dimension and length. So breaking the scheme amounts to find for a permuted version of the Polar code a permutation that gives the original Polar code. What makes this problem difficult for Polar codes is that the standard algorithm for solving it, namely the Support Splitting Algorithm [Sen00] is too complex to be used in this context due to the very large size of the hull of the Polar code (see Section 3.4). What makes the problem even more intricate is the fact that a Polar code have a very large permutation group (see Section 3.6) which complicates the task significantly.

So it seems that this variant is not threatened by any known key recovery attack, at least not an obvious one. Moreover a proper choice of parameters seems to be possible, parameters that give the security of the scheme with respect to the MRA attack. Despite all these arguments for the use of Polar codes, we will reveal the existence of a flawless in this case, which is represented by the set of minimum weight codewords. In the case of a $[2048, 614]$ Polar code the minimum distance equals 32, which is way lower than the minimum error weight that has to be considered for a proper security level. On top of that we have just seen in the previous section that the structure of the minimum weight codewords is given by the action of the lower triangular affine group. Therefore there might be a distinguisher on the maximum degree monomials using this group action, which is exactly the idea that we will use against this McEliece variant.

4.3 Solving the code equivalence problem for Polar codes

In order to solve the code equivalence problem for Polar codes we will need to define the notion of signature for a linear code.

4.3.1 Definitions

Definition 4.3.1 (Signature)**.** *Let $\mathscr{C}$ be a code of length n. Let $\mathcal{G}$ be a subgroup of permutations of $\mathscr{C}$ and* W *be a subset of $\mathscr{C}$ globally invariant under $\mathcal{G}$. We say that a function $\Sigma(\boldsymbol{c}, \mathscr{C})$ where $\boldsymbol{c}$ belongs to $\mathscr{C}$ is a signature for the action of $\mathcal{G}$ on* W *if and only if:*

1. *$\Sigma(\boldsymbol{c}, \mathscr{C}) = \Sigma(\boldsymbol{c}^\pi, \mathscr{C}^\pi)$ for π from S_n (i.e. Σ is invariant by permutation),*
2. *$\Sigma(\boldsymbol{c}, \mathscr{C}) \neq \Sigma(\boldsymbol{c}', \mathscr{C})$ if $\boldsymbol{c}$ and $\boldsymbol{c}'$ both belong to* W *but are not in the same orbit under $\mathcal{G}$ (i.e. Σ takes distinct values for each orbit).*

Notice here that a signature always takes the same value on an orbit under $\mathcal{G}$ since if we take $\boldsymbol{c}$ in W and γ is an element of $\mathcal{G}$, then $\Sigma(\boldsymbol{c}, \mathscr{C}) = \Sigma(\boldsymbol{c}^\gamma, \mathscr{C}^\gamma) = \Sigma(\boldsymbol{c}^\gamma, \mathscr{C})$ since γ belongs to the permutation group of the code.

The main idea of our attack is that we are able to find an efficient signature for the Polar codes $\mathscr{C}(I)$. This signature will allow us to determine the orbit of a particular monomial $f \in I$ under the $\mathrm{LTA}(m, 2)$. Furthermore we will proceed by induction, more exactly we will determine the monomials f_i which divide f and solve the code equivalence problem for the shortened code on the support of f_i.

4.3.2 Preliminaries

Before we detail the attack we recall some of the properties of Polar codes that are going to be used. We will suppose that the Polar code of length $n = 2^m$ and dimension k is defined by the set of monomials I and that the maximum degree of monomials in I equals r_+. We will denote as before the set of monomials of degree r_+ by $I_{r_+} \subset I$. Next we recall the necessary ingredients for a successful cryptanalysis of Polar codes:

- From Theorem 3.6.6 we know that the permutation group of a decreasing monomial code in m variables contains $\mathrm{LTA}(m, 2)$.
- The minimum weight codewords of a Polar code $\mathscr{C}(I)$ are given by the evaluation of the polynomials in the orbits $\mathrm{LTA}(m, 2) \cdot f$ for any maximum degree monomial $f \in I_{r_+}$.
- Since any Polar code is a decreasing monomial code and $I_{r_+} \neq \emptyset$ we have that $x_0 \dots x_{r_+-1} \in I_{r_+}$. We denote the corresponding codeword by $\boldsymbol{c}_{\min} \stackrel{\text{def}}{=} \mathsf{ev}(x_0 \cdots x_{r_+-1})$

Notation 4.3.2. *Let $\boldsymbol{c}_{min} \stackrel{def}{=} \mathsf{ev}(x_0 \cdots x_{r_+-1})$, than we denote by $\mathcal{I}$ be the support of $\boldsymbol{c}_{min}$, and $\mathcal{J}$ be the complementary set (that is the set of position for which $\boldsymbol{c}_{min}$ takes the value 0).*

Let $\boldsymbol{c}^i = \mathsf{ev}(x_0 \dots, x_{i-1})$ with $\boldsymbol{c}^0$ being $\mathsf{ev}(1)$, that is the all-one codeword and $\boldsymbol{c}_{min} = \boldsymbol{c}^{r_+}$. We denote by $\mathcal{I}^i$ the support of $\boldsymbol{c}^i$ and $\mathcal{J}^i$ be the complementary set. We remark that we have $\mathcal{I}^{r_+} = \mathcal{I}$ and $\mathcal{J}^{r_+} = \mathcal{J}$.

4.3.3 Attack algorithm.

The algorithm for performing the attack can now be summarized as follows:

Step 1. (Minimum weight codewords searching) Search the non-zero minimum weight vectors of $\mathscr{C}(I)$ and $\mathscr{C}(I)^\pi$, that we denote by $\mathrm{W}_{\min}$ and ${\mathrm{W}_{\min}}^\pi$ respectively. We know from Properties 4.3.2 that $\boldsymbol{c}_{\min}$ belongs to $\mathrm{W}_{\min}$.

Step 2. (Signature of orbits in $\mathrm{W}_{\min}$) Compute the orbits of $\mathrm{W}_{\min}$ under the action of $\mathrm{LTA}(m, 2)$ and find a signature for these orbits. This signature is based on shortening the dual $\mathscr{C}(I)^\perp$ on the support of $\boldsymbol{c}$ (where $\boldsymbol{c}$ belongs to $\mathrm{W}_{\min}$) and computing the dimension of this code and the number of codewords of minimum weight in it.

Step 3. (Computation of orbits in ${\mathrm{W}_{\min}}^\pi$) Use this signature to decompose ${\mathrm{W}_{\min}}^\pi$ into distinct orbits under the group $\pi^{-1}\mathrm{LTA}(m, 2)\pi$ and use it to find the orbit of $\boldsymbol{c}^\pi_{\min}$.

Step 4. (Identification of affine spaces) Without loss of generality, we may take any codeword in the orbit of $\boldsymbol{c}^\pi_{\min}$ and declare that it is equal to $\boldsymbol{c}^\pi_{\min}$. The structure of the orbit of $\boldsymbol{c}_{\min}$ is such that the supports of all the codewords in this orbit are affine spaces of the form $x_0 = \varepsilon_0,\ x_1 = \varepsilon_1, \dots, x_{r_+-1} = \varepsilon_{r_+-1}$, where the ε_i's are arbitrary elements in $\mathbb{F}_2$. Denote this affine space by $A(\varepsilon_0, \dots, \varepsilon_{r_+-1})$ and let $\boldsymbol{c}_{\min}(\varepsilon_0, \dots, \varepsilon_{r_+-1})$ be the corresponding codeword. Up to a permutation of $\mathscr{C}^\pi$, we identify all the elements $\boldsymbol{c}_{\min}(\varepsilon_0, \dots, \varepsilon_{r_+-1})^\pi$. This gives all the affine spaces permuted by π, that is $A(\varepsilon_0, \dots, \varepsilon_{r_+-1})^\pi \stackrel{\text{def}}{=} \{\pi^{-1}(i) \mid i \in A(\varepsilon_0, \dots, \varepsilon_{r_+-1})\}$.

Step 5. (Equivalence problem for a short code) We compute the codes $\mathscr{D} \stackrel{\text{def}}{=} \mathcal{P}_{\mathcal{J}}(\mathscr{C})$ and $\mathscr{D}^{\pi} \stackrel{\text{def}}{=} \mathcal{P}_{\mathcal{J}^{\pi}}(\mathscr{C}^{\pi})$ and solve the code equivalence problem for $\mathscr{D}$ and $\mathscr{D}^{\pi'}$ where π' is the restriction of the permutation π to the affine space $\mathcal{I}$. Notice that this problem is solved for much shorter codes than the original system.

Step 6. (Induction step) We compute the punctured code $\mathscr{D}^i = \mathcal{P}_{\mathcal{J}^i}(\mathscr{C})$. Then solve for $i = r_+ - 1, \ldots, 0$ the code equivalence problem for the pair $(\mathscr{D}^i, (\mathscr{D}^i)^{\pi^i})$ by using the solution to the code equivalence problem $(\mathscr{D}^{i+1}, (\mathscr{D}^{i+1})^{\pi^{i+1}})$ where π^i is the restriction of π to the set of positions of $\mathscr{D}^i$.

The last code equivalence problem we solve here (namely for $i = 0$) is just a solution to the original code equivalence problem.

4.4 Cryptanalyze of the McEliece variant based on Polar codes

4.4.1 Step 1 – Minimum weight codewords searching.

Finding the minimum weight codewords of $\mathscr{C}(I)^{\pi}$ can be performed by applying any variants of the ISD algorithm.On the other hand, all the minimum codewords of $\mathscr{C}(I)$ are easily obtained by using Theorem 3.7.14: $\mathrm{W}_{\min}$ decomposes into orbits under the action of $\mathrm{LTA}(m, 2)$ where each orbit contains one of the monomials of I of degree r_+.

Example 4.4.1. *The example that we give here will be considered through all the cryptanalysis. We will consider the* $[256, 79]$ *Polar code. It is defined by the set* $I = [1, x_6x_7]_{\preceq} \cup [1, x_1x_4x_7]_{\preceq} \cup [1, x_0x_5x_6]_{\preceq} \cup [1, x_2x_4x_6]_{\preceq} \cup [1, x_3x_4x_5]_{\preceq} \cup [1, x_0x_1x_2x_5]_{\preceq} \cup [1, x_0x_1x_3x_4]_{\preceq}$.

The Polar code is such that $\mathscr{R}(2, 8) \subset \mathscr{C}(I) \subset \mathscr{R}(4, 8)$. *Therefore we have that the minimum weight equals* $2^{8-4} = 16$ *and the set of maximum degree monomials is:* $I_4 = \{x_0x_1x_2x_3, x_0x_1x_2x_4, x_0x_1x_2x_5, x_0x_1x_3x_4\}$.

Using the minimum weight counting method we have that there are 16 words in the orbit of $x_0x_1x_2x_3$, *32 in the orbit of* $x_0x_1x_2x_4$ *and 64 in the orbit of* $x_0x_1x_2x_5$ *and* $x_0x_1x_3x_4$, *which makes a total of 176 minimum weight codewords.*

4.4.2 Step 2 – Signature of orbits in $\mathrm{W}_{\min}$

To distinguish between the codewords of $\mathrm{W}_{\min}$ we have first chosen a monomial in each of the orbits under $\mathrm{LTA}(m, 2)$ that decompose $\mathrm{W}_{\min}$. For each of such monomials g we have computed the dual of the shortened code $\mathscr{D} \stackrel{\text{def}}{=} \left(\mathcal{S}_{\mathcal{J}}(\mathscr{C}(I))\right)^{\perp}$ with respect to the support $\mathcal{J}$ of $\mathsf{ev}(g)$. It has turned out that, for the Polar codes we have considered, the pair (number of codewords of weight 2^{r_-} in $\mathscr{D}$, dimension of $\mathscr{D}$) was discriminant enough to yield a signature of the orbit. This critical quantity 2^{r_-} occurs because we have

Theorem 4.4.2. *Let* $g = x_{i_1} \ldots x_{i_{r_+}}$ *be a monomial of degree* r_+ *in* I. *Denote by* $\mathsf{supp}(g)$ *the support of* $\mathsf{ev}(g)$, *then the minimum distance of* $\left(\mathcal{S}_{\mathsf{supp}(g)}(\mathscr{C}(I))\right)^{\perp}$ *is equal to* 2^{r_-} *if and only if there exists a monomial* h *in* $\mathcal{M}_m \setminus \check{I}$ *such that:*

1. $\deg(\gcd(h, g)) = r_+ - 1$
2. $\deg(\gcd(h, \check{g})) = m - r_- - r_+$

We will first begin this demonstration by proving a general result about the dual of shortened monomial codes.

Lemma 4.4.3. *Let $\mathscr{C}(I)$ be a decreasing monomial code and $g \in I$. Let $\mathsf{supp}(g)$ be the support of $\mathsf{ev}(g)$. We denote by $E\left(\mathcal{S}_{\mathsf{supp}(g)}(\mathscr{C}(I))\right)^\perp$ the dual of the shortened code in $\mathsf{supp}(g)$ that we have extended by zeros in the positions in which we have shortened the code. Then*

$$E\left(\mathcal{S}_{\mathsf{supp}(g)}(\mathscr{C}(I))\right)^\perp = \left\{\mathsf{ev}((1+g)f) : f \in \mathcal{M}_m \setminus \check{I}\right\}$$

Proof. Recall that we have

$$\left(\mathcal{S}_{\mathsf{supp}(g)}(\mathscr{C}(I))\right)^\perp = \mathcal{P}_{\mathsf{supp}(g)}\left(\mathscr{C}(I)^\perp\right)$$

We know that $\mathscr{C}(I)^\perp = \mathscr{C}(\mathcal{M}_m \setminus \check{I})$. The lemma follows from this and the fact the $\mathsf{ev}(1+g)$ takes value 1 on the complementary of $\mathsf{supp}(g)$ and 0 on $\mathsf{supp}(g)$. □

We will also need the following result that is only a slight generalization of [Min07, Prop. 6, p.69] (and our proof will follow closely the proof of this proposition).

Lemma 4.4.4. *Let g be some monomial of degree $s \geqslant 1$. Denote by $\mathsf{supp}(g)$ the support of $\mathsf{ev}(g)$, then the minimum distance of $\left(\mathcal{S}_{\mathsf{supp}(g)}(\mathscr{C}(I))\right)^\perp$ is greater than or equal to 2^{r_-}. If the minimum distance is equal to 2^{r_-} then there exists a monomial h in $\mathcal{M}_m \setminus \check{I}$ such that*

1. $\deg(\gcd(h, g)) = s - 1$
2. $\deg(\gcd(h, \check{g})) = m - r_- - s$

Proof. Let us take a nonzero codeword of $\mathscr{C}(I)^\perp$, say that is the evaluation of some polynomial f, which is in this case of degree at most $m - 1 - r_-$. Write $f = \sum_j m_j$ as a sum of monomials. Then $\tilde{f} \stackrel{\text{def}}{=} \sum_{j : g \nmid m_j} m_j$ is defined as the polynomial where we have removed from the monomial expression of f all monomials that are divisible by g. Since $(\mathcal{S}_{\mathsf{supp}(g)}(\mathscr{C}(I)))^\perp = \mathcal{P}_{\mathsf{supp}(g)}\left(\mathscr{C}(I)^\perp\right)$, we want to prove that the evaluation of f on $\{0,1\}^m \setminus \mathsf{supp}(g)$ is either zero or of weight $\geqslant 2^{r_-}$. Notice that the evaluation on $\{0,1\}^m \setminus \mathsf{supp}(g)$ coincides with the evaluation of $\tilde{f}$.

Let us assume that $g = x_0 \dots x_{s-1}$. With this choice, let us pick a monomial of $\tilde{f}$ that has maximum degree in $x_s, \dots, x_{m-1}$. Let d be this degree (in $x_s, \dots, x_{m-1}$). $\tilde{f}$ can be written as

$$\tilde{f} = mu(x_0, \dots, x_{s-1}) + v(x_0, \dots, x_{m-1}),$$

where m is a monomial of degree d in $x_s, \dots, x_{m-1}$. We take here in the monomials whose sum is equal to $\tilde{f}$ all monomials that are divisible by m and u is just the sum of these monomials divided by m. Let d' be the degree of u which is necessarily smaller than s since $\tilde{f}$ does not contain any monomial divisible by g.

Notice that $u(x_0 \dots x_{s-1})$ is non zero in at least $2^{s-d'}-1$ entries if we do not count the $(1,\dots,1)$ entry, since its evaluation is a codeword of $\mathscr{R}(d',s)$.

Call a "block" the set of points $(x_0,\dots,x_{m-1})$ which take a prescribed value on $x_0,\dots,x_{s-1}$. The support $\mathsf{supp}(g)$ of g corresponds to the block $x_0=1,\dots,x_{s-1}=1$. Notice that the weight of $\mathsf{ev}(\tilde{f})$ restricted to a block (with the exception of the block $x_0=1,\dots,x_{s-1}=1$) is at least 2^{m-s-d}, since this restriction is a codeword of $\mathscr{R}(d,m-s)$. In other words the weight of $\mathsf{ev}(\tilde{f}(1+g))$ is lower-bounded by

$$|\mathsf{ev}(\tilde{f})(1+g)| \geqslant 2^{m-s-d}(2^{s-d'}-1) \geqslant 2^{m-s-d}2^{s-d'}\frac{1}{2} = 2^{m-d-d'-1}.$$

Notice that we have $d+d' \leqslant m-r_- -1$ and therefore we finally obtain

$$|\mathsf{ev}(\tilde{f})| \geqslant 2^{m-(m-r_- -1)-1} = 2^{r_-}.$$

This proves the statement about the minimum distance in this case. A quick inspection of this proof shows that the only fact we used on g was that is is different from 1 (the particular form of g was only here to simplify notation), and therefore it also holds for all monomials g different from 1.

Assume now that the minimum distance of $\left(\mathcal{S}_{\mathsf{supp}(g)}\left(\mathscr{C}(I)\right)\right)^{\perp}$ is equal to 2^{r_-}. By a quick inspection of this proof this means that $\deg u = s-1$ and $\deg m = m-r_- -1-(s-1) = m-r_- -s$. Write u as a set of monomials $u=\sum_j m'_j$ and choose m' as any monomial in this sum that is of degree $s-1$. Obviously $h \stackrel{\text{def}}{=} mm'$ is a monomial of degree $s-1+m-r_- -s = m-r_- -1$ that appears as a monomial in the sum $f=\sum_j m_j$. Therefore h is in $\mathcal{M}_m \setminus \check{I}$. Such an h has the aforementioned form. □

We will now use this to prove Theorem 4.4.2.

Proof. First of all let us notice that the minimum distance of $E\left(\mathcal{S}_{\mathsf{supp}(g)}\left(\mathscr{C}(I)\right)\right)^{\perp}$ is the same as the minimum distance of $\left(\mathcal{S}_{\mathsf{supp}(g)}\left(\mathscr{C}(I)\right)\right)^{\perp}$. From Lemma 4.4.3 we know that any codeword in the first code can be written as $\mathsf{ev}((1+g)f))$ where f is polynomial which is a linear combination of monomials in $\mathcal{M}_m \setminus \check{I}$. Consider now that there is a monomial h satisfying the conditions above. Let us prove that the weight of $\mathsf{ev}((1+g)h)$ is equal to 2^{r_-}. Let i_0 be the only index that is in $\mathrm{ind}(g)$ but not in $\mathrm{ind}(g\wedge h)$. Observe now that

$$\begin{aligned}
(1+g)h &= (1+x_{i_1}\dots x_{i_{r_+}}) \prod_{i\in\mathrm{ind}(\gcd(g,h))} x_i \prod_{i\in\mathrm{ind}(\gcd(\check{g},h))} x_i \\
&= (1+x_{i_0}) \prod_{i\in\mathrm{ind}(\gcd(g,h))} x_i \prod_{i\in\mathrm{ind}(\gcd(\check{g},h))} x_i \\
&= (1+x_{i_0})h.
\end{aligned}$$

Thus

$$|\mathsf{ev}((1+g)h))| = |(\mathsf{ev}((1+x_{j_0})h)| = 2^{m-(m-r_- -1+1)} = 2^{r_-}.$$

By using the lower-bound on the minimum distance coming from Lemma 4.4.4 we obtain that the minimum distance of $\left(\mathcal{S}_{\mathsf{supp}(g)}\left(\mathscr{C}(I)\right)\right)^{\perp}$ is equal to 2^{r_-}.

Assume now that the minimum distance of $\left(\mathcal{S}_{\mathsf{supp}(g)}\left(\mathscr{C}(I)\right)\right)^{\perp}$ is equal to 2^{r_-}, then we can use Lemma 4.4.4 and obtain the aforementioned claim. □

Example 4.4.5. *If we consider the dual of the Polar code we have that* $\mathscr{R}(3,8) \subset \mathscr{C}(I)^{\perp} \subset \mathscr{R}(5,8)$. *The maximum degree monomials in the dual code are*

$$\{g \in \mathcal{M}_m \ , \ g \preceq x_0x_1x_2x_5x_7 \ , or \ g \preceq x_0x_1x_3x_5x_6 \ , or \ g \preceq x_0x_2x_3x_4x_7 \ , or \ g \preceq x_1x_2x_3x_4x_6\}.$$

Recall that the minimum weight codewords of the Polar code are given by the monomials in $I_4 = \{x_0x_1x_2x_3, x_0x_1x_2x_4, x_0x_1x_2x_5, x_0x_1x_3x_4\}$. *So we obtain that the minimum distance of the dual of the shortened code is*

- 2^{4+2} *for* $x_0x_1x_2x_3$ *since* $(1 + x_0x_1x_2x_3)x_0x_1x_2x_5x_7 = x_0x_1x_2(1 + x_3)x_5x_7$.
- 2^{4+2} *for* $x_0x_1x_2x_4$ *since* $(1 + x_0x_1x_2x_4)x_0x_1x_2x_5x_7 = x_0x_1x_2(1 + x_4)x_5x_7$.
- 2^{4+2} *for* $x_0x_1x_2x_5$ *since* $(1 + x_0x_1x_2x_5)x_0x_1x_2x_4x_7 = x_0x_1x_2(1 + x_5)x_4x_7$.
- 2^{4+2} *for* $x_0x_1x_3x_4$ *since* $(1 + x_0x_1x_3x_4)x_0x_1x_2x_4x_7 = x_0x_1x_4(1 + x_3)x_2x_7$.

If we apply the minimum weight counting method from Section 3.7 to the four monomials we obtain that in the dual of the shortened code there are 372 *minimum weight codewords when we consider the monomial* $x_0x_1x_2x_3$, 340 *codewords for* $x_0x_1x_2x_4$, 308 *codewords for* $x_0x_1x_2x_5$ *and* 148 *codewords for* $x_0x_1x_3x_4$.

Therefore, here the number of minimum weight codewords in the dual of the shortened code can be considered as signature for the monomials.

Step 3 – Computation of orbits in $\mathrm{W_{min}}^{\pi}$

The signature Σ that has been found in the previous step is now applied to $\mathrm{W_{min}}^{\pi}$. It gives the orbits of $\mathrm{W_{min}}^{\pi}$ with respect to the conjugate group $\pi^{-1}\mathcal{G}\pi$. Indeed, it can be verified that

Proposition 4.4.6. $\mathrm{W_{min}}^{\pi}$ *is invariant by the action of* $\pi^{-1}\mathrm{LTA}(m,2)\pi$ *and if* Σ *is a signature for* $\mathrm{W_{min}}$ *under the action of* $\mathrm{LTA}(m,2)$, *then it is also a signature for the action of* $\pi^{-1}\mathrm{LTA}(m,2)\pi$ *on* $\mathrm{W_{min}}^{\pi}$.

We use this signature for finding the orbit of $\boldsymbol{c}_{\min}$.

Proposition 4.4.7. *The orbit of* $\boldsymbol{c}_{min}$ *under* $\mathrm{LTA}(m,2)$ *consists of* 2^{r_+} *codewords that are of the form* $\boldsymbol{c}_{min}(\varepsilon_0, \ldots, \varepsilon_{r_+-1})$ *where the* ε_i*'s are arbitrary elements of* $\mathbb{F}_2$. *The orbit of* $\boldsymbol{c}^{\pi}_{min}$ *under* $\pi^{-1}\mathrm{LTA}(m,2)\pi$ *is given by* 2^{r_+} *codewords of weight* 2^{m-r_+} *that have disjoint supports which are the permuted versions* $A(\varepsilon_0, \ldots, \varepsilon_{r_+-1})^{\pi}$ *of the affine spaces* $A(\varepsilon_0, \ldots, \varepsilon_{r_+-1})$.

In other words, finding this orbit in $\mathrm{W_{min}}^{\pi}$ and looking at the support of the codewords that we have found in this way allows us to find the support of the permuted versions $A(\varepsilon_0, \ldots, \varepsilon_{r_+-1})^{\pi}$ of the affine spaces $A(\varepsilon_0, \ldots, \varepsilon_{r_+-1})$.

4.4.3 Step 4 – Identification of affine spaces

There are several ways to identify the permuted versions of the affine spaces we are interested in. One of the simplest way is by computing the dimensions of certain spaces. First we take any codeword in the orbit of $\boldsymbol{c}_{\min}$. Such codeword is of the form $\boldsymbol{c}_{\min}^{\gamma\pi}$ where γ is a permutation leaving $\mathscr{C}(I)$ invariant. In other words, up to applying the permutation group, we can safely declare that this codeword is $\boldsymbol{c}_{\min}^{\pi}$. Let $\mathcal{I}_0$ be the support of $\boldsymbol{c}_{\min} = \boldsymbol{c}(1,\dots,1)$. We choose $\mathcal{I}_0'$ be the support of the codeword $\boldsymbol{c}(\underbrace{1,\dots,1}_{(r_+-1)\text{ times}},0)$. Notice that $\mathcal{I} \stackrel{\text{def}}{=} \mathcal{I}_0 \cup \mathcal{I}_0'$ is the support of the codeword $\mathsf{ev}(x_0 \dots x_{r_+-2})$. We compute the dimension of the code $\mathcal{P}_{\mathcal{I}}(\mathscr{C}(I))$. Now, we let $\mathcal{J}_0,\dots,\mathcal{J}_{2^{r_+}-1}$ be the supports of the codewords that are in the orbit of $\boldsymbol{c}_{\min}^{\pi}$, with $\mathcal{J}_0$ being the support of the codeword $\boldsymbol{c}_{\min}^{\gamma\pi}$ that has been chosen. We compute the dimensions of the codes $\mathcal{P}_{\mathcal{J}_0\cup\mathcal{J}_i}(\mathscr{C}(I)^{\pi})$ for $i = 1,\dots,2^{r_+}-1$. It turns out that there is generally a single space $\mathcal{J}_i$ such that $\dim\left(\mathcal{P}_{\mathcal{J}_0\cup\mathcal{J}_i}(\mathscr{C}(I)^{\pi})\right) = \dim\left(\mathcal{P}_{\mathcal{I}}(\mathscr{C}(I))\right)$. We pair these two spaces $\mathcal{J}_0$ and $\mathcal{J}_i$ together. This process can be used to pair together all the spaces $A(\varepsilon_0,\dots,\varepsilon_{r_+-2},0)^{\gamma\pi}$ and $A(\varepsilon_0,\dots,\varepsilon_{r_+-2},1)^{\gamma\pi}$ by pairing together $\mathcal{J}_i$ and $\mathcal{J}_j$ when $\mathcal{J}_j$ is the only space for a given i such that

$$\dim\left(\mathcal{P}_{\mathcal{J}_i\cup\mathcal{J}_j}(\mathscr{C}(I)^{\pi})\right) = \dim\left(\mathcal{P}_{\mathcal{I}}(\mathscr{C}(I))\right).$$

In such a case, $\mathcal{J}_i$ and $\mathcal{J}_j$ necessarily correspond to

$$A(\varepsilon_0,\dots,\varepsilon_{r_+-2},0)^{\gamma\pi} \text{ and } A(\varepsilon_0,\dots,\varepsilon_{r_+-2},1)^{\gamma\pi}$$

for a certain $(\varepsilon_0,\dots,\varepsilon_{r_+-2}) \in \mathbb{F}_2^{r_+-1}$. In other words, we know after this process all the spaces $A(\varepsilon_0,\dots,\varepsilon_{r_+-2})^{\gamma\pi} = A(\varepsilon_0,\dots,\varepsilon_{r_+-2},0)^{\gamma\pi} \cup A(\varepsilon_0,\dots,\varepsilon_{r_+-2},1)^{\gamma\pi}$. We can carry on this process with the codeword $\boldsymbol{c} = \mathsf{ev}(x_0 \dots x_{r_+-1})$ instead of $\boldsymbol{c}_{\min}$ and recover all the permuted affines spaces $A(1)^{\gamma\pi}, A(1,1)^{\gamma\pi},\dots, A(\underbrace{1,1,\dots,1}_{r_+\text{ times}})^{\gamma\pi}$ for some permutation γ leaving $\mathscr{C}(I)$ invariant.

Example 4.4.8. *If we consider the Polar code and we puncture the code on the support of the vector* $\mathsf{ev}(x_0x_1x_2) = \mathsf{ev}(x_0x_1x_2x_3 + x_0x_1x_2(1+x_3))$ *we obtain a code of dimension 75. If we puncture now on the other* 2^4-1 *possible supports we obtain each time a dimension strictly greater than 75. Which means that there is only one minimum weight codeword, namely* $\mathsf{ev}(x_0x_1x_2(x_3+1))$*, that can be paired with* $\mathsf{ev}(x_0x_1x_2x_3)$.

We continue and consider the vector $\mathsf{ev}(x_0x_1)$ *and remark that this is the only codeword for which the dimension of the punctured code equals* 68*. It means that there is only one vector, namely* $\mathsf{ev}(x_0x_1(x_2+1))$*, than can be paired with* $\mathsf{ev}(x_0x_1x_2)$.

In the last step we find the support of the permuted of the codeword $\mathsf{ev}(x_0)$.

4.4.4 Step 5 – Equivalence problem for a short decreasing monomial code

We now have to solve the code equivalence problem for $\mathscr{D}$ which is a code of length 2^{m-r_+} which is much shorter than the original code. It is also straightforward to check that it is a decreasing monomial code.

4.4.5 Step 6 – Induction step

The idea here is to reconstruct the permutation $\hat{\pi}$ given that we already know its action on the support of $\boldsymbol{c}_{\min}$. More precisely, the code equivalence problem that we solve here is:

Problem 4.4.9 (Code equivalence search problem with side information)**.** *Given* $(\mathscr{C}, \mathscr{C}^{\pi})$ *and* t *pairs of code positions* (i_0, j_0), $(i_1, j_1), \ldots, (i_{t-1}, j_{t-1})$, *find* $\hat{\pi}$ *such that* $\mathscr{C}^{\hat{\pi}} = \mathscr{C}^{\pi}$ *and* $\hat{\pi}(i_s) = j_s$ *for all* $s \in \{0, 1, \ldots, t-1\}$

We use the following algorithm for solving this problem (we let here $\mathcal{I} \stackrel{\text{def}}{=} \{i_0, \ldots, i_{t-1}\}$ and $\mathcal{J} \stackrel{\text{def}}{=} \{j_0, \ldots, j_{t-1}\}$)

1. we pick a certain number ℓ of codewords $\boldsymbol{c}(0), \ldots, \boldsymbol{c}(\ell-1)$ of $\mathscr{C}$.
2. Let $\mathscr{C}(j)$ the set of codewords of $\mathscr{C}$ which coincide with $\boldsymbol{c}(j)$ on the positions belonging to $\mathcal{J}$. We also define $\mathscr{C}(i)^{\pi}$ as the set of codewords of $\mathscr{C}^{\pi}$ that coincide with $\boldsymbol{c}(i)^{\pi}$ on $\mathcal{I}$.
3. We compute for all i in $0, 1, \ldots, \ell-1$ and all positions j which are not in $\mathcal{J}$, the number $\Sigma(i, j)$ which is the number of codewords of minimum weight in $\mathcal{P}_j(\mathscr{C}(i))$, and similarly for all all positions j that are not in $\mathcal{I}$, the number $\Sigma^{\pi}(i, j)$ which is the number of codewords of minimum weight in $\mathcal{P}_j(\mathscr{C}(i)^{\pi})$.
4. We declare for u which is not in $\mathcal{I}$ that $\hat{\pi}(u) = v$ if there exists a unique v which does not belong to $\mathcal{J}$ such that $\Sigma(i, v) = \Sigma^{\pi}(i, u)$ for all i in $\{0, 1, \ldots, \ell-1\}$.

It is straightforward to verify that this algorithm outputs the unique $\hat{\pi}$ solving the problem in this case. We have also encountered cases, where even with the knowledge we have on $\hat{\pi}$, we have different solutions. In such a case, we were able to compute how many solutions we had and add to the set of pairs (i_s, j_s) an additional pair (or additional pairs) which gives a unique solution.

4.5 Implementation

We implemented the attack on the $[2048, 614]$-Polar code.Such a code is able to correct more than 200 errors with a small error probability- this should be compared to the 130 errors that a Goppa code of the same rate is able to tolerate. In the case of a Goppa code we have about 70 bits of security against message attacks based on generic linear codes decoding algorithms, whereas we have more than 105 bits of security for the Polar code.

We first checked that this code $\mathscr{C}$ and its dual $\mathscr{C}^{\perp}$ are both Decreasing Monomial codes and computed all the minimum weight codewords by using Proposition 3.7.12. We have also checked that the code was weakly self-dual $\mathscr{C} \subset \mathscr{C}^{\perp}$. The minimum distance of $\mathscr{C}$ turned out to be equal to 32 and there were 42176 codewords of this weight, whereas the minimum distance of $\mathscr{C}^{\perp}$ was 8 and there were 6912 codewords of this weight in the dual, numbers that were computed using Theorem 3.7.14. The same number of codewords were found by Dumer's algorithm in $\mathscr{C}^{\pi}$ and in $(\mathscr{C}^{\pi})^{\perp}$. It tooks 27 seconds to find these codewords in $\mathscr{C}^{\pi}$ and 3 seconds to find these codewords in $(\mathscr{C}^{\pi})^{\perp}$ on a 8-core XEON E3-1240 running at 3.40 GHz.

But the most time consuming part was Step 6 of the attack when we have to compute the various $\Sigma(i,j)$'s that are needed. This is done again by using Dumer's algorithm. The difference with obtaining codewords of minimum weight of the Polar code is that in the polar case we know beforehand the number of minimum weight codewords by using a counting procedure based on Theorem 3.7.14 and we can stop the search procedure once we have the right amount of different codewords. However when we compute $\Sigma(i,j)$ we do not know beforehand the number of minimum weight codewords in $\mathcal{P}_j(\mathscr{C}(i))$ and we use a probabilistic procedure based on the coupon collector problem : once we have found n different minimal codewords, where on average we have found each codeword $\alpha \ln n$ times we stop the procedure for a certain value of α greater than 1. Here we have taken α to be equal to 3. In this case, to speed up the computation we chose the $c(i)$'s to be minimum weight codewords of $\mathscr{C}$. More than 80% of the total computation is actually taken for the last step of induction where we recover a permutation for the whole $[2048, 614]$ code from the partial permutation acting on half its positions. This takes about 227 hours and the total computation time is about 280 hours. This part of the attack is very likely to be improved significantly if need be.

4.6 Perspectives

Even though we managed to totally break the McEliece scheme based on Polar codes there is another variant using Polar codes which is not vulnerable to our attack. In [HSEA14] the authors propose to choose a subcode of the Polar code instead of the whole code, which is equivalent to choosing a Monomial code from a subset of monomials in $\mathcal{M}_m$. Therefore our attack can not be directly implemented on this code and different techniques have to be investigated. We notice that this variant is similar to [Rin15], in which the authors propose to choose a subcode of a Reed-Muller code. Remark that choosing subcodes of Decreasing Monomial codes rises a bigger difficulty in finding the structure of the code since in general the subcodes lose all the decreasing properties and more important properties related to the permutation group. Another fundamental difference is that there is no more only one private code but an exponential number of codes since there are $\binom{k_D}{k_S}$ where k_D is the dimension of the decreasing code and k_S is the dimension of the subcode. Therefore the code equivalence problem becomes even more complicated to solve in this case. So if one manges to solve the code equivalence problem for Monomial codes or even Weakly Decreasing Monomial codes it might make a step towards the complete elimination of this family of codes from public key cryptography.

The second aspect we want to reveal is the efficiency of the attack. Since the main step in our algorithm is to search for the minimum weight codewords one might suggest to propose parameters for which the minimum distance is big enough in order to make the attack unfeasible in practice. But this is not always a good solution since the minimum distance of the dual code might decrease and thus we can apply our attack on the dual code, since it is a Decreasing Monomial code. Nonetheless, there is another type of attack, much more efficient, that worked on the family of Reed-Muller codes [CB13]. The idea is to use the square code and the dual code in order to decrease the dimension of the initial Reed-Muller code down to the first order Reed-Muller code. The only issue is that for this family of codes the square code has the same permutation group as the initial code, which

is not the case for a Decreasing Monomial code in general. In the case of Polar codes, we tried this attack but we did not managed to find the whole structure of the code. By that we mean that only a part of the generating monomials of the Polar code were found.

5 Weak keys in the QC-MDPC McEliece

5.1 Introduction

Moderate Density Parity-Check codes, or shortly MDPC codes, were introduces in 2009 [OB09] and became famous for their application in cryptography, in a McEliece type scheme [MTSB13].

The security assessment for the QC-MDPC is based on a reduction towards the syndrome decoding problem for quasi-cyclic codes. The main ingredient for the reduction is the assumption that a QC-MDPC code can not be distinguish from a random QC linear code. More details on this issue can be found in the original paper [MTSB13] and in [Sen10].

The scientific community presented a real interest in this variant and thus several efficient implementations were proposed [HvMG13, vMG14a, MOG15, Cho16] as well as a structural attack against the cyclosymmetric MDPC codes [Per14], side-channel attacks [vMG14b, GJS16] and improved decoders for the scheme [CS16]. But this variant using QC-MDPC codes, made a big step towards a global recognition at The Seventh International Conference on Post-Quantum Cryptography PQCrypto2016, where the scientific community seemed to agree on the fact that this variant could be among the finalists in the POST-QUANTUM CRYPTO Project, initiated by the NIST.

Our contribution. We propose here to analyze the security of the MDPC-McEliece cryptosystem from another point of view, more exactly to identify a subset of private keys, that we will call weak keys, which can be efficiently retrieved from the corresponding public keys. For that we describe the Key Generation step using an algebraic formalism for quasi-cyclic codes and show that the Key Recovery Attack is just a particular instance of the *Rational Reconstruction Problem.* Hence we propose to use a modified version of the Extended Euclidean Algorithm, or shortly EEA as a Key Recovery Attack against the QC-MDPC scheme. The main advantage of this approach is the low complexity of the algorithm that we use, since the EEA runs in $O(n^2)$ bit operations, where n is the length of the code.

The most challenging part of our study was to estimate the proportion of weak and search for different techniques that might increase the proportion of weak keys. So the first step was to determine the exact proportion of private keys that can be recovered

using the EEA. In our study we have considered the case of a 2-quasi-cyclic MDPC scheme with parameters $(2p, p, \omega)$, where ω is the Hamming weight of the two blocks and p is the length of each block. We have proved that the proportion of weak keys is asymptotically dominated by $2^{-c\omega}$ where $0 < c \leqslant 1$ is a constant. Thus the chances of finding a weak key are very close to the security level for the scheme, which is given by the cost of the best variant of Information Set Decoding.

In our study we also investigate different methods to improve our initial results. A successful approach is to consider the problem of equivalence of quasi-cyclic codes and use it in order to increase the proportion of weak keys. For that we use two group actions that preserve the 2-quasi-cyclic structure of the $(2p, p, \omega)$ QC-MDPC code, namely the additive group $\mathbb{F}_p$ and the multiplicative group $\mathbb{F}_p{}^*$. We prove that using the additive group we increase the proportion of weak keys by a factor equal to ω^2. As for the multiplicative group $\mathbb{F}_p{}^*$, we extend the proportion of vulnerable keys with a linear factor in p.

We confront our result with the proposed parameters for the QC-MDPC scheme and remark that we obtain probability values that are bigger than the announced security level. Hence we estimate that some of the parameters must be reviewed in order to at least decrease this probability below the security level. In order to do so one must increase the weight of the MDPC, which comes with a degradation of the decoding error probability. Therefore we propose another step in the Key Generation which completely eliminates weak keys from the possible set of keys. The results that we detail in this Chapter were published in [BDLO16].

5.2 Preliminaries on QC-MDPC Codes

5.2.1 Cyclic and Quasi-Cyclic codes

Cyclic codes were studied for the first time by Prange [Pra57] and benefit of simple encoding and decoding techniques. Part of this family of codes are Bose-Chaudhuri-Hocquenghem [Hoc59, BRC60] and Reed-Solomon codes [ISR60]. Since here we are rather interested in the quasi-cyclic structure we prefer to recall only a small part of the properties of cyclic codes. For a detailed presentation of cyclic codes we address the reader to the well-known [MS86]. As for the structural results on quasi-cyclic codes many of the result we recall here come from the work of San Lin and Patrick Solé, more precisely from the third part [LS05] of a four series of articles entitled *On the algebraic structure of quasi-cyclic codes*, as well as the phd thesis of Christophe Chabot [Cha09].

Preliminaries

In order to define cyclic codes we need to introduce the notion of cyclic shift of a n length vector. We will use the usual convention, that is we consider the shifts to the right, and as in the previous chapter, the most significant bit of a vector $(c_0, \dots, c_{n-1})$ is the rightmost bit, here c_{n-1}.

Definition 5.2.1 (Cyclic Shift)**.** *We define the cyclic shift of a binary n-length vector $\boldsymbol{c} = (c_0, \dots, c_{n-1})$ by $T(\boldsymbol{c})$, where*

$$\begin{array}{ccc} T : \mathbb{F}_2^n & \to & \mathbb{F}_2^n \\ (c_0, c_1, \dots, c_{n-1}) & \mapsto & (c_{n-1}, c_0, \dots, c_{n-2}) \end{array}$$

For any $0 \leqslant i \leqslant n-1$ the i^{th} cyclic shift of a vector $\boldsymbol{c}$ is $T^i(\boldsymbol{c}) = (c_{n-i}, \dots, c_0, \dots, c_{n-i-1})$.

Definition 5.2.2 (Cyclic and Quasi-Cyclic codes)**.** *Let n be a strictly positive integer and $\mathscr{C}$ be a $[n, k]$ binary linear code.*

- *We say that $\mathscr{C}$ is a cyclic code if for any codeword $\boldsymbol{c} \in \mathscr{C}$ we have $T(\boldsymbol{c}) \in \mathscr{C}$.*
- *We say that $\mathscr{C}$ is a n_0-quasi-cyclic code if $\exists\ 0 < n_0 < n$ with $n_0 | n$, such that for any codeword $\boldsymbol{c} \in \mathscr{C}$ we have $T^{n_0}(\boldsymbol{c}) \in \mathscr{C}$.*

Let's recall the main properties regarding cyclic and quasi-cyclic codes:

Proposition 5.2.3.

- *([MS86]) Let $\mathscr{C}$ be a $[n, k]$ binary cyclic code, then for any codeword $\boldsymbol{c} \in \mathscr{C}$ all the cyclic shifts $T^i(\boldsymbol{c})$, for $1 \leqslant i \leqslant n-1$ are codewords of $\mathscr{C}$.*
- *([MS86]) Any $[n, k]$ binary cyclic code admits as parity-check matrix a $n \times n$ circulant matrix, that we denote $\boldsymbol{H}$, such that $\mathrm{rank}(\boldsymbol{H}) = n - k$, where a circulant matrix is a $n \times n$ matrix obtained by cyclically right shifting its first row $\boldsymbol{h} = (h_0, h_1, \dots, h_{n-1})$*

$$\boldsymbol{H} = \begin{pmatrix} h_0 & h_1 & \cdots & h_{n-1} \\ h_{n-1} & h_0 & \cdots & h_{n-2} \\ \vdots & \vdots & \ddots & \vdots \\ h_1 & h_2 & \cdots & h_0 \end{pmatrix}. \tag{5.1}$$

- *([Bal14]) Any binary n_0-quasi-cyclic code of length $n_0 p$ and rate k_0/n_0 admits a parity-check matrix $\boldsymbol{H}$ such that*

$$\boldsymbol{H} = \begin{pmatrix} \boldsymbol{H}_{1,1} & \boldsymbol{H}_{1,2} & \cdots & \boldsymbol{H}_{1,n_0} \\ \vdots & \vdots & \vdots & \vdots \\ \boldsymbol{H}_{n_0-k_0,1} & \boldsymbol{H}_{n_0-k_0,2} & \cdots & \boldsymbol{H}_{n_0-k_0,n_0} \end{pmatrix},$$

where for each $1 \leqslant i \leqslant n_0 - k_0$ and $1 \leqslant j \leqslant n_0$ the matrix $\boldsymbol{H}_{i,j}$ is $p \times p$ circulant.

An interesting case that is used for the QC-LDPC and QC-MDPC codes is when $k_0 = n_0 - 1$. In this case we have a n_0-quasi-cyclic code of length $n_0 p$ and rate $(n_0 - 1)/n_0$ that admits a parity check matrix

$$\boldsymbol{H} = \begin{pmatrix} \boldsymbol{H}_1 & \boldsymbol{H}_2 & \cdots & \boldsymbol{H}_{n_0} \end{pmatrix}.$$

Algebraic formalism

It is already known that cyclic codes can be defined using an algebraic formalism [MS86]. For that we associate to any binary n length vector $\boldsymbol{c} = (c_0, \ldots, c_{n-1})$ the polynomial over $\mathbb{F}_2[x]$, that is $c(x) = \sum_{i=0}^{n-1} c_i x^i$. Now we can define a cyclic code in an equivalent manner

Proposition 5.2.4. *[MS86] A binary cyclic code of length n is an ideal of the polynomial algebra $\mathbb{F}_2[x]/(x^n - 1)$.*

Since both cyclic and quasi-cyclic codes are strongly related to the algebra of circulant matrices we recall some well-known facts concerning this algebra.

Theorem 5.2.5. *[Dav79] The algebra of $n \times n$ binary circulant matrices denoted by $\left(\mathfrak{C}_n(\mathbb{F}_2), +, \times\right)$ is isomorphic to the polynomial algebra $\left(\mathbb{F}_2[x]/(x^n - 1), +, \cdot\right)$ through the mapping*

$$\begin{array}{rcl} \Phi : \mathfrak{C}_n(\mathbb{F}_2) & \longrightarrow & \mathbb{F}_2[x]/(x^n - 1) \\ \\ \boldsymbol{M} & \longmapsto & m(x) = \displaystyle\sum_{i=0}^{n-1} m_i x^i \pmod{x^n - 1}. \end{array}$$

where $\boldsymbol{m} = (m_0, \ldots, m_{n-1})$ is the first row defining $\boldsymbol{M}$.

Proposition 5.2.6. *Let $\boldsymbol{M} \in \mathfrak{C}_n(\mathbb{F}_2)$ be a circulant matrix and $m(x) \in \mathbb{F}_2[x]/(x^n - 1)$ the corresponding polynomial. Then $\boldsymbol{M}$ is invertible if and only if $m(x)$ and $x^n - 1$ are coprime.*

Furthermore the inverse of $\boldsymbol{M}$ is given by the polynomial $m^{-1}(x) \in \mathbb{F}_2[x]/(x^n - 1)$. An efficient algorithm to compute $m^{-1}(x)$ is the Extended Euclidean Algorithm.

Corollary 5.2.7 ([Bal14])**.** *Let $\boldsymbol{M} \in \mathfrak{C}_n(\mathbb{F}_2)$ be a circulant matrix and $\boldsymbol{m}$ the first row vector defining $\boldsymbol{M}$. If $\boldsymbol{m}$ has an even Hamming weight then $\boldsymbol{M}$ is rank deficient, in other words $\boldsymbol{M}$ is not invertible.*

Remark 5.2.8. *From now on we will consider the case $n = p$ a prime number, choice that is also considered in [MTSB13]. Our choice has several motivations that we will point out at adequate moment, see for example Section 5.3.2.*

Proposition 5.2.9. *[LS05] The mapping Φ defined in Theorem 5.2.5 can be extended to quasi circulant matrices in a natural manner*

$$\begin{aligned} (\mathfrak{C}_p(\mathbb{F}_2))^{n_0} &\longrightarrow (\mathbb{F}_2[x]/(x^p-1))^{n_0} \\ (\boldsymbol{M}_0, \ldots, \boldsymbol{M}_{n_0-1}) &\longmapsto (m_0(x), \ldots, m_{n_0-1}(x)) \,. \end{aligned}$$

Then application Φ induces a one-to-one correspondence between binary quasi-cyclic codes of length $n_0 p$ and linear codes over $\mathbb{F}_2[x]/(x^p-1)$ of length n_0.

Since we deal with polynomials in the algebra $\left(\mathbb{F}_2[x]/(x^p-1), +, \cdot\right)$ we recall in Appendix C the main facts about the factorization of $x^p - 1$ over $\mathbb{F}_2[x]$.

5.2.2 QC-MDPC codes

Gallager discovered the low density parity check (LDPC) codes during his phd thesis [Gal63]. He was motivated by the problem of finding "random-like" codes that could be decoded near the capacity with quasi-optimal performance and feasible complexity. Since LDPC were too complex for the technology at that time, they were forgotten for more than 30 years, and rediscovered by MacKay [Mac99] and Sipser and Spielman [SS96]. These codes were extended in a natural way to moderate density parity check (MDPC) codes in [OB09].

Definition 5.2.10. *A (n, r, ω)-code is a linear code defined by a $r \times n$ parity-check matrix ($r < n$) where each row has weight ω.*

- *a LDPC code is a (n, r, ω) with $\omega = O(1)$ when $n \to \infty$*
- *a MDPC code is a (n, r, ω) with $\omega = O(\sqrt{n \log n})$ when $n \to \infty$*

Definition 5.2.11. *A $(n_0 p, p, \omega)$* Quasi-Cyclic MDPC (QC-MDPC) code *is a MDPC code defined by a block parity-check matrix*

$$\boldsymbol{H} = \begin{pmatrix} \boldsymbol{H}_1 & \boldsymbol{H}_2 & \cdots & \boldsymbol{H}_{n_0} \end{pmatrix}, \tag{5.2}$$

where each block $\boldsymbol{H}_i$ is a $p \times p$ circulant matrix.

Remark 5.2.12. *Using the isomorphism defined in Proposition 5.2.5 we can define a $(n_0 p, p, w)$ QC-MDPC code in an equivalent manner. Indeed, we can consider $h_i(x) \in \mathbb{F}_2[x]/(x^p-1)$ such that $\sum_{i=1}^{n_0} \|h_i\| = \omega$, where $\|h\|$ denotes the Hamming weight of $h(x)$, that is the number of non zero coefficients of $h(x)$.*

Bit flipping decoding algorithm. The decoding techniques for MDPC codes are slightly modified version of the original bit flipping algorithm, due to Gallager [Gal63]. This technique is known to provide an error-correction capability that increase linearly with the length of the code, but decreases with the weight of the parity-checks. Therefore MDPC codes come with a degradation of the performances for decoding, compared to a LDPC code.

The bit flipping algorithm takes for input a vector $\boldsymbol{y} \in \mathbb{F}_2^n$ and the parity-check matrix $\boldsymbol{H}$ of the MDPC code, as well as a threshold b. At each iteration it computes the syndrome of $\boldsymbol{y}$ with respect to $\boldsymbol{H}$ and for each bit i of the vector $\boldsymbol{y}$ it computes the number of parity equations unsatisfied by i. If the number of unsatisfied equations is greater than or equal to the threshold b then the corresponding bit is flipped and the procedure restarts with the new vector and syndrome. The algorithm stops when the syndrome is equal to zero or the number of iterations has reached a fixed upper bound. A full description of the algorithm is given in [Gal63] and in [MTSB13, CS16] for the latest variants.

Remark 5.2.13. *Since the performance and the correctness of the bit flipping algorithm depends on the density of the parity-check matrix $\boldsymbol{H}$, any equivalent parity-check matrix that respects the density condition enables a correct decoding algorithm. In other words given a LDPC/MDPC code $\mathscr{C}$ and a vector $\boldsymbol{y}$, then any equivalent LDPC/MDPC code of $\mathscr{C}$ can decode the vector $\boldsymbol{y}$ with the same decoding algorithm, namely the bit flipping algorithm.*

5.3 QC-MDPC McEliece

5.3.1 Description

We begin by describing the Key Generation of the QC-MDPC McEliece as defined in [MTSB13]. The private key of an $(n_0 p, p, \omega)$ QC-MDPC code is a parity check matrix

$$\boldsymbol{H} = \begin{pmatrix} \boldsymbol{H}_1 & \boldsymbol{H}_2 & \cdots & \boldsymbol{H}_{n_0} \end{pmatrix}, \tag{5.3}$$

where $\boldsymbol{H}_i$ are $p \times p$ circulant matrices for $1 \leqslant i \leqslant n_0$. This private key is obtained by taking at random the first row of $\boldsymbol{H}$ until $\boldsymbol{H}_{n_0}$ is invertible. The public key is the block parity-check matrix $\boldsymbol{F} \stackrel{\text{def}}{=} \boldsymbol{H}_{n_0}^{-1}\boldsymbol{H}$, or equivalently

$$\boldsymbol{F} = \begin{pmatrix} \boldsymbol{H}_{n_0}^{-1}\boldsymbol{H}_1 & \cdots & \boldsymbol{H}_{n_0}^{-1}\boldsymbol{H}_{n_0-1} & \boldsymbol{I}_p \end{pmatrix} \stackrel{\text{def}}{=} \begin{pmatrix} \boldsymbol{F}_1 & \cdots & \boldsymbol{F}_{n_0-1} & \boldsymbol{I}_p \end{pmatrix}. \tag{5.4}$$

Hence, the main steps in the QC-MDPC scheme can be summarized as follows

1. *Key-Generation*
   - **Private key.** Pick at random a valid parity check matrix matrix $\boldsymbol{H}$ of a $(n_0 p, p, \omega)$ QC-MDPC code.
   - **Public key.** The parity check matrix $\boldsymbol{F}$ (or equivalently the generator matrix $\boldsymbol{G}$, where $\boldsymbol{G}\boldsymbol{F}^T = \boldsymbol{0}$).
2. *Encryption* of a plaintext $\boldsymbol{m} \in \mathbb{F}_2^{(n_0-1)p}$ into $\boldsymbol{z} \in \mathbb{F}_2^{n_0 p}$:
   - choose a random error vector $\boldsymbol{e} \in \mathbb{F}_2^{n_0 p}$ with weight $\|\boldsymbol{e}\| \leqslant t$, where t is the decoding capacity of the code.
   - compute the ciphertext $\boldsymbol{z} = \boldsymbol{m}\boldsymbol{G} + \boldsymbol{e}$
3. *Decryption* of the ciphertext $\boldsymbol{z}$ using the private key and the bit-flipping decoder:

- compute $\boldsymbol{mG}$ using the bit-flipping algorithm applied to $\boldsymbol{z}$ with the knowledge of $\boldsymbol{H}$.
- extract the first $(n_0 - 1)p$ positions of $\boldsymbol{mG}$ in order to obtain $\boldsymbol{m}$.

Remark 5.3.1. *Using the isomorphism defined in Proposition 5.2.5, the private and public keys are fully described by the sequences $h_1, \ldots, h_{n_0}$ and $f_1, \ldots, f_{n_0-1}$ of polynomials in $\mathbb{F}_2[x]/(x^p - 1)$ such that for all $i \in \{1, \ldots, n_0 - 1\}$,*

$$f_i = \frac{h_i}{h_{n_0}} \pmod{x^p - 1}, \tag{5.5}$$

where the private polynomials are taken so that $\sum_{i=1}^{n_0} \|h_i\| = \omega$.

5.3.2 The choice of parameters for the scheme

There are several factors to be considered in the choice of the parameters for the QC-MDPC McEliece scheme.

- The parameter p has to be prime in order to avoid attacks using the decomposition of $x^p - 1$ into irreducible [Loi01, FL08] or squaring type attack [LJS+16].
- Against Folding type attacks [Gen01, FOP+14] we need to choose values for p such that the number of irreducible factors of $x^p - 1$ in $\mathbb{F}_2[x]$ different from $x - 1$, denoted d, is as small as possible, for example $d = O(1)$. In Figure 5.1 we detail the values for d for all the proposed parameters in [MTSB13].
- During the Key Generation step one must randomly choose the polynomials h_i until at least one of them is invertible. So we might expect, for security reasons, that the designers selected those parameters for which the set of invertible polynomials in the polynomial algebra $\mathbb{F}_2[x]/(x^p - 1)$ is the largest possible. Using a ring isomorphism we give the number of invertible polynomials and thus show which are the proper parameters to be selected.

Proposition 5.3.2. *Let p be a prime number and assume $(x - 1) \prod_{i=1}^{d} g_i(x)$ is the decomposition of $x^p - 1$ into irreducible polynomials over $\mathbb{F}_2[x]$ for some $d \geqslant 1$. Then*

- *an invertible polynomial has necessarily an odd Hamming weight.*
- *the number of invertible polynomials in $\mathbb{F}_2[x]/(x^p - 1)$ equals $\left(2^{\frac{p-1}{d}} - 1\right)^d$.*

.

Hence, if we take into consideration that $d = O(1)$ when p tends to infinity, we obtain that the number of invertible polynomials in $\mathbb{F}_2[x]/(x^p - 1)$ tends to $2^{p-1}$when p tends to infinity, which is exactly the number of polynomials with an odd Hamming weight. So

Security level	n_0	n	p	ω	t	d
80	2	9602	4801	90	84	4
	3	10779	3593	153	53	2
	4	12316	3079	220	42	2
128	2	19714	9857	142	134	2
	3	22299	7433	243	85	2
	4	27212	6803	340	68	1
256	2	65542	32771	274	264	1
	3	67593	22531	465	167	1
	4	81932	20483	644	137	7

Figure 5.1 – Suggested parameters for the QC-MDPC scheme in [MTSB13]

the probability of choosing an invertible polynomial from the set of polynomials with an odd Hamming weight is $\left(1 - 2^{-\frac{p-1}{d}}\right)^d$, which tends to 1 when $d = O(1)$.

We notice from Figure 5.1 that the designers of the scheme considered in most of the cases $d = 1$ or $d = 2$. Therefore they maximized the cardinality of the set of invertible polynomials.

Remark 5.3.3. *Furthermore we restrict the study to the case of a two blocks $(2p, p, \omega)$ QC-MDPC scheme that is to say $n_0 = 2$ with p a prime number such that $d = 1$ or possibly $d = 2$. We consider that for $1 \leqslant i \leqslant 2$ we have $\|h_i\| = \omega_i$ with ω_i an odd integer and $\omega_1 + \omega_2 = \omega$. Thus we maximize our chances that h_i is invertible.*

In [MTSB13] it is also recommended to choose the polynomials h_i with $1 \leqslant i \leqslant 2$, using a *smooth* distribution of their Hamming weight, by that we understand that $\omega_1 = \omega_2 = \omega/2$. We analyze in detail in the next section the consequence of a *smooth* distribution as well as *non-smooth* distribution.

5.4 Weak keys for the QC-MDPC scheme

5.4.1 The key recovery attack

We are interested in a *key-recovery under a chosen plaintext attack* when applied on a $(2p, p, \omega)$ QC-MDPC scheme.

Definition 5.4.1 (QC-MDPC Key Recovery Problem)**.** *Given the public key $f \in \mathbb{F}_2[x]/(x^p - 1)$, of a $(2p, p, \omega)$ QC-MDPC scheme, find a pair of polynomials (h_1, h_2), with $h_i \in \mathbb{F}_2[x]/(x^p - 1)$ satisfying*

$$f = \frac{h_1}{h_2} \pmod{x^p - 1} \text{ and } \|h_1\| + \|h_2\| \leqslant \omega. \tag{5.6}$$

This problem can be tackled by applying classical techniques based on exponential algorithms seeking low-weight codewords, which is the idea used in [MTSB13] in order to set up the security level of the scheme. It can also be recast as the problem of solving the *rational reconstruction problem* that is described in full details in Sec. 5.4.2. The extended Euclidean algorithm solves (5.6) when there exists an integer $t > 0$ such that $\deg(h_1) < t \leqslant p$ and $\deg(h_2) \leqslant p - t$.

5.4.2 The Rational Reconstruction Problem

Definition 5.4.2 (Rational reconstruction)**.** *Let g and f be polynomials in $\mathbb{K}[x]$ where $\mathbb{K}$ is a field such that $0 < \deg(f) < \deg(g)$. For a given integer r satisfying $1 \leqslant r \leqslant \deg(g)$, the* rational reconstruction *of f modulo g consists in finding φ and ψ in $\mathbb{K}[x]$ such that $\gcd(\varphi, g) = 1$, $\deg(\psi) < r$ and $\deg(\varphi) \leqslant \deg(g) - r$ and satisfying*

$$\frac{\psi}{\varphi} = f \pmod{g}. \tag{RR}$$

Remark 5.4.3. *When $g = x^p$ then we rather speak of Padé approximation.*

Note that if (RR) has a solution (φ, ψ) then $\frac{\psi}{\varphi}$ is unique. Furthermore if $(\varphi, \psi) \in \mathbb{K}[x]^2$ is a solution of the problem (RR), then it is also a solution to the following problem.

Definition 5.4.4. *Let g and f be polynomials in $\mathbb{K}[x]$ where $\mathbb{K}$ is a field such that $0 < \deg(f) < \deg(g) = p$. For a given integer r satisfying $1 \leqslant r \leqslant p$, the SRR problem consists in finding φ and ψ in $\mathbb{K}[x]$ such that $(\varphi, \psi) \neq (0, 0)$ and $\deg(\psi) < r$ and $\deg(\varphi) \leqslant p - r$ and satisfying*

$$\varphi f = \psi \pmod{g} \tag{SRR}$$

Clearly, any solution to (SRR) is solution to (RR) if and only if $\gcd(\varphi, g) = 1$. Moreover, (SRR) always has a non-trivial solution since recovering φ and ψ can be done by solving a linear system of p equations with $r + (p - r + 1) = p + 1$ unknowns representing the coefficients of φ and ψ.

A very efficient way to solve (RR) is to apply the Extended Euclidean Algorithm (EEA) to (f, g). Recall that if we denote by $(\varphi_i, \delta_i, \psi_i)$, with $i \geqslant 0$, the polynomials obtained at the i-th step of $\mathsf{EEA}(f, g)$ then we have $\psi_0 \stackrel{\text{def}}{=} g$, $\psi_1 \stackrel{\text{def}}{=} f$ and for all $i \geqslant 0$:

$$\begin{cases} \psi_i = Q_{i+1}\psi_{i+1} + \psi_{i+2} & \text{with } 0 \leqslant \deg(\psi_{i+2}) < \deg(\psi_{i+1}), \\ \\ \psi_i = \varphi_i f + \delta_i g & \text{with } (\varphi_0, \varphi_1) \stackrel{\text{def}}{=} (0, 1) \text{ and } (\delta_0, \delta_1) \stackrel{\text{def}}{=} (1, 0). \end{cases}$$

We also have the relations $\varphi_{i+2} = -Q_{i+1}\varphi_{i+1} + \varphi_i$ and $\delta_{i+2} = -Q_{i+1}\delta_{i+1} + \delta_i$.

Input: Two polynomials $f, g \in \mathbb{K}[x]$
Output: The polynomial $\psi \in \mathbb{K}[x]$ s.t. $\psi = \gcd(f, g)$ together with $\varphi, \delta \in \mathbb{K}[x]$ s.t. $\psi = \varphi f + \delta g$

$\psi_0 = g,\ \varphi_0 = 0,\ \delta_0 = 1;$
$\psi_1 = f,\ \varphi_0 = 1,\ \delta_0 = 0;$
$i = 0;$
while $\psi_{i+2} \neq 0$ **do**
  $Q_{i+1} = \psi_i \text{ quo } \psi_{i+1}$;
  $\psi_{i+2} = -Q_{i+1}\psi_{i+1} + \psi_i;$
  $\varphi_{i+2} = -Q_{i+1}\varphi_{i+1} + \varphi_i;$
  $\delta_{i+2} = -Q_{i+1}\delta_{i+1} + \delta_i$
end
return $(\psi_i, \varphi_i, \delta_i)$

Algorithm 1: The Extended Euclidean Algorithm

We are now able to prove that this approach provides a non-trivial solution. Let's denote by j the smallest index such that $\deg(\psi_j) < r \leqslant \deg(\psi_{j-1})$. We require the following proposition.

Proposition 5.4.5 ([Pan12, Chapter 2]). *At each step $i \geqslant 0$ of $\mathsf{EEA}(f,g)$ it holds that*

$$\deg(\varphi_{i+1}) = p - \deg(\psi_i). \tag{5.7}$$

The following proposition characterizes a solution to (RR) when it exists.

Proposition 5.4.6 ([Pan12]). *Let $(\varphi_i, \delta_i, \psi_i)$, with $i \geqslant 0$, be the polynomials obtained at the i-th step of $\mathsf{EEA}(f,g)$. Let j be the smallest integer such that $\deg(\psi_j) < r$ then (φ_j, ψ_j) is a non-trivial solution to (SRR). Furthermore, if (φ, ψ) is a solution to (RR) then there exists λ in $\mathbb{K}\backslash\{0\}$ such that $\varphi = \lambda\varphi_j$ and $\psi = \lambda\psi_j$.*

5.4.3 Weak keys

This section is devoted to the identification of private keys (h_1, h_2) that can be recovered from the public key f by means of the Extended Euclidean Algorithm.

Main idea of the attack. Since $f = \dfrac{h_1}{h_2} \pmod{x^p - 1}$, the idea of our attack is to find a rational reconstruction of f_1 modulo $x^p - 1$. At each step t of algorithm $\mathsf{EEA}(f, x^p - 1)$, the attacker checks if the ongoing computed polynomials denoted by (ψ_t, φ_t) where $\psi_t = f\varphi_t$, satisfy the inequality

$$\|\varphi_t\| + \|f_1\varphi_t\| \leqslant \omega. \tag{5.8}$$

If such a solution is found then by Proposition 5.4.6 we have found (equivalent) secret polynomials. With these polynomials we built an equivalent parity-check matrix for the QC-MDPC code and thus we are able to decrypt any ciphertext (see Remark 5.2.13). The main question we want to answer is to estimate precisely the number of keys that can be recovered with this technique.

Definition 5.4.7. *Let p be a prime number and ω an even integer with $1 < \omega < p$. Let $(\omega_1, \omega_2) \in \mathbb{N}^2$ be odd integers such that $\omega_1 + \omega_2 = \omega$. We define the set of* private pairs with fixed weights *by*

$$\mathcal{P}_{\omega_1,\omega_2} = \left\{ (h_1, h_2) \in \left(\mathbb{F}_2[x]/(x^p - 1)\right)^2 \mid \|h_i\| = \omega_i \text{ and } \omega_i \text{ odd} \right\},$$

and the set of all private pairs *of a $(2p, p, \omega)$ QC-MDPC scheme by $\mathcal{P}_\omega = \bigcup_{\omega_1+\omega_2=\omega} \mathcal{P}_{\omega_1,\omega_2}$.*

Private pairs that can be recovered using the Extended Euclidean Algorithm are declared weak pairs.

Definition 5.4.8. *A pair $(h_1, h_2) \in \mathcal{P}_\omega$ is called a* weak pair *if*

$$\deg(h_1) + \deg(h_2) < p. \tag{5.9}$$

The set of weak pairs is denoted by $\mathcal{W}_\omega = \{(h_1, h_2) \in \mathcal{P}_\omega \mid \deg(h_1) + \deg(h_2) < p\}$. Similarly, $\mathcal{W}_{\omega_1,\omega_2}$ is defined as $\mathcal{W}_\omega \cap \mathcal{P}_{\omega_1,\omega_2}$.

Remark 5.4.9.

- *The exact collection of private keys of a general $(2p, p, \omega)$ QC-MDPC scheme is actually the set* $\mathcal{P}_\omega{}^* = \bigcup_{\omega_1+\omega_2=\omega} \mathcal{P}^*_{\omega_1,\omega_2}$ *where*

$$\mathcal{P}^*_{\omega_1,\omega_2} = \left\{ (h_1, h_2) \in \mathcal{P}_{\omega_1,\omega_2} \mid \gcd(h_2, x^p - 1) = 1 \right\}.$$

- *Since we consider parameters for which $d = 1$ and $d = 2$, where d comes from the factorization of $x^p - 1$, we have*
  - *when $d = 1$*
$$\mathcal{P}_{\omega_1,\omega_2} = \mathcal{P}^*_{\omega_1,\omega_2} \text{ and } \quad \mathcal{P}_\omega = \mathcal{P}_\omega{}^*.$$
  - *when $d = 2$ (see Proposition 5.4.14)*
$$\lim_{p\to\infty} \sum_{\substack{\omega=2 \\ \omega \text{ even}}}^{p} |\mathcal{W}_\omega{}^*| = \lim_{p\to\infty} \sum_{\substack{\omega=2 \\ \omega \text{ even}}}^{2p} |\mathcal{W}_\omega|.$$

Proposition 5.4.10. *Let p be a prime number and ω an even integer with $1 < \omega < p$. Let $(\omega_1, \omega_2) \in \mathbb{N}^2$ be odd integers such that $\omega_1 + \omega_2 = \omega$. Then we have*

$$|\mathcal{W}_{\omega_1,\omega_2}| = \binom{p+1}{\omega_1+\omega_2} \quad \text{and} \quad |\mathcal{W}_\omega| = \frac{\omega}{2}\binom{p+1}{\omega}. \tag{5.10}$$

$$|\mathcal{P}_{\omega_1,\omega_2}| = \binom{p}{\omega_1}\binom{p}{\omega_2} \quad \text{and} \quad |\mathcal{P}_\omega| = \frac{1}{2}\left(\binom{2p}{\omega} - (-1)^{\frac{\omega}{2}}\binom{p}{\frac{\omega}{2}}\right). \tag{5.11}$$

Proof. Let $(h_1, h_2) \in \mathcal{P}_{\omega_1,\omega_2}$. Then h_i has ω_i non-zero coefficients, and a degree less than p, hence $|\mathcal{P}_{\omega_1,\omega_2}| = \binom{p}{\omega_1}\binom{p}{\omega_2}$. For $(h_1, h_2) \in \mathcal{W}_{\omega_1,\omega_2}$ we have $\deg(h_1) + \deg(h_2) < p$. If $k = \deg(h_1)$, then h_1 has a leading coefficient x^k and $\omega_1 - 1$ non-zero coefficients between x^0 and x^{k-1}. The number of such polynomials is $\binom{k}{\omega_1-1}$.

Furthermore the number of polynomials h_2 with ω_2 non-zero coefficients and $\deg(h_2) < p - k$ equals $\binom{p-k}{\omega_2}$. Using the Gould's formulae [Gou72], we get

$$|\mathcal{W}_{\omega_1,\omega_2}| = \sum_{k=0}^{p-1} \binom{k}{\omega_1 - 1}\binom{p-k}{\omega - \omega_1} = \binom{p+1}{\omega},$$

and

$$|\mathcal{P}_\omega| = \sum_{\substack{\omega_1+\omega_2=\omega \\ \omega_i \text{ odd}}} \binom{p}{\omega_1}\binom{p}{\omega_2} = \frac{1}{2}\left[\binom{2p}{\omega} - (-1)^{\frac{\omega}{2}}\binom{p}{\frac{\omega}{2}}\right].$$

As for $\mathcal{W}_\omega$ we obtain:

$$|\mathcal{W}_\omega| = \sum_{\substack{\omega_1+\omega_2=\omega \\ \omega_i \text{ odd}}} \binom{p+1}{\omega} = \binom{p+1}{\omega} \sum_{\substack{\omega_1+\omega_2=\omega \\ \omega_i \text{ odd}}} 1 = \frac{\omega}{2}\binom{p+1}{\omega}.$$

□

Corollary 5.4.11. *Let p be a prime number and ω an even integer with $1 < \omega < p$ and $H(\alpha) = -\alpha \log_2 \alpha - (1-\alpha)\log_2(1-\alpha)$ be the binary entropy function for any $0 \leqslant \alpha \leqslant 1$. Let $(\omega_1, \omega_2) \in \mathbb{N}^2$ be odd integers such that $\omega_1 + \omega_2 = \omega$ and $\omega = o(p)$. Then we have*

$$\frac{|\mathcal{W}_{\omega_1,\omega_2}|}{|\mathcal{P}_{\omega_1,\omega_2}|} = \sqrt{2\pi\alpha(1-\alpha)\omega}\,2^{-\omega H(\alpha)} \times \begin{cases} e^{-2\sqrt{c_1 c_2}}\left(1 + O(\frac{1}{\sqrt{p}})\right) & \text{if } \frac{\omega_i^2}{2p} = c_i + O(\frac{1}{\sqrt{p}}) \\ p^{-2\sqrt{c_1 c_2}}\left(1 + O(\sqrt{\frac{\log^3 p}{p}})\right) & \text{if } \frac{\omega_i^2}{2p} = c_i \log p + O(\sqrt{\frac{\log p}{p}}) \end{cases}$$

with $\alpha = \dfrac{1}{1+\sqrt{\frac{c_2}{c_1}}}$.

$$\frac{|\mathcal{W}_{\omega}|}{|\mathcal{P}_{\omega}|} = \omega 2^{-\omega} \times \begin{cases} e^{-\frac{c}{2}}\left(1 + O(\frac{1}{\sqrt{p}})\right) & \text{if } \frac{\omega^2}{2p} = c + O(\frac{1}{\sqrt{p}}), \\ p^{-\frac{c}{2}}\left(1 + O(\sqrt{\frac{\log^3 p}{p}})\right) & \text{if } \frac{\omega^2}{2p} = c\log p + O(\sqrt{\frac{\log p}{p}}). \end{cases}$$

For all the asymptotic expansion that we derive in this Chapter we mainly use the Stirling formula. In Appendix D we detail the asymptotic expansion of the binomial coefficient and deduce all the asymptotic expansions that we state here.

Corollary 5.4.12. *In particular for a smooth $(2p, p, \omega)$ QC-MDPC, we have*

$$\frac{\left|\mathcal{W}_{\omega/2,\omega/2}\right|}{\left|\mathcal{P}_{\omega/2,\omega/2}\right|} = \frac{\binom{p+1}{\omega}}{\binom{p}{\omega/2}^2},$$

with asymptotic equivalence

$$\frac{\left|\mathcal{W}_{\omega/2,\omega/2}\right|}{\left|\mathcal{P}_{\omega/2,\omega/2}\right|} \sim \begin{cases} \sqrt{\pi} p^{\frac{1}{4}} e^{-2} 2^{\frac{1}{4} - 2\sqrt{2p}} & \text{if } \omega = 2\sqrt{2p}, \\ \sqrt{\pi} p^{\frac{1}{4}-2} \log^{\frac{1}{4}} p 2^{\frac{1}{4} - 2\sqrt{2p\log p}} & \text{if } \omega = 2\sqrt{2p\log p}. \end{cases}$$

Remark 5.4.13. *We notice from Proposition 5.4.10 that the probability of a weak key with fixed weight is asymptotically dominated by $2^{-\omega H(\alpha)}$. Therefore the most secure choice for the parameters is when α equals 1, which is equivalent to $\omega_1 = \omega_2 = \omega/2$, in other words the smooth distribution. Nevertheless the impact of a non* smooth *distribution is not measured in [MTSB13] but now it can be determined by the means of Proposition 5.4.10 (see Figure 5.3 for numerical values).*

In the next proposition we demonstrate that the cardinality of the set of all weak pairs for all the possible values of ω tends to the cardinality of the set of all weak keys when $d = 2$. Hence our choice of counting weak pairs, when the length goes to infinity, is a justified solution, for $d = 1$ and $d = 2$.

Proposition 5.4.14. *Let $x^p - 1 = (x-1)g_1(x)g_2(x)$ be the decomposition of $x^p - 1$ into irreducible factors over $\mathbb{F}_2[x]$. Then we have:*

$$\lim_{p\to\infty} \sum_{\substack{\omega=2 \\ \omega \text{ even}}}^{p} |\mathcal{W}_\omega{}^*| = \lim_{p\to\infty} \sum_{\substack{\omega=2 \\ \omega \text{ even}}}^{p} |\mathcal{W}_\omega|$$

Proof. Since the set of weak keys equals the set of weak pairs minus the set of non invertible pairs of odd weight polynomials (h_1, h_2) we have:

$$\begin{array}{rcl} \mathcal{W}_\omega \setminus (\mathcal{P}_\omega \setminus {\mathcal{P}_\omega}^*) & = & {\mathcal{W}_\omega}^* \subseteq \mathcal{W}_\omega \\ \bigcup\limits_{\substack{\omega=2 \\ \omega \text{ even}}}^{p} \mathcal{W}_\omega \setminus (\mathcal{P}_\omega \setminus {\mathcal{P}_\omega}^*) & = & \bigcup\limits_{\substack{\omega=2 \\ \omega \text{ even}}}^{p} {\mathcal{W}_\omega}^* \subseteq \bigcup\limits_{\substack{\omega=2 \\ \omega \text{ even}}}^{p} \mathcal{W}_\omega \\ \sum\limits_{\substack{\omega=2 \\ \omega \text{ even}}}^{p} \left(|\mathcal{W}_\omega| - |\mathcal{P}_\omega \setminus {\mathcal{P}_\omega}^*| \right) & \leqslant & \sum\limits_{\substack{\omega=2 \\ \omega \text{ even}}}^{p} |{\mathcal{W}_\omega}^*| \leqslant \sum\limits_{\substack{\omega=2 \\ \omega \text{ even}}}^{p} |\mathcal{W}_\omega| \end{array}$$

We compute the cardinal of pairs of non invertible polynomials with odd Hamming weight (h_1, h_2): $\sum\limits_{\substack{\omega=2 \\ \omega \text{ even}}}^{p} |\mathcal{P}_\omega \setminus {\mathcal{P}_\omega}^*| = \left(2^{p-1} - (2^{\frac{p-1}{2}} - 1)^2\right)^2 = (2^{\frac{p+1}{2}} - 1)^2$.

From Proposition 5.4.10 we deduce

$$\begin{aligned} \sum_{\substack{\omega=2 \\ \omega \text{ even}}}^{p} |\mathcal{W}_\omega| &= \sum_{\substack{\omega=2 \\ \omega \text{ even}}}^{p} \frac{\omega}{2} \binom{p+1}{\omega} \\ &= \sum_{\substack{\omega=2 \\ \omega \text{ even}}}^{p} \frac{p+1}{2} \binom{p}{\omega-1} \\ &= \frac{p+1}{2} 2^{p-1} \end{aligned}$$

So when $d = 2$ we have that:

$$\lim_{p\to\infty} \frac{\sum\limits_{\substack{\omega=2 \\ \omega \text{ even}}}^{p} |\mathcal{P}_\omega \setminus {\mathcal{P}_\omega}^*|}{\sum\limits_{\substack{\omega=2 \\ \omega \text{ even}}}^{p} |\mathcal{W}_\omega|} \leqslant \lim_{p\to\infty} \frac{8}{p+1} = 0.$$

□

Numerical results. In Figure 5.2 we plot the numerical results for the suggested parameters in [MTSB13] and [Bal14]. We notice that the values we obtain for the proportion of weak keys are lower than the security level, which is a good security argument in favor of the parameters that were chosen for the scheme. The values were computed in PARI/GP as well as in MAPLE software.

In Figure 5.3 we compute the proportion of weak keys for the 80 bit security parameters, namely the $(9602, 4801, 90)$ QC-MDPC. In this case we consider all the possible values for (ω_1, ω_2) such that $\omega_1 + \omega_2 = 90$ with $21 \leqslant \omega_1 \leqslant 45$. We notice that a non smooth distribution of the Hamming weight has a negative impact on the security of the scheme, reason for which we suggest to choose $\omega_1 = \omega_2 = \omega/2$.

Since the proportion of weak keys is lower that the security level for a smooth $(2p, p, \omega)$ QC-MDPC, we focus our interest in finding additional techniques that allow us to increase this proportion. For that we naturally begin by considering the equivalence problem for quasi-cyclic codes.

Security level	p	$\frac{\omega}{2}$	$\frac{\lvert\mathcal{W}_{\omega/2,\omega/2}\rvert}{\lvert\mathcal{P}_{\omega/2,\omega/2}\rvert}$ Corollary 5.4.12
80	4801	45	2^{-87}
100	6851	56	2^{-109}
128	9857	71	2^{-139}
160	15101	87	2^{-171}
256	32771	132	2^{-260}

Figure 5.2 – Proportion of weak key for the $(2p, p, \omega)$ QC-MDPC when $\omega_1 = \omega_2 = \frac{\omega}{2}$.

ω_1	45	43	41	39	37	35	33	31	29	27	25	23	21
$\log \frac{\lvert\mathcal{W}_{\omega_1,\omega_2}\rvert}{\lvert\mathcal{P}_{\omega_1,\omega_2}\rvert}$	−87.0	−86.9	−86.5	−85.8	−84.9	−83.8	−82.3	−80.6	−78.6	−76.3	−73.7	−70.8	−67.6

Figure 5.3 – Proportion of weak keys for the $(9602, 4801, 90)$ QC-MDPC for $\omega_1 = 90 - \omega_2$ with $21 \leqslant \omega_1 \leqslant 45$.

5.5 Equivalence of codes

5.5.1 Equivalence of cyclic codes

The equivalence of cyclic codes is presented in [MS86, Chapter8]. There are two group actions that preserves a cyclic code $\mathscr{C}$, group actions that we redefine here using the algebraic formalism for cyclic codes.

Preliminaries

Definition 5.5.1. *Let $\mathscr{C}$ be a $[p, k]$ binary cyclic code and $\pi \in \mathsf{Perm}(\mathscr{C})$. Then for all $\boldsymbol{c} \in \mathscr{C}$ the permutation π acts on the polynomial $c(x)$ as in Definition 2.2.26*

$$c^{\pi}(x) = \sum_{i=0}^{p-1} c_{\pi^{-1}(i)} x^i.$$

Next we define the first group that acts as a permutation and leaves a cyclic code invariant. It comes directly from the definition of cyclic codes and consists of the cyclic shifts:

Definition 5.5.2. *The additive group $(\mathbb{F}_p, +)$ acts on the set of polynomials $\mathbb{F}_2[x]/(x^p - 1)$ as:*

$$\begin{array}{ccccccc} \sigma^+ : \mathbb{F}_p & \times & \mathbb{F}_2[x]/(x^p-1) & \to & \mathbb{F}_2[x]/(x^p-1) \\ (\alpha & , & c(x)) & \mapsto & \sigma^+_\alpha(c(x)) = x^\alpha c(x). \end{array}$$

We denote by $\mathbb{F}_p.c$ the orbit of $c(x)$ under the action of $\mathbb{F}_p$.

Remark 5.5.3.

- *We notice that σ^+ is a group action since we have $\sigma^+_0(c(x)) = c(x)$ for any $c(x) \in \mathbb{F}_2[x]/(x^p - 1)$ and $\sigma^+_\alpha(\sigma^+_\beta(c(x))) = \sigma^+_{\alpha+\beta}(c(x))$ for any $c(x) \in \mathbb{F}_2[x]/(x^p - 1)$ and α and $\beta \in (\mathbb{F}_p, +)$.*

- *We also have that σ_α^+ induces a permutation on the coefficients of any polynomial in $\mathbb{F}_2[x]/(x^p-1)$ and thus doesn't modify the weight of the polynomial*

$$\sigma_\alpha^+(c(x)) = \sum_{i=0}^{p-1} c_{i-\alpha \pmod p} x^i. \tag{5.12}$$

Proposition 5.5.4. *Let $\mathscr{C}$ be a $[p,k]$ binary cyclic code. Then for any $\alpha \in (\mathbb{F}_p, +)$ we have*

$$\sigma_\alpha^+(\mathscr{C}) = \mathscr{C}.$$

Proof. Let $c(x)$ be a polynomial in $\mathbb{F}_2[x]/(x^p-1)$ and $\mathscr{C}$ be the ideal $\langle c(x)\rangle$. By definition of the group action we have that $\sigma_\alpha^+(c(x))$ is an element of the ideal $\langle c(x)\rangle$. Conversely we have that $c(x)$ is an element of the ideal generated by $< \sigma_\alpha^+(c(x)) >$. Indeed, since x^α and x^p-1 are coprime then x^α admits an inverse modulo x^p-1 and thus we can write $c(x) = (x^\alpha)^{-1}(x^\alpha c(x))$. □

The second group action that plays an important role in the equivalence of cyclic codes is:

Definition 5.5.5. *The action of the multiplicative group $(\mathbb{F}_p{}^*, \cdot)$ over $\mathbb{F}_2[x]/(x^p-1)$ can be defined as follow:*

$$\begin{array}{ccccc} \sigma^* : \mathbb{F}_p{}^* & \times & \mathbb{F}_2[x]/(x^p-1) & \to & \mathbb{F}_2[x]/(x^p-1) \\ (\alpha & , & c(x)) & \mapsto & \sigma_\alpha^*(c(x)), \end{array}$$

where $\sigma_\alpha^*\left(\sum_{i=0}^{p-1} c_i x^i\right) = \sum_{i=0}^{p-1} c_i x^{\alpha i} \pmod{x^p-1}$.

We denote by $\mathbb{F}_p{}^*.c$ the orbit of $c(x)$ under the action of $\mathbb{F}_p{}^*$.

Remark 5.5.6.

- *σ^* is well defined as a group action on the set of polynomials in $\mathbb{F}_2[x]/(x^p-1)$ since $\sigma_1^*(x^i) = x^i, \forall i$ and for any pair α and β elements of the group $(\mathbb{F}_p{}^*, \cdot)$ we have*

$$\begin{aligned} \sigma_\alpha^*(\sigma_\beta^*(x^i)) &= x^{\alpha\beta i} \pmod{x^p-1} \\ &= \sigma_{\alpha\beta}^*(x^i), \quad \forall 0 \leqslant i \leqslant p-1. \end{aligned}$$

- *We also have that σ_α^* induce a permutation on the coefficients of any polynomial in $\mathbb{F}_2[x]/(x^p-1)$ and thus doesn't modify the weight of the polynomial*

$$\sigma_\alpha^*(c(x)) = \sum_{i=0}^{p-1} c_{i\alpha^{-1} \pmod p} x^i, \tag{5.13}$$

It is known from the literature that if two cyclic codes $\mathscr{C}_1$ and $\mathscr{C}_2$ are such that $\exists \alpha \in \mathbb{F}_p{}^*$ satisfying $\mathscr{C}_1 = \sigma_\alpha^*(\mathscr{C}_2)$, then we say that $\mathscr{C}_1$ and $\mathscr{C}_2$ are multiplier equivalent.

Proposition 5.5.7. *Let $\mathscr{C}$ be a $[p,k]$ cyclic code. Then for any α in the multiplicative subgroup $\langle 2\rangle \subseteq {\mathbb{F}_p}^*$ we have*

$$\sigma_\alpha^*(\mathscr{C}) = \mathscr{C}.$$

Proof. Let $c(x)$ be a polynomial in $\mathbb{F}_2[x]/(x^p-1)$ and $\mathscr{C}$ be the ideal $\langle c(x)\rangle$. By definition of the group action and using the field characteristic we have that $\sigma_2^*(c(x)) = c(x)^2$. Hence $\sigma_2^*(c(x))$ is an element of the ideal $\langle c(x)\rangle$. Conversely we have to prove that $c(x) \in \langle \sigma_2^*(c(x))\rangle$. First notice that for any power of two we have $c(x)^{2^i} \in \langle \sigma_2^*(c(x))\rangle$. Since $c(x) = c(x)^{2^{\mathsf{ord}(2)}}$, where $\mathsf{ord}(2)$ is the multiplicative order of 2, we have that $\mathscr{C} \in \sigma_2^*(\mathscr{C})$. □

When the two group actions are combined we use the following notation for the orbits

Notation 5.5.8. *We denote the orbit of a polynomial $c(x)$ under the action of $\mathbb{F}_p$ and ${\mathbb{F}_p}^*$ by ${\mathbb{F}_p}^*.(\mathbb{F}_p.c) = \cup_{f(x)\in\mathbb{F}_p.c}\{\sigma_\alpha^*(f(x)) \mid \alpha \in {\mathbb{F}_p}^*\}$ respectively $\mathbb{F}_p.({\mathbb{F}_p}^*.c) = \cup_{f(x)\in{\mathbb{F}_p}^*.c}\{\sigma_\alpha^+(f(x)) \mid \alpha \in \mathbb{F}_p\}$.*

5.5.2 Equivalence of quasi-cyclic codes.

Since we deal with $(2p, p, \omega)$ QC-MDPC codes we detail in this section only the case of 2-quasi-cyclic codes. Hence we consider a pair of polynomials $(h_1(x), h_2(x)) \in (\mathbb{F}_2[x]/(x^p-1))^2$ and study the group actions that act as a permutation and preserve the quasi-cyclic structure, notion that we define as follows:

Definition 5.5.9. *Let p be a prime number. We say that a permutation $\pi \in \mathfrak{S}_{2p}$ preserves the 2-quasi-cyclic structure if for any 2-quasi-cyclic code $\mathscr{C}$, $\mathscr{C}^\pi$ is a 2-quasi-cyclic code.*

The action of $\mathbb{F}_p$ and ${\mathbb{F}_p}^*$ can be extended in a natural manner to $(\mathbb{F}_2[x]/(x^p-1))^2$, by considering the action on each component.

Definition 5.5.10. *For any pair $(\alpha_1,\alpha_2) \in \mathbb{F}_p^2$ and any pair of polynomials $(h_1(x), h_2(x)) \in (\mathbb{F}_2[x]/(x^p-1))^2$ we define*

$$\sigma^+_{(\alpha_1,\alpha_2)}(h_1(x), h_2(x)) = \left(\sigma^+_{\alpha_1}(h_1(x)), \sigma^+_{\alpha_2}(h_2(x))\right).$$

Proposition 5.5.11. *For any pair $(\alpha_1,\alpha_2) \in \mathbb{F}_p^2$ the permutation induced by $\sigma^+_{(\alpha_1,\alpha_2)}$ is an element of the symmetric group $\mathfrak{S}_{2p}$ that preserves the 2-quasi-cyclic structure.*

Moreover for any $\alpha \in \mathbb{F}_p$ and any 2-quasi-cyclic code $\mathscr{C}$ we have

$$\sigma^+_{(\alpha,\alpha)}(\mathscr{C}) = \mathscr{C}.$$

As for the second group action we have

Definition 5.5.12. *For any pair $(\alpha_1,\alpha_2) \in ({\mathbb{F}_p}^*)^2$ and any pair of polynomials $(h_1, h_2) \in (\mathbb{F}_2[x]/(x^p-1))^2$ we define*

$$\sigma^*_{(\alpha_1,\alpha_2)}(h_1, h_2) = \left(\sigma^*_{\alpha_1}(h_1), \sigma^*_{\alpha_2}(h_2)\right).$$

Proposition 5.5.13. *For any $\alpha \in {\mathbb{F}_p}^*$ the permutation induced by $\sigma^*_{(\alpha,\alpha)}$ is an element of the symmetric group $\mathfrak{S}_{2p}$ that preserves the 2-quasi-cyclic structure.*

Since the later Proposition states that only $\sigma^*_{\alpha,\beta}$ with $\alpha = \beta$ preserves the 2-quasi-cyclic structure, we simplify the notations and rather say that ${\mathbb{F}_p}^*$ acts on the set of polynomials $\left(\mathbb{F}_2[x]/(x^p-1)\right)^2$.

Proposition 5.5.14. *For any $\pi \in \mathfrak{S}_2$, $(\beta_1, \beta_2) \in \mathbb{F}_p^2$ and $\alpha \in {\mathbb{F}_p}^*$ we have that $\pi \circ \sigma^+_{\beta_1,\beta_2} \circ \sigma^*_{\alpha,\alpha}$ preserves the 2-quasi-cyclic structure.*

Moreover for a given 2-quasi-cyclic code $\mathscr{C}$ there are at most $2p^2(p-1)$ equivalent 2-quasi-cyclic codes.

For the orbits we use the usual notations

Notation 5.5.15. $\left(\mathbb{F}_p\right)^2.(h_1,h_2) = \{\sigma^+_{(\alpha_1,\alpha_2)}(h_1(x),h_2(x)) \mid (\alpha_1,\alpha_2) \in \left(\mathbb{F}_p\right)^2\}$ *and* ${\mathbb{F}_p}^*.(h_1,h_2) = \{\sigma^*_{(\alpha,\alpha)}(h_1(x),h_2(x)) \mid \alpha \in {\mathbb{F}_p}^*\}$.

5.6 Attacking equivalent public keys

Considering the equivalence of quasi-cyclic codes for the QC-MDPC scheme can be used to extend the initial key recovery attack. In this part we explain how to deploy the Extended Euclidean Algorithm on the equivalent public keys.

5.6.1 The modified EEA for the attack

Equivalence under the action of $\left(\mathbb{F}_p\right)^2$. Let $f \in \mathbb{F}_2[x]/(x^p-1)$ be a public key of a $(2p,p,\omega)$ QC-MDPC scheme, and $(h_1,h_2) \in \mathbb{F}_2[x]/(x^p-1) \times \mathbb{F}_2[x]/(x^p-1)$ the corresponding private key. So we have that $f = \frac{h_1}{h_2} \mod (x^p-1)$. Now assume that h_1 and h_2 can not be recovered only by applying $\mathsf{EEA}(x^p-1,f)$ but instead there exists $(\alpha_1,\alpha_2) \in \mathbb{F}_p^2$ such that $(x^{\alpha_1}h_1, x^{\alpha_2}h_2)$ is a weak key. Then the public key $x^{\alpha_1-\alpha_2}f = \frac{x^{\alpha_1}h_1}{x^{\alpha_2}h_2}$ can be attacked by EEA, which is equivalent to say that

$$\exists \alpha_1,\alpha_2 \in \mathbb{F}_p^2 \text{ such that } \deg(x^{\alpha_1}h_1) + \deg(x^{\alpha_2}h_2) < p. \tag{5.14}$$

Using this idea if our attack does not work on f we repeat it on all $p-1$ cyclic shifts of f, namely $xf, x^2f, \ldots, x^{p-1}f$. If there is a shift such that the outgoing polynomials satisfy the weight conditions in (5.8) then we have successfully recovered (equivalent) secret polynomials by Proposition 5.4.6. In other words we extend our technique to equivalent private keys, more exactly equivalent under the action of the additive group $\left(\mathbb{F}_p,+\right)^2$. We call these solutions *weak orbits* and formally define them:

Definition 5.6.1 (Weak orbit)**.** *The set $\left(\mathbb{F}_p\right)^2.(h_1,h_2)$ defined by a private key $(h_1,h_2) \in \mathbb{F}_2[x]/(x^p-1)^2$ of a $(2p,p,\omega)$ QC-MDPC is called a* weak orbit *if it contains at least one weak key, i.e. satisfies (5.14).*

We denote $\widetilde{\mathcal{W}}_{\omega_1,\omega_2} = \{(h_1,h_2) \in \mathcal{P}_{\omega_1,\omega_2} \mid (h_1,h_2)$ satisfy (5.14)$\}$ the set of private keys that define weak orbits.

In the case of all $(2p,p,\omega)$ QC-MDPC codes we denote

$$\widetilde{\mathcal{W}}_{\omega} = \bigcup_{\substack{\omega_1+\omega_2=\omega \\ \omega_1 \text{ odd}}} \widetilde{\mathcal{W}}_{\omega_1,\omega_2}.$$

We deduce immediately that

$$\mathcal{W}_{\omega_1,\omega_2} \subseteq \widetilde{\mathcal{W}}_{\omega_1,\omega_2} \quad \text{and} \quad |\mathcal{W}_{\omega_1,\omega_2}| \leqslant \left|\widetilde{\mathcal{W}}_{\omega_1,\omega_2}\right| \leqslant p^2 \left|\mathcal{W}_{\omega}\right|. \tag{5.15}$$

Equivalence under the action of $\mathbb{F}_p{}^*$. Let $f \in \mathbb{F}_2[x]/(x^p-1)$ be the public key of a $(2p, p, \omega)$ QC-MDPC scheme, and $(h_1, h_2) \in \mathbb{F}_2[x]/(x^p-1) \times \mathbb{F}_2[x]/(x^p-1)$ the corresponding private key. Now suppose that $\nexists (\alpha_1, \alpha_2) \in \mathbb{F}_p^2$ such that $(x^{\alpha_1}h_1, x^{\alpha_2}h_2)$ is a weak key, in other words (h_1, h_2) can not be recovered by means of EEA applied to f or any of the cyclic shifts of f.

The idea to increase our chances of finding the private key or an equivalent one is to use the second group action, namely $(\mathbb{F}_p{}^*, \cdot)$. Therefore we start our attack by fixing $\alpha \in \mathbb{F}_p{}^*$ and try to find a rational reconstruction of $\sigma_\alpha^*(f)$ modulo x^p-1. If the algorithm finds a solution (ψ_t, φ_t) where $\psi_t = \alpha \cdot f\varphi_t$ satisfy the inequality

$$\|\varphi_t\| + \|\psi_t\| \leqslant \omega. \tag{5.16}$$

then we have found as before (equivalent) secret polynomials.

Otherwise, the attacker performs the same attack to all shifts of f, namely $\alpha \cdot x^j f$. If the attack fails, another α is chosen and the procedure is repeated until the good combination of α and shifts are founded. We call these solutions *extended weak orbits* and define them

Definition 5.6.2 (Extended weak orbits)**.** *Let $(h_1, h_2) \in (\mathbb{F}_2[x]/(x^p-1))^2$ be a private key of a $(2p, p, \omega)$ QC-MDPC. We say that the set $\mathbb{F}_p{}^*.\left((\mathbb{F}_p)^2.(h_1, h_2)\right)$ is an extended weak orbit if and only if it contains at least one weak orbit.*

We denote $\widetilde{\widetilde{\mathcal{W}}}_{\omega_1,\omega_2} = \{(h_1, h_2) \in \mathcal{P}_{\omega_1,\omega_2} \mid \exists \alpha \in \mathbb{F}_p{}^* \text{ s.t. } \sigma^*_{\alpha,\alpha}(h_1, h_2) \text{ satisfy } (5.14)\}$ the set of private keys that define extended weak orbits.

In the case of all $(2p, p, \omega)$ QC-MDPC codes we denote

$$\widetilde{\widetilde{\mathcal{W}}}_{\omega} = \bigcup_{\substack{\omega_1+\omega_2=\omega \\ \omega_1 \text{ odd}}} \widetilde{\widetilde{\mathcal{W}}}_{\omega_1,\omega_2}.$$

We deduce immediately that

$$\mathcal{W}_{\omega_1,\omega_2} \subseteq \widetilde{\mathcal{W}}_{\omega_1,\omega_2} \subseteq \widetilde{\widetilde{\mathcal{W}}}_{\omega_1,\omega_2} \quad \text{and} \quad \left|\widetilde{\mathcal{W}}_{\omega_1,\omega_2}\right| \leqslant \left|\widetilde{\widetilde{\mathcal{W}}}_{\omega_1,\omega_2}\right| \leqslant (p-1)\left|\widetilde{\mathcal{W}}_{\omega_1,\omega_2}\right|. \tag{5.17}$$

We are left to determine how these two group actions increase the proportion of equivalent private keys that can be recovered using the Extended Euclidean Algorithm.

5.6.2 Orbits of the QC-MDPC private keys

So far we have seen that applying the Extended Euclidean Algorithm on the public key f of a $(2p, p, \omega)$ QC-MDPC scheme reveals a proportion of weak keys that is lower than the security level given by the designers in [MTSB13]. Thus we have defined two group actions that could potentially increase the proportion of weak keys. In order to achieve our goal, namely to determine how these group actions might increase the proportion of weak keys we proceed in two steps

1. we determine the cardinality of the orbits of $(h_1(x), h_2(x)) \in \mathcal{P}_{\omega_1,\omega_2}$ under the action of $(\mathbb{F}_p)^2$ and $\mathbb{F}_p{}^*$.

2. we compute the cardinality of the orbits of $(h_1(x), h_2(x)) \in \widetilde{\widetilde{\mathcal{W}}}_{\omega_1,\omega_2}$ under the action of $(\mathbb{F}_p)^2$ and ${\mathbb{F}_p}^*$.

Before we detail how to compute the cardinality of the orbits we give an upper bound on the proportion of weak keys that might be obtained using these two group actions. Using Equation (5.15) and (5.17) we have

$$\frac{\left|\widetilde{\widetilde{\mathcal{W}}}_{\omega_1,\omega_2}\right|}{|\mathcal{P}_{\omega_1,\omega_2}|} \leqslant p^2(p-1)\frac{|\mathcal{W}_{\omega_1,\omega_2}|}{|\mathcal{P}_{\omega_1,\omega_2}|}. \tag{5.18}$$

In order to better illustrate the potential effect of the group action that we propose here, we plot in Figure 5.4 the numerical values for proportion of weak keys and the upper bound from (5.18) for the proportion of extended weak orbits for the $(2p, p, \omega)$ QC-MDPC scheme.

Security level	p	$\frac{\omega}{2}$	$\frac{\lvert\mathcal{W}_{\omega/2,\omega/2}\rvert}{\lvert\mathcal{P}_{\omega/2,\omega/2}\rvert}$ Corollary 5.4.12	$p^2(p-1)\frac{\lvert\mathcal{W}_{\omega/2,\omega/2}\rvert}{\lvert\mathcal{P}_{\omega/2,\omega/2}\rvert}$ Equation (5.18)
80	4801	45	2^{-87}	2^{-49}
100	6851	56	2^{-109}	2^{-70}
128	9857	71	2^{-139}	2^{-98}
160	15101	87	2^{-171}	2^{-128}
256	32771	132	2^{-260}	2^{-214}

Figure 5.4 – Proportion of weak key and upper bound for the proportion of weak keys under the action of $(F_p)^2$ and ${\mathbb{F}_p}^*$ for the $(2p, p, \omega)$ QC-MDPC when $\omega_1 = \omega_2 = \frac{\omega}{2}$.

The results in Figure 5.4 are a strong motivation for computing the exact value of the proportion of weak keys under the action of the two groups, $(\mathbb{F}_p)^2$ and ${\mathbb{F}_p}^*$. Therefore the remaining part of this section as well as Section 5.7, Section 5.8 and Section 5.9 are dedicated to solving the aforementioned problem. In the rest of this section we give some general properties regarding the orbits of polynomials in $\mathbb{F}_2[x]/(x^p-1)$.

Proposition 5.6.3. *Let $(h_1, h_2) \in (\mathbb{F}_2[x]/(x^p-1))^2$. Then we have*

$$ {\mathbb{F}_p}^*.\left((\mathbb{F}_p)^2.(h_1,h_2)\right) = (\mathbb{F}_p)^2.({\mathbb{F}_p}^*.(h_1,h_2)). $$

Proof. Let $c(x) = \sum_{i=0}^{p-1} c_i x^i$ be a polynomial in $\mathbb{F}_2[x]/(x^p-1)$. Then by definition we have

$$\begin{aligned}
\sigma^*_\alpha(\sigma^+_\beta(c(x))) &= \sigma^*_\alpha\left(x^\beta \sum_{i=0}^{p-1} c_i x^i\right) \\
&= \sum_{i=0}^{p-1} c_i x^{(i+\beta)\alpha} \\
&= x^{\alpha\beta} \sum_{i=0}^{p-1} c_i x^{i\alpha} \\
&= \sigma^+_{\alpha\beta}(\sigma^*_\alpha(c(x))).
\end{aligned}$$

Use the later result in the case of $(h_1, h_2) \in (\mathbb{F}_2[x]/(x^p-1))^2$. By definition of the two group actions we have

$$\begin{aligned}\sigma^*_{(\alpha,\alpha)}(\sigma^+_{(\beta_1,\beta_2)}(h_1,h_2)) &= (\sigma^*_\alpha(\sigma_{\beta_1}(h_1)), \sigma^*_\alpha(\sigma_{\beta_2}(h_2))) \\ &= \left(\sigma^+_{\alpha\beta_1}(\sigma^*_\alpha(h_1)), \sigma^+_{\alpha\beta_2}(\sigma_\alpha(h_2))\right) \\ &= \sigma^+_{(\alpha\beta_1,\alpha\beta_2)}(\sigma^*_{(\alpha,\alpha)}(h_1,h_2))\end{aligned}$$

□

Since the later Proposition states that we obtain the same orbit regardless of the order in which we decided to act on the set of polynomials, we will begin by considering the action of the additive group. Hence in the next section we estimate the proportion of weak keys under the action of $(\mathbb{F}_p)^2$.

5.7 Weak orbits and Extended weak orbits

5.7.1 Orbits under the action of $(\mathbb{F}_p)^2$

Proposition 5.7.1. *Let $c(x)$ be a polynomial in $\mathbb{F}_2[x]/(x^p-1)$, then if $0 < \|c\| < p-1$ the cardinality of the orbit of h under the action of the additive group $(\mathbb{F}_p, +)$ equals*

$$|\mathbb{F}_p.\, c| = p.$$

Proof. First of all recall that for a prime number p the only subgroups of the additive group $(\mathbb{F}_p, +)$ are the trivial group $(\{0\}, +)$ or the whole group. Therefore if for a polynomial c the stabilizer subgroup of $(\mathbb{F}_p, +)$ with respect to c is reduced to the trivial group we obtain

$$|\mathbb{F}_p.\, c| = p.$$

Now notice that the only polynomials c for which the stabilizer subgroup is the hole group are either $c(x) = 0$ or $c(x) = x^{p-1} + x^{p-2} \cdots + x + 1$, which ends our proof. □

Straightforward we obtain

Corollary 5.7.2. *Let (h_1, h_2) be a pair of polynomials in $\mathbb{F}_2[x]/(x^p-1)^2$, then if $0 < \|h_i\| < p-1$ we have*

$$\left|(\mathbb{F}_p)^2.\,(h_1,h_2)\right| = p^2.$$

In particular

$$\left|\mathcal{P}_{\omega_1,\omega_2}/\,(\mathbb{F}_p)^2\right| = 1/p^2\,|\mathcal{P}_{\omega_1,\omega_2}| = 1/p^2\binom{p}{\omega_1}\binom{p}{\omega_2}$$

and

$$\frac{\left|\widetilde{\mathcal{W}}_{\omega_1,\omega_2}/\,(\mathbb{F}_p)^2\right|}{\left|\mathcal{P}_{\omega_1,\omega_2}/\,(\mathbb{F}_p)^2\right|} = \frac{\left|\widetilde{\mathcal{W}}_{\omega_1,\omega_2}\right|}{|\mathcal{P}_{\omega_1,\omega_2}|} \quad \textit{and} \quad \frac{\left|\left(\widetilde{\widetilde{\mathcal{W}}}_{\omega_1,\omega_2}/\,(\mathbb{F}_p)^2\right)/{\mathbb{F}_p}^*\right|}{\left|\left(\mathcal{P}_{\omega_1,\omega_2}/\,(\mathbb{F}_p)^2\right){\mathbb{F}_p}^*\right|} = \frac{\left|\widetilde{\widetilde{\mathcal{W}}}_{\omega_1,\omega_2}\right|}{|\mathcal{P}_{\omega_1,\omega_2}|}.$$

5.7.2 Proportion of weak orbits

Computing the cardinality of $\left|\widetilde{\mathcal{W}}_{\omega_1,\omega_2}/\left(\mathbb{F}_p\right)^2\right|$ turns out to be a challenging task. The technique that we use to achieve our goal requires a background in Combinatorics of Words, facts that we details in Section 5.8. We rather give here the results and show how the cyclic shifts increased the proportion of weak keys on practical parameters.

Theorem 5.7.3. *Let $p, omega$ and ω_1, ω_2 be the parameters of a $(2p, p, \omega)$ QC-MDPC scheme with $\omega_i^2/2p = c_i \log p + O(\sqrt{\frac{\log p}{p}})$, where c_1 and c_2 are constant such that $c_1 \geqslant c_2$. Then we have:*

$$\frac{\left|\widetilde{\mathcal{W}}_{\omega_1,\omega_2}/\left(\mathbb{F}_p\right)^2\right|}{\left|\mathcal{P}_{\omega_1,\omega_2}/\left(\mathbb{F}_p\right)^2\right|} \sim \frac{\binom{p-1}{\omega-2}}{1/p^2\binom{p}{\omega_1}\binom{p}{\omega_2}} \quad \text{when } p \to \infty, \tag{5.19}$$

with asymptotic equivalence

$$\frac{\left|\widetilde{\mathcal{W}}_{\omega_1,\omega_2}/\left(\mathbb{F}_p\right)^2\right|}{\left|\mathcal{P}_{\omega_1,\omega_2}/\left(\mathbb{F}_p\right)^2\right|} \sim \omega^2\sqrt{2\pi\alpha(1-\alpha)}p^{-2\sqrt{c_1c_2}}\omega^{\frac{1}{2}}2^{-\omega H(\alpha)}$$

where $\alpha = \dfrac{1}{1+\sqrt{\frac{c_2}{c_1}}}$.

The proof of this Theorem is given in details in Section 5.8.

Corollary 5.7.4. *Let p ω and ω_1, ω_2 be the parameters of a $(2p, p, \omega)$ QC-MDPC scheme with $\omega_i^2/2p = c_i \log p + O(\sqrt{\frac{\log p}{p}})$, where c_1 and c_2 are constant such that $c_1 \geqslant c_2$. Then using the action of $\left(\mathbb{F}_p\right)^2$ we have increased the proportion of weak keys by a factor equivalent to ω^2 when p goes to infinity, more exactly we have*

$$\left|\widetilde{\mathcal{W}}_{\omega_1,\omega_2}/\left(\mathbb{F}_p\right)^2\right| \sim \omega^2\left|\mathcal{W}_{\omega_1,\omega_2}/\left(\mathbb{F}_p\right)^2\right|$$

Numerical values. In Figure 5.5 we plot the numerical values obtained for the proportion of weak keys and weak orbits for a $(2p, p, \omega)$ QC-MDPC scheme. We notice that for all the proposed parameters the improvement coming from the group action of $\left(\mathbb{F}_p\right)^2$ has overpassed the announced security level. In other words we managed to increase the chances of finding a private key above the security level.

Security level	p	$\frac{\omega}{2}$	$\frac{\left\|\mathcal{W}_{\omega_1,\omega_2}\right\|}{\left\|\mathcal{P}_{\omega_1,\omega_2}\right\|}$ Corollary 5.4.12	$\frac{\left\|\widetilde{\mathcal{W}}_{\omega_1,\omega_2}\right\|}{\left\|\mathcal{P}_{\omega_1,\omega_2}\right\|}$ Theorem 5.7.3
80	4801	45	2^{-87}	2^{-74}
100	6851	56	2^{-109}	2^{-95}
128	9857	71	2^{-139}	2^{-124}
160	15101	87	2^{-171}	2^{-155}
256	32771	132	2^{-260}	2^{-244}

Figure 5.5 – Proportion of weak key and weak orbits for the smooth $(2p, p, \omega)$ QC-MDPC

5.7.3 Orbits under the action of $\mathbb{F}_p{}^*$

The action of the multiplicative group $\mathbb{F}_p{}^*$ is studied in detail in Section 5.9. Here we synthesize the main results concerning the orbits under the action of $\mathbb{F}_p{}^*$. The first difficulty is to determine the cardinality of an orbit $\mathbb{F}_p{}^*.\left((\mathbb{F}_p)^2.(h_1,h_2)\right)$ where $(h_1,h_2)\in\mathcal{P}_{\omega_1,\omega_2}$. As for the multiplicative group we begin by studying the action of $\mathbb{F}_p{}^*$ on a polynomial $c\in\mathbb{F}_2[x]/(x^p-1)$ and obtain:

Proposition (5.9.3). *Let $c\in\mathbb{F}_2[x]/(x^p-1)$ and Γ_c be the subgroup of $(\mathbb{F}_p{}^*,\cdot)$ which stabilizes $\mathbb{F}_p.c$. Then the cardinality of the orbit $\mathbb{F}_p{}^*.(\mathbb{F}_p.c)$ is*

$$|\mathbb{F}_p{}^*.(\mathbb{F}_p.c)|=\frac{(p-1)}{|\Gamma_c|}. \tag{5.20}$$

Since the polynomials we consider here satisfy a Hamming weight condition we prove that:

Proposition (5.9.9). *Let $\alpha\in\mathbb{F}_p{}^*$ and $c\in\mathbb{F}_2[x]/(x^p-1)$ so that $1<\|c\|<p$ and $\sigma_\alpha^*(\mathbb{F}_p.c)=\mathbb{F}_p.c$. Then the order of α in the multiplicative group $\mathbb{F}_p{}^*$ divides either $\|c\|$ or $\|c\|-1$.*

From Proposition 5.9.3 and 5.9.9 we deduce that:

Corollary (5.9.12). *Let $(h_1,h_2)\in\mathcal{P}_{\omega_1,\omega_2}$ with $\|h_i\|=\omega_i$ and denote by $\Gamma_{(h_1,h_2)}$ the subgroup of $\mathbb{F}_p{}^*$ that stabilizes $(\mathbb{F}_p)^2.(h_1,h_2)$. Then we have*

- *any $\alpha\in\Gamma_{(h_1,h_2)}$ is such that $\mathsf{ord}(\alpha)|\gcd(p-1,\gcd(l_1,l_2))$, where l_i runs through $\{\omega_i,\omega_i-1\}$, for $1\leqslant i\leqslant 2$.*
- *The cardinality of an orbit equals*

$$\left|\mathbb{F}_p{}^*.\left((\mathbb{F}_p)^2.(h_1,h_2)\right)\right|=\frac{(p-1)}{|\Gamma_{(h_1,h_2)}|}. \tag{5.21}$$

- *When (h_1,h_2) is a private key of a $(2p,p,\omega)$ QC-MDPC code with $\omega_i=O(\sqrt{p\log p})$ we have*

$$O\left(\sqrt{\frac{p}{\log p}}\right)\leqslant\left|\mathbb{F}_p{}^*.\left((\mathbb{F}_p)^2.(h_1,h_2)\right)\right|\leqslant p-1. \tag{5.22}$$

5.7.4 Proportion of extended weak orbits

Estimating the cardinality of the set $\widetilde{\widetilde{\mathcal{W}}}_{\omega_1,\omega_2}$ highly depends on the subgroups of the multiplicative group $\mathbb{F}_p{}^*$. Hence it seems difficult to obtain a general theorem that characterize the cardinality of the set of extended weak orbits for any prime number p. Nonetheless we obtain some results regarding this problem that we state bellow.

Proposition (5.9.13). *Let (h_1,h_2) be a private key of a $(2p,p,\omega)$ QC-MDPC code. If $(h_1,h_2)\in\widetilde{\mathcal{W}}_{\omega_1,\omega_2}$ then $\sigma_{-1}^*\left((\mathbb{F}_p)^2.(h_1,h_2)\right)$ is a weak orbit.*

Moreover if $-1\in\Gamma_{(h_1,h_2)}$ then $|\mathbb{F}_p{}^*.((\mathbb{F}_p)^2.(h_1,h_2))|\leqslant\frac{p-1}{2}$.

In general for a smooth QC-MDPC code we obtain

Corollary (5.9.14). *Let p and ω be the parameters of a smooth $(2p, p, \omega)$ QC-MDPC code. Then we have*

$$\frac{p-1}{\omega/2}\frac{\left|\widetilde{\mathcal{W}}_{\omega_1,\omega_2}\right|}{|\mathcal{P}_{\omega_1,\omega_2}|} \leqslant \frac{\left|\widetilde{\widetilde{\mathcal{W}}}_{\omega_1,\omega_2}\right|}{|\mathcal{P}_{\omega_1,\omega_2}|} \leqslant (p-1)\frac{\left|\widetilde{\mathcal{W}}_{\omega_1,\omega_2}\right|}{|\mathcal{P}_{\omega_1,\omega_2}|}. \quad (5.51)$$

Numerical values. We plot in Figure 5.6 the numerical values obtained for the proportion of weak keys, weak orbits and extended weak orbits for a $(2p, p, \omega)$ QC-MDPC scheme.

Security level	p	$\frac{\omega}{2}$	$\frac{\left\|\mathcal{W}_{\omega_1,\omega_2}\right\|}{\left\|\mathcal{P}_{\omega_1,\omega_2}\right\|}$ Corollary 5.4.12	$\frac{\left\|\widetilde{\mathcal{W}}_{\omega_1,\omega_2}\right\|}{\left\|\mathcal{P}_{\omega_1,\omega_2}\right\|}$ Theorem 5.7.3	$\frac{\left\|\widetilde{\widetilde{\mathcal{W}}}_{\omega_1,\omega_2}\right\|}{\left\|\mathcal{P}_{\omega_1,\omega_2}\right\|}$ Lower bound (5.51)	Upper bound (5.51)
80	4801	45	2^{-87}	2^{-74}	2^{-67}	2^{-62}
100	6851	56	2^{-109}	2^{-95}	2^{-88}	2^{-82}
128	9857	71	2^{-139}	2^{-124}	2^{-117}	2^{-111}
160	15101	87	2^{-171}	2^{-155}	2^{-148}	2^{-141}
256	32771	132	2^{-260}	2^{-244}	2^{-236}	2^{-229}

Figure 5.6 – Proportion of weak key, weak orbits and extended weak orbits for the smooth $(2p, p, \omega)$ QC-MDPC

We notice that the proportion of extended weak orbits are big enough to raise the question whether the designer of this scheme can assume this risk. Therefore we propose a solution to avoid this type of vulnerabilities, solution that we detail in Section 5.11.4 and that consists in a secure Key Generation algorithm (see Figure 2).

5.8 Computing the proportion of weak orbits

5.8.1 Redefining the problem

The problem we want to solve here, is to estimate the cardinality of
$\widetilde{\mathcal{W}}_{\omega_1,\omega_2} = \{(h_1, h_2) \in \mathcal{P}_{\omega_1,\omega_2} \mid (h_1, h_2) \text{ satisfy } (5.14)\}$, where Equation (5.14) states

$$\exists \alpha_1, \alpha_2 \in \mathbb{F}_p^2 \text{ such that } \deg(x^{\alpha_1} h_1) + \deg(x^{\alpha_2} h_2) < p.$$

In order to achieve our goal we will introduce the concept of longest run and redefine the problem in an equivalent manner.

Lemma 5.8.1. *Let $\overline{h_i} = \min(\mathbb{F}_p . h_i)$ be the minimum polynomial for the lexicographical order of $h_i \in \mathbb{F}_2[x]/(x^p - 1)$. Then the set $(\mathbb{F}_p)^2 . (h_1, h_2)$ is a weak orbit if and only if $(\overline{h_1}) + \deg(\overline{h_2}) < p$.*

Proof. By definition we have that $\overline{h_i} = \min(\mathbb{F}_p . h_i)$ implies that $\deg(\overline{h_i}) < x^\alpha \overline{h_i}$ for any $1 \leqslant \alpha < p$. Since a weak orbit has to satisfy equation (5.14) we obtain the wanted result. □

Definition 5.8.2. *We define the longest run of zeros of a polynomial c in $\mathbb{F}_2[x]/(x^p - 1)$ by the longest sequence of consecutive zero coefficients of c.*

Remark 5.8.3. *We remark that there is a relation connecting the degree of the minimum polynomial and the longest run of zeros. If k_i denotes the longest run of zeros of $h_i \in \mathbb{F}_2[x]/(x^p-1)$ we have that* $\deg(\overline{h_i}) = p - k_i - 1$.

Example 5.8.4. *Let $p = 7$ and consider the polynomial $c(x) = 1 + x + x^5$. Then the orbit equals in this case*

$$\mathbb{F}_p.\, c = \{1+x+x^5, x+x^2+x^6, 1+x^2+x^3, x+x^3+x^4, x^2+x^4+x^5, x^3+x^5+x^6, 1+x^4+x^6\}.$$

The smallest polynomial in lexicographic order is $\bar{c} = 1 + x^2 + x^3$. We also notice that the longest run of zeros equals $k = 3$ and $\deg(\bar{c}) = p - k - 1 = 3$.

Since we have the relation between the degree and the longest run of zeros for the minimal polynomial in the equivalence class we can redefine a weak orbit in terms of longest run:

Proposition 5.8.5. *The set $(\mathbb{F}_p)^2 . (h_1, h_2)$ defined by a private key $(h_1, h_2) \in \mathcal{P}_{\omega_1,\omega_2}$ of a $(2p, p, \omega)$ QC-MDPC code is a* weak orbit *if and only if it satisfies the equation:*

$$k_1 + k_2 \geqslant p - 1. \tag{5.23}$$

Proof. Use Lemma 5.8.1 and Remark 5.8.3. □

At this point we have reduced our key recovery attack to another problem. To count all pairs (h_1, h_2) with the restriction mentioned above, we have to answer the following question: *What is the distribution of the longest run of zeros for the equivalence class of all cyclic shifts of a $\mathbb{F}_2^p$ vector with fixed Hamming weight?* We will answer this question using the concept of Lyndon words, that we explain in the next paragraph.

5.8.2 Lyndon words

Before we define the notion of Lyndon words we recall some basic definitions from the field of Combinatorics of Words.

Definition 5.8.6.

- *A finite alphabet $\mathcal{A}_k = \{a_0, a_1, \ldots, a_k\}$ is a finite set of symbols that we call* letters.
- *Any finite sequence of letters belonging to the same alphabet is called a finite* word.
- *The free monoid generated by the alphabet $\mathcal{A}_k$,usually denoted by $\mathcal{A}_k^*$, is defined as the monoid whose elements are all the finite sequences of zero or more elements from $\mathcal{A}_k$, with string concatenation as the monoid operation. We will denote the empty word by ε. This statement can be written in terms of formal series as*

$$\sum_{w \in \mathcal{A}_k^\star} w = \frac{1}{1 - \sum\limits_{i=0}^{k} a_i}. \tag{5.24}$$

Example 5.8.7. *Let $\mathcal{B} = \{0, 1\}$ denote the binary alphabet. Then the free monoid on $\mathcal{B}$ is*

$$\mathcal{B}^* = \varepsilon + 0 + 1 + 00 + 01 + 10 + 11 + 000 + \dots .$$

The same set can be generated by means of equation (5.24)

$$\begin{aligned}\mathcal{B}^* &= (0+1)^0 + (0+1)^1 + (0+1)^2 + (0+1)^3 + \dots \\ &= \varepsilon + 0 + 1 + 00 + 01 + 10 + 11 + 000 + \dots\end{aligned}$$

Definition 5.8.8. *The set of all cyclic shifts of a word is called its conjugacy class. The conjugacy class of a word can be represented by a necklace, also known as circular word.*

There are two types of necklaces, namely periodic and aperiodic (or primitive). Take for example the word $w = 0101$ which is periodic since $w = (01)^2$ and $w = 001$ which is aperiodic.

Definition 5.8.9. *[Lot02]A Lyndon word l is a word satisfying the conditions:*

- *l is a primitive word (i.e. it cannot be written $l = uv$, where u and v commute and $u, v \neq \varepsilon$)*
- *l is the lexicographically smallest word in its conjugacy class*

Example 5.8.10.

1. *Let $\mathbb{F}_p$. $00011 = \{00011, 00110, 01100, 11000, 10001\}$. The Lyndon word here is 00011 since it is the strictly smallest than all the cyclic shifts. and it is primitive.*
2. *Let $\mathbb{F}_p$. $0101 = \{0101, 1010, 0101, 1010\}$. There is no Lyndon word here, since the smallest element in the conjugacy class is 0101 but it is not primitive since $0101 = (01)^2$.*

Lyndon words are named after the mathematician Roger Lyndon, who introduced them in 1954 under the name of standard lexicographic sequences. Since a Lyndon word is the only word which is lexicographically smaller than all its the cyclic shifts it implies that a Lyndon word is different from all of its non-trivial shifts, and is therefore primitive. So we have an equivalent way of defining a Lyndon word, that is

Remark 5.8.11. *We notice that the existence of a Lyndon word is closely related to the size of the conjugacy class. Indeed if the set of all cyclic shifts of p-length word w, has exactly p different elements, in other words the size of the orbit equals p, then we have the existence of a Lyndon word.*

When p is prime we obtain the result in Proposition 5.7.1, namely for any word w with Hamming weight different from zero or p the number of elements in the conjugacy class of w equals p.

Lyndon words have many applications in algebra and combinatorics. For instance in number theory we know that the set of monic irreducible polynomials of degree n over a field of characteristic p is in bijection with the set of Lyndon words of length n over an alphabet of size p (see [Lot02] for more details). We also have an application of Lyndon words in cryptography [Per05].

Since our goal here is to use the Lyndon words to count weak orbits for the QC-MDPC scheme, we will explain the basic enumeration methods related to Lyndon words and recall the existing results in this area.

5.8.3 Longest run of Lyndon words with fixed weight

The longest run problem was previously studied from a probabilistic point of view by Feller [Fel68] and Schilling, Gordon and Waterman [GSW86]. The problem studied here is slightly different from our in the sense that the authors give the distribution of the longest run of binary words with fixed weight and not binary words under the cyclic shift action. Their approach is to consider an p trial of Bernoulli variables and search for the probability of the maximum longest run of "heads" or "coins". Results are given for a fixed length p as well as in the asymptotic case. Same results for the latter case were obtained using analytical combinatorics by Flajolet [FS09] and Wilson and Permantle [PW08].

Now if we inspect the case of Lyondon words, many results are known about the number of Lyndon words with different restrictions. We recall the results on the number of Lyndon words over a q-ary alphabet with length p given by Witt [Wit13] or the number of Lyndon words over the q-ary alphabet with fixed weight given by Gilbert in [GR61]. The first article analyzing the longest run of Lyndon word is by Bassino, Clément and Nicaudin [BCN05]. Nonetheless the authors do not take into consideration the longest run of Lyndon words with fixed weight. In a sense our problem is the both an extension of Gilbert's result and Bassino, Clément and Nicaudin's result.

Counting techniques related to Lyndon words. In order to achieve our goal we need to recall the main properties and techniques used for enumerating problems related to Lyndon words. For that we will recall the result in [Wit13] with a full proof. For that we begin by a well known theorem on the factorization of words into product of Lyndon words

Theorem 5.8.12. *[KTC58, Lot02] Any word $w \in \mathcal{A}_k^*$ may be uniquely written as a unique non-increasing product of Lyndon words, i.e.,*

$$w = l_1 l_2 \dots l_n$$

where the l_i are Lyndon words such that $l_n \preceq l_{n-1} \preceq \cdots \preceq l_1$.

Example 5.8.13. *Let $w = 01101001$, then the factorization into product of Lyndon words is $w = (011)(01)(001)$.*

There is an efficient algorithm that computes the factorization into Lyndon words due to Duval [Duv83], algorithm that has a linear complexity in the length of the word to be factorized.

Corollary 5.8.14. *Let $L(\mathcal{A}_k)$ denote the set of Lyndon words over the alphabet $\mathcal{A}_k$. Then we have*

$$\sum_{w \in \mathcal{A}_k^*} w = \prod_{l \in L(\mathcal{A}_k)}^{\searrow} \frac{1}{1-l}, \tag{5.25}$$

where the $\searrow$ denotes the fact that the elements $l \in L(\mathcal{A}_k)$ are considered in decreasing lexicographical order.

Example 5.8.15. *Let $\mathcal{B} = \{0, 1\}$ be the binary alphabet totally ordered by $0 < 1$. We will generate all the binary words in $\mathcal{B}^*$ of length at most 3, denoted by $B_{\leqslant 3}$, ordered by length*

$$B_{\leqslant 3} = \{\varepsilon, 0, 1, 00, 01, 10, 11, 000, 001, 010, 011, 100, 101, 110, 111\}.$$

Now let us consider the set of Lyndon words of length less than or equal to 3*, in decreasing lexicographical order*

$$L(\mathcal{B}, \leqslant 3) = \{1, 011, 01, 001, 0\}$$

and compute the product in Equation 5.25

$$\begin{aligned}\prod_{l \in L(\mathcal{B}, \leqslant 3)}^{\searrow} \frac{1}{1-l} &= (\varepsilon + 1 + 11 + 111)(\varepsilon + 011)(\varepsilon + 01)(\varepsilon + 001)(\varepsilon + 0 + 00 + 000) \\ &= \varepsilon + 0 + 00 + 000 + 001 + 01 + 010 + 011 + 111 \\ &+ 11 + 110 + 1 + 10 + 100 + 101 + \dots\end{aligned}$$

Now if we extract from the result of this product only the elements with a length less that or equal to 3 *we obtain the set* $B_{\leqslant 3}$.

The number of Lyndon words over the binary alphabet is a well known sequence, that can be found in *The On-Line Encyclopedia of Integer Sequences*

$$1, 2, 1, 2, 3, 6, 9, 18, 30, 56, 99, 186, 335, \dots \text{ (sequence A001037 in the Oeis)}$$

The last technique needed in enumeration problems are the generating functions. We do not recall the definitions regarding the concepts that are involved in this area, we will only explain how to apply these techniques. For a detailed reference in this field we direct the reader to the book of Flajolet and Sedgewick [FS09], which turned out to be highly useful.

The procedure that we follow to obtain the generating function is composed of three steps:

- the first step is to formally represent each letter of the alphabet $\mathcal{A}_k$ by a formal variable z that represents the length of the letter $\forall 0 \leqslant i \leqslant k, a_i \to z$
- since any finite word is a finite sequence of letters we associate to any word w in $\mathcal{A}_k^*$ the formal variable $z^{|w|}$, representing the length of the word w. Here $|w|$ denotes the length of the word w.
- the univariate generating function

  $$\Phi(z) = \sum_{w \in \mathcal{A}_k^*} z^{|w|}$$

  is so that the coefficient of $[z^n]\,\Phi(z)$ represents the number of works of length n over the alphabet $\mathcal{A}_k$.

Proposition 5.8.16. *Let* $L(\mathcal{A}_k, n)$ *denote the set of Lyndon words of length* n *over* $\mathcal{A}_k$*. Then we have*

$$|L(\mathcal{A}_k, n)| = \frac{1}{n} \sum_{d|n} \mu(d)(k+1)^{\frac{n}{d}}, \tag{5.26}$$

where μ *is the Möbius function, defined by* $\mu(j) = 0$ *if* j *has a squared prime factor,* $\mu(j) = 1$ *if* j *is square-free with an even number of prime factors and* $\mu(j) = -1$ *otherwise.*

Proof. From Equation (5.24) the generating function for the words in $\mathcal{A}_k^*$ equals $\Phi(z) = \dfrac{1}{1-\sum\limits_{i=0}^{k} z} = \dfrac{1}{1-(k+1)z}$.

Based on Equation (5.25) and the cyclotomic identity from [Wit13] we have that the same generating function equals $\Phi(z) = \prod\limits_{i=1}^{\infty} \left(\dfrac{1}{1-z^i}\right)^{|L(\mathcal{A}_k,i)|}$

So we deduce

$$\frac{1}{1-(k+1)z} = \prod_{i\geqslant 1} \left(\frac{1}{1-z^i}\right)^{|L(\mathcal{A}_k,i)|}$$

Taking the logarithm we obtain

$$\log(1-(k+1)z) = \sum_{i\geqslant 1}^{\infty} |L(\mathcal{A}_k,i)| \log(1-z^i)$$

We develop the later equation using the Taylor-Young formula

$$\sum_{n\geqslant 1} \frac{((k+1)z)^n}{n} = \sum_{i\geqslant 1}^{\infty} |L(\mathcal{A}_k,i)| \sum_{d\geqslant 1} \frac{z^{id}}{d}.$$

We extract the coefficient of z^n on both sides of the equation and obtain

$$\frac{(k+1)^n}{n} = \sum_{d|n} \frac{\left|L(\mathcal{A}_k, \frac{n}{d})\right|}{d}.$$

The last step is to use the Möbius inversion formula [Möb32, Lan09] and obtain the wanted result. □

Remark 5.8.17. *A particular case of this formula is for the binary alphabet $\mathcal{B}$, case for which we obtain $|L(\mathcal{B},n)| = \frac{1}{n}\sum\limits_{d|n} \mu(d) 2^{\frac{n}{d}}$. Remember that in our case we have $n = p$ is a prime number, case for which we have $|L(\mathcal{B},p)| = \frac{1}{p}(2^p - 2)$ (which is a positive integer, fact that can be proved using the binomial theorem).*

Based on the later technique we will prove the following theorem:

Theorem 5.8.18. *Let p, k, ω be integers, such that $1 \leqslant \omega \leqslant p$ and $k \leqslant p - \omega$. The number of binary Lyndon words with length p, longest run less than or equal to k and weight equal to ω is:*

$$\left|L^{\leqslant k}(\mathcal{B},p,\omega)\right| = \frac{1}{\omega} \sum_{j\in\mathbb{N}^*,\, j|gcd(p,\omega)} \mu(j) \binom{\frac{\omega}{j}}{\frac{p}{j} - \frac{\omega}{j}}_k, \tag{5.27}$$

where $\binom{j}{i}_k$ is known as the standard multinomial coefficient and is defined as the coefficient of x^i in $\left(1 + x + \cdots + x^k\right)^j$.

Proof. In the first place we define a well known morphism between $\mathcal{A}_k^*$ and $\mathcal{B}^*$. The monoids $\mathcal{A}_k^*$ and $\mathcal{B}^*$ are endowed with the lexicographic orders satisfying $0 < 1$ and $a_k < \cdots < a_0$. The morphism

$$\begin{aligned} \varphi : \mathcal{A}_k^* &\to (0^*1)^* \subset \mathcal{B}^* \\ a_i &\to 0^i 1 \end{aligned}$$

is an order preserving isomorphism (see [Ric03] for details). We deduce that $w \in \mathcal{A}_k^*$ is a Lyndon word if and only if $\varphi(w)$ is a Lyndon word. The purpose of the application φ is that it allows us to compute the cardinality of the Lyndon words over the binary alphabet by handling only the set of Lyndon words over the alphabet $\mathcal{A}_k$, which is much easier to compute.

The next step is to deduce the generating function, which in our case is a bivariate function.

- Each word in $\mathcal{A}_k^*$ can be written as follows:

$$w = a_{l_0} a_{l_2} \dots a_{l_{j-1}} \quad where \quad \forall m \in \{0, 1, \dots j-1\} \quad a_{l_m} \in \mathcal{A}_k \text{ and } j \geqslant 0$$

- Each *letter* of the alphabet $\mathcal{A}_k$ is formally represented by the translation rule:

$$a_{l_m} \to z x^{l_m+1}$$

  where the formal variable z represents the length of the *letter* and the formal variable x represents the sum $1 + l_m$.

- We have the following translation rule for any *word* generated by the specified alphabet:

$$w \to z^{|w|} x^{\psi(w)} \quad where \quad \psi(w) = |w| + \sum_{m=0}^{|w|-1} l_m$$

- The bivariate generating function

$$\Phi(z, x) = \sum_{w \in \mathcal{A}^*} z^{|w|} x^{\psi(w)}$$

  is so that the coefficient of $z^\omega x^p$ in $\Phi(z, x)$ represents the number of words of length ω and ψ-parameter equal to p.

We notice that the we have for any finite word $w \in \mathcal{A}_k^*$

$$\psi(w) = |\varphi(w)|.$$

In other words $\psi(w)$ is the equivalent measure for the length of the binary word $\varphi(w)$ and $|w|$ is the equivalent measure for the Hamming weight of the binary word $\varphi(w)$.

If we set $L_\psi(\mathcal{A}_k, \omega, p) = \left\{ l \in L(\mathcal{A}_k) \;\middle|\; |l| = \omega \text{ and } \psi(l) = p \right\}$ then

$$\varphi\left(L_\psi(\mathcal{A}_k, \omega, p)\right) = L^{\leqslant k}(\mathcal{B}, p, \omega).$$

Hence, it suffices to compute $|L_\psi(\mathcal{A}_k, \omega, p)|$, which can be done using the same technique as in Proposition 5.8.16.

Using Equation (5.25) and (5.24) we obtain the equality

$$\Phi(z,x) = \frac{1}{1 - z\sum_{i=1}^{k+1} x^i} = \prod_{1\leqslant j\leqslant i}^{\infty} \left(\frac{1}{1-x^i z^j}\right)^{|L_\psi(\mathcal{A}_k,j,i)|}. \tag{5.28}$$

We apply the logarithm in each side of the equality above and develop using the Taylor expansion. In the resulting formula we compare the coefficient of $z^\omega x^p$ in the left hand side and the right hand side and obtain

$$\sum_{\substack{j|\omega \\ \frac{\omega}{j}|p}} j\left|L_\psi(\mathcal{A}_k, j, \frac{p}{\omega}j)\right| = \binom{\omega}{p-\omega}_k, \tag{5.29}$$

where $\binom{\omega}{p}_k = \sum_{\substack{l_1+2l_2+\ldots kl_k=p \\ l_0+l_1+\ldots l_k=\omega}} \frac{\omega!}{l_0!l_1!\ldots l_k!}$ denotes the coefficient of x^p in $(1+x+x^2+\cdots+x^k)^\omega$.

We rewrite the last equation as

$$\sum_{\substack{j|\omega \\ j|p}} \frac{\omega}{j}\left|L_\psi(\mathcal{A}_k, \frac{\omega}{j}, \frac{p}{j})\right| = \binom{\omega}{p-\omega}_k, \tag{5.30}$$

and finally apply the Möbius Inversion □

Since from now on we deal only with binary words we will simplify the notations and drop the name of the alphabet from the $L(\mathcal{B}, p)$. So starting from this point we denote the set of Lyndon words over the binary alphabet of length p by $L(p)$.

Corollary 5.8.19. *The number of Lyndon words of length p and Hamming weight equal to ω over the binary alphabet (result already found in [GR61] by Gilbert and Riordan) is:*

$$|L(p,\omega)| = \frac{1}{p}\sum_{j|\gcd(p,\omega)} \mu(j)\binom{\frac{p}{j}}{\frac{\omega}{j}}. \tag{5.31}$$

Proof. We notice that the for fixed p and ω the longest run satisfies $k \leqslant p-\omega$. Hence $L^{\leqslant p-\omega}(p,\omega) = L(p,\omega)$ and using the later theorem we have

$$|L(p,\omega)| = \frac{1}{\omega}\sum_{j\in\mathbb{N}^*, j|\gcd(p,\omega)} \mu(j)\binom{\frac{\omega}{j}}{\frac{p}{j}-\frac{\omega}{j}}_{p-\omega} \tag{5.32}$$

From [BBK08] we have that

$$\forall k \geqslant p, \binom{\omega}{p}_k = \binom{p+\omega-1}{p} \tag{5.33}$$

Combining the last two equations we obtain the wanted result. □

Corollary 5.8.20. *When p is prime,*

$$\left|L^{\leqslant k}(p,\omega)\right| = \frac{1}{\omega}\binom{\omega}{p-\omega}_k, \tag{5.34}$$

$$|L(p,\omega)| = \frac{1}{p}\binom{p}{\omega}. \tag{5.35}$$

Remark 5.8.21. *Since we deal only with p a prime number we have that $L(p) = \bigcup_{\omega=1}^{p-1} L(p,\omega)$, fact that we verify by computing the size of each set:*

$$\begin{aligned}
|L(p)| &= \frac{1}{p}(2^p-2)\\
&= \frac{1}{p}\left(\sum_{\omega=0}^{p}\binom{p}{\omega} - \binom{p}{0} - \binom{p}{p}\right)\\
&= \frac{1}{p}\left(\sum_{\omega=1}^{p-1}\binom{p}{\omega}\right)\\
&= \left|\bigcup_{\omega=1}^{p-1} L(p,\omega)\right|.
\end{aligned}$$

ω	l	k	$L^k(7,\omega)$	$\lvert L(7,\omega)\rvert$	$\lvert L(7)\rvert$
1	0000001	6	1	1	
	0000011	5	1		
2	0000101	4	1	3	
	0001001	3	1		
	0000111	4	1		
	0001101	3	2		
3	0001011	3		5	
	0010011	2	2		
	0010101	2			
	0001111	3	1		18
	0011101	2			
4	0011011	2	3	5	
	0010111	2			
	0101011	1	1		
	0011111	2	1		
5	0101111	1	2	3	
	0110111	1			
6	0111111	1	1	1	

Figure 5.7 – The Lyndon words for $p = 7$. The result from [Wit13] -$|L(7)|$, the result from [GR61] -$|L(7,\omega)|$ and our result Theorem 5.8.18 - $\left|L^k(7,\omega)\right|$

5.8.4 Probability of weak orbits

Recall that our initial problem was to determine the proportion $\left|\widetilde{\mathcal{W}}_{\omega_1,\omega_2}/\left(\mathbb{F}_p\right)^2\right| / \left|\mathcal{P}_{\omega_1,\omega_2}/\left(\mathbb{F}_p\right)^2\right|$. Since we have associated to any polynomial c in $\mathbb{F}_2[x]/(x^p-1)$ with fixed hamming weight

the corresponding Lyndon word in $L(p, \|c\|)$ we obtain

$$\frac{\left|\widetilde{\mathcal{W}}_{\omega_1,\omega_2}/\left(\mathbb{F}_p\right)^2\right|}{\left|\mathcal{P}_{\omega_1,\omega_2}/\left(\mathbb{F}_p\right)^2\right|} = \frac{\left|\bigcup_{k_1+k_2\geqslant p-1} L^{k_1}(p,\omega_1)L^{k_2}(p,\omega_2)\right|}{|L(p,\omega_1)L(p,\omega_2)|}.$$

In order to simplify the proofs, in the next part we define a discrete random variable that represents the longest run of Lyndon words.

Definition 5.8.22. *Let $L(p,\omega)$ be the probability space for our model and consider that each Lyndon word in $L(p,\omega)$ has the same probability to be chosen, namely $p/\binom{p}{\omega}$.*

Now, let $X_{p,\omega}$ be a discrete random variable that represents the longest run of zeros of Lyndon words with length p and weight ω, which takes values in the set $\{\lfloor\frac{p-1}{\omega}\rfloor,\ldots,p-\omega\}$. We define the cumulative distribution and mass function for $X_{p,\omega}$, for any $\lfloor\frac{p-1}{\omega}\rfloor \leqslant k \leqslant p-\omega$

$$F_{X_{p,\omega}}(k) = \frac{\left|L^{\leqslant k}(p,\omega)\right|}{|L(p,\omega)|} \quad \textit{and} \quad f_{X_{p,\omega}}(k) = \frac{\left|L^{k}(p,\omega)\right|}{|L(p,\omega)|}.$$

In other words for our probability model we write the set of binary Lyndon words of length p and weight ω as $L(p,\omega) = \bigcup_{k=\lfloor\frac{p-1}{\omega}\rfloor}^{p-\omega} L^k(p,\omega)$. Since for the QC-MDPC code we study pairs of Lyndon words with a certain condition (see Equation (5.23)), we define

Definition 5.8.23. *Let $Y_{p,\omega_1,\omega_2} \overset{def}{=} X_{p,\omega_1} + X_{p,\omega_2}$ be a discrete random variable that represents the sum of two independent random variables X_{p,ω_1} and X_{p,ω_2}.*

Remark 5.8.24. *The initial problem, that is estimating the proportion of weak orbits, becomes now a probability problem, namely*

$$\frac{\left|\widetilde{\mathcal{W}}_{\omega_1,\omega_2}/\left(\mathbb{F}_p\right)^2\right|}{\left|\mathcal{P}_{\omega_1,\omega_2}/\left(\mathbb{F}_p\right)^2\right|} = P\left(Y_{p,\omega_1,\omega_2} \geqslant p-1\right) = \sum_{k_1+k_2\geqslant p-1} f_{X_{p,\omega_1}}(k_1) f_{X_{p,\omega_2}}(k_2).$$

As p is prime, using Corollary 5.8.20 and Definition 5.8.22 we get the exact value:

$$P\left(Y_{p,\omega_1,\omega_2} \geqslant p-1\right) = \sum_{k_1+k_2\geqslant p-1} \frac{\binom{\omega_1}{p-\omega_1}_{k_1} - \binom{\omega_1}{p-\omega_1}_{k_1-1}}{\binom{p-1}{\omega_1-1}} \frac{\binom{\omega_2}{p-\omega_2}_{k_2} - \binom{\omega_2}{p-\omega_2}_{k_2-1}}{\binom{p-1}{\omega_2-1}} \tag{5.36}$$

The first case that seems interesting is when each variable has a longest run greater than or equal to half of the wanted quantity $\frac{p-1}{2}$.

Proposition 5.8.25. *Let ω_1 and $\omega_2 \geqslant 2$, then we have:*

$$P\left(X_{p,\omega_1} \geqslant \frac{p-1}{2}\right) P\left(X_{p,\omega_2} \geqslant \frac{p-1}{2}\right) = \omega_1\omega_2 \times \frac{\binom{\frac{p-1}{2}}{\omega_1-1}\binom{\frac{p-1}{2}}{\omega_2-1}}{\binom{p-1}{\omega_1-1}\binom{p-1}{\omega_2-1}}, \tag{5.37}$$

with asymptotic equivalence

$$\omega_1\omega_2 2^{-\omega} \times \begin{cases} p^{-\frac{c_2}{2}} & \textit{if } \frac{\omega_2^2}{p} = c_2\log p + O(\sqrt{\frac{\log p}{p}}) \quad \textit{and } \omega_1 = O(1), \\ e^{-\frac{c_1+c_2}{2}} & \textit{if } \frac{\omega_i^2}{p} = c_i + O(\frac{1}{\sqrt{p}}), \\ e^{-\frac{c_1}{2}}p^{-\frac{c_2}{2}} & \textit{if } \frac{\omega_1^2}{p} = c_1 + O(\frac{1}{\sqrt{p}}) \quad \textit{and } \frac{\omega_2^2}{p} = c_2\log p + O(\sqrt{\frac{\log p}{p}}), \\ p^{-\frac{c_1+c_2}{2}} & \textit{if } \frac{\omega_i^2}{p} = c_i\log p + O(\sqrt{\frac{\log p}{p}}). \end{cases}$$

Proof. Apply the formula for the standard multinomial coefficient from [Lot02, BBK08]:

$$\binom{\omega}{p-\omega}_k = \sum_{j=0}^{\lfloor \frac{p-\omega}{k+1} \rfloor} (-1)^j \binom{\omega}{j} \binom{p-j(k+1)-1}{\omega-1}.$$

For asymptotic expansion as before use the Stirling approximation for factorials. □

Remark 5.8.26. *Using the results from Proposition 5.8.25 and 5.4.10 when* $\omega_1 = \omega_2 = \frac{\omega}{2}$ *we have that*

$$P\left(X_{p,\omega_1} \geqslant \frac{p-1}{2}\right) P\left(X_{p,\omega_2} \geqslant \frac{p-1}{2}\right) \sim \omega^{\frac{3}{2}} \frac{|\mathcal{W}_{\omega_1,\omega_2}|}{|\mathcal{P}_{\omega_1,\omega_2}|}.$$

We step forward and analyze the probability for a weak orbit in the general case.

Remark 5.8.27. *We notice that if* ω_1 *or* ω_2 *equals 1, or* $\omega_1 = \omega_2 = 2$ *then the probability of a weak orbit equals*

$$P\left(Y_{p,\omega_1,\omega_2} \geqslant p-1\right) = 1.$$

But the interesting analysis for the QC-MDPC scheme, is when ω_1 and ω_2 are relatively close and $\omega = O\left(\sqrt{p \log p}\right)$.

Proposition 5.8.28. *Let* $\omega_1 \geqslant \omega_2$ *satisfy the relation* $\lim_{p \to +\infty} e^{\frac{\omega_2^2}{p} - \log \omega_1} = +\infty$. *Then we have:*

$$P(Y_{p,\omega_1,\omega_2} \geqslant p-1) \sim \omega_1 \omega_2 \frac{\binom{p-1}{\omega-2}}{\binom{p-1}{\omega_1-1}\binom{p-1}{\omega_2-1}} \quad \text{when } p \to \infty, \tag{5.38}$$

Moreover for $\frac{\omega_i^2}{2p} = c_i \log p + O(\sqrt{\frac{\log p}{p}})$, *we have*

$$P(Y_{p,\omega_1,\omega_2} \geqslant p-1) \sim \omega^2 \sqrt{2\pi\alpha(1-\alpha)} p^{-2\sqrt{c_1 c_2}} \omega^{\frac{1}{2}} 2^{-\omega H(\alpha)}$$

where $\alpha = \dfrac{1}{1+\sqrt{\frac{c_2}{c_1}}}$.

Proof. By definition we have:

$$P\left(Y_{p,\omega_1,\omega_2} \geqslant p-1\right) = \sum_{\omega_2 - 1 \leqslant k \leqslant p-\omega_1} f_{X_{p,\omega_1}}(k) \left(1 - F_{X_{p,\omega_2}}(p-k-1-1)\right).$$

In order to estimate the probability $P(Y_{p,\omega_1,\omega_2} \geqslant p-1)$ we begin by some properties on the distribution function of the random variable $X_{p,\omega}$.

Lemma 5.8.29. *Let* $\omega \geqslant 2$ *and* p *prime. Then for* $k > \lfloor \frac{p-\omega}{2} \rfloor$ *we have*

$$f_{X_{p,\omega}}(k) = \frac{\omega\binom{p-k-2}{\omega-2}}{\binom{p-1}{\omega-1}}, \quad F_{X_{p,\omega}}(k-1) = 1 - \frac{\omega\binom{p-k-1}{\omega-1}}{\binom{p-1}{\omega-1}}. \tag{5.39}$$

For $k \leqslant \lfloor \frac{p-\omega}{2} \rfloor$ the bounds are

$$\frac{\omega\binom{p-k-2}{\omega-2} - \binom{\omega}{2}\left[\binom{p-2k-1}{\omega-1} - \binom{p-2k-3}{\omega-1}\right]}{\binom{p-1}{\omega-1}} \leqslant f_{X_{p,\omega}}(k) \leqslant \frac{\omega\binom{p-k-2}{\omega-2}}{\binom{p-1}{\omega-1}}, \tag{5.40}$$

$$\frac{\omega\binom{p-k-1}{\omega-1} - \binom{\omega}{2}\binom{p-2k-1}{\omega-1}}{\binom{p-1}{\omega-1}} \leqslant 1 - F_{X_{p,\omega}}(k-1) \leqslant \frac{\omega\binom{p-k-1}{\omega-1}}{\binom{p-1}{\omega-1}}. \tag{5.41}$$

For the upper bound, this gives

$$P\left(Y_{p,\omega_1,\omega_2} \geqslant p-1\right) \leqslant \sum_{k=\omega_2-1}^{p-\omega_1} \omega_1 \frac{\binom{p-k-2}{\omega_1-2}}{\binom{p-1}{\omega_1-1}} \omega_2 \frac{\binom{k}{\omega_2-1}}{\binom{p-1}{\omega_2-1}} = \frac{\omega_1\omega_2\binom{p-1}{\omega_1+\omega_2-2}}{\binom{p-1}{\omega_1-1}\binom{p-1}{\omega_2-1}}. \tag{5.42}$$

For the lower bound, we separate our sum into three different sums, for $k \leqslant \lfloor \frac{p-\omega_1}{2} \rfloor$, $\lfloor \frac{p-\omega_1}{2} \rfloor < k < p-1-\lfloor \frac{p-\omega_2}{2} \rfloor = \lceil \frac{p+\omega_2}{2} \rceil - 1$ and $\lceil \frac{p+\omega_2}{2} \rceil - 1 \leqslant k \leqslant p-\omega_1$ and use relations (5.39), (5.40) and (5.41):

$$\begin{aligned} P\left(Y_{p,\omega_1,\omega_2} \geqslant p-1\right) \geqslant & \sum_{k=\omega_2-1}^{p-\omega_1} \omega_1 \frac{\binom{p-k-2}{\omega_1-2}}{\binom{p-1}{\omega_1-1}} \omega_2 \frac{\binom{k}{\omega_2-1}}{\binom{p-1}{\omega_2-1}} \\ & - \sum_{k=\omega_2-1}^{\lfloor \frac{p-\omega_1}{2} \rfloor} \binom{\omega_1}{2} \frac{\binom{p-2k-1}{\omega_1-1} - \binom{p-2k-3}{\omega_1-1}}{\binom{p-1}{\omega_1-1}} \omega_2 \frac{\binom{k}{\omega_2-1}}{\binom{p-1}{\omega_2-1}} \\ & - \sum_{k=\lceil \frac{p+\omega_2}{2} \rceil - 1}^{p-\omega_1} \binom{\omega_2}{2} \frac{\binom{p-k-2}{\omega_1-2}}{\binom{p-1}{\omega_1-1}} \omega_1 \frac{\binom{2k-p+1}{\omega_2-1}}{\binom{p-1}{\omega_2-1}} \end{aligned}$$

We use the relations $\binom{p-2k-1}{\omega_1-1} - \binom{p-2k-3}{\omega_1-1} = \binom{p-2k-2}{\omega_1-2} + \binom{p-2k-3}{\omega_1-2} \leqslant 2\binom{p-2k-2}{\omega_1-2}$ (as $\omega_1 \geqslant 2$), $\frac{\omega_1\omega_2}{\binom{p-1}{\omega_1-1}\binom{p-1}{\omega_2-1}} = \frac{p^2}{\binom{p}{\omega_1}\binom{p}{\omega_2}}$ and a change of variable $k \to p-k-2$ in the last sum to get

$$\begin{aligned} \frac{\binom{p}{\omega_1}\binom{p}{\omega_2}}{p^2} P\left(Y_{p,\omega_1,\omega_2} \geqslant p-1\right) \geqslant & \binom{p-1}{\omega-2} - \omega_1 \sum_{k=\omega_2-1}^{\lfloor \frac{p-\omega_1}{2} \rfloor} \binom{p-2k-2}{\omega_1-2}\binom{k}{\omega_2-1} \\ & - \frac{1}{2}\omega_2 \sum_{k=\omega_1-2}^{\lfloor \frac{p-\omega_2}{2} \rfloor - 1} \binom{p-2k-3}{\omega_2-1}\binom{k}{\omega_1-2} \end{aligned}$$

Now we use the following bound on the product $\binom{p-2k-2}{\omega_1-2}\binom{k}{\omega_2-1} \leqslant \binom{p-k-2}{\omega-3}$ and the relation from [Gou72] $\sum_{k=r}^{s}\binom{a-k}{b} = \binom{a-r+1}{b+1} - \binom{a-s}{b+1} \leqslant \binom{a-r+1}{b+1}$ to get

$$\begin{aligned} \frac{\binom{p}{\omega_1}\binom{p}{\omega_2}}{p^2} P\left(Y_{p,\omega_1,\omega_2} \geqslant p-1\right) &\geqslant \binom{p-1}{\omega-2} - \omega_1\binom{p-\omega_2}{\omega-2} - \frac{1}{2}\omega_2\binom{p-\omega_1}{\omega-2} \\ &\geqslant \binom{p-1}{\omega-2} - \frac{3}{2}\omega_1\binom{p-\omega_2}{\omega-2}. \end{aligned}$$

if $\omega_1 = \max(\omega_1, \omega_2)$. We finally get the bounds

$$1 - \frac{3\omega_1}{2}\frac{\binom{p-\omega_2}{\omega-2}}{\binom{p-1}{\omega-2}} \leqslant \frac{P\left(Y_{p,\omega_1,\omega_2} \geqslant p-1\right)}{\frac{p^2\binom{p-1}{\omega-2}}{\binom{p}{\omega_1}\binom{p}{\omega_2}}} \leqslant 1. \tag{5.43}$$

We check that the lower bound tends to 1 when $\omega_i = O(\sqrt{p\log p})$. □

The proof of Theorem 5.7.3 comes directly from the later proposition and the fact that $\frac{\left|\widetilde{\mathcal{W}}_{\omega_1,\omega_2}/\left(\mathbb{F}_p\right)^2\right|}{\left|\mathcal{P}_{\omega_1,\omega_2}/\left(\mathbb{F}_p\right)^2\right|} = P\left(Y_{p,\omega_1,\omega_2} \geqslant p-1\right).$

Corollary 5.8.30. *For a smooth $(2p, p, \omega)$ QC-MDPC scheme we have*

$$P(Y_{p,\omega/2,\omega/2} \geqslant p-1) \sim \left(\frac{\frac{\omega}{2}}{\binom{p-1}{\frac{\omega}{2}-1}}\right)^2 \binom{p-1}{\omega-2} \quad \text{when } p \to \infty \quad \text{and} \quad \omega = O(\sqrt{p\log p}). \tag{5.44}$$

If we recall the results obtained with the first method in Proposition 5.4.10 and Corollary 5.4.12 we conclude that we can attack ω^2 times more private keys

$$P(Y_{p,\omega_1,\omega_2} \geqslant p-1) \sim \omega^2 \times \frac{|\mathcal{W}_{\omega_1,\omega_2}|}{|\mathcal{P}_{\omega_1,\omega_2}|}.$$

Our result might be extended to the set of all $(2p, p, \omega)$ QC-MDPC codes.

Proposition 5.8.31. *Let p be a prime number and ω an even integer such that $\omega = O(\sqrt{p\log p})$. Then we have*

$$\left(\frac{\frac{\omega}{2}}{\binom{p-1}{\frac{\omega}{2}-1}}\right)^2 \binom{p-1}{\omega-2} \leqslant \frac{\left|\widetilde{\mathcal{W}}_\omega\right|}{|\mathcal{P}_\omega|} \leqslant \omega p^2 \frac{\binom{p-1}{\omega-2}}{\binom{2p}{\omega} + (-1)^{\frac{\omega}{2}+1}\binom{p}{\frac{\omega}{2}}}. \tag{5.45}$$

Remark 5.8.32. *If we recall the result in Proposition 5.4.10 we obtain a gain factor that is close to ω^2 when we consider the cyclic shifts.*

5.9 Extended weak orbits

5.9.1 General Properties

Proposition 5.9.1. *Let $\alpha \in {\mathbb{F}_p}^* \setminus \{1\}$ and $c \in \mathbb{F}_2[x]/(x^p-1)$. The following equivalence holds:*

$$\sigma_\alpha^*\left(\mathbb{F}_p.\, c\right) = \mathbb{F}_p.\, c \Leftrightarrow \exists!\ \beta \in \mathbb{F}_p,\ \sigma_\alpha^*(\sigma_\beta^+(c)) = \sigma_\beta^+(c). \tag{5.46}$$

For $\alpha = 1$ we have $\forall c^* \in \mathbb{F}_p.\, c,\ \sigma_\alpha^*(c^*) = c^*$.

Proof. For $\alpha = 1$ the relation comes directly from the definition of the group action. Now assume that $\alpha \neq 1$.

The ($\Leftarrow$) implication comes from the definition of the orbits.

For the ($\Rightarrow$) implication let c be such that $\sigma_\alpha^*(\mathbb{F}_p.\, c) = \mathbb{F}_p.\, c$. This implies that there exits $j < p$ so that $\sigma_\alpha^*(c) = x^j c = \sigma_j^+(c)$. Recall from Proposition 5.6.3 that

$$\forall(\alpha, \beta) \in {\mathbb{F}_p}^* \times \mathbb{F}_p \quad \sigma_\alpha^*(\sigma_\beta^+(c)) = \sigma_{\alpha\beta}^+(\sigma_\alpha^*(c)).$$

In particular for $\beta = -j(\alpha - 1)^{-1}$ the later equality also holds, fact that implies

$$\begin{aligned}
\sigma_\alpha^*(\sigma_{-j(\alpha-1)^{-1}}^+(c)) &= \sigma_{\alpha(-j)(\alpha-1)^{-1}}^+(\sigma_\alpha^*(c)) \\
&= \sigma_{\alpha(-j)(\alpha-1)^{-1}}^+(\sigma_j^+(c)) \\
&= \sigma_{\alpha(-j)(\alpha-1)^{-1}+j}^+(c) \\
&= \sigma_{(-j)(\alpha-1)^{-1}(\alpha+(-1)(\alpha-1))}^+(c) \\
&= \sigma_{-j(\alpha-1)^{-1}}^+(c).
\end{aligned}$$

Let's now suppose that β is not unique, in other words there are two polynomials in the orbit $\mathbb{F}_p.\, c$ such that $\sigma_\alpha^*(\sigma_i^+(c)) = \sigma_i^+(c)$ and $\sigma_\alpha^*(\sigma_l^+(c)) = \sigma_l^+(c)$, with $0 \leqslant i < l \leqslant p-1$. But this implies

$$\begin{aligned}
\sigma_\alpha^*(\sigma_l^+(c)) &= \sigma_\alpha^*\left(\sigma_{l-i}^+(\sigma_i^+(c))\right) \\
&= \sigma_{\alpha(l-i)}^+\left(\sigma_\alpha^*(\sigma_i^+(c))\right) \\
&= \sigma_{\alpha(l-i)}^+\left(\sigma_i^+(c)\right) \\
&= \sigma_{\alpha(l-i)+i}^+(c).
\end{aligned}$$

Since $\sigma_\alpha^*(\sigma_l^+(c)) = \sigma_l^+(c)$, we deduce

$$\sigma_l^+(c) = \sigma_{\alpha(l-i)+i}^+(c),$$

fact that implies $(\alpha - 1)(l - i) = 0$. So unless $\alpha = 1$ or $l = i$ the equation is not satisfied, which ends the proof

□

Let's recall a well-known result about the sugbroups of a finite cyclic group, theorem that will be applied to ${\mathbb{F}_p}^*$,

Theorem 5.9.2 ([BC68, Theorem 4.9])**.** *Let $G = \langle g \rangle$ be a finite cyclic group, with size m. Then any subgroup of G has the form $\langle g^d \rangle$, where d is a positive divisor of m. Different values of d give subgroups with different sizes, so there is just one subgroup of G having a given size.*

With this theorem at hand we directly deduce

Proposition 5.9.3. *Let $c \in \mathbb{F}_2[x]/(x^p - 1)$ and Γ_c be the subgroup of $({\mathbb{F}_p}^*, \cdot)$ which stabilizes $\mathbb{F}_p.\, c$. Then the cardinality of the orbit ${\mathbb{F}_p}^*.(\mathbb{F}_p.\, c)$ is*

$$|{\mathbb{F}_p}^*.(\mathbb{F}_p.\, c)| = \frac{(p-1)}{|\Gamma_c|}. \tag{5.47}$$

But in our case we have to consider polynomials with a given Hamming weight, fact that seems to complicate the computations. We demonstrate that orbits of polynomials with a given weight are in fact easier to compute. For that we need to introduce some notations.

Notation 5.9.4. *We denote the set of polynomial in* $\mathbb{F}_2[x]/(x^p-1)$ *that are fixed by an element* $\alpha \in {\mathbb{F}_p}^*$ *by*

$$R_p^\alpha = \{c \in \mathbb{F}_2[x]/(x^p-1) \mid \sigma_\alpha^*(c) = c\}.$$

$C_i^\alpha = \{i, i\alpha, \ldots, i\alpha^{\mathsf{ord}(\alpha)-1}\}$, *is the* α*-cyclotomic class of* i *where* $\mathsf{ord}(\alpha)$ *is the order of* α *in* ${\mathbb{F}_p}^*$. *For a given* α *we denote by* $I_p^\alpha = \{i_1, i_2, \ldots, i_l\}$ *the set containing one representative of each* α*-cyclotomic class.*

When p is prime for a given $\alpha \in {\mathbb{F}_p}^*$ there are $(p-1)/\mathsf{ord}(\alpha)$ cyclotomic classes having the same cardinality plus the class of zero which is reduced to only one element. Hence $I_p^\alpha = \{0, i_1, i_2, \ldots, i_{(p-1)/\mathsf{ord}(\alpha)}\}$.

Example 5.9.5. *Let* $p = 7$. *Then we have*

- *for* $\alpha = 1$ *we have* $C_i^1 = \{i\}$ *for any* $0 \leqslant i \leqslant 6$.
- *for* $\alpha = 2$ *and* $\alpha = 4$ *we have* $\mathsf{ord}(2) = \mathsf{ord}(4) = 3$, *hence we obtain* $C_0^2 = \{0\}, C_1^2 = \{1, 2, 4\}$ *and* $C_3^2 = \{3, 5, 6\}$
- *for* $\alpha = 3$ *and* $\alpha = 5$ *we have* $\mathsf{ord}(3) = \mathsf{ord}(5) = 6$ *and* $C_0^3 = \{0\}$ *and* $C_1^3 = \mathbb{F}_7^*$.
- *for* $\alpha = 6$ *we have* $\mathsf{ord}(6) = 2$. *We obtain* $C_0^6 = \{0\}$ *and* $C_i^6 = \{i, 7-i\}$ *for any* $1 \leqslant i \leqslant 3$.

Proposition 5.9.6. *The set of polynomials in* $\mathbb{F}_2[x]/(x^p-1)$ *that are fixed by an element* $\alpha \in {\mathbb{F}_p}^*$ *equals*

$$R_p^\alpha = \{c(x) = \sum_{i=0}^{p-1} c_i x^i \in \mathbb{F}_2[x]/(x^p-1) \mid \forall j \in C_i^\alpha \quad c_j = c_i, \text{ for } i \in I_p^\alpha\}.$$

Proof. By the definition we have that $\sigma_\alpha^*(c) = c$ is equivalent to

$$\forall i \in \mathbb{F}_p \quad c_{\alpha i \mod p} = c_i. \tag{5.48}$$

So if we choose one element $i \in I_p^\alpha$ then $\alpha i \mod p \in C_i^\alpha$ has to satisfy the later equation. But this means that $\alpha(\alpha i) \in C_i^\alpha$ is also an index that has to satisfy the same equation. In other words all the indices in C_i^α must satisfy the same equation and thus $\forall j \in C_i^\alpha$ we have $c_i = c_j$.

□

Example 5.9.7. *Let* $p = 7$. *Then we have*

- *for* $\alpha = 1$, $R_7^1 = \mathbb{F}_2[x]/(x^p-1)$.
- *for* $\alpha = 2$ *and* $\alpha = 4$, $R_7^2 = R_7^4 = \{c_0 + c_1(x+x^2+x^4) + c_3(x^3+x^5+x^6)\ ,\ c_0, c_1, c_3 \in \mathbb{F}_2\}$.

- *for* $\alpha = 3$ *and* $\alpha = 5$, $R_7^3 = R_7^5 = \{c_0 + c_1(x + x^2 + x^3 + x^4 + x^5 + x^6)\ ,\ c_0, c_1 \in \mathbb{F}_2\}$.
- *for* $\alpha = 6$, $R_7^6 = \{c_0, c_1(x + x^6) + c_2(x^2 + x^5) + c_3(x^3 + x^4)\ ,\ c_0, c_1, c_2, c_3 \in \mathbb{F}_2\}$.

Remark 5.9.8. *We notice that*

- *for a fixed* $i \in {\mathbb{F}_p}^*$, *for any element* $j \in C_i^\alpha$ *we have* $R_p^i = R_p^j$. *Fact that can be expressed in an equivalent manner*

$$\text{for } \langle i \rangle \subset {\mathbb{F}_p}^*, \forall j \in \langle i \rangle \text{ we have } R_p^i = R_p^j.$$

- *the classes defined by the element* -1 *are* $C_0^{p-1} = \{0\}$ *and* $C_i^{p-1} = \{i, p-i\}$ *for all* $1 \leqslant i < \frac{p-1}{2}$.
- *when* p *is odd* ${\mathbb{F}_p}^*$ *admits at least three subgroups: the trivial group, the whole group* ${\mathbb{F}_p}^*$ *and* $\langle -1 \rangle$, *which is a group of order two.*

Proposition 5.9.9. *Let* $\alpha \in {\mathbb{F}_p}^*$ *and* $c \in \mathbb{F}_2[x]/(x^p - 1)$ *so that* $1 < \|c\| < p$ *and* $\sigma_\alpha^*(c) = c$. *Then the order of* α *divides either* $\|c\|$ *or* $\|c\| - 1$.

Proof. From Proposition 5.9.6 we know that any polynomial $c \in R_p^\alpha$ is fully described $c = \sum_{i \in I_p^\alpha} c_i \sum_{j_i \in C_i^\alpha} (x^{j_i})$.But for a fixed α the cardinality of any equivalence class C_i^α equals the order of the element $|C_i^\alpha| = \mathsf{ord}(\alpha)$, from which we deduce two possible cases:

- either $c_0 = 0$ and we obtain that $\mathsf{ord}(\alpha) | \|c\|$.
- or $c_0 = 1$ and then we obtain $\mathsf{ord}(\alpha) | \|c\| - 1$.

□

So only group elements that respect the property given above can fix elements in the set of polynomials with weight restrictions. Since the order of any element $\alpha \in {\mathbb{F}_p}^*$ also satisfies the condition $\mathsf{ord}(\alpha) | (p-1)$ we deduce that only $\alpha \in {\mathbb{F}_p}^*$ satisfying $\mathsf{ord}(\alpha) | \gcd(\|c\|, p-1)$ or $\mathsf{ord}(\alpha) | \gcd(\|c\| - 1, p-1)$ can stabilize c. Thus a natural consequence is that we can use the Burnside lemma for counting the number of orbits in this case.

Theorem 5.9.10. *Let* p *be a prime number and* $1 \leqslant \omega \leqslant p-1$. *Then we have*

$$|L(p,\omega)/{\mathbb{F}_p}^*| = \frac{1}{p-1}\left(\frac{1}{\omega}\binom{p-1}{\omega-1} + \sum_{\substack{\alpha \in {\mathbb{F}_p}^* \setminus \{1\} \\ \mathsf{ord}(\alpha) | \gcd(p-1,\omega)}} \binom{\frac{p-1}{\mathsf{ord}(\alpha)}}{\frac{\omega}{\mathsf{ord}(\alpha)}} + \sum_{\substack{\alpha \in {\mathbb{F}_p}^* \setminus \{1\} \\ \mathsf{ord}(\alpha) | \gcd(p-1,\omega-1)}} \binom{\frac{p-1}{\mathsf{ord}(\alpha)}}{\frac{\omega-1}{\mathsf{ord}(\alpha)}}\right)$$

Example 5.9.11.

For $p = 7$ *and* $\omega = 3$ *we have:*

$$L(7,3) = \{0000111, 0001011, 0010011, 0001101, 0010101\}$$

We also have $|\mathbb{F}_7^*| = 6$ *and* $|L(7,3)| = 5$. *We count the number of fixed points for all the elements* $\alpha \in \mathbb{F}_7^*$.

$$\begin{aligned}
&\alpha = 1, \mathsf{ord}(1) = 1 \Longrightarrow L(7,3)\\
&\alpha = 2, \mathsf{ord}(2) = 3 \Longrightarrow \{0001011, 0001101\}\\
&\alpha = 3, \mathsf{ord}(3) = 6 \Longrightarrow \emptyset\\
&\alpha = 4, \mathsf{ord}(4) = 3 \Longrightarrow \{0001011, 0001101\}\\
&\alpha = 5, \mathsf{ord}(5) = 6 \Longrightarrow \emptyset\\
&\alpha = 6, \mathsf{ord}(6) = 2 \Longrightarrow \{0000111, 0010011, 0010101\}
\end{aligned}$$

So using Theorem 5.9.10 we obtain

$$|L(7,3)/\mathbb{Z}_7^*| = \frac{1}{6}\left(1/3\binom{6}{2} + \binom{6/3}{3/3} + \binom{6/3}{3/3} + \binom{6/2}{2/2}\right) = \frac{5+2+2+3}{6} = 2$$

Verification:

$$\mathbb{F}_p{}^*.0000111 = \{0000111, 0010011, 0010101\} \text{ and } \mathbb{F}_p{}^*.0001011 = \{0001011, 0001101\}$$

5.9.2 Proportion of extended weak orbits

The case of a 2-quasi-cyclic MDPC code is a direct generalization of the results obtained in Proposition 5.9.3 and 5.9.9.

Corollary 5.9.12. *Let $(h_1, h_2) \in \mathcal{P}_{\omega_1,\omega_2}$ with $\|h_i\| = \omega_i$ and denote by $\Gamma_{(h_1,h_2)}$ the subgroup of $\mathbb{F}_p{}^*$ that stabilizes $(\mathbb{F}_p)^2.(h_1, h_2)$. Then we have*

- *any $\alpha \in \Gamma_{(h_1,h_2)}$ is such that $\mathsf{ord}(\alpha) | \gcd(p-1, \gcd(l_1, l_2))$, where l_i runs through $\{\omega_i, \omega_i - 1\}$, for $1 \leqslant i \leqslant 2$.*
- *The cardinality of an orbit equals*

$$\left|\mathbb{F}_p{}^*.\left((\mathbb{F}_p)^2.(h_1,h_2)\right)\right| = \frac{(p-1)}{|\Gamma_{(h_1,h_2)}|}. \tag{5.49}$$

- *When (h_1, h_2) is a private key of a $(2p, p, \omega)$ QC-MDPC code with $\omega_i = O(\sqrt{p \log p})$ we have*

$$O\left(\sqrt{\frac{p}{\log p}}\right) \leqslant \left|\mathbb{F}_p{}^*.\left((\mathbb{F}_p)^2.(h_1,h_2)\right)\right| \leqslant p-1. \tag{5.50}$$

Proposition 5.9.13. *Let (h_1, h_2) be a private key of a $(2p, p, \omega)$ QC-MDPC code. If $(h_1, h_2) \in \widetilde{\mathcal{W}}_{\omega_1,\omega_2}$ then $\sigma_{-1}^*\left((\mathbb{F}_p)^2.(h_1, h_2)\right)$ is a weak orbit.*

Moreover if $-1 \in \Gamma_{(h_1,h_2)}$ then $|\mathbb{F}_p{}^*.((\mathbb{F}_p)^2.(h_1, h_2))| \leqslant \frac{p-1}{2}$.

In general for a smooth QC-MDPC code we obtain

Corollary 5.9.14. *Let p and ω be the parameters of a smooth $(2p, p, \omega)$ QC-MDPC code. Then we have*

$$\frac{p-1}{\omega/2}\frac{\left|\widetilde{\mathcal{W}}_{\omega/2,\omega/2}\right|}{\left|\mathcal{P}_{\omega/2,\omega/2}\right|} \leqslant \frac{\left|\widetilde{\widetilde{\mathcal{W}}}_{\omega/2,\omega/2}\right|}{\left|\mathcal{P}_{\omega/2,\omega/2}\right|} \leqslant (p-1)\frac{\left|\widetilde{\mathcal{W}}_{\omega/2,\omega/2}\right|}{\left|\mathcal{P}_{\omega/2,\omega/2}\right|}. \tag{5.51}$$

Now if we consider the set of all $(2p, p\omega)$ QC-MDPC codes we obtain:

Proposition 5.9.15. *Let p be a prime number and ω an even integer such that $\omega = O(\sqrt{p \log p})$. Then we have*

$$\frac{p-1}{\omega/2}\left(\frac{\frac{\omega}{2}}{\binom{p-1}{\frac{\omega}{2}-1}}\right)^2\binom{p-1}{\omega-2} \leqslant \frac{\left|\widetilde{\widetilde{\mathcal{W}}}_{\omega}\right|}{|\mathcal{P}_{\omega}|} \leqslant \frac{\omega p^3}{2}\frac{\binom{p-1}{\omega-2}}{\binom{2p}{\omega}+(-1)^{\frac{\omega}{2}+1}\binom{p}{\frac{\omega}{2}}}. \tag{5.52}$$

5.10 Numerical Results

The parameters chosen for the experimental part are those suggested by the designers of the scheme [MTSB13]. The security levels correspond to the best known attacks given in [MTSB13] and the probabilities displayed in Figure 5.8 and 5.9 are computed directly from the formulas given in Corollary 5.4.12, Proposition 5.8.25, Corollary 5.8.30 and Proposition 5.4.10.

In Figure 5.8 we compute the exact values directly from Corollary 5.4.12 and Proposition 5.8.25 for the first and the second probability. In the last column we give the asymptotic value of the probability of a weak orbit from Corollary 5.8.30. The asymptotic value approaches very precisely the exact value, at least when the exact computation is possible. We used the following procedure to obtain our results:

- We generate the list $L := \left[\binom{\frac{\omega}{2}}{p-\frac{\omega}{2}}_k - \binom{\frac{\omega}{2}}{p-\frac{\omega}{2}}_{k-1}\right]_{k\in\{(p-1)/\frac{\omega}{2},\ldots,p-\frac{\omega}{2}\}}$.
- We compute the probability from Equation 5.36

$$P\left(Y_{p,\omega_1,\omega_2} \geqslant p-1\right) = \sum_{\substack{k_1+k_2\geqslant p-1 \\ k_1,k_2\in\{(p-1)/\frac{\omega}{2},\ldots,p-\frac{\omega}{2}\}}} L[k_1]L[k_2].$$

In Figure 5.9, we display the probability values for all $\omega_1 + \omega_2 = \omega$. In the first column we compute the exact value of the probability from Proposition 5.4.10. Whereas in the next column we compute the asymptotic value of lower bound and the upper bound. In the last column we give only the asymptotic value for the upper bound. One might think that the upper bound is not very tight and that the exact value of the probability is way lower than the value of the upper bound. Even though we share this concern we want to insist on the fact that for achieving sharp bounds we need to deal with complex summations involving the generalized Pascal-DeMoivre triangles.

In order to cope with that, we experimented for several values of the parameters, which show that the probability is quite close to the upper bound. As p goes to infinity and $\omega = O\left(\sqrt{p \log p}\right)$ the difference between the two values tends to zero. We compute the probabilities for the first cryptographic parameters $p = 4801$ and $\omega = 90$. The exact value for the probability equals $2^{-71.26}$ whereas the upper bound equals $2^{-71.12}$.

Security level	p	$\frac{\omega}{2}$	$\frac{\lvert\mathcal{W}_{\omega/2,\omega/2}\rvert}{\lvert\mathcal{P}_{\omega/2,\omega/2}\rvert}$ Corollary 5.4.12 exact value	$P(X_{p,\frac{\omega}{2}} \geqslant \frac{p-1}{2})^2$ Proposition 5.8.25 exact value	$P(Y_{p,\frac{\omega}{2},\frac{\omega}{2}} \geqslant p-1)$ Equation 5.36 exact value	$P(Y_{p,\frac{\omega}{2},\frac{\omega}{2}} \geqslant p-1)$ Corollary 5.8.30 asympt. value
80	4801	45	2^{-87}	2^{-78}	$2^{-74.04}$	$2^{-74.04}$
	3593	51	2^{-99}	2^{-90}	$2^{-86.02}$	$2^{-86.02}$
	3079	55	2^{-108}	2^{-98}	$2^{-94.12}$	$2^{-94.12}$
128	9857	71	2^{-139}	2^{-128}	$2^{-124.52}$	$2^{-124.52}$
	7433	81	2^{-159}	2^{-149}	$2^{-145.58}$	$2^{-144.58}$
	6803	85	2^{-167}	2^{-157}	$2^{-153.67}$	$2^{-152.67}$
256	32771	132	2^{-260}	2^{-249}		$2^{-244.3}$
	22531	155	2^{-307}	2^{-295}		$2^{-290.5}$
	20483	161	2^{-319}	2^{-307}		$2^{-302.7}$

Figure 5.8 – Probability of a weak key (orbit) for the QC-MDPC when $\omega_1 = \omega_2 = \frac{\omega}{2}$.

Security level	p	$\frac{\omega}{2}$	$\frac{\lvert\mathcal{W}_{\omega}\rvert}{\lvert\mathcal{P}_{\omega}\rvert}$ Proposition 5.4.10 exact value	$\frac{\lvert\widetilde{\mathcal{W}}_{\omega}\rvert}{\lvert\mathcal{P}_{\omega}\rvert}$ Proposition 5.8.31 bounds Eq. (5.45)	$\frac{\lvert\widetilde{\widetilde{\mathcal{W}}}_{\omega}\rvert}{\lvert\mathcal{P}_{\omega}\rvert}$ Proposition 5.9.15 upper bound
80	4801	45	2^{-84}	$[2^{-74}, 2^{-71}]$	2^{-60}
	3593	51	2^{-96}	$[2^{-86}, 2^{-83}]$	2^{-72}
	3079	55	2^{-105}	$[2^{-94}, 2^{-91}]$	2^{-80}
128	9857	71	2^{-136}	$[2^{-125}, 2^{-121}]$	2^{-109}
	7433	81	2^{-156}	$[2^{-145}, 2^{-141}]$	2^{-129}
	6803	85	2^{-164}	$[2^{-153}, 2^{-149}]$	2^{-137}
256	32771	132	2^{-257}	$[2^{-244}, 2^{-241}]$	2^{-227}
	22531	155	2^{-303}	$[2^{-291}, 2^{-287}]$	2^{-273}
	20483	161	2^{-315}	$[2^{-303}, 2^{-299}]$	2^{-285}

Figure 5.9 – Probability of a weak key, extended weak pairs and improvements on extended weak pairs for the QC-MDPC for all $\omega_1 + \omega_2 = \omega$.

5.11 Complexity and Experimental Timings

5.11.1 Preliminaries

In general for the Key Recovery Attack the model is quite different in the sense that for any given public key there is an algorithm able to recover the corresponding private key. So the complexity of the KRA is given by the Working Factor (WF) of the best generic attack, which in the case of QC-MDPC is exponential in the parameters of the scheme. In our case the attack algorithm recovers a private key from the public key only if it's applied on a weak key, weak orbit or extended weak orbit. Therefore in this model the WF of the attack is given by the complexity of the algorithm that solves the Rational Reconstruction Problem.

Extended Euclidean Algorithm complexity. The main brick in our algorithm is the Rational Reconstruction, that can be performed using the Extended Euclidean Algorithm. The original version of the EEA has a time complexity which is quadratic in the length of the input, here $O(p^2)$. The first optimizations were proposed by Lehmer [Leh38] in 1938, when the constant factor was improved but the complexity was still quadratic. The first sub-quadratic algorithm was proposed in 1970 by Knuth [Knu71b] with complexity $O(p(\log p)^5 \log\log p)$ and shortly after revisited by Schönhage in 1971 [Sch71] who obtained

a better complexity $O(p(\log p)^2 \log\log p)$. The Least-Significant-Bit version of the Knuth-Schönhage algorithm is due to Stehlé and Zimmermann in 2004 [SZ04]. Even though the time complexity of this algorithm is not improved the description and the proof of their algorithm is significantly simpler in this case. The average behavior of the EEA was studied in [LV08, CCD+09].

We remark that here we employ a slightly modified EEA. Indeed, in our version the algorithm stops when the weight of the two outgoing polynomials satisfies the relation in (5.6). This means that the number of iterations might possibly be less than the maximum number of steps in EEA. But this advantage is balanced by the fact that we check the weight of the polynomials. The complexity of checking the weight of the polynomials equals $\omega + \varepsilon$ bit operations, where ε can vary in the set $\{1, \ldots, p - 1 - \omega\}$. This fact comes from the degree condition of the outgoing polynomials in the EEA, that is $\deg(h_1) + \deg(h_2) < p$. Now if (h_1, h_2) is a weak key we might have to check $p - 1$ coefficients and if not, in the best case only ω plus a constant number of coefficients.

5.11.2 Complexity analysis

Worst case.

The worst case scenario is when the attack is performed on a key that is not weak, with respect to our definition. It is also the case when the attacker manages to find a weak key at the last operation, more exactly when all the $p - 1$ actions of the multiplicative group were shot and all the p actions of the additive group were tried. In this case we perform $p - 1$ actions of ${\mathbb{F}_p}^*$ times p actions of the $\mathbb{F}_p$ times the EEA. So in the worst case the work factor equals $p(p-1) \cdot \mathrm{WF}_{\mathsf{EEA}}$. We stress out the fact that due to the values of the weak keys, weak orbits and extended weak orbits, for the cryptographic purpose in [MTSB13] the worst case scenario work factor complexity is also valid in the case of a random key.

Now let's suppose that the private key of a QC-MDPC is a weak key or a weak orbit or an extended weak orbit. We will study the work factor of the attack for each configuration in the best case and in average.

Best case.

The best case for an adversary is when the private key is a weak key and in this case we have a work factor $\mathrm{WF}_{\mathsf{EEA}}$. Slightly different but still at a constant close work factor from the best case is when a *small* number of cyclic shifts is needed to reach a weak key.

Average case.

We consider here two scenarios

- *Average case for a weak orbit.* Let (h_1, h_2) be the private key if a QC-MDPC scheme such that it is not weak, meanwhile the orbit under the additive group $\mathbb{F}_p$ is a weak one. In other words $k_1 + k_2 \geqslant p - 1$, where k_i represents as before, the longest run of h_i. We first notice that the private key cannot be recovered by applying directly the modified EEA on the public key. So we need to test a several number of cyclic shifts in order to find the weak pair.

In order to estimate the average number of shifts needed to retrieve a weak pair we set $k_1 + k_2 = p - 1 + \varepsilon$, where $0 \leqslant \varepsilon \leqslant p + 1 - \omega$. Then we have

Proposition 5.11.1. *Let $(h_1, h_2) \in \mathbb{F}_2[x]/(x^p - 1)$ with $\|h_1\| + \|h_2\| = \omega$ and $k_1 + k_2 = p - 1 + \varepsilon$, where $0 \leqslant \varepsilon \leqslant p + 1 - \omega$. Let $f \in \mathbb{F}_2[x]/(x^p - 1)$ be such that $f = h_2 h_1^{-1} \pmod{x^p - 1}$.*

Then there are $2\varepsilon+1$ shifts of f such that the outgoing polynomials of $\mathsf{EEA}(x^p-1, x^i f)$ are weak pairs.

Proof. The first weak key given (h_1, h_2) with $k_1 + k_2 = p - 1 + \varepsilon$ is the minimum pair $(\overline{h_1}, \overline{h_2})$. Then for all $1 \leqslant i \leqslant \varepsilon$ we have that $(x^i\overline{h_1}, \overline{h_2})$ and $(\overline{h_1}, x^i\overline{h_2})$ are also weak pairs. Therefore we have $2\varepsilon + 1$ different shifts which give weak pairs. □

Thus in order to estimate the average number of shifts we put a simple probability problem.

Proposition 5.11.2. *Consider a set of p balls (representing the total number of shifts), composed of two type of balls, $2\varepsilon + 1$ white balls and the rest are black balls.*

Then the probability of picking the first white ball at the i^{th} step equals for all $i \in \{1, \ldots, p - 2\varepsilon - 1\}$.

$$\frac{\binom{p-i}{2\varepsilon}}{\binom{p}{2\varepsilon+1}}$$

So in average the first white ball is picked at the $\dfrac{p+1}{2(\varepsilon+1)}^{th}$ step.

Proof. The problem is a classic urn process without replacement. In fact is the analogue of the geometric distribution, law that describes the first arrival probability in a urn process with replacement. To simplify the computations we denote the number of white balls by $d = 2\varepsilon + 1$. So the probability of choosing the first white ball at the i^{th} step is the ratio between the total number of white balls at the i^{th} step in the urn, which equals d and the number of balls in the urn, which equals $p - i + 1$, multiplied by the probability of choosing only black balls in the previous $(i - 1)$ steps, which equals $\frac{p-d}{p} \times \cdots \times \frac{p-d-i+2}{p-i+2}$. From this we deduce that the probability of the choosing the first white ball at the i^{th} step equals

$$\forall 1 \leqslant i \leqslant p - d + 1, \quad \frac{\binom{p-i}{d-1}}{\binom{p}{d}}.$$

We also need to verify that our distribution is well defined

$$\begin{aligned}
\sum_{i=1}^{p-d-1} \frac{\binom{p-i}{d-1}}{\binom{p}{d}} &= \frac{1}{\binom{p}{d}} \sum_{j=0}^{p-d} \binom{j+d-1}{d-1} \\
&= \frac{\binom{p}{d} - \binom{d-1}{d}}{\binom{p}{d}} \\
&= 1.
\end{aligned}$$

We used here the following formula from [Gou72] $\sum_{i=0}^{p} \binom{i+k}{r} = \binom{p+k+1}{r+1} - \binom{k}{r+1}$.

To compute the average we have

$$\begin{aligned}
\sum_{i=1}^{p-d-1} i\frac{\binom{p-i}{d-1}}{\binom{p}{d}} &= \sum_{j=0}^{p-d}(n-d+1-j)\frac{\binom{j+d-1}{d-1}}{\binom{p}{d}} \\
&= \frac{n-d+1}{\binom{p}{d}}\sum_{j=0}^{p-d}\binom{j+d-1}{d-1} - \frac{1}{\binom{p}{d}}\sum_{j=0}^{p-d} j\binom{j+d-1}{d-1} \\
&= n-d+1-\frac{1}{\binom{p}{d}}\sum_{j=0}^{p-d} d\binom{j+d-1}{d} \\
&= n-d+1-d\frac{\binom{p}{d+1}}{\binom{p}{d}} \\
&= n-d+1-d\frac{n-d}{d+1} \\
&= \frac{n+1}{d+1}.
\end{aligned}$$

□

When $\varepsilon = 0$ in other words only one shift works, we obtain the discrete uniform distribution and obtain an average number of shifts equal to $\frac{p+1}{2}$ (the working factor equals $\frac{p+1}{2}\mathrm{WF}_{\mathsf{EEA}}$). Whereas for $\varepsilon = \frac{p+1}{2c} - 1$, where c a positive constant, the average number of shifts is constant and equals c (the working factor equals $c\mathrm{WF}_{\mathsf{EEA}}$).

- *The average case for an extended weak orbit.* The last part of the analysis is estimating the average number of extension (actions of ${\mathbb{F}_p}^*$) needed to retrieve a weak pair. Since the relation between the action of this group and the longest run seems more complicated we just give a lower bound for the searched quantity. We suppose that there is at least one shift on $f \equiv \frac{h_1}{h_2} \mod (x^p - 1)$ such that two shifted h_1 and h_2 can be obtained using the EEA. By Proposition 5.9.13 that $\sigma_{-1}^*(f)$ can also be attacked with the same algorithm, so for each weak key there are at least two good extensions for our attack, which makes an average quantity upper bounded by $\frac{p}{3}$.

5.11.3 Numerical results

The first set of parameters that we used were not in the scale of the cryptographic values. More precisely we considered $p = 101$ and $\omega_1 = \omega_2 = 9$. The purpose was to confront the theoretical values for the probabilities of a weak keys and the experimental results. In this sense using MAGMA's random generator we computed 10^5 pair of polynomials for the QC-MDPC scheme and executed the attack on the shifted keys. In theory the

probability of finding a weak orbit equals 0.0032. Meanwhile in practice we obtained a probability equal to 0.00317 and the time needed to test all the orbits was approximately 6000 seconds.

In the second part we used the first parameters for the 2^{80} security level which are $p = 4801$ and $\omega = 90$ and consider the most frequent case $\max_{i\in\{1,2\}} \omega_i = 47$. In the first case we applied the EEA on a weak key. In the second part we generated a weak key that we shifted. Therefore we randomly choose an integer $i \in \mathbb{F}_p$ and applied the EEA on the i^{th} shift. We repeated the procedure until a weak key was found. In the worst case we had to compute all the p shifts, whereas in average we only needed a small number of trials until the weak key was discovered. The last column corresponds to the following experience. We generated a weak key, then we applied the action of $\mathbb{F}_p^*$ and the we shifted. In this case the procedure is the same: we randomly pick an element of the group $\mathbb{F}_p^*$ and consider the key under the action of this element. Then we apply the Shifted(EEA) until the proper pair of shift and extension in founded. In the worst case we compute all the possible combinations of shifts and extensions.

On a 4-core Intel(R) Xeon(R) CPU ES-2690 @ 2.90 GHz, using MAGMA V2.19-9 we applied two variants of the EEA : the recursive original variant with complexity $O(p^2)$ and the MAGMA implementation using the Knuth–Schönhage version.

	EEA	Shifted(EEA)		Extended(Shifted(EEA))	
	Best	Average	Worst	Average	Worst
Recursive Version	0.12 s	4.5 min	9.5 min	5.3 days	1 month
MAGMA Version	0.86 ms	2 s	4.1 s	1 h	5h30 min

Figure 5.10 – Experimental timings for the first set of parameters at a 2^{80} security level for the QC-MDPC scheme.

5.11.4 Secure QC-MDPC

In order to prevent the scheme from such weakness we have to check whether we can retrieve an equivalent private pair from the public key. The first solution is to apply our attack on all the elements in the orbit $\mathbb{F}_p{}^*.\mathbb{F}_p.f(x)$. But we can do much better using the longest run property from (5.23). Indeed we only need to check if the sum of the longest runs of $h_1(x)$ and $h_2(x)$ is bigger than $p-1$. Based on this remark our algorithm generates two private polynomials for the QC-MDPC code and then checks for each equivalent pair $\sigma^*_{(\alpha,\alpha)}(h_1, h_2)$ for all $\alpha \in \mathbb{F}_p{}^*$, if the longest run condition is satisfied. If the answer is positive then we restart from the beginning. Figure 2 illustrates the new Key Generation algorithm for the QC-MDPC scheme.

Proposition 5.11.3. *Let p be a prime number such that 2 is a primitive element in the group $\mathbb{F}_p{}^*$. Then the time complexity of Algorithm 2 is dominated by $N_f(p-1)(2\omega+\omega^2)$ binary operations, where N_f is the number of times we repeat Step 2.*

Proof. The time complexity of Algorithm 2 equals $p-1$ (the number of steps in the loop) times the working factor of computing the two lists and their corresponding longest run. Computing the lists takes ω modular multiplications, at which we add the *Sort* function

Input: Two odd integers ω_1 and ω_2 s.t. $\omega_1 + \omega_2 = \omega$
Output: A secure private pair $(h_1(x), h_2(x))$ for the QC-MDPC scheme
1 $S \leftarrow \{0, \ldots, p-1\}$;
2 $L_1 \leftarrow RandomSubset(S, \omega_1)$, $L_2 \leftarrow RandomSubset(S, \omega_2)$;
3 // $RandomSubset(E, \omega)$ randomly chooses a subset of E of cardinality ω;
4 **for** $\alpha \in {\mathbb{F}_p}^*$ **do**
5 $L_1^\alpha = Sort([\alpha i \pmod p : i \in L_1])$ and $L_2^\alpha = Sort([\alpha i \pmod p : i \in L_2])$;
6 $k_{L_j^\alpha} = \max(\max_{1 \leqslant i \leqslant \omega_j - 1}(L_j^\alpha[i+1] - L_j^\alpha[i]), p - 1 - L_j^\alpha[\omega_j] + L_j^\alpha[1])$, for $1 \leqslant j \leqslant 2$;
7 **if** $k_{L_1^\alpha} + k_{L_2^\alpha} \geqslant p - 1$ **then**
8 go back to Step 2;
9 **end**
10 $h_1(x) \leftarrow \sum_{i \in L_1} x^i$ and $h_2(x) \leftarrow \sum_{i \in L_2} x^i$

Algorithm 2: Secure Key Generation of a QC-MDPC McEliece scheme

that takes ω_i^2 operations plus ω additions which corresponds to the computation of the longest run of the two lists L_1^α and L_2^α. The result has to be multiplied by the number of failures, N_f which ends the proof. □

5.12 Perspectives

Our technique might also be useful for other public key encryption schemes like for example the NTRU scheme [HPS98]. Here the description is similar to the QC-MDPC case. The Key Generation algorithm takes as input the parameters (p, q, l) and outputs the private key $\mathsf{sk} = (h_1(x), h_2(x))$ and the public key $\mathsf{pk} = f(x)$ where

- $h_1(x), h_2(x) \in \mathbb{Z}_q[x]/(x^p - 1)$ are such that
  1. h_1 and h_2 are sparse polynomials with respect to the norm $\mathcal{L}_\infty$. In general the polynomials have coefficients in $\{-1, 0, +1\}$.
  2. h_1 and h_2 are moderately sparse polynomials with respect to the Hamming distance. For example when $p = 251$ the weight equals $\|h_1\| = \|h_2\| = 72$.
  3. h_1 admits an inverse modulo p and modulo q.
- $f(x) \in \mathbb{Z}_q[x]/(x^p - 1)$ is such that

$$h_1(x)f(x) = ph_2(x) \pmod q. \tag{5.53}$$

So if we apply the weak keys technique we must take into consideration Lyndon words of length p over the alphabet $\{-1, 0, 1\}$. Secondly it is rather unclear how the restriction on the Hamming weight is considered. Nevertheless the problem seems quite similar to the QC-MDPC and the probabilities might be computed in this case. But it appears at least for the $(251, 128, 3)-$NTRU that the probabilities are significantly smaller than in the first two security level of the DC-MDPC. Our statement is based on a simple argument, since the Hamming weight of the two polynomials equals 72, a rough estimation gives a probability close to $3^{-2\times 72} = 3^{-144}$.

6 Conclusion

Motivated by the up-growing weight that code-based cryptography has lately developed in the public key cryptography, we proposed to study during the thesis the security of the latest McEliece variants.

We started with the QC-MDPC McEliece, which is among the favorite candidates for quantum resistant encryption solutions. We proposed a key recovery attack based on an algorithm that solves the Rational Reconstruction Problem. The main advantage of our approach is the complexity of the attack, which is quadratic in the code length. Since only particular configurations of private keys can be retrieved with our algorithm we analyzed the proportion of such configurations, that we call weak keys. We continued our analysis and investigated different techniques for extending the proportion of weak keys. We developed two methods based on the equivalence of quasi-cyclic codes that allowed us to multiply the proportion of weak keys by a factor equal to the length of the code times the square of the Hamming weight of the private key. We finished this part by proposing an efficient and secure Key Generation algorithm for the QC-MDPC McEliece scheme.

Secondly we analyzed the security of the McEliece variant based on Polar codes. The idea of using Polar codes in cryptography came in a natural manner since Polar codes posses interesting properties like: they are capacity achieving codes for the class of Binary Discrete Memoryless channels, they admit efficient encoding and decoding procedures etc. Even though Polar codes are closely related to Reed-Muller codes the techniques used for the cryptanalysis of Reed-Muller codes do not work on Polar codes. Therefore we started to study in detail the structure of Polar codes. If at the beginning our purpose was purely cryptographic we finally answered to several non-trivial coding theory questions as well as to our initial problem. The first contribution was to propose an algebraic framework for both Polar and Reed-Muller codes. It is based on a partial order that we defined, order that allowed us to determine many of the hidden structural properties of Polar codes. Using some of the these properties, we proposed a successful cryptanalysis against the McEliece scheme based on Polar codes.

Perspectives in code-based cryptography Since most of the McEliece variants were successfully cryptanalyzed, several questions regarding this scheme were raised, among which a general security question of indistinguishability of the public code from a random code. And even though some variants remain secured against existing attacks there is no

theoretical guaranty of their security. By that we mean there is no security proof for the aforementioned variants. For instance there is no formal proof of the indistinguishability of the public code from a random one. Following McEliece's idea a possible solution for this problem would be to find a new masking technique for which there is a formal proof of the indistinguishability of the public code from a random one. In [Wan16] the author propose a masking technique for which he proves that the public code is equivalent to a random code and thus reintroduce in the context all the structural codes that have been broken. He mention that the attack we propose against Polar codes do not work any more in this new context. Another solution was already proposed by Alekhnovich who proposed an innovative approach based on the difficulty of decoding purely random codes [Ale11]. Several authors were inspired by his work [DMN12, DV13, KMP14, ABD+16]. This two approaches open a new perspective for code-based cryptography.

Appendices

A

Permutation group of linear codes

Proposition A.0.1. *Let $\mathscr{C}$ be a $[n,k,d]$ binary linear code and $\pi \in \mathfrak{S}_n$. Then we have*

$$\left(\mathscr{C}^{\perp}\right)^{\pi} = \left(\mathscr{C}^{\pi}\right)^{\perp}. \tag{A.1}$$

$$\mathcal{H}\left(\mathscr{C}^{\pi}\right) = \left(\mathcal{H}(\mathscr{C})\right)^{\pi}. \tag{A.2}$$

$$\left(\mathscr{C}^{2}\right)^{\pi} = \left(\mathscr{C}^{\pi}\right)^{2}. \tag{A.3}$$

Proof.

- Proof of Equation (A.1).

  First of all we remark that the scalar product is invariant by permutation of the vectors positions, namely $\boldsymbol{xy} \stackrel{\text{def}}{=} \sum_{i=1}^{n} x_i y_i = \sum_{i=1}^{n} x_{\pi(i)} y_{\pi(i)} \stackrel{\text{def}}{=} \boldsymbol{x}^{\pi^{-1}} \boldsymbol{y}^{\pi^{-1}}$.

  Now we can prove the first identity $\left(\mathscr{C}^{\perp}\right)^{\pi} = \left(\mathscr{C}^{\pi}\right)^{\perp}$. Let $\boldsymbol{y}$ be a codeword in $\left(\mathscr{C}^{\perp}\right)^{\pi}$. Then $\boldsymbol{y} = \boldsymbol{x}^{\pi}$ with $\boldsymbol{x} \in \mathscr{C}^{\perp}$ which means that $\forall\, \boldsymbol{c} \in \mathscr{C}$ we have that $\boldsymbol{y}\boldsymbol{c}^{\pi} = \boldsymbol{x}^{\pi}\boldsymbol{c}^{\pi} = 0$ by definition of $\boldsymbol{x}$. Therefore $\left(\mathscr{C}^{\perp}\right)^{\pi} \subset \left(\mathscr{C}^{\pi}\right)^{\perp}$.

  Moreover $\dim\left((\mathscr{C}^{\perp})^{\pi}\right) = \dim\left(\mathscr{C}^{\perp}\right) = \dim\left((\mathscr{C}^{\pi})^{\perp}\right)$, fact that ends the proof.

- Proof of Equation (A.2).

  We begin by recalling that for any n length binary linear codes $\mathscr{C}_1$ and $\mathscr{C}_2$ we have

  $$\mathscr{C}_1^{\pi} \cap \mathscr{C}_2^{\pi} = \left(\mathscr{C}_1 \cap \mathscr{C}_2\right)^{\pi}. \tag{A.4}$$

  Now we can use the previous result and obtain

  $$\left(\mathcal{H}(\mathscr{C})\right)^{\pi} \stackrel{\text{def}}{=} \left(\mathscr{C} \cap \mathscr{C}^{\perp}\right)^{\pi} \stackrel{(A.4)}{=} \mathscr{C}^{\pi} \cap \left(\mathscr{C}^{\perp}\right)^{\pi} \stackrel{(A.1)}{=} \mathscr{C}^{\pi} \cap \left(\mathscr{C}^{\pi}\right)^{\perp} \stackrel{\text{def}}{=} \mathcal{H}\left(\mathscr{C}^{\pi}\right).$$

- Proof of Equation (A.3).

  For the square code we use the fact that the component-wise product is invariant by permutation, $(\boldsymbol{c}_1 \star \boldsymbol{c}_2)^{\pi} = \boldsymbol{c}_1^{\pi} \star \boldsymbol{c}_2^{\pi}$.

Let $\boldsymbol{y} \in (\mathscr{C}^2)^\pi$, then we have that $\boldsymbol{y} = \left(\sum_{(i,j)} \alpha_{i,j} \boldsymbol{c}_i \star \boldsymbol{c}_j\right)^\pi$ with $\boldsymbol{c}_i$ and $\boldsymbol{c}_j$ codewords in $\mathscr{C}$ and $\alpha_{i,j} \in \mathbb{F}_2$. Using the previous property of the component-wise product we have that $\boldsymbol{y} = \sum_{(i,j)} \alpha_{i,j} \boldsymbol{c}_i^\pi \star \boldsymbol{c}_j^\pi \in (\mathscr{C}^\pi)^2$. For the second inclusion we use the dimension argument and obtain the wanted result.

□

Corollary A.0.2. *Let $\mathscr{C}$ be a binary linear code $[n, k, d]$ and $\mathsf{Perm}(\mathscr{C})$ be its permutation group. Then we have that*

$$\mathsf{Perm}(\mathscr{C}) = \mathsf{Perm}\left(\mathscr{C}^\perp\right). \tag{A.5}$$

$$\mathsf{Perm}(\mathscr{C}) \subseteq \mathsf{Perm}(\mathcal{H}(\mathscr{C})). \tag{A.6}$$

$$\mathsf{Perm}(\mathscr{C}) \subseteq \mathsf{Perm}\left(\mathscr{C}^2\right). \tag{A.7}$$

Proof. The proof for all the three $\subseteq$ inclusions works exactly the same. Let $\pi \in \mathsf{Perm}(\mathscr{C})$ be an element of the permutation group of $\mathscr{C}$. Then we have by Proposition A.0.1 that

$$\begin{aligned}
\left(\mathscr{C}^\perp\right)^\pi &= (\mathscr{C}^\pi)^\perp = (\mathscr{C})^\perp &&\Rightarrow \pi \in \mathsf{Perm}\left(\mathscr{C}^\perp\right)\\
(\mathcal{H}(\mathscr{C}))^\pi &= \mathcal{H}(\mathscr{C}^\pi) = \mathcal{H}(\mathscr{C}) &&\Rightarrow \pi \in \mathsf{Perm}(\mathcal{H}(\mathscr{C}))\\
\left(\mathscr{C}^2\right)^\pi &= (\mathscr{C}^\pi)^2 = (\mathscr{C})^2 &&\Rightarrow \pi \in \mathsf{Perm}\left(\mathscr{C}^2\right)
\end{aligned}$$

In order to prove Equation (A.5) we choose an element $\pi \in \mathsf{Perm}\left(\mathscr{C}^\perp\right)$ and obtain

$$\mathscr{C}^\perp = \left(\mathscr{C}^\perp\right)^\pi \overset{(A.1)}{=} (\mathscr{C}^\pi)^\perp \Rightarrow \mathscr{C} = (\mathscr{C}^\pi).$$

Hence we have that $\pi \in \mathsf{Perm}(\mathscr{C})$. □

Remark A.0.3.

- *If $\mathscr{C}$ is weakly self-dual then we have that $\mathsf{Perm}(\mathscr{C}) = \mathsf{Perm}(\mathcal{H}(\mathscr{C}))$ since $\mathcal{H}(\mathscr{C}) = \mathscr{C}$.*
- *There are code families for which $\mathsf{Perm}(\mathscr{C}) = \mathsf{Perm}(\mathcal{H}(\mathscr{C})) = \mathsf{Perm}(\mathscr{C}^2)$, for example the Reed-Muller codes $\mathscr{R}(r, m)$ with $r \leqslant \lfloor (m-2)/2 \rfloor$. (see 2.2.27)*

Proposition A.0.4. *Any binary linear code $\mathscr{C}$ of dimension 1 or 2 has $\mathsf{Perm}(\mathscr{C}) = \mathsf{Perm}(\mathscr{C}^2)$.*

Proof. For a binary linear code $\mathscr{C}$ of dimension 1 we have that $\mathscr{C} = \mathscr{C}^2$.

For a binary linear code $\mathscr{C}$ of dimension 2 we set $\mathscr{C} = \{\mathbf{0}_n, \boldsymbol{c}, \boldsymbol{b}, \boldsymbol{c} + \boldsymbol{b}\}$, with $\boldsymbol{b} \neq \boldsymbol{c}$. So the square code of $\mathscr{C}$ is $\mathscr{C}^2 = \{\mathbf{0}_n, \boldsymbol{c}, \boldsymbol{b}, \boldsymbol{c} + \boldsymbol{b}, \boldsymbol{c} \star \boldsymbol{b}, \boldsymbol{c} \star \boldsymbol{b} + \boldsymbol{b}, \boldsymbol{c} \star \boldsymbol{b} + \boldsymbol{c}, \boldsymbol{c} \star \boldsymbol{b} + \boldsymbol{c} + \boldsymbol{b}\}$. We notice that

Remark A.0.5. *If one of the following conditions is satisfied*

- $\mathbf{1}_n \in \mathscr{C}$
- $\boldsymbol{c} \star \boldsymbol{b} = \mathbf{0}_n$
- $\boldsymbol{c} \star \boldsymbol{b} = \mathbf{1}_n$
- $\boldsymbol{b} \star \boldsymbol{c} = \boldsymbol{b}$
- $\boldsymbol{b} \star \boldsymbol{c} = \boldsymbol{c}$

then $\mathscr{C} = \mathscr{C}^2$.

So let $\pi \in \mathsf{Perm}\left(\mathscr{C}^2\right)$, then we have the following possible cases:

- $\boldsymbol{c}^\pi \in \mathscr{C}$, fact that implies that $\pi \in \mathsf{Perm}\left(\mathscr{C}\right)$.
- $\boldsymbol{c}^\pi = \boldsymbol{c} \star \boldsymbol{b}$. Since π preserves the Hamming weight we have that $\mathsf{wt}(\boldsymbol{c}) = \mathsf{wt}(\boldsymbol{c} \star \boldsymbol{b})$. But this implies that either $\boldsymbol{b} = \mathbf{1}_n$ or $\boldsymbol{b} = \boldsymbol{c}$. For the former case we satisfy the first condition in A.0.5 and thus $\mathscr{C} = \mathscr{C}^2$ so $\pi \in \mathsf{Perm}\left(\mathscr{C}\right)$. As for the later case $\boldsymbol{b} = \boldsymbol{c}$ we have a contradiction since $\boldsymbol{b} \neq \boldsymbol{c}$.
- $\boldsymbol{c}^\pi = \boldsymbol{c} \star \boldsymbol{b} + \boldsymbol{c}$. Using the fact that π preserves the Hamming weight and Equation (2.2.9) we have that $\mathsf{wt}(\boldsymbol{c}) = \mathsf{wt}(\boldsymbol{c} \star \boldsymbol{b} + \boldsymbol{c}) \overset{(2.2.9)}{=} \mathsf{wt}(\boldsymbol{c}) + \mathsf{wt}(\boldsymbol{c} \star \boldsymbol{b}) - 2\mathsf{wt}(\boldsymbol{c} \star \boldsymbol{b} \star \boldsymbol{c})$. This equation implies that $\mathsf{wt}(\boldsymbol{c}) = \mathsf{wt}(\boldsymbol{c}) - \mathsf{wt}(\boldsymbol{c} \star \boldsymbol{b})$. But unless $\boldsymbol{c} \star \boldsymbol{b} = \mathbf{0}_n$ this equation is not possible, so we have by A.0.5 that $\mathscr{C} = \mathscr{C}^2$ and thus $\pi \in \mathsf{Perm}\left(\mathscr{C}\right)$.
- $\boldsymbol{c}^\pi = \boldsymbol{c} \star \boldsymbol{b} + \boldsymbol{c} + \boldsymbol{b}$. Since π preserves the Hamming weight and using Equation (2.2.9) we have

$$\begin{aligned}
\mathsf{wt}(\boldsymbol{c}) &= \mathsf{wt}(\boldsymbol{c} \star \boldsymbol{b} + \boldsymbol{c} + \boldsymbol{b}) \\
&= \mathsf{wt}(\boldsymbol{c} + \boldsymbol{b}) + \mathsf{wt}(\boldsymbol{c} \star \boldsymbol{b}) - 2\mathsf{wt}((\boldsymbol{b} + \boldsymbol{c}) \star (\boldsymbol{c} \star \boldsymbol{b})) \\
&= \mathsf{wt}(\boldsymbol{c} + \boldsymbol{b}) + \mathsf{wt}(\boldsymbol{c} \star \boldsymbol{b}) \\
&\overset{(2.2.9)}{=} \mathsf{wt}(\boldsymbol{c}) + \mathsf{wt}(\boldsymbol{b}) - 2\mathsf{wt}(\boldsymbol{c} \star \boldsymbol{b}) + \mathsf{wt}(\boldsymbol{c} \star \boldsymbol{b}) \\
&= \mathsf{wt}(\boldsymbol{c}) + \mathsf{wt}(\boldsymbol{b}) - \mathsf{wt}(\boldsymbol{c} \star \boldsymbol{b})
\end{aligned}$$

  This fact implies that $\mathsf{wt}(\boldsymbol{b}) = \mathsf{wt}(\boldsymbol{c} \star \boldsymbol{b})$. But this equality holds only if $\boldsymbol{b} = \boldsymbol{b} \star \boldsymbol{c}$ from which by A.0.5 we deduce as before that $\pi \in \mathsf{Perm}\left(\mathscr{C}\right)$.
- $\boldsymbol{c}^\pi = \boldsymbol{c} \star \boldsymbol{b} + \boldsymbol{b}$. Using Equation (2.2.9) we deduce that

$$\mathsf{wt}(\boldsymbol{c}^\pi) = \mathsf{wt}(\boldsymbol{b}) + \mathsf{wt}(\boldsymbol{c} \star \boldsymbol{b}) - 2\mathsf{wt}(\boldsymbol{c} \star \boldsymbol{b}),$$

  fact that implies

$$\mathsf{wt}(\boldsymbol{b}) = \mathsf{wt}(\boldsymbol{c}) + \mathsf{wt}(\boldsymbol{c} \star \boldsymbol{b}).$$

Now, based on this equation we obtain

$$\begin{aligned}\mathsf{wt}(\boldsymbol{b}+\boldsymbol{c}) &= 2\mathsf{wt}(\boldsymbol{c}) - \mathsf{wt}(\boldsymbol{c}\star\boldsymbol{b})\\ \mathsf{wt}(\boldsymbol{c}\star\boldsymbol{b}+\boldsymbol{b}) &= \mathsf{wt}(\boldsymbol{c})\\ \mathsf{wt}(\boldsymbol{c}\star\boldsymbol{b}+\boldsymbol{c}) &= \mathsf{wt}(\boldsymbol{c}) - \mathsf{wt}(\boldsymbol{c}\star\boldsymbol{b})\\ \mathsf{wt}(\boldsymbol{c}\star\boldsymbol{b}+\boldsymbol{b}+\boldsymbol{c}) &= 2\mathsf{wt}(\boldsymbol{c})\end{aligned}$$

If $\mathsf{wt}(\boldsymbol{c}\star\boldsymbol{b}) \in \{0, \mathsf{wt}(\boldsymbol{c})\}$ then we deduce as before that either $\boldsymbol{c}\star\boldsymbol{b} = \mathbf{0}_n$ or $\boldsymbol{c}\star\boldsymbol{b} = \boldsymbol{c}$ and by A.0.5 we have $\pi \in \mathsf{Perm}(\mathscr{C})$.

So we consider that $0 < \mathsf{wt}(\boldsymbol{c}\star\boldsymbol{b}) < \mathsf{wt}(\boldsymbol{c})$. Since the maximum weight is given by $\boldsymbol{c}+\boldsymbol{b}+\boldsymbol{c}\star\boldsymbol{b}$ and it is strictly bigger than all the other vector weight (because $0 < \mathsf{wt}(\boldsymbol{c}\star\boldsymbol{b}) < \mathsf{wt}(\boldsymbol{c})$) we have that

$$\begin{aligned}\boldsymbol{c}\star\boldsymbol{b}+\boldsymbol{c}+\boldsymbol{b} &= (\boldsymbol{c}\star\boldsymbol{b}+\boldsymbol{c}+\boldsymbol{b})^\pi\\ &= (\boldsymbol{c}\star\boldsymbol{b}+\boldsymbol{b})^\pi + \boldsymbol{c}^\pi\\ &= (\boldsymbol{c}\star\boldsymbol{b}+\boldsymbol{b})^\pi + \boldsymbol{c}\star\boldsymbol{b}+\boldsymbol{b}\end{aligned}$$

But this implies that $\boldsymbol{c} = (\boldsymbol{c}\star\boldsymbol{b}+\boldsymbol{b})^\pi$. So for the moment we have determined how π acts on $\boldsymbol{c}, \boldsymbol{c}\star\boldsymbol{b}+\boldsymbol{b}$ and $\boldsymbol{c}\star\boldsymbol{b}+\boldsymbol{b}+\boldsymbol{c}$. We continue and see what happens for the rest of the vectors and we distinguish three different possibilities

1. If $0 < \mathsf{wt}(\boldsymbol{c}\star\boldsymbol{b}) < \mathsf{wt}(\boldsymbol{c})/2$ we obtain

   $$\mathsf{wt}(\boldsymbol{c}\star\boldsymbol{b}) < \mathsf{wt}(\boldsymbol{c}\star\boldsymbol{b}+\boldsymbol{c}) < \mathsf{wt}(\boldsymbol{b}) < \mathsf{wt}(\boldsymbol{c}+\boldsymbol{b}).$$

   This implies that each of the vectors in the later inequality is mapped by π into itself. Therefore we have

   $$\begin{aligned}\boldsymbol{c}+\boldsymbol{b} &= (\boldsymbol{c}+\boldsymbol{b})^\pi\\ &= \boldsymbol{c}^\pi + \boldsymbol{b}^\pi\\ &= \boldsymbol{b}+\boldsymbol{b}+\boldsymbol{c}\star\boldsymbol{b}\\ &= \boldsymbol{c}\star\boldsymbol{b}.\end{aligned}$$

   Which is impossible unless $\boldsymbol{c} = \boldsymbol{b} = \mathbf{0}_n$, from which we obtain that $\mathscr{C} = \mathscr{C}^2$.
2. If $\mathsf{wt}(\boldsymbol{c})/2 < \mathsf{wt}(\boldsymbol{c}\star\boldsymbol{b}) < \mathsf{wt}(\boldsymbol{c})$ we obtain

   $$\mathsf{wt}(\boldsymbol{c}\star\boldsymbol{b}+\boldsymbol{c}) < \mathsf{wt}(\boldsymbol{c}\star\boldsymbol{b}) < \mathsf{wt}(\boldsymbol{c}+\boldsymbol{b}) < \mathsf{wt}(\boldsymbol{b}).$$

   This implies that each of the vectors in the later inequality is mapped by π into itself and we use the same argument as before to prove that $\mathscr{C} = \mathscr{C}^2$.
3. If $\mathsf{wt}(\boldsymbol{c}\star\boldsymbol{b}) = \mathsf{wt}(\boldsymbol{c})/2$ we obtain that $(\boldsymbol{c}\star\boldsymbol{b}+\boldsymbol{b})^\pi = \boldsymbol{c}\star\boldsymbol{b}+\boldsymbol{b}$. We also have that $\boldsymbol{b}+\boldsymbol{c}$ can be mapped by π into itself or into $\boldsymbol{b}$. We have seen before that if $(\boldsymbol{c}+\boldsymbol{b})^\pi = \boldsymbol{c}+\boldsymbol{b}$ and $\boldsymbol{b}^\pi = \boldsymbol{b}$ then we have that $\mathscr{C}^2 = \mathscr{C}$. So we can consider the remaining case, for which we have

   $$\begin{aligned}\boldsymbol{b}+\boldsymbol{c} &= (\boldsymbol{b}+\boldsymbol{c})^\pi\\ &= \boldsymbol{b}^\pi + \boldsymbol{c}^\pi\\ &= \boldsymbol{c}+\boldsymbol{b}+\boldsymbol{b}+\boldsymbol{c}\star\boldsymbol{b}\\ &= \boldsymbol{c}+\boldsymbol{c}\star\boldsymbol{b}\end{aligned}$$

Fact that implies $\boldsymbol{b} = \boldsymbol{c} \star \boldsymbol{b}$ which is impossible by the Hamming weight condition.

□

In the next paragraph we give an example of a binary linear code for which the permutation group of the code is different from both the permutation group of Hull and permutation group of the square code.

Example A.0.6. *Let $\mathscr{C}$ be a $[6, 3, 2]$ binary linear code defined by the generator matrix* $\begin{pmatrix} 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}$. *Then we have that* $\mathsf{Perm}(\mathscr{C}) = \{id; (3,4); (5,6); (3,4)(5,6)\}$.

We also have that $\mathscr{C}^{\perp}$ is a $[6, 3, 2]$ binary linear code defined by the generator matrix $\begin{pmatrix} 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 \end{pmatrix}$ *and has the same permutation group as $\mathscr{C}$.*

The $\mathcal{H}(\mathscr{C})$ is the $[6, 0]$ binary linear code. Indeed the only codeword that belong to $\mathscr{C}$ and $\mathscr{C}^{\perp}$ is the $\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$ *vector. Therefore the permutation group is the symmetric group* $\mathsf{Perm}(\mathcal{H}(\mathscr{C})) = \mathfrak{S}_6$.

The square code $\mathscr{C}^2$ is the $[6, 5, 1]$ binary linear code defined by the generator matrix $\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$. *The permutation group* $\mathsf{Perm}(\mathscr{C}^2)$ *is a group of order* 48 *generated by* $\{(1,2); (2,5); (5,6); (3,4)\}$.

B

Decoding Polar codes

Successive cancellation decoder. We consider the simplest case, namely $m = 1$. Suppose that a pair of bits (u_2, u_1) is encoded into a pair of bits (z_2, z_1) using W_1, in other words $(u_2, u_1) \to (z_2, z_1) = (u_1 \oplus u_2, u_1)$. Then each element is sent over a channel W to obtain a pair of symbols (y_1, y_2). Now given (y_1, y_2) we want to be able to decode, more exactly to recover (u_1, u_2).

The first step consists in computing $p_1 \stackrel{\text{def}}{=} \mathsf{prob}(z_1 = 1 \mid y_1)$ and $p_2 \stackrel{\text{def}}{=} \mathsf{prob}(z_2 = 1 \mid y_2)$, given (y_1, y_2) and the channel W. Remark that computing p_1 and $_2$ can be done by inverting the probability matrix corresponding to the channel W.

Secondly we detail for each synthetic channel the probability of recovering the initial bit

1. The first case (the channel W^-):

   Given p_1 and p_2 find $q_2 = \mathsf{prob}(u_2 = 1 \mid y_1, y_2)$.

   Lemma B.0.1.
   $$q_2 = \frac{1 - (1 - 2p_1)(1 - 2p_2)}{2}. \tag{B.1}$$

   Proof. Since $u_2 = z_1 \oplus z_2$ we obtain

   $$\mathsf{prob}(z_1 \oplus z_2 = 1) = \mathsf{prob}(z_1 = 1) + \mathsf{prob}(z_1 = 1) - 2\mathsf{prob}(z_1 = 1, z_2 = 1).$$

   As we know, the variables z_1 and z_2 are independent, therefore we obtain

   $$\begin{aligned}
   \mathsf{prob}(u_2 = 1 \mid y_1, y_2) &= \mathsf{prob}(z_1 \oplus z_2 = 1 \mid y_1, y_2) \\
   &= \mathsf{prob}(z_1 = 1 \mid y_1, y_2) + \mathsf{prob}(z_2 = 1 \mid y_1, y_2) \\
   &- 2\mathsf{prob}(z_1 = 1, z_2 = 1 \mid y_1, y_2) \\
   &= \mathsf{prob}(z_1 = 1 \mid y_1) + \mathsf{prob}(z_2 = 1 \mid y_2) \\
   &- 2\mathsf{prob}(z_1 = 1 \mid y_1)\mathsf{prob}(z_2 = 1 \mid y_2) \\
   &= p_1 + p_2 - 2p_1p_2 \\
   &= \frac{1 - (1 - 2p_1)(1 - 2p_2)}{2}.
   \end{aligned}$$

   □

2. The second case (the channel W^+):

   Given p_1, p_2 and u_2 find $q_1 = \mathsf{prob}(u_1 = 1 \mid y_1, y_2, u_2)$.

   Lemma B.0.2.

$$q_1 = \frac{p_1 p_2}{p_1 p_2 + (1-p_1)(1-p_2)} \quad if \quad u_2 = 0 \tag{B.2}$$

$$q_1 = \frac{p_1(1-p_2)}{(1-p_2)p_1 + p_2(1-p_1)} \quad if \quad u_2 = 1 \tag{B.3}$$

   Proof. Given $u_2 = 0$ we have that $z_1 = u_1$ and $z_2 = u_1$, in other words $z_1 = z_2 = u_1$. Since we search for the probability that $u_1 = 1$ we have $q_1 = \dfrac{p_1 p_2}{p_1 p_2 + (1-p_1)(1-p_2)}$ when $u_2 = 0$.

   The second case $u_2 = 1$ works the same. We have that $z_1 = u_1$ and $z_2 = u_1 + 1$, in other words $z_1 = z_2 + 1 = u_1$. Therefore the total number of possibilities is: either $z_1 = u_1 = 0, z_2 = 1$ or $z_1 = u_1 = 1, z_2 = 0$. Among them there is one good configuration for the probability $\mathsf{prob}(u_1 = 1 \mid y_1, y_2)$, namely when $z_1 = u_1 = 1, z_2 = 0$. Thus the probability $q_1 = \dfrac{p_1(1-p_2)}{(1-p_2)p_1 + p_2(1-p_1)}$ when $u_2 = 1$.

   □

Figure B.1 – The decoding algorithm over the two synthetic channels (left) $W^- : \mathbb{F}_2 \to \mathcal{Y} \times \mathcal{Y}$, (right) $W^+ : \mathbb{F}_2 \to \mathcal{Y} \times \mathcal{Y} \times \mathbb{F}_2$

We notice that for W^+ the value of u_2 is necessary, therefore the two synthetic channels are not independent. Hence the decoder works as follows:

1. Compute the probability q_2 corresponding to the channel W^-.
2. If $q_2 \geqslant 1/2$ then decode $u_2 = 1$ else decode $u_2 = 0$.
3. Use the value of u_2 to compute q_1 and decode u_1.

Example B.0.3.

- *Let $W = \mathrm{BEC}(p)$. Then W^- can be viewed as binary erasure channel. The SC decoder fails to decode u_2 only when at least one of the symbols z_1 or z_2 were erased. This fact arrives with probability $1 - (1-p)^2$. Thus the decoder recovers u_2 with probability $(1-p)^2$.*

  W^+ can also be viewed as a binary erasure channel. In this case the value of u_2 is provided to the decoder and therefore, the decoder fails to recover the value of u_1 when both of the symbols z_1 or z_2 were erased. This arrives with a probability p^2 and thus the decoder recovers u_1 with probability $1 - p^2$.

- *Let* $W = \mathrm{BSC}(p)$. W^- *corresponds to* $u_2 \to (y_1, y_2)$ *and if we denote the errors by* (e_1, e_2) *we have that* $(y_1, y_2) = (z_1 \oplus e_1, z_2 \oplus e_2)$. *Since* $z_1 \oplus z_2 = u_2$ *we state that* W^- *can be viewed as a* BSC *with* $u_2 \to u_2 \oplus e_1 \oplus e_2$. *Therefore we obtain*

$$\begin{aligned}\mathsf{prob}(u_2 = 1 | y_1, y_2) &= \mathsf{prob}(x_1 \oplus x_2 = 1 | y_1, y_2) \\ &\overset{(B.1)}{=} \frac{1 - (1-2p)^2}{2} \\ &= 2p(1-p).\end{aligned}$$

W^+ corresponds to $u_1 \to (y_1, y_2, u_2)$ where the value of u_2 is provided by the decoder of W^-. This implies that $(y_1, y_2 \oplus u_2) = (u_1 \oplus e_1, y_2 + u_2)$. But since we know the value of u_2 and that $y_2 = u_1 \oplus u_2 \oplus e_2$ we say that the channel W^+ outputs $(u_1 \oplus e_1, u_1 \oplus e_2)$, which is a BSC *of diversity 2. Here the probabilities are:*

$$\begin{aligned}\mathsf{prob}(u_1 = 1 | y_1, y_2, u_2) &\overset{(B.2)}{=} \frac{p^2}{p^2 + (1-p)^2} = \frac{p^2}{1 - 2p + 2p^2} \quad \text{if} \quad u_2 = 0 \\ \mathsf{prob}(u_1 = 1 | y_1, y_2, u_2) &\overset{(B.3)}{=} \frac{p(1-p)}{2p(1-p)} = 1/2 \quad \text{if} \quad u_2 = 1\end{aligned}$$

C

Factorization of $x^p - 1$

Notation C.0.1. *We denote for any $1 \leqslant \alpha \leqslant p-1$ the α-cyclotomic coset of the integer i mod p by $C_i^\alpha = \{i, i\alpha, \ldots, i\alpha^{\mathsf{ord}(\alpha)-1}\}$, where $\mathsf{ord}(\alpha)$ is the order of α in the multiplicative group $\mathbb{F}_p^*$.*

Theorem C.0.2 ([MS86, Chapter 7])**.** *Let $p>2$ be a prime number. The decomposition of $x^p - 1$ over $\mathbb{F}_2$ is*

$$x^p - 1 = (x-1)\prod_{i=1}^{d} g_i(x),$$

where $g_i(x)$ are irreducible polynomials over $\mathbb{F}_2[x]$, with $\deg g_i = \frac{p-1}{d}$ for all $i \in \{1, \ldots, d\}$ and

$$g_i(x) = \prod_{j \in C_{s_i}^2} (x - \theta^j),$$

with θ a primitive p^{th} root of unity and $s_i \neq 0$ runs through a set of coset representatives mod p.

The polynomials g_i in the factorization of $x^p - 1$ are known as the minimal polynomials of θ^{s_i}. The degree of each minimal polynomial g_i equals the order $\mathsf{ord}(2)$ in the multiplicative group ${\mathbb{F}_p}^*$. In general if we search for the factors of $x^p - 1$ over $\mathbb{F}_q[x]$ with q coprime with p the minimal polynomials are given by the q-cyclotomic cosets $C_{s_i}^q$, that is $g_i(x) = \prod_{j \in C_{s_i}^\alpha} (x - \theta^j)$.

Corollary C.0.3. *Let p be a prime number such that 2 is a primitive element of the multiplicative group ${\mathbb{F}_p}^*$. Then the factorization of $x^p - 1$ over $\mathbb{F}_2[x]$ is*

$$x^p - 1 = (x-1)(x^{p-1} + \cdots + x + 1).$$

D

Binomial Coefficient - Asymptotics

We begin by recalling the Stirling formula for factorials:

$$p! = \sqrt{2\pi p}\left(\frac{p}{e}\right)^p \left[1 + \frac{1}{12p} + \frac{1}{288p^2} + O(\frac{1}{p^3})\right].$$

and the main asymptotic expansion derived from it:

Proposition D.0.1. *When $k = \alpha p$ with α constant we have:*

$$\binom{p}{k} = \sqrt{\frac{1}{2\pi\alpha(1-\alpha)p}}\left(\alpha^\alpha(1-\alpha)^{1-\alpha}\right)^{-p}\left(1 - \frac{1}{12p}\left(\frac{1}{\alpha} + \frac{1}{1-\alpha} - 1\right) + O\left(\frac{1}{p^2}\right)\right).$$

or using the Entropy function:

$$\log\binom{p}{k} = -\frac{1}{2}\log 2\pi\alpha(1-\alpha)p + pH(\alpha) - \frac{1}{12p}\left(\frac{1}{\alpha} + \frac{1}{1-\alpha} - 1\right) + O\left(\frac{1}{p^2}\right).$$

Proof.

$$\begin{aligned}
\binom{p}{k} &= \frac{p!}{k!(p-k)!} \\
&= \sqrt{\frac{p}{2\pi k(p-k)}}\left(\frac{p}{e}\right)^p \left(\frac{e}{p-k}\right)^{p-k}\left(\frac{e}{k}\right)^k f(p,k) \\
&= \sqrt{\frac{1}{2\pi\alpha(1-\alpha)p}}\frac{p^p}{\alpha^{\alpha p}p^{\alpha p}\left((1-\alpha)\right)^{(1-\alpha)p}p^{(1-\alpha)p}} f(p,k) \\
&= \sqrt{\frac{1}{2\pi\alpha(1-\alpha)p}}\left(\alpha^\alpha(1-\alpha)^{1-\alpha}\right)^{-p} f(p,\alpha)
\end{aligned}$$

where

$$f(p,k) = \frac{1 + \frac{1}{12p} + \frac{1}{288p^2} + O(\frac{1}{p^3})}{\left(1 + \frac{1}{12(p-k)} + \frac{1}{288(p-k)^2} + O(\frac{1}{(p-k)^3})\right)\left(1 + \frac{1}{12k} + \frac{1}{288k^2} + O(\frac{1}{k^3})\right)}$$
$$= 1 - \frac{1}{12p}\left(\frac{1}{\alpha} + \frac{1}{1-\alpha} - 1\right) + O\left(\frac{1}{p^2}\right)$$

So

$$\binom{p}{k} = \sqrt{\frac{1}{2\pi\alpha(1-\alpha)p}}\left(\alpha^\alpha(1-\alpha)^{1-\alpha}\right)^{-p}\left(1 - \frac{1}{12p}\left(\frac{1}{\alpha} + \frac{1}{1-\alpha} - 1\right) + O\left(\frac{1}{p^2}\right)\right).$$

or using the Entropy function:

$$\log\binom{p}{k} = -\frac{1}{2}\log 2\pi\alpha(1-\alpha)p + pH(\alpha) - \frac{1}{12p}\left(\frac{1}{\alpha} + \frac{1}{1-\alpha} - 1\right) + O\left(\frac{1}{p^2}\right).$$

□

Corollary D.0.2. *When $\alpha = \frac{1}{2}$ we obtain the famous formula for the central coefficient:*

$$\binom{p}{\frac{p}{2}} = \sqrt{\frac{2}{\pi p}}2^p\left(1 - \frac{1}{4p} + O\left(\frac{1}{p^2}\right)\right).$$

In the second part we assume that $k = o(p)$ and we develop the formula for the binomial coefficient and replace the factorials using the Stirling formula:

Proposition D.0.3. *When $k = o(p)$ we have:*

$$\binom{p}{k} = \begin{cases} \frac{p^k}{k!}\left(1 + O(\frac{1}{p})\right) & \text{if } k = O(1) \\ \frac{p^k}{k!}e^{-c}\left(1 + O(\frac{1}{\sqrt{p}})\right) & \text{if } \frac{k^2}{2p} = c + O(\frac{1}{\sqrt{p}}) \\ \frac{p^k}{k!}p^{-c}\left(1 + O(\sqrt{\frac{\log^3 p}{p}})\right) & \text{if } \frac{k^2}{2p} = c\log p + O(\sqrt{\frac{\log p}{p}}) \end{cases}$$

Proof.

$$\binom{p}{k} = \frac{p!}{k!(p-k)!}$$
$$= \frac{1}{k!}\sqrt{\frac{2\pi p}{2\pi(p-k)}}\left(\frac{p}{e}\right)^p\left(\frac{e}{p-k}\right)^{p-k}\frac{1 + \frac{1}{12p} + \frac{1}{288p^2} + O(\frac{1}{p^3})}{1 + \frac{1}{12(p-k)} + \frac{1}{288(p-k)^2} + O(\frac{1}{(p-k)^3})}$$
$$= \frac{p^k}{k!}\sqrt{\frac{p}{(p-k)}}\left(1 + \frac{k}{p-k}\right)^{p-k}e^{-k}f(p,k)$$

We study each element apart and give the asymptotic expansion for several cases:

$$\begin{aligned} f(p,k) &= \frac{1+\frac{1}{12p}+o(\frac{1}{p})}{1+\frac{1}{12(p-k)}+o(\frac{1}{(p-k)})} \\ &= \left(1+\frac{1}{12p}+o(\frac{1}{p})\right)\left(1-\frac{1}{12(p-k)}+o(\frac{1}{p})\right) \\ &= 1-\frac{1}{12}\frac{k}{p^2}+o(\frac{1}{p}) \\ &= 1+o\left(\frac{1}{p}\right) \end{aligned}$$

Since

$$\sqrt{\frac{p}{p-k}} = 1+\frac{k}{2p}+o\left(\frac{k}{p}\right)$$

we obtain:

$$\sqrt{\frac{p}{p-k}}f(p,k) = 1+\frac{k}{2p}+o\left(\frac{k}{p}\right)+o\left(\frac{1}{p}\right) = 1+O\left(\frac{k}{p}\right)$$

For the exponential factor we have:

$$\begin{aligned} \left(1+\frac{k}{p-k}\right)^{p-k}e^{-k} &= e^{-k}e^{p(1-\frac{k}{p})\log\left(\frac{1}{1-\frac{k}{p}}\right)} \\ &= e^{-k}e^{p\left(\frac{k}{p}-\frac{k^2}{2p^2}-\frac{k^3}{6p^3}+o\left(\frac{k^3}{p^3}\right)\right)} \\ &= e^{-\frac{k^2}{2p}-\frac{k^3}{6p^2}+o\left(\frac{k^3}{p^2}\right)} \\ &= e^{-\frac{1}{2}\frac{k^2}{p}\left(1+\frac{1}{3}\frac{k}{p}+o\left(\frac{k}{p}\right)\right)} \end{aligned}$$

If $k = O(1)$ we have:

$$\begin{aligned} \binom{p}{k} &= \frac{p^k}{k!}e^{O(\frac{1}{p})}\left(1+O\left(\frac{1}{p}\right)\right) \\ &= \frac{p^k}{k!}\left(1+O\left(\frac{1}{p}\right)\right). \end{aligned}$$

If $k = \sqrt{2cp+O(\sqrt{p})}$ where $c>0$ is a constant, we have:

$$\begin{aligned} \binom{p}{k} &= \frac{p^k}{k!}e^{-(c+O(\frac{1}{\sqrt{p}}))(1+O(\frac{1}{\sqrt{p}}))}\left(1+O\left(\frac{1}{\sqrt{p}}\right)\right) \\ &= \frac{p^k}{k!}e^{-c}e^{O\left(\frac{1}{\sqrt{p}}\right)}\left(1+O\left(\frac{1}{\sqrt{p}}\right)\right) \\ &= \frac{p^k}{k!}e^{-c}\left(1+O\left(\frac{1}{\sqrt{p}}\right)\right). \end{aligned}$$

If $k = \sqrt{2cp\log p + O(\sqrt{p\log p})}$ where $c > 0$ is a constant, we have:

$$\begin{aligned}\binom{p}{k} &= \frac{p^k}{k!} e^{-(c\log p + O(\sqrt{\frac{\log p}{p}}))(1+O(\sqrt{\frac{\log p}{p}}))}\left(1 + O\left(\sqrt{\frac{\log p}{p}}\right)\right) \\ &= \frac{p^k}{k!} e^{-c\log p} e^{O\left(\sqrt{\frac{\log^3 p}{p}}\right)}\left(1 + O\left(\sqrt{\frac{\log p}{p}}\right)\right) \\ &= \frac{p^k}{k!} p^{-c}\left(1 + O\left(\sqrt{\frac{\log^3 p}{p}}\right)\right).\end{aligned}$$

□

The first consequence of this analysis is the quotient of two binomial coefficients:

Corollary D.0.4.
$$\frac{\binom{p}{k}}{\binom{2p}{k}} = \begin{cases} 2^{-k}\left(1 + O(\frac{1}{p})\right) & \text{if } k = O(1) \\ 2^{-k}e^{-\frac{c}{2}}\left(1 + O(\frac{1}{\sqrt{p}})\right) & \text{if } \frac{k^2}{2p} = c + O(\frac{1}{\sqrt{p}}) \\ 2^{-k}p^{-\frac{c}{2}}\left(1 + O(\sqrt{\frac{\log^3 p}{p}})\right) & \text{if } \frac{k^2}{2p} = c\log p + O(\sqrt{\frac{\log p}{p}}) \end{cases}$$

	p	100	500	1000	5000	10000
$k = \sqrt{2p}$	$\frac{\binom{p}{k}}{\binom{2p}{k}}$	$0.3357 * 10^{-4}$	$0.1835 * 10^{-9}$	$0.2091 * 10^{-13}$	$0.4784 * 10^{-30}$	$0.1624 * 10^{-42}$
	$2^{-k}e^{-\frac{1}{2}}$	$0.3354 * 10^{-4}$	$0.1834 * 10^{-9}$	$0.2091 * 10^{-13}$	$0.4784 * 10^{-30}$	$0.1624 * 10^{-42}$
$k = \sqrt{2p\log p}$	$\frac{\binom{p}{k}}{\binom{2p}{k}}$	$0.5267 * 10^{-10}$	$0.6631 * 10^{-25}$	$0.1087 * 10^{-36}$	$0.1769 * 10^{-89}$	$0.5764 * 10^{-131}$
	$2^{-k}p^{-\frac{1}{2}}$	$0.7314 * 10^{-10}$	$0.8307 * 10^{-25}$	$0.1309 * 10^{-36}$	$0.1982 * 10^{-89}$	$0.6309 * 10^{-131}$

Figure D.1 – Difference between the real value of $\binom{p}{k}/\binom{2p}{k}$ and the asymptotic approximation for $k = \sqrt{2p}$ and $k = \sqrt{2p\log p}$ when p ranges in the set $\{100, 500, 1000, 5000, 10000\}$.

Proposition D.0.5. *When $\omega_1 = O(1)$ and $\omega = o(p)$ we have:*

$$\frac{\binom{p}{\omega}}{\binom{p}{\omega_1}\binom{p}{\omega-\omega_1}} = \begin{cases} \frac{1}{\binom{\omega}{\omega_1}}\left(1+O(\frac{1}{p})\right) & \text{if } \omega = O(1) \\ \frac{1}{\binom{\omega}{\omega_1}}\left(1+O(\frac{1}{\sqrt{p}})\right) & \text{if } \frac{\omega^2}{2p} = c + O(\frac{1}{\sqrt{p}}) \quad \text{and } \omega_1 = O(1) \\ \frac{1}{\binom{\omega}{\omega_1}}\left(1+O(\sqrt{\frac{\log^3 p}{p}})\right) & \text{if } \frac{\omega^2}{2p} = c\log p + O(\sqrt{\frac{\log p}{p}}) \quad \text{and } \omega_1 = O(1) \end{cases}$$

	p	500	1000	5000	10000
$(\omega_1 = 1, \omega = \sqrt{2p})$	$\frac{\binom{p}{\omega}}{\binom{p}{\omega_1}\binom{p}{\omega-\omega_1}}$	0.029	0.0213	0.0098	$0.6971 * 10^{-2}$
	$\frac{1}{\binom{\omega}{\omega_1}}$	0.031	0.0223	0.010	$0.7071 * 10^{-2}$
$(\omega_1 = 3, \omega = \sqrt{2p})$	$\frac{\binom{p}{\omega}}{\binom{p}{\omega_1}\binom{p}{\omega-\omega_1}}$	$0.175 * 10^{-3}$	$0.631 * 10^{-4}$	$0.583 * 10^{-5}$	$0.2078 * 10^{-5}$
	$\frac{1}{\binom{\omega}{\omega_1}}$	$0.209 * 10^{-3}$	$0.718 * 10^{-4}$	$0.618 * 10^{-5}$	$0.2167 * 10^{-5}$

Figure D.2 – Difference between the real value of $\binom{p}{\omega}/\binom{p}{\omega_1}\binom{p}{\omega-\omega_1}$ and the asymptotic approximation for $(\omega_1 = 1, \omega = \sqrt{2p})$ and $(\omega_1 = 3, \omega = \sqrt{2p\log p})$ when p ranges in the set $\{100, 500, 1000, 5000\}$.

A natural question to ask is what happens when ω_1 is up to a constant close to ω. In this case we have

Proposition D.0.6. *When $\omega_1 = o(p)$ and $\omega = o(p)$ we have:*

$$\frac{\binom{p}{\omega}}{\binom{p}{\omega_1}\binom{p}{\omega-\omega_1}} = \begin{cases} \frac{1}{\binom{\omega}{\omega_1}}e^{-2\sqrt{cc_1}}\left(1+O(\frac{1}{\sqrt{p}})\right) & \text{if } \frac{\omega_1^2}{2p} = c_1 + O(\frac{1}{\sqrt{p}}) \quad ; \frac{(\omega-\omega_1)^2}{2p} = c + O(\frac{1}{\sqrt{p}}) \\ \frac{1}{\binom{\omega}{\omega_1}}p^{-2\sqrt{cc_1}}\left(1+O(\sqrt{\frac{\log^3 p}{p}})\right) & \text{if } \frac{\omega_1^2}{2p} = c_1\log p + O(\sqrt{\frac{\log p}{p}}) \quad ; \frac{(\omega-\omega_1)^2}{2p} = c\log p + O(\sqrt{\frac{\log p}{p}}) \end{cases}$$

Proof. First case $\frac{\omega_1^2}{2p} = c_1 + O(\frac{1}{\sqrt{p}})$ and $\frac{(\omega-\omega_1)^2}{2p} = c + O(\frac{1}{\sqrt{p}})$

$$\begin{aligned}
\omega &= \omega - \omega_1 + \omega_1 \\
&= \sqrt{2cp + O(\sqrt{p})} + \sqrt{2c_1 p + O(\sqrt{p})} \\
\frac{\omega^2}{2p} &= \frac{1}{2p}\left(2cp + 2c_1 p + O(\sqrt{p}) + 2\sqrt{2c_1 p\left(1 + O\left(\frac{1}{\sqrt{p}}\right)\right)}\sqrt{2cp\left(1 + O\left(\frac{1}{\sqrt{p}}\right)\right)}\right) \\
\frac{\omega^2}{2p} &= c + c_1 + O\left(\frac{1}{\sqrt{p}}\right) + 2\sqrt{cc_1}\left(1 + O\left(\frac{1}{\sqrt{p}}\right)\right) \\
\frac{\omega^2}{2p} &= c + c_1 + 2\sqrt{c_1 c} + O\left(\frac{1}{\sqrt{p}}\right)
\end{aligned}$$

$$\frac{\binom{p}{\omega}}{\binom{p}{\omega_1}\binom{p}{\omega-\omega_1}} = \frac{1}{\binom{\omega}{\omega_1}} e^{-2\sqrt{cc_1}}\left(1 + O\left(\frac{1}{\sqrt{p}}\right)\right).$$

Second case is when $\frac{(\omega-\omega_1)^2}{2p} = c\log p + O\left(\sqrt{\frac{\log p}{p}}\right)$ and $\frac{\omega_1^2}{2p} = c_1 \log p + O\left(\sqrt{\frac{\log p}{p}}\right)$.

$$\begin{aligned}
\omega &= \omega - \omega_1 + \omega_1 \\
&= \sqrt{2cp\log p + O(\sqrt{p\log p})} + \sqrt{2c_1 p\log p + O(\sqrt{p\log p})} \\
\frac{\omega^2}{2p} &= (c + c_1)\log p + O\left(\sqrt{\frac{\log p}{p}}\right) \\
&\quad + \frac{1}{p}\left(\sqrt{2c_1 p\log p\left(1 + O\left(\frac{1}{\sqrt{p\log p}}\right)\right)}\sqrt{2cp\log p\left(1 + O\left(\frac{1}{\sqrt{p\log p}}\right)\right)}\right) \\
\frac{\omega^2}{2p} &= (c + c_1)\log p + O\left(\sqrt{\frac{\log p}{p}}\right) + 2\sqrt{cc_1}\log p\left(1 + O\left(\frac{1}{\sqrt{p}}\right)\right) \\
\frac{\omega^2}{2p} &= (c + c_1 + 2\sqrt{c_1 c})\log p + O\left(\sqrt{\frac{\log^2 p}{p}}\right)
\end{aligned}$$

So

$$\frac{\binom{p}{\omega}}{\binom{p}{\omega_1}\binom{p}{\omega-\omega_1}} = \frac{1}{\binom{\omega}{\omega_1}} p^{-2\sqrt{cc_1}}\left(1 + O\left(\sqrt{\frac{\log^3 p}{p}}\right)\right).$$

□

With these results at hand we can prove all the asymptotic facts related to the binomial coefficient. We begin with Corollary 5.4.11 that states the following

Corollary. *5.4.11 Let p be a prime number and ω an even integer with $1 < \omega < p$ and $H(\alpha) = -\alpha \log_2 \alpha - (1-\alpha)\log_2(1-\alpha)$ be the binary entropy function for any $0 \leqslant \alpha \leqslant 1$. Let $(\omega_1, \omega_2) \in \mathbb{N}^2$ be odd integers such that $\omega_1 + \omega_2 = \omega$ and $\omega = o(p)$. Then we have*

$$\frac{|\mathcal{W}_{\omega_1,\omega_2}|}{|\mathcal{P}_{\omega_1,\omega_2}|} = \sqrt{2\pi\alpha(1-\alpha)}\omega^{\frac{1}{2}}2^{-\omega H(\alpha)} \times \begin{cases} e^{-2\sqrt{c_1c_2}}\left(1+O(\frac{1}{\sqrt{p}})\right) & \text{if } \frac{\omega_i^2}{2p} = c_i + O(\frac{1}{\sqrt{p}}) \\ p^{-2\sqrt{c_1c_2}}\left(1+O(\sqrt{\frac{\log^3 p}{p}})\right) & \text{if } \frac{\omega_i^2}{2p} = c_i \log p + O(\sqrt{\frac{\log p}{p}}) \end{cases}$$

with $\alpha = \dfrac{1}{1+\sqrt{\frac{c_2}{c_1}}}$.

$$\frac{|\mathcal{W}_{\omega}|}{|\mathcal{P}_{\omega}|} = \omega 2^{-\omega} \times \begin{cases} e^{-\frac{c}{2}}\left(1+O(\frac{1}{\sqrt{p}})\right) & \text{if } \frac{\omega^2}{2p} = c + O(\frac{1}{\sqrt{p}}), \\ p^{-\frac{c}{2}}\left(1+O(\sqrt{\frac{\log^3 p}{p}})\right) & \text{if } \frac{\omega^2}{2p} = c\log p + O(\sqrt{\frac{\log p}{p}}). \end{cases}$$

Proof. In the first case $\dfrac{|\mathcal{W}_{\omega_1,\omega_2}|}{|\mathcal{P}_{\omega_1,\omega_2}|}$ replace $\binom{\omega}{\omega_1}$ in Proposition D.0.6 by its expansion detailed in Proposition D.0.1.

The second part is $\dfrac{|\mathcal{W}_{\omega}|}{|\mathcal{P}_{\omega}|}$. Here the proof is a simple application of Corollary D.0.4. □

The next analysis on our list is Corollary 5.4.12. Here we have to prove that

$$\frac{\binom{p+1}{\omega}}{\binom{p}{\omega/2}^2} \sim \begin{cases} \sqrt{\pi}p^{\frac{1}{4}}e^{-2}2^{\frac{1}{4}-2\sqrt{2p}} & \text{if } \omega = 2\sqrt{2p} \\ \sqrt{\pi}p^{\frac{1}{4}-2}\log^{\frac{1}{4}} p 2^{\frac{1}{4}-2\sqrt{2p\log p}} & \text{if } \omega = 2\sqrt{2p\log p} \end{cases}$$

Proof. This is a particular case of Proposition D.0.6 when $c = c_1$.

So when $\omega = 2\sqrt{2p}$ we have:

$$\frac{\binom{p}{\omega}}{\binom{p}{\frac{\omega}{2}}\binom{p}{\frac{\omega}{2}}} \sim \sqrt{\pi}p^{\frac{1}{4}}e^{-2}2^{\frac{1}{4}-2\sqrt{2p}}.$$

and when $\omega = 2\sqrt{2p\log p}$ we have:

$$\frac{\binom{p}{\omega}}{\binom{p}{\frac{\omega}{2}}\binom{p}{\frac{\omega}{2}}} \sim \sqrt{\pi}p^{\frac{1}{4}-2}\log^{\frac{1}{4}} p 2^{\frac{1}{4}-2\sqrt{2p\log p}}.$$

□

	p	1000	5000	10000	20000
$\omega = 2\sqrt{2p}$	$\frac{\binom{p}{\omega}}{\binom{p}{\frac{\omega}{2}}^2}$	$0.1741 * 10^{-26}$	$0.1434 * 10^{-59}$	$0.1990 * 10^{-84}$	$0.1288 * 10^{-119}$
	$\frac{1}{e^2\binom{\omega}{\frac{\omega}{2}}}$	$0.1912 * 10^{-26}$	$0.1496 * 10^{-59}$	$0.2048 * 10^{-84}$	$0.1314 * 10^{-119}$
	$\sqrt{\pi}p^{\frac{1}{4}}e^{-2}2^{\frac{1}{4}-2\sqrt{2p}}$	$0.1906 * 10^{-26}$	$0.1492 * 10^{-59}$	$0.2046 * 10^{-84}$	$0.1313 * 10^{-119}$
$\omega = 2\sqrt{2n\log n}$	$\frac{\binom{n}{\omega}}{\binom{n}{\frac{\omega}{2}}^2}$	$0.1741 * 10^{-26}$	$0.1434 * 10^{-59}$	$0.1990 * 10^{-84}$	$0.1288 * 10^{-119}$
	$\frac{1}{n^2\binom{\omega}{\frac{\omega}{2}}}$	$0.1912 * 10^{-26}$	$0.1496 * 10^{-59}$	$0.2048 * 10^{-84}$	$0.1314 * 10^{-119}$
	$\sqrt{\pi}n^{\frac{1}{4}-2}\log^{\frac{1}{4}} n 2^{\frac{1}{4}-2\sqrt{2n\log n}}$	$0.1906 * 10^{-26}$	$0.1492 * 10^{-59}$	$0.2046 * 10^{-84}$	$0.1313 * 10^{-119}$

Figure D.3 – Difference between the real value of $\binom{p}{\omega}/\binom{p}{\omega/2}^2$ and the asymptotic approximations for $\omega = \sqrt{2p}$ and $\omega = 2\sqrt{2p\log p}$ when p ranges in the set $\{1000, 5000, 10000, 20000\}$.

List of Figures

2.1 Channel coding 2
2.2 (left) BSC(p) and (right) BEC(p) 4
2.3 BEC(δ) seen as a degradation of BEC(ε) for $0 \leqslant \varepsilon < \delta < 1$. 5
2.4 Key Generation of the original McEliece scheme - $\mathsf{KeyGen}(n, k, t) = (\mathsf{pk}, \mathsf{sk})$ 13
2.5 The Encryption function of the original McEliece scheme - $\mathsf{Encrypt}(\boldsymbol{m}, \mathsf{pk}) = \boldsymbol{z}$ 13
2.6 The Decryption function of the original McEliece scheme - $\mathsf{Decrypt}(\boldsymbol{z}, \mathsf{sk}) = \boldsymbol{m}$ 13
2.7 Parameters for McEliece with Goppa codes from [BLP08] 15

3.1 The channel combining for $m = 1$ 24
3.2 The combined circuit for $m = 3$ and the binary expansion of each row. The index order is used for the row numbers (Arıkan's notation) 24
3.3 The two synthetic channels (left) $W^- : \mathbb{F}_2 \to \mathcal{Y} \times \mathcal{Y}$, (right) $W^+ : \mathbb{F}_2 \to \mathcal{Y} \times \mathcal{Y} \times \mathbb{F}_2$ 25
3.4 The combined circuit for $m = 3$ with the monomial representation and the corresponding synthetic channels. The rows are in decreasing index order. 26
3.5 The Hasse diagram for the weakly decreasing monomial order when $m = 4$ 29
3.6 The Hasse diagram for the decreasing monomial order (left) $m = 3$ and (right) $m = 4$ 31
3.7 The Bhattacharyya parameter for all the synthetic channels when $m = 4$ 33
3.8 The Bhattacharyya parameter for all the synthetic channels when $m = 5$ 34
3.9 Young diagrams inside a 2×3 grid 51
3.10 Main properties of Monomial, Weakly Decreasing Monomial and Decreasing Monomial codes 55
3.11 The decreasing order over the Young diagrams corresponding to the monomials when $m = 4$ 56

4.1 Error probability for a $[2048, 614]$ Polar code in function of the error weight 60
4.2 Security level against the MRA of the $[2048, 614]$ Polar code in function of the error weight 60

5.1 Suggested parameters for the QC-MDPC scheme in [MTSB13] 78

5.2 Proportion of weak key for the $(2p, p, \omega)$ QC-MDPC when $\omega_1 = \omega_2 = \frac{\omega}{2}$. . 84
5.3 Proportion of weak keys for the $(9602, 4801, 90)$ QC-MDPC for $\omega_1 = 90 - \omega_2$ with $21 \leqslant \omega_1 \leqslant 45$. 84
5.4 Proportion of weak key and upper bound for the proportion of weak keys under the action of $(F_p)^2$ and $\mathbb{F}_p{}^*$ for the $(2p, p, \omega)$ QC-MDPC when $\omega_1 = \omega_2 = \frac{\omega}{2}$. 89
5.5 Proportion of weak key and weak orbits for the smooth $(2p, p, \omega)$ QC-MDPC 91
5.6 Proportion of weak key, weak orbits and extended weak orbits for the smooth $(2p, p, \omega)$ QC-MDPC 93
5.7 The Lyndon words for $p = 7$. The result from [Wit13] -$|L(7)|$, the result from [GR61] -$|L(7, \omega)|$ and our result Theorem 5.8.18 - $\left|L^k(7, \omega)\right|$ 101
5.8 Probability of a weak key (orbit) for the QC-MDPC when $\omega_1 = \omega_2 = \frac{\omega}{2}$. . . 111
5.9 Probability of a weak key, extended weak pairs and improvements on extended weak pairs for the QC-MDPC for all $\omega_1 + \omega_2 = \omega$. 111
5.10 Experimental timings for the first set of parameters at a 2^{80} security level for the QC-MDPC scheme. 115

B.1 The decoding algorithm over the two synthetic channels (left) $W^- : \mathbb{F}_{\mathbf{2}} \to \mathcal{Y} \times \mathcal{Y}$, (right) $W^+ : \mathbb{F}_{\mathbf{2}} \to \mathcal{Y} \times \mathcal{Y} \times \mathbb{F}_2$ 128

D.1 Difference between the real value of $\binom{p}{k}/\binom{2p}{k}$ and the asymptotic approximation for $k = \sqrt{2p}$ and $k = \sqrt{2p \log p}$ when p ranges in the set $\{100, 500, 1000, 5000, 10000\}$. 136
D.2 Difference between the real value of $\binom{p}{\omega}/\binom{p}{\omega_1}\binom{p}{\omega-\omega_1}$ and the asymptotic approximation for $(\omega_1 = 1, \omega = \sqrt{2p})$ and $(\omega_1 = 3, \omega = \sqrt{2p \log p})$ when p ranges in the set $\{100, 500, 1000, 5000\}$. 137
D.3 Difference between the real value of $\binom{p}{\omega}/\binom{p}{\omega/2}^2$ and the asymptotic approximations for $\omega = \sqrt{2p}$ and $\omega = 2\sqrt{2p \log p}$ when p ranges in the set $\{1000, 5000, 10000, 20000\}$. 140

Bibliography

[ABD+16] Carlos Aguilar, Olivier Blazy, Jean-Christophe Deneuville, Philippe Gaborit, and Gilles Zémor. Efficient encryption from random quasi-cyclic codes. *arXiv preprint arXiv:1612.05572*, 2016. (Cited on page 118.)

[AD97] Miklós Ajtai and Cynthia Dwork. A public-key cryptosystem with worst-case/average-case equivalence. In *Proceedings of the Twenty-ninth Annual ACM Symposium on Theory of Computing*, STOC '97, pages 284–293, New York, NY, USA, 1997. ACM. (Cited on page xii.)

[AHPT11] Roberto Avanzi, Simon Hoerder, Dan Page, and Michael Tunstall. Side-channel attacks on the McEliece and Niederreiter public-key cryptosystems. *J. Cryptographic Engineering*, 1(4):271–281, 2011. (Not cited.)

[Ajt96] M. Ajtai. Generating hard instances of lattice problems (extended abstract). In *Proceedings of the Twenty-eighth Annual ACM Symposium on Theory of Computing*, STOC '96, pages 99–108, New York, NY, USA, 1996. ACM. (Cited on page xii.)

[Ale11] Michael Alekhnovich. More on average case vs approximation complexity. *Computational Complexity*, 20(4):755–786, 2011. (Cited on page 118.)

[Ari08a] E. Arikan. Channel polarization: A method for constructing capacity-achieving codes. In *2008 IEEE International Symposium on Information Theory*, pages 1173–1177, July 2008. (Cited on page 26.)

[Arı08b] Erdal Arıkan. A performance comparison of polar codes and reed-muller codes. *IEEE Commun. Lett*, 12(6):447–449, 2008. (Cited on page 22.)

[Arı09] Erdal Arıkan. Channel polarization: a method for constructing capacity-achieving codes for symmetric binary-input memoryless channels. *IEEE Trans. Inform. Theory*, 55(7):3051–3073, 2009. (Cited on pages i, ii, xiii, 19, 20, 23, and 26.)

[Bal14] Marco Baldi. *QC-LDPC Code-Based Cryptography*. Springer Science & Business, 2014. (Cited on pages 16, 74, and 83.)

[BBC08] Marco Baldi, Marco Bodrato, and Franco Chiaraluce. A new analysis of the TMcEliece cryptosystem based on QC-LDPC codes. In *Proceedings of the 6th international conference on Security and Cryptography for Networks*, SCN '08, pages 246–262, Berlin, Heidelberg, 2008. Springer-Verlag. (Cited on page 16.)

[BBC+16] Marco Baldi, Marco Bianchi, Franco Chiaraluce, Joachim Rosenthal, and Davide Schipani. Enhanced public key security for the McEliece cryptosystem. *Journal of Cryptology*, 29(1):1–27, 2016. (Cited on page 15.)

[BBK08] Hacène Belbachir, Sadek Bouroubi, and Abdelkader Khelladi. Connection between ordinary multinomials, Fibonacci numbers, Bell polynomials and discrete uniform distribution. *Ann. Math. Inform.*, 35:21–30, 2008. (Cited on pages 100 and 103.)

[BC68] Benjamin Baumslag and Bruce Chandler. Schaum's outline of theory and problems of group theory. 1968. (Cited on page 106.)

[BC07] Marco Baldi and Franco Chiaraluce. Cryptanalysis of a new instance of McEliece cryptosystem based on QC-LDPC codes. In *Proc. IEEE Int. Symposium Inf. Theory - ISIT*, pages 2591–2595, Nice, France, June 2007. (Cited on page 16.)

[BCD+16] Magali Bardet, Julia Chaulet, Vlad Dragoi, Ayoub Otmani, and Jean-Pierre Tillich. Cryptanalysis of the McEliece public key cryptosystem based on polar codes. In *Post-Quantum Cryptography2016*, Lecture Notes in Comput. Sci., Fukuoka, Japan, February 2016. (Cited on pages i, ii, and 59.)

[BCN05] Frédérique Bassino, Julien Clément, and Cyril Nicaud. The standard factorization of lyndon words: an average point of view. *Discrete Mathematics*, 290(1):1 – 25, 2005. (Cited on page 96.)

[BCS13] Daniel J. Bernstein, Tung Chou, and Peter Schwabe. Mcbits: Fast constant-time code-based cryptography. In Guido Bertoni and Jean-Sébastien Coron, editors, *Cryptographic Hardware and Embedded Systems - CHES 2013*, volume 8086 of *Lecture Notes in Comput. Sci.*, pages 250–272. Springer, 2013. (Cited on page 1.)

[BDLO16] Magali Bardet, Vlad Dragoi, Jean-Gabriel Luque, and Ayoub Otmani. Weak keys for the quasi-cyclic MDPC public key encryption scheme. In *Progress in Cryptology - AFRICACRYPT 2016 - 8th International Conference on Cryptology in Africa, Fes, Morocco, April 13-15, 2016, Proceedings*, pages 346–367, 2016. (Cited on pages i, ii, and 72.)

[BDOT16] Magali Bardet, Vlad Dragoi, Ayoub Otmani, and Jean-Pierre Tillich. Algebraic properties of polar codes from a new polynomial formalism. In *IEEE International Symposium on Information Theory, ISIT 2016, Barcelona, Spain, July 10-15, 2016*, pages 230–234, 2016. (Cited on pages i, ii, and 20.)

[BGJT14] Razvan Barbulescu, Pierrick Gaudry, Antoine Joux, and Emmanuel Thomé. A heuristic quasi-polynomial algorithm for discrete logarithm in finite fields of small characteristic. In *Advances in Cryptology - EUROCRYPT 2014*, volume 8441 of *Lecture Notes in Comput. Sci.*, pages 1–16, Copenhagen, Denmark, May 2014. Springer. (Cited on page xi.)

[BL05] Thierry P. Berger and Pierre Loidreau. How to mask the structure of codes for a cryptographic use. *Des. Codes Cryptogr.*, 35(1):63–79, 2005. (Cited on page 15.)

[BLP08] Daniel J. Bernstein, Tanja Lange, and Christiane Peters. Attacking and defending the McEliece cryptosystem. In *Post-Quantum Cryptography 2008*, volume 5299 of *Lecture Notes in Comput. Sci.*, pages 31–46, 2008. (Cited on pages 15 and 141.)

[BLP10] Daniel J. Bernstein, Tanja Lange, and Christiane Peters. Wild McEliece. In Alex Biryukov, Guang Gong, and DouglasR. Stinson, editors, *Selected Areas in Cryptography*, volume 6544 of *Lecture Notes in Comput. Sci.*, pages 143–158, 2010. (Cited on page 16.)

[BLP11] Daniel J. Bernstein, Tanja Lange, and Christiane Peters. Wild McEliece Incognito. In Bo-Yin Yang, editor, *Post-Quantum Cryptography 2011*, volume 7071 of *Lecture Notes in Comput. Sci.*, pages 244–254. Springer Berlin Heidelberg, 2011. (Cited on page 16.)

[BMvT78] Elwyn Berlekamp, Robert McEliece, and Henk van Tilborg. On the inherent intractability of certain coding problems. *IEEE Trans. Inform. Theory*, 24(3):384–386, May 1978. (Cited on pages xii, 1, and 8.)

[BRC60] R. C. Bose and D. K. Ray-Chaudhuri. On a class of error correcting binary group codes. *Inform. and Control*, 3:68–79, 1960. (Cited on page 73.)

[BS08] Bhaskar Biswas and Nicolas Sendrier. McEliece cryptosystem implementation: theory and practice. In Johannes Buchmann and Jintai Ding, editors, *Post-Quantum Cryptography, Second International Workshop, PQCrypto 2008, Cincinnati, OH, USA, October 17-19, 2008, Proceedings*, volume 5299 of *Lecture Notes in Comput. Sci.*, pages 47–62. Springer, 2008. (Cited on page 1.)

[Car10a] Claude Carlet. Boolean functions for cryptography and error correcting codes. *Boolean models and methods in mathematics, computer science, and engineering*, 2:257, 2010. (Cited on page 21.)

[Car10b] Claude Carlet. Vectorial boolean functions for cryptography. *Boolean models and methods in mathematics, computer science, and engineering*, 134:398–469, 2010. (Cited on page 21.)

[CB13] Ivan Vladimirovich Chizhov and Mikhail Alekseevich Borodin. The failure of mceliece pkc based on reed-muller codes. *IACR Cryptology ePrint Archive*, 2013:287, 2013. (Cited on pages 16 and 69.)

[CB14] Ivan V. Chizhov and Mikhail A. Borodin. Effective attack on the McEliece cryptosystem based on Reed-Muller codes. *Discrete Math. Appl.*, 24(5):273–280, 2014. (Cited on page 16.)

[CCD+09] Eda Cesaratto, Julien Clément, Benoît Daireaux, Loïck Lhote, Véronique Maume-Deschamps, and Brigitte Vallée. Regularity of the Euclid Algorithm, Application to the analysis of fast GCD Algorithms. *Journal of Symbolic Computation*, 44(7):726, 2009. (Cited on page 112.)

[CGG+14] Alain Couvreur, Philippe Gaborit, Valérie Gauthier-Umaña, Ayoub Otmani, and Jean-Pierre Tillich. Distinguisher-based attacks on public-key cryptosystems using Reed-Solomon codes. *Des. Codes Cryptogr.*, 73(2):641–666, 2014. (Cited on page 15.)

[Cha09] Christophe Chabot. *Reconnaissance de codes, structure des codes quasi-cycliques*. PhD thesis, 2009. Thèse de doctorat dirigée par Berger, Thierry et Sendrier, Nicolas Mathématiques et applications Limoges 2009. (Cited on page 73.)

[Cho16] Tung Chou. Qcbits: constant-time small-key code-based cryptography. In *International Conference on Cryptographic Hardware and Embedded Systems*, pages 280–300. Springer, 2016. (Cited on page 71.)

[CHP12] Pierre-Louis Cayrel, Gerhard Hoffmann, and Eduardo Persichetti. Efficient implementation of a CCA2-secure variant of McEliece using generalized Srivastava codes. In *Public-Key Cryptography - PKC 2012*, volume 7293 of *Lecture Notes in Comput. Sci.*, pages 138–155. Springer, 2012. (Cited on page 1.)

[CMCP14] Alain Couvreur, Irene Márquez-Corbella, and Ruud Pellikaan. A polynomial time attack against algebraic geometry code based public key cryptosystems. In *Proc. IEEE Int. Symposium Inf. Theory - ISIT 2014*, pages 1446–1450, June 2014. (Cited on page 16.)

[Com12] Louis Comtet. *Advanced Combinatorics: The art of finite and infinite expansions*. Springer Science & Business Media, 2012. (Cited on page 51.)

[COT14a] Alain Couvreur, Ayoub Otmani, and Jean-Pierre Tillich. New identities relating wild Goppa codes. *Finite Fields Appl.*, 29:178–197, 2014. (Cited on page 16.)

[COT14b] Alain Couvreur, Ayoub Otmani, and Jean-Pierre Tillich. Polynomial time attack on wild McEliece over quadratic extensions. In Phong Q. Nguyen and Elisabeth Oswald, editors, *Advances in Cryptology - EUROCRYPT 2014*, volume 8441 of *Lecture Notes in Comput. Sci.*, pages 17–39. Springer Berlin Heidelberg, 2014. (Cited on page 16.)

[CS16] Julia Chaulet and Nicolas Sendrier. Worst case QC-MDPC decoder for McEliece cryptosystem. In *IEEE International Symposium on Information Theory, ISIT 2016, Barcelona, Spain, July 10-15, 2016*, pages 1366–1370, 2016. (Cited on pages 71 and 76.)

[Dav79] P.J. Davis. *Circulant matrices*. Pure and applied mathematics. Wiley, 1979. (Cited on page 74.)

[DH76] Whitfield Diffie and Martin Hellman. New directions in cryptography. *IEEE Trans. Inform. Theory*, 22(6):644–654, November 1976. (Cited on page xi.)

[DMN12] Nico Döttling, Jörn Müller-Quade, and Anderson C. A. Nascimento. IND-CCA secure cryptography based on a variant of the LPN problem. In *Advances in Cryptology - ASIACRYPT 2012*, volume 7658 of *Lecture Notes in Comput. Sci.*, pages 485–503, Beijing, China, 2012. Springer. (Cited on page 118.)

[Duv83] Jean Pierre Duval. Factorizing words over an ordered alphabet. *Journal of Algorithms*, 4(4):363 – 381, 1983. (Cited on page 96.)

[DV13] Alexandre Duc and Serge Vaudenay. HELEN: A public-key cryptosystem based on the LPN and the decisional minimal distance problems. In *Progress in Cryptology - AFRICACRYPT 2013*, volume 7918 of *Lecture Notes in Comput. Sci.*, pages 107–126. Springer, 2013. (Cited on page 118.)

[Fel68] William Feller. *An introduction to probability theory and its applications. Vol. I.* Third edition. John Wiley & Sons, Inc., New York-London-Sydney, 1968. (Cited on page 96.)

[FGO+10] Jean-Charles Faugère, Valérie Gauthier, Ayoub Otmani, Ludovic Perret, and Jean-Pierre Tillich. A distinguisher for high rate McEliece cryptosystems. IACR Cryptology ePrint Archive, Report2010/331, 2010. http://eprint.iacr.org/. (Cited on pages 1, 14, and 15.)

[FL08] Pierre-Alain Fouque and Gaëtan Leurent. Cryptanalysis of a hash function based on quasi-cyclic codes. In *Topics in Cryptology - CT-RSA 2008, The Cryptographers' Track at the RSA Conference 2008, San Francisco, CA, USA, April 8-11, 2008. Proceedings*, volume 4964 of *Lecture Notes in Comput. Sci.*, pages 19–35. Springer, 2008. (Cited on page 77.)

[FM08] Cédric Faure and Lorenz Minder. Cryptanalysis of the McEliece cryptosystem over hyperelliptic curves. In *Proceedings of the eleventh International Workshop on Algebraic and Combinatorial Coding Theory*, pages 99–107, Pamporovo, Bulgaria, June 2008. (Cited on page 16.)

[FOP+14] Jean-Charles Faugère, Ayoub Otmani, Ludovic Perret, Frédéric de Portzamparc, and Jean-Pierre Tillich. Folding alternant and Goppa codes with non-trivial automorphism groups. to appear in the Transactions on Information Theor, 2014. arXiv:1405.5101 [cs.IT]. (Cited on pages 16 and 77.)

[FS09] Philippe Flajolet and Robert Sedgewick. *Analytic combinatorics*. Cambridge University Press, Cambridge, 2009. (Cited on pages 96 and 97.)

[Gab05] Philippe Gaborit. Shorter keys for code based cryptography. In *Proceedings of the 2005 International Workshop on Coding and Cryptography (WCC 2005)*, pages 81–91, Bergen, Norway, March 2005. (Cited on page 16.)

[Gal63] R. G. Gallager. *Low Density Parity Check Codes*. M.I.T. Press, Cambridge, Massachusetts, 1963. (Cited on pages 75 and 76.)

[Gen01] Craig Gentry. Key recovery and message attacks on NTRU-composite. In Birgit Pfitzmann, editor, *Advances in Cryptology - EUROCRYPT 2001*, volume 2045 of *Lecture Notes in Comput. Sci.*, pages 182–194, 2001. (Cited on page 77.)

[GGH97] Oded Goldreich, Shafi Goldwasser, and Shai Halevi. Public-key cryptosystems from lattice reduction problems. In *Annual International Cryptology Conference*, pages 112–131. Springer, 1997. (Cited on page xii.)

[GJS16] Qian Guo, Thomas Johansson, and Paul Stankovski. A key recovery attack on mdpc with cca security using decoding errors. In *Advances in Cryptology–ASIACRYPT 2016: 22nd International Conference on the Theory and Application of Cryptology and Information Security, Hanoi, Vietnam, December 4-8, 2016, Proceedings, Part I 22*, pages 789–815. Springer, 2016. (Cited on page 71.)

[GM13] Cheikh Thiecoumba Gueye and El Hadji Modou Mboup. Secure cryptographic scheme based on modified Reed Muller codes. *International Journal of Security and Its Applications*, 7(3):55–64, 2013. (Cited on page 16.)

[GMRZ13] Philippe Gaborit, Gaétan Murat, Olivier Ruatta, and Gilles Zémor. Low rank parity check codes and their application to cryptography. In *Proceedings of the Workshop on Coding and Cryptography WCC'2013*, Bergen, Norway, 2013. Available on `www.selmer.uib.no/WCC2013/pdfs/Gaborit.pdf`. (Cited on page xii.)

[Gop70] Valerii Denisovich Goppa. A new class of linear correcting codes. *Problemy Peredachi Informatsii*, 6(3):24–30, 1970. (Cited on page 11.)

[Gou72] H.W. Gould. *Combinatorial identities: a standardized set of tables listing 500 binomial coefficient summations*. Morgantown, W Va, 1972. (Cited on pages 81, 104, and 114.)

[GR61] E. N. Gilbert and John Riordan. Symmetry types of periodic sequences. *Illinois J. Math.*, 5:657–665, 1961. (Cited on pages 96, 100, 101, and 142.)

[GSW86] Louis Gordon, Mark F. Schilling, and Michael S. Waterman. An extreme value theory for long head runs. *Probab. Theory Relat. Fields*, 72(2):279–287, 1986. (Cited on page 96.)

[GZ14] Philippe Gaborit and Gilles Zémor. On the hardness of the decoding and the minimum distance problems for rank codes. *CoRR*, abs/1404.3482, 2014. (Cited on page xii.)

[Hoc59] A. Hocquenghem. Codes correcteurs d'erreurs. *Chiffres*, 2:147–158, 1959. (Cited on page 73.)

[Hor19] W.G. Horner. A new method of solving numerical equations of all orders,by continuous approximation. *Philosophical transactions, of the Royal Society of London*, 109:308–335, 1819. (Not cited.)

[HPS98] Jeffrey Hoffstein, Jill Pipher, and Joseph H. Silverman. NTRU: A ring-based public key cryptosystem. In Joe Buhler, editor, *Algorithmic Number Theory, Third International Symposium, ANTS-III, Portland, Oregon, USA, June 21-25, 1998, Proceedings*, volume 1423 of *Lecture Notes in Comput. Sci.*, pages 267–288. Springer, 1998. (Cited on pages xii and 116.)

[HSA13] Reza Hooshmand, M Koochak Shooshtari, and Mohammad Reza Aref. Secret key cryptosystem based on polar codes over binary erasure channel. In *2013 10th International ISC Conference on Information Security and Cryptology (ISCISC)*, pages 1–6. IEEE, 2013. (Cited on page 59.)

[HSEA14] R Hooshmand, M Koochak Shooshtari, T Eghlidos, and MR Aref. Reducing the key length of McEliece cryptosystem using polar codes. In *2014 11th International ISC Conference on Information Security and Cryptology (ISCISC)*, pages 104–108. IEEE, 2014. (Cited on pages 17, 19, 59, and 69.)

[Huf98] W Cary Huffman. Codes and groups. *Handbook of coding theory*, 2(Part 2):1345–1440, 1998. (Cited on page 10.)

[HvMG13] Stefan Heyse, Ingo von Maurich, and Tim Güneysu. Smaller keys for code-based cryptography: QC-MDPC McEliece implementations on embedded devices. In Guido Bertoni and Jean-Sébastien Coron, editors, *Cryptographic Hardware and Embedded Systems - CHES 2013*, volume 8086 of *Lecture Notes in Comput. Sci.*, pages 273–292. Springer, 2013. (Cited on pages 1 and 71.)

[ISR60] G. Solomon I. S. Reed. Polynomial codes over certain finite fields. *Journal of the Society for Industrial and Applied Mathematics*, 8(2):300–304, 1960. (Cited on pages 11 and 73.)

[JM96] Heeralal Janwa and Oscar Moreno. McEliece public key cryptosystems using algebraic-geometric codes. *Des. Codes Cryptogr.*, 8(3):293–307, 1996. (Cited on page 16.)

[KMP14] Eike Kiltz, Daniel Masny, and Krzysztof Pietrzak. Simple chosen-ciphertext security from low-noise lpn. In *International Workshop on Public Key Cryptography*, pages 1–18. Springer, 2014. (Cited on page 118.)

[Knu71a] Donald E Knuth. Subspaces, subsets, and partitions. *Journal of Combinatorial Theory, Series A*, 10(2):178–180, 1971. (Cited on pages 51 and 55.)

[Knu71b] Donald E. Knuth. The analysis of algorithms. In *Actes Congr. internat. Math. 1970, 3, 269-274 (1971).*, pages 269–274, 1971. URL: `http://cr.yp.to/bib/entries.html#1971/knuth-gcd`. (Cited on page 111.)

[Kob09] Kazukuni Kobara. Code-based public-key cryptosystems and their applications. In *Information Theoretic Security, 4th International Conference, ICITS*, volume 5973 of *Lecture Notes in Comput. Sci.*, pages 45–55, Shizuoka, Japan, December 2009. Springer. (Cited on page 35.)

[Kor09] Satish Babu Korada. *Polar Codes for Channel and Source Coding*. PhD thesis, 'Ecole Polytechnique Fédérale de Lausanne (EPFL), July 2009. (Cited on page 45.)

[KT70] Tadao Kasami and Nobuki Tokura. On the weight structure of Reed-Muller codes. *Information Theory, IEEE Transactions on*, 16(6):752–759, 1970. (Cited on page 11.)

[KTA76] Tadao Kasami, Nobuki Tokura, and Saburo Azumi. On the weight enumeration of weights less than $2.5d$ of Reed-Muller codes. *Information and Control*, 30(4):380–395, 1976. (Cited on page 53.)

[KTC58] R. C. Lyndon K. T. Chen, R. H. Fox. Free differential calculus, iv. the quotient groups of the lower central series. *Annals of Mathematics*, 68(1):81–95, 1958. (Cited on page 96.)

[Lan09] Edmund Landau. *Handbuch der Lehre von der Verteilung der Primzahlen*. Teubner, 1909. (Cited on page 98.)

[Leh38] D. H. Lehmer. Euclid's algorithm for large numbers. *American Mathematical Monthly*, 45:227–233, 1938. (Cited on page 111.)

[LJ12] Carl Löndahl and Thomas Johansson. A new version of McEliece PKC based on convolutional codes. In *Information and Communications Security, ICICS*, volume 7168 of *Lecture Notes in Comput. Sci.*, pages 461–470. Springer, 2012. (Cited on page 17.)

[LJS+16] Carl Löndahl, Thomas Johansson, Masoumeh Koochak Shooshtari, Mahmoud Ahmadian-Attari, and Mohammad Reza Aref. Squaring attacks on mceliece public-key cryptosystems using quasi-cyclic codes of even dimension. *Designs, Codes and Cryptography*, 80(2):359–377, 2016. (Cited on page 77.)

[Loi01] Pierre Loidreau. Codes derived from binary Goppa codes. *Probl. Inf. Transm.*, 2001. (Cited on page 77.)

[Lot02] M. Lothaire. *Algebraic combinatorics on words*. Encyclopedia of mathematics and its applications. Cambridge university press, New York, 2002. (Cited on pages 95, 96, and 103.)

[LPR10] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On ideal lattices and learning with errors over rings. In *Advances in Cryptology - EUROCRYPT2010*, volume 6110 of *Lecture Notes in Comput. Sci.*, pages 1–23. Springer, 2010. (Cited on page xii.)

[LS01] Pierre Loidreau and Nicolas Sendrier. Weak keys in the McEliece public-key cryptosystem. *IEEE Trans. Inform. Theory*, 47(3):1207–1211, 2001. (Cited on page 15.)

[LS05] San Ling and Patrick Solé. On the algebraic structure of quasi-cyclic codes III: generator theory. *IEEE Trans. Information Theory*, 51(7):2692–2700, 2005. (Cited on pages 73 and 75.)

[LT13] Grégory Landais and Jean-Pierre Tillich. An efficient attack of a McEliece cryptosystem variant based on convolutional codes. In P. Gaborit, editor, *Post-Quantum Cryptography'13*, volume 7932 of *Lecture Notes in Comput. Sci.*, pages 102–117. Springer, June 2013. (Cited on page 17.)

[LV08] Loïck Lhote and Brigitte Vallée. Gaussian Laws for the Main Parameters of the Euclid Algorithms. *Algorithmica*, 50(4):497–554, 2008. (Cited on page 112.)

[Mac99] David JC MacKay. Good error-correcting codes based on very sparse matrices. *Information Theory, IEEE Transactions on*, 45(2):399–431, 1999. (Cited on page 75.)

[MB09] Rafael Misoczki and Paulo Barreto. Compact McEliece keys from Goppa codes. In *Selected Areas in Cryptography*, Calgary, Canada, August 13-14 2009. (Cited on page 15.)

[McE78] Robert J. McEliece. *A Public-Key System Based on Algebraic Coding Theory*, pages 114–116. Jet Propulsion Lab, 1978. DSN Progress Report 44. (Cited on pages xii, 1, 12, and 15.)

[Mil85] Victor S Miller. Use of elliptic curves in cryptography. In *Conference on the Theory and Application of Cryptographic Techniques*, pages 417–426. Springer, 1985. (Cited on page xi.)

[Min07] Lorenz Minder. *Cryptography based on error correcting codes*. PhD thesis, Ecole Polytechnique Fédérale de Lausanne, 2007. (Cited on page 64.)

[Mit51] N Mitani. On the transmission of numbers in a sequential computer. In *National Convention of the Institute of Electrical Communication Engineers of Japan, November*, 1951. (Cited on page 10.)

[Möb32] A.F. Möbius. Über eine besondere art von umkehrung der reihen. *Journal für die reine und angewandte Mathematik*, 9:105–123, 1832. (Cited on page 98.)

[MOG15] Ingo Von Maurich, Tobias Oder, and Tim Güneysu. Implementing QC-MDPC McEliece encryption. *ACM Trans. Embed. Comput. Syst.*, 14(3):44:1–44:27, April 2015. (Cited on pages 1 and 71.)

[MRAS00] Chris Monico, Joachim Rosenthal, and Amin A. Shokrollahi. Using low density parity check codes in the McEliece cryptosystem. In *Proc. IEEE Int. Symposium Inf. Theory - ISIT*, page 215, Sorrento, Italy, 2000. (Cited on page 16.)

[MS86] Florence J. MacWilliams and Neil J. A. Sloane. *The Theory of Error-Correcting Codes*. North–Holland, Amsterdam, fifth edition, 1986. (Cited on pages 4, 11, 73, 74, 84, and 131.)

[MS07] Lorenz Minder and Amin Shokrollahi. Cryptanalysis of the Sidelnikov cryptosystem. In *Advances in Cryptology - EUROCRYPT 2007*, volume 4515 of *Lecture Notes in Comput. Sci.*, pages 347–360, Barcelona, Spain, 2007. (Cited on page 15.)

[MT09a] R. Mori and T. Tanaka. Performance of polar codes with the construction using density evolution. *IEEE Communications Letters*, 13(7):519–521, July 2009. (Cited on page 26.)

[MT09b] Ryuhei Mori and Toshiyuki Tanaka. Performance and construction of polar codes on symmetric binary-input memoryless channels. In *Information Theory, 2009. ISIT 2009. IEEE International Symposium on*, pages 1496–1500. IEEE, 2009. (Cited on page 27.)

[MTSB13] Rafael Misoczki, Jean-Pierre Tillich, Nicolas Sendrier, and Paulo S. L. M. Barreto. MDPC-McEliece: New McEliece variants from moderate density parity-check codes. In *Proc. IEEE Int. Symposium Inf. Theory - ISIT*, pages 2069–2073, 2013. (Cited on pages i, xiv, xv, 17, 71, 74, 76, 77, 78, 82, 83, 88, 110, 112, and 141.)

[Mul54] D. E. Muller. Application of boolean algebra to switching circuit design and to error detection. *Transactions of the I.R.E. Professional Group on Electronic Computers*, EC-3(3):6–12, Sept 1954. (Cited on page 10.)

[MV15] Hessam Mahdavifar and Alexander Vardy. Explicit capacity achieving codes for defective memories. In *IEEE International Symposium on Information Theory, ISIT 2015, Hong Kong, China, June 14-19, 2015*, pages 641–645, 2015. (Cited on page 41.)

[Nie86] Harald Niederreiter. Knapsack-type cryptosystems and algebraic coding theory. *Problems of Control and Information Theory*, 15(2):159–166, 1986. (Cited on page 15.)

[OB09] Samuel Ouzan and Yair Be'ery. Moderate-density parity-check codes. *arXiv preprint arXiv:0911.3262*, 2009. (Cited on pages 71 and 75.)

[OK15] Ayoub Otmani and Hervé Talé Kalachi. Square code attack on a modified sidelnikov cryptosystem. In *Codes, Cryptology, and Information Security*, pages 173–183. Springer, 2015. (Cited on page 16.)

[OS09] Raphael Overbeck and Nicolas Sendrier. Code-based cryptography. In Daniel J. Bernstein, Johannes Buchmann, and Erik Dahmen, editors, *Post-quantum cryptography*, pages 95–145. Springer, 2009. (Cited on page 1.)

[OT12] Ayoub Otmani and Jean-Pierre Tillich. On the Design of Code-Based Signatures. In *Code-based Cryptography Workshop (CBC 2012)*, Lyngby, Denmark, May 2012. (Cited on page 59.)

[OTD08] Ayoub Otmani, Jean-Pierre Tillich, and Léonard Dallot. Cryptanalysis of McEliece cryptosystem based on quasi-cyclic LDPC codes. In *Proceedings of First International Conference on Symbolic Computation and Cryptography*, pages 69–81, Beijing, China, April 28-30 2008. LMIB Beihang University. (Cited on page 16.)

[Pan12] Victor Pan. *Structured matrices and polynomials: unified superfast algorithms*. Springer Science & Business Media, 2012. (Cited on page 80.)

[Per05] Ludovic Perret. A chosen ciphertext attack on a public key cryptosystem based on lyndon words. Cryptology ePrint Archive, Report 2005/014, 2005. http://eprint.iacr.org/2005/014. (Cited on page 95.)

[Per12] Edoardo Persichetti. Compact McEliece keys based on quasi-dyadic Srivastava codes. *J. Math. Cryptol.*, 6(2):149–169, 2012. (Cited on page 16.)

[Per14] Ray A. Perlner. Optimizing information set decoding algorithms to attack cyclosymmetric MDPC codes. In *Post-Quantum Cryptography 2014*, volume 8772 of *Lecture Notes in Comput. Sci.*, pages 220–228. Springer, 2014. (Cited on page 71.)

[PR97] Erez Petrank and Ron. Roth. Is code equivalence easy to decide? *IEEE Trans. Inform. Theory*, 43(5):1602–1604, 1997. (Cited on page 14.)

[Pra57] E. Prange. *Cyclic error-correcting codes in two symbols*. Electronics Research Directorate, Air Force Cambridge Research Center, September 1957. No. AFCRC-TN-57-103. ASTIA Document No. AD133749. (Cited on page 73.)

[PW08] Robin Pemantle and Mark C. Wilson. Twenty combinatorial examples of asymptotics derived from multivariate generating functions. *SIAM Rev.*, 50(2):199–272, 2008. (Cited on page 96.)

[Ree54] I. S. Reed. A class of multiple-error-correcting codes and the decoding scheme. *IRE Trans.*, IT-4:38–49, 1954. (Cited on page 10.)

[Ric03] G. Richomme. Lyndon morphisms. *Bulletin of the Belgian Mathematical Society - Simon Stevin*, 10(5):761–785, 12 2003. (Cited on page 99.)

[Rin15] Revenko Rina. A new post-quantum cryptosystem based on newly discovered decoding algorithm of reed-muller codes. september 2015. http://itas2015.iitp.ru/pdf/03.pdf. (Cited on page 69.)

[Rot06] Ron M. Roth. *Introduction to Coding Theory*. Cambridge University Press, New York, NY, USA, 2006. (Cited on page 4.)

[RSA78] Ronald L. Rivest, Adi Shamir, and Leonard M. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM*, 21(2):120–126, 1978. (Cited on page xi.)

[RU08] Tom Richardson and Ruediger Urbanke. *Modern Coding Theory*. Cambridge University Press, 2008. (Cited on pages 4 and 35.)

[Ruf04] Paolo Ruffini. *Sopra la determinazione delle radici nelle equazioni numeriche di qualungue grado*. presso la societa'tipografica, 1804. (Not cited.)

[Sch71] A. Schönhage. Schnelle Berechnung von Kettenbruchentwicklungen. (German) [Fast calculation of expansions of continued fractions]. *ACTA-INFO*, 1:139–144, 1971. (Cited on page 111.)

[Sch16] C. Schürch. A partial order for the synthesized channels of a polar code. In *2016 IEEE International Symposium on Information Theory (ISIT)*, pages 220–224, July 2016. (Cited on page 32.)

[Sen94] Nicolas Sendrier. On the structure of a randomly permuted concatenated code. In *EUROCODE'94*, pages 169–173, 1994. (Cited on page 16.)

[Sen98] Nicolas Sendrier. On the concatenated structure of a linear code. *Appl. Algebra Eng. Commun. Comput. (AAECC)*, 9(3):221–242, 1998. (Cited on page 16.)

[Sen00] Nicolas Sendrier. Finding the permutation between equivalent linear codes: The support splitting algorithm. *IEEE Trans. Inform. Theory*, 46(4):1193–1203, 2000. (Cited on pages 14, 59, and 61.)

[Sen10] Nicolas Sendrier. On the use of structured codes in code based cryptography. In L. Storme S. Nikova, B. Preneel, editor, *Coding Theory and Cryptography III*, pages 59–68. The Royal Flemish Academy of Belgium for Science and the Arts, 2010. (Cited on page 71.)

[Sha48] C. E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27(3):379–423, 1948. (Cited on page 2.)

[Sho94] P.W. Shor. Algorithms for quantum computation: Discrete logarithms and factoring. In S. Goldwasser, editor, *FOCS*, pages 124–134, 1994. (Cited on page xi.)

[Sid94] Vladimir Michilovich Sidelnikov. A public-key cryptosytem based on Reed-Muller codes. *Discrete Math. Appl.*, 4(3):191–207, 1994. (Cited on pages 15 and 60.)

[SK14] Sujan Raj Shrestha and Young-Sik Kim. New McEliece cryptosystem based on polar codes as a candidate for post-quantum cryptography. In *2014 14th International Symposium on Communications and Information Technologies (ISCIT)*, pages 368–372. IEEE, 2014. (Cited on pages i, 17, 19, 59, and 60.)

[SLR+16] Alexandre Soro, Jérôme Lacan, Vincent Roca, Valentin Savin, and Mathieu Cunche. Enhanced Recursive Reed-Muller Erasure Decoding. In IEEE, editor, *IEEE International Symposium on Information Theory (ISIT)*, IEEE International Symposium on Information Theory (ISIT), Barcelona, Spain, July 2016. (Cited on page 48.)

[SS92] Vladimir Michilovich Sidelnikov and S.O. Shestakov. On the insecurity of cryptosystems based on generalized Reed-Solomon codes. *Discrete Math. Appl.*, 1(4):439–444, 1992. (Cited on pages 15 and 16.)

[SS96] Michael Sipser and Daniel A. Spielman. Expander codes. *IEEE TRANSACTIONS ON INFORMATION THEORY*, 42:1710–1722, 1996. (Cited on page 75.)

[SS13] Nicolas Sendrier and Dimitris E. Simos. The hardness of code equivalence over and its application to code-based cryptography. In *Post-Quantum Cryptography 2013*, volume 7932 of *Lecture Notes in Comput. Sci.*, pages 203–216. Springer, 2013. (Cited on pages 10 and 14.)

[SSMS09] Abdulhadi Shoufan, Falko Strenzke, H. Gregor Molter, and Marc Stöttinger. A timing attack against patterson algorithm in the McEliece PKC. In *Information, Security and Cryptology - ICISC 2009, 12th International Conference*, volume 5984 of *Lecture Notes in Comput. Sci.*, pages 161–175, Seoul, Korea, December 2009. Springer. (Not cited.)

[STM+08] Falko Strenzke, Erik Tews, H. Gregor Molter, Raphael Overbeck, and Abdulhadi Shoufan. Side channels in the McEliece PKC. In Johannes Buchmann and Jintai Ding, editors, *Post-Quantum Cryptography 2008*, volume 5299 of *Lecture Notes in Comput. Sci.*, pages 216–229. Springer, 2008. (Not cited.)

[Str10a] Falko Strenzke. A Timing Attack against the Secret Permutation in the McEliece PKC. In Nicolas Sendrier, editor, *Post-Quantum Cryptography 2010*, volume 6061 of *Lecture Notes in Comput. Sci.*, pages 95–107. Springer, 2010. (Not cited.)

[Str10b] Falko Strenzke. A smart card implementation of the McEliece PKC. In *Information Security Theory and Practices. Security and Privacy of Pervasive Systems and Smart Devices*, volume 6033 of *Lecture Notes in Comput. Sci.*, pages 47–59. Springer, 2010. (Cited on page 1.)

[Str13] Falko Strenzke. Timing attacks against the syndrome inversion in code-based cryptosystems. In *Post-Quantum Cryptography 2013*, volume 7932 of *Lecture Notes in Comput. Sci.*, pages 217–230, Limoges, France, June 2013. Springer. (Not cited.)

[SZ04] Damien Stehlé and Paul Zimmermann. A Binary Recursive Gcd Algorithm. In Duncan Buell, editor, *6th International Symposium on Algorithmic Number Theory - ANTS VI*, volume 3076 of *Lecture notes in Computer Science*, pages 411–425, Burligton, US, 2004. Springer. Colloque avec actes et comité de lecture. internationale. (Cited on page 112.)

[TS16] Rodolfo Canto Torres and Nicolas Sendrier. Analysis of information set decoding for a sub-linear error weight. In *Post-Quantum Cryptography - 7th International Workshop, PQCrypto 2016, Fukuoka, Japan, February 24-26, 2016, Proceedings*, pages 144–161, 2016. (Cited on pages 1 and 14.)

[TV13] Ido Tal and Alexander Vardy. How to construct polar codes. *IEEE Trans. Inform. Theory*, 59(10):6562–6582, 2013. (Cited on pages 26 and 35.)

[Var97] Alexander Vardy. The intractability of computing the minimum distance of a code. *IEEE Trans. Inform. Theory*, 43(6):1757–1766, November 1997. (Cited on page 6.)

[vMG14a] Ingo von Maurich and Tim Güneysu. Lightweight code-based cryptography: QC-MDPC McEliece encryption on reconfigurable devices. In *Design, Automation & Test in Europe Conference & Exhibition, DATE 2014, Dresden, Germany, March 24-28, 2014*, pages 1–6, 2014. (Cited on page 71.)

[vMG14b] Ingo von Maurich and Tim Güneysu. Towards side-channel resistant implementations of QC-MDPC mceliece encryption on constrained devices. In *Post-Quantum Cryptography 2014*, volume 8772 of *Lecture Notes in Comput. Sci.*, pages 266–282. Springer, 2014. (Cited on page 71.)

[Wan16] Yongge Wang. Quantum resistant random linear code based public key encryption scheme RLCE. In *Information Theory (ISIT), 2016 IEEE International Symposium on*, pages 2519–2523. IEEE, 2016. (Cited on page 118.)

[Wie06a] Christian Wieschebrink. An attack on a modified Niederreiter encryption scheme. In Moti Yung, Yevgeniy Dodis, Aggelos Kiayias, and Tal Malk, editors, *Public-Key Cryptography - PKC 2006*, volume 3958 of *Lecture Notes in Comput. Sci.*, pages 14–26. Springer, 2006. (Cited on page 15.)

[Wie06b] Christian Wieschebrink. Two NP-complete problems in coding theory with an application in code based cryptography. In *Proc. IEEE Int. Symposium Inf. Theory - ISIT*, pages 1733–1737, 2006. (Cited on page 15.)

[Wie09] Christian Wieschebrink. Cryptanalysis of the Niederreiter public key scheme based on GRS subcodes. IACR Cryptology ePrint Archive, Report 2009/452, 2009. (Cited on page 15.)

[Wit13] Ernst Witt. *Collected papers/Gesammelte Abhandlungen*. Springer Collected Works in Mathematics. Springer, Heidelberg, 2013. Edited by Ina Kersten, Reprint of the 1998 edition. (Cited on pages 96, 98, 101, and 142.)