UNIVERSITÉ DE LIMOGES
École doctorale no 521 - Sciences et Ingéniérie pour l'Information, Mathématiques
Faculté des Sciences et Techniques
Département de mathématiques et d'informatique - Laboratoire XLIM-DMI

Thèse de Doctorat

Pour obtenir le titre de

DOCTEUR DE L'UNIVERSITÉ DE LIMOGES

Spécialité : Informatique

Présentée et soutenue par

Gaétan MURAT

le 09/12/2014

Résultants de polynômes de Ore et Cryptosystèmes de McEliece sur des Codes Rang faiblement structurés

Jury			
Rapporteurs			
	Jean-Pierre TILLICH	INRIA	Directeur de recherche
	Felix ULMER	Université de Rennes 1	Professeur
Directeurs de thèse			
	Philippe GABORIT	Université de Limoges	Professeur
	Olivier RUATTA	Université de Limoges	Maître de conférences
Examinateurs			
	Thierry BERGER	Université de Limoges	Professeur
	Ayoub OTMANI	Université de Rouen	Professeur

Résumé

Les techniques de chiffrement les plus utilisées en *cryptographie*, basées sur des problèmes de théorie des nombres, présentent malgré leur efficacité des défauts notamment une vulnérabilité aux attaques menées à l'aide d'ordinateur quantiques. Il est donc pertinent d'étudier d'autres familles de cryptosystèmes. Nous nous intéressons ici aux *cryptosystèmes basés sur les codes correcteurs*, introduits par McEliece en 1978 dans [McEl], qui, étant basés sur des problèmes difficiles de théorie des codes, ne présentent pas cette vulnérabilité. Ces cryptosystèmes présentent des inconvénients, qui font qu'ils sont peu utilisés en pratique. Selon le code choisi, ils peuvent être vulnérables aux attaques structurelles, mais surtout ils nécessitent des clés de taille très importante.

Récemment [Mis], [MTSB] une nouvelle famille de codes appelés codes MDPC a été introduite ainsi qu'un cryptosystème basé sur cette famille de codes. Les codes MDPC semblent être distinguables seulement en trouvant des mots de poids faibles dans leur dual, les affranchissant ainsi d'une éventuelle vulnérabilité aux attaques structurelles. De plus, en utilisant une des matrices quasi-cycliques, ils obtiennent des clés de taille très compacte.

Nous avons pour notre part, travaillé dans le contexte de la *métrique rang*, une nouvelle métrique introduite en 1985 [Gab1] par Gabidulin qui semble bien adaptée à une utilisation en cryptographie :

- Nous avons commencé par travailler autour de la notion de *polynôme de Ore* [Ore1] et le cas particulier important des *q-polynômes* [Ore2]. Ces derniers sont des combinaisons linéaires des itérés de l'automorphisme de Frobenius sur un corps fini.

  Ces polynômes constituent un objet d'étude important en métrique rang, de par leur utilisation dans les premiers cryptosystèmes dans cette métrique. Nous présentons sous une nouvelle forme des résultats déjà connus, et de nouveaux algorithmes pour le calcul du PGCD de deux polynômes de Ore et le calcul des résultants et sous-résultants de polynômes de Ore (ainsi que de polynômes usuels en généralisant au calcul des sous-résultants la formule déjà connue pour les résultants) en utilisant une *matrice de multiplication à droite* plus petite que la matrice de Sylvester utilisée habituellement.

  Ces résultats peuvent être réexploités indirectement dans le cryptosystème présenté par la suite bien que celui-ci ne soit pas basé sur les q-polynômes.

- La partie suivante de notre travail est consacrée à l'introduction d'une nouvelle famille de codes en métrique rang appelés codes LRPC (pour Low Rank Parity Check codes). Ces codes ont la particularité d'avoir une matrice de parité de poids rang faible (et peuvent donc être vus comme une généralisation des codes LDPC ou MDPC à la métrique rang).

  Nous présentons le cryptosystème LRPC, un cryptosystème de type Mc Eliece en métrique rang basé sur les codes LRPC. Ces codes sont très peu structurés et sont donc vraisemblablement résistants aux attaques structurelles. La matrice de parité peut être choisie doublement circulante (on parle alors de codes DC-LRPC) ce qui diminue considérablement la taille de la clé.

  Ainsi, le cryptosystème DC-LRPC cumule les avantages d'offrir une bonne sécurité en étant basé sur un problème difficile (comme tous les cryptosystèmes basés sur les codes correcteurs), d'être faiblement structurés, de disposer d'une clé de taille assez petite (quelques milliers de bits au plus) et d'un algorithme de décodage efficace.

  Une attaque a été trouvée dans le cas du cryptosystème DC-LRPC. Cette attaque basée sur la notion de *code replié* permet de baisser significativement la sécurité du cryptosystème dans le cas où le polynôme $X^{k-1} + X^{k-2} + \cdots + 1$ est scindable (k désignant la dimension du code). Cependant ce n'est pas le cas pour les paramètres présentés où le cryptosystème reste valide.

Mots-Clés : Cryptographie, codes correcteurs d'erreurs, cryptosystème de McEliece, q-polynômes, matrice de multiplication à droite, métrique rang, matrice doublement circulante

Abstract

Weakly Structured Error Correcting Codes in Rank Metric and their Application to Cryptography

The most commonly used encryption techniques in *cryptography* are based on problems in number theory. Despite their efficiency, they are vulnerable to post-quantum cryptographic attack. Therefore it is relevant to study other types of cryptosystems. In this work we study *error-corrector codes based cryptosystmems*, introduced by McEliece in 1978 [McEl] ; being based on hard problems in coding theory, these cryptosystems do not have this weakness. However these cryptosystems are almost not used in practice because they are vulnerable to strucural attacks and they require a key with very big length.

Recently [Mis], [MTSB] a new family of codes named MDPC codes has been introduced as well as a cryptosystem that is based on these codes. It seems that MDPC codes are distinguishable only by finding words with weak weight in their dual, thus preventing them from structural attacks. Furthermore, they can have compact keys by using quasi-cyclic matrices.

In the present paper we use the *rank metric*, a new metric for codes that was introduced by Gabidulin in [Gab1] and seems suited for a cryptographic use :

- At first we studied *Ore Polynomials* [Ore1] and the special case of *q-polynomials* [Ore2], the latter being iterates of the Fobenius automorphism on a finite field.

  These polynomials are widely in rank metric due to their use in the first code-based cryptosystems in rank metric. We reformulate already known results and give new results regarding the computation of GCD, resultants and subresultants of two Ore polynomials (as well as usual polynomials for which we give a generalization of the resultant computation to subresultants) using a *right-hand multiplication matrix* which is smaller than the well-known Sylvester matrix.

  These results may be reused in the cryptosystem we introduce in the next chapters, though this cryptosystem is not based on q-polynomials.

- In the next part of our work we define the LRPC codes (for Low Rank Parity Check Codes), a new family of codes in rank metric. These codes have a parity check matrix whose rank weight is low (and thus they can be seen as a generalization of LDPC or MDPC codes to rank metric).

  We present the LRPC cryptosystem, a McEliece cryptosystem in rank metric based on LRPC codes. These codes are weakly structured and so are likely to resist structural attacks. We can choose a double-circulant parity check matrix which greatly lowers the key size (we name these particular codes DC-LRPC codes).

  Thus the DC-LRPC cryptosystems have a good security (being based on a hard problem in coding theory), are weakly structured, have small public keys and can be quickly decoded.

  An attack was found for DC-LRPC cryptosystem. This attack relies on *folded codes* and may greatly lower the security of the cryptosystem, however it works only when the polynomial $X^{k-1} + X^{k-2} + \cdots + 1$ has a divisor with big degree. We give parameters for which the cryptosystem remains valid.

Keywords : Cryptography, Error-Correcting Codes, McEliece Cryptosystem, Ore Polynomials, q-Polynomials, Right-Hand Multiplication Matrix, Rank Metric, Double Circulant Matrix

Remerciements

Je souhaite remercier l'ensemble des personnes qui m'ont côtoyé et soutenu pendant l'élaboration et la rédaction de cette thèse.

Je tiens à remercier en premier lieu Philippe Gaborit et Olivier Ruatta pour m'avoir donné l'opportunité de faire ce travail de recherche ainsi que pour les conseils, la confiance et le soutien qu'ils m'ont apportés pendant ces quatre ans.

Je remercie Felix Ulmer et Jean-Pierre Tillich d'avoir accepté d'être rapporteurs, ainsi que Thierry Berger et Ayoub Otmani d'avoir accepté de faire partie du jury.

Je remercie le personnel du département de Mathématiques et d'Informatique de l'université de Limoges pour la qualité des formations qu'il dispense et pour m'avoir permis de travailler dans ces domaines passionnants que sont la cryptographie et la théorie des codes correcteurs d'erreurs pendant ma thèse, et avant cela, il y a déjà quelques années, pour mon Master.

J'en profite pour saluer les autres étudiants et doctorants que j'ai pu côtoyer à l'université de Limoges, avec une pensée particulière pour Christophe, Elsa, Daouda, Adrien et Samuel. Merci à Julien Schreck et Guillaume Quintin pour leur aide précieuse en programmation.

Remontons encore un peu plus avant ; je souhaiterais remercier les enseignants qui m'ont donné le goût des Mathématiques et l'envie de les étudier. Merci en particulier à Michel Ferrière, Agnès Bonnier-Rigny et Sylvie Massonnet. Maintenant que je suis prof à mon tour, j'espère être capable d'enseigner avec autant de passion, de gentillesse et de professionnalisme que vous le faisiez.

Fin du flashback récursif, je voudrais pour finir saluer et remercier les gens qui comptent le plus pour moi et que je suis heureux d'avoir à mes côtés.

Salut à mes potes, Guigui, Alex, Oana, François, Judith, Germain, Cécile, Philippe, Marie-Pierre, Romane et tous les autres. Merci d'avoir réussi à me sortir de ma tanière de temps en temps.

Salut à ma famille, mon p'tit frère, ma p'tite soeur, mon Papa et ma Maman, qui m'ont donné l'envie et les moyens de faire des études et m'ont toujours soutenu dans les choix que j'ai faits.

Merci à Gustin et Zadig, deux petits garçons très importants avec qui il me tarde de pouvoir repasser plus de temps.

Merci à Hélène pour tellement, tellement, tellement de choses qu'il serait vain de vouloir essayer de les détailler ici.

Table des matières

Résumé 4

Abstract 5

Remerciements 6

Index des notations 13

Introduction 14

1 Contexte d'étude 14

2 Travaux précédents 16

2.1 En théorie de la complexité 16
2.2 En cryptographie basée sur les codes correcteurs (depuis les travaux de Gabidulin) . 17

3 Notre contribution 18

4 Plan 19

Première partie : Généralités 22

Chapitre I :

Notions essentielles en théorie des codes ;

Applications à la cryptographie 24

1 Théorie de l'information 24

1.1 Transmission d'un signal 24
1.2 Codage de canal 25

2 Théorie des codes correcteurs d'erreurs 26

2.1 Code et principales caractéristiques d'un code 27
2.2 Structure d'espace métrique sur $\mathcal{C}$ 27
2.3 Décodage 28
2.4 Codes linéaires 30
2.4.1 Généralités 30
2.4.2 Bornes sur les paramètres d'un code 30
2.4.3 Matrice génératrice, matrice de contrôle de parité 31
2.4.4 Code dual 32
2.4.5 Exemples d'algorithmes de calcul pour la matrice de parité 33
2.5 Aspects algorithmiques du décodage ; exemples 35
2.5.1 Décodage d'un code linéaire par syndrome 35
2.5.2 Le problème du décodage par syndrome et sa complexité 35
2.5.3 Décodage d'un code linéaire par ensemble d'information 36
2.6 Codes cycliques 36

3 Utilisation des codes correcteurs dans un contexte cryptographique 37

3.1 Notion de cryptosystème . . . 37
3.1.1 Définition . . . 37
3.1.2 Cryptosystèmes symétriques, cryptosystèmes asymétriques . . . 38
3.1.3 Sécurité d'un cryptosystème . . . 38
3.1.4 Notion de problème difficile . . . 38
3.2 Application des codes correcteurs à la cryptographie : le cryptosystème de McEliece . 39
3.2.1 Présentation du cryptosystème . . . 39
3.2.2 Sécurité du cryptosystème de McEliece . . . 41
3.2.3 Avantages et inconvénients de la cryptographie basée sur les codes correcteurs . 41

4 Conclusion . . . 41

Chapitre II :

Construction des corps finis . . . 43

1 Idée de base : quotient d'un anneau de polynômes par un polynôme irréductible . . . 43

2 Cardinal d'un corps fini . . . 44

3 Groupe multiplicatif d'un corps fini . . . 44

4 Unicité d'un corps fini à isomorphisme près . . . 44

5 Construction effective . . . 44

6 Cas particulier important : construction d'un corps par adjonction d'un élément primitif à son sous-corps premier . . . 45

7 Implémentation . . . 48

Deuxième partie : Polynômes de Ore, q-polynômes et applications . 50

Chapitre III :

Polynômes de Ore et q-polynômes . . . 52

1 Automorphismes des corps finis . . . 52

2 Polynômes de Ore . . . 53

2.1 Définition . . . 53
2.2 Sous-corps des éléments commutatifs pour la multiplication . . . 54
2.3 Exemples de calculs . . . 55
2.4 Degré d'un polynôme de Ore . . . 55
2.5 Division euclidienne à droite . . . 56
2.6 Division euclidienne à gauche . . . 57
2.7 Evaluation d'un élément par un polynôme de Ore . . . 58
2.8 Racine d'un polynôme de Ore . . . 58

3 Anneaux de q-polynômes (ou polynômes linéarisés) . . . 58

3.1 Définitions et premières propriétés . . . 58
3.2 Produit de q-polynômes . . . 59
3.3 Les q-polynômes vus comme des cas particuliers des polynômes de Ore . . . 60
3.3.1 Bijection entre un anneau de Ore et un anneau de q-polynômes . . . 60

3.3.2 Conséquences . . . 61
3.4 Complexité des opérations arithmétiques dans $\mathbb{F}_{q^m}[X^q, \theta]$ . . . 61
3.5 Racines de q-polynômes . . . 62
3.5.1 Structure de l'ensemble des racines d'un q-polynôme . . . 62
3.5.2 Polynôme associé à un espace vectoriel . . . 63
3.5.3 Complexité . . . 65
3.5.4 Structure des racines du RGCD, du RLCM de deux q-polynômes . . . 65
3.6 Application : utilisation des q-polynômes pour déterminer l'intersection de deux sous-espaces vectoriels . . . 66

Chapitre IV :

Matrice de multiplication à droite pour des q-polynômes et applications . . . 68

1 Résultants et sous-résultants sur un anneau commutatif . . . 69

1.1 Résultant de deux polynômes . . . 69
1.1.1 Définition à partir de la matrice de Sylvester et propriétés . . . 69
1.1.2 Matrice de multiplication et application au calcul du résultant . . . 70
1.2 Suites des sous-résultants . . . 70
1.3 Lien entre sous-résultants et PGCD . . . 72

2 Calcul des coefficients sous-résultants à partir d'une matrice extraite de la matrice de multiplication . . . 73

2.1 Polynôme déterminantal et application au calcul des sous-résultants . . . 73
2.2 Utilisation de la matrice de multiplication . . . 74

3 Résultant de deux polynômes de Ore . . . 76

3.1 Utilisation de la matrice de multiplication à droite . . . 76
3.1.1 Idée générale . . . 76
3.1.2 Matrice de multiplication à droite et nouvelle définition du résultant . . . 77
3.2 Propriétés du résultant de deux polynômes de Ore . . . 78

4 Application : algorithme de recherche du PGCD de deux polynômes de Ore . 79

4.1 Algorithme de recherche du PGCD . . . 79
4.2 Etude de la complexité . . . 79

5 Sous-résultants de deux polynômes de Ore . . . 80

5.1 Opérateurs de Ore . . . 80
5.2 Sous-résultants de deux polynômes de Ore . . . 82

6 Calcul des coefficients sous-résultants de deux polynômes de Ore à partir d'une matrice de multiplication . . . 83

6.1 Liens entre les deux points de vue sur les polynômes de Ore . . . 83
6.2 Généralisation de la formule vue dans le cas commutatif . . . 83

Troisième partie : Métrique rang et cryptosystème LRPC . . . 86

Chapitre V :

Codes correcteurs en métrique rang, applications à la cryptographie . 88

1 Définitions . . . 88

2 Codes en métrique rang . . . 90

2.1 Définitions . . . 90
2.2 Bornes sur les codes en métrique rang . . . 90
2.2.1 Borne de Singleton . . . 90
2.2.2 Borne de Gilbert-Varshamov . . . 91

3 Décodage en métrique rang . . . 92

3.1 Problèmes liés au décodage en métrique rang . . . 92
3.1.1 Problème du décodage par syndrome en métrique rang . . . 92
3.1.2 L'algorithme d'Ourivski-Johanson . . . 92
3.1.3 Un nouvel algorithme . . . 93
3.2 Importance de la complexité algorithmique du décodage des codes en métrique rang . 94

4 Un exemple fondamental : les codes de Gabidulin . . . 95

4.1 Définition . . . 95
4.2 Propriétés . . . 96
4.3 Décodage des codes de Gabidulin . . . 96

5 Cryptographie en métrique rang . . . 97

5.1 Introduction . . . 97
5.2 Le cryptosystème GPT . . . 97
5.3 Attaque structurelle . . . 98

6 Conclusion . . . 99

Chapitre VI :

Codes LRPC ; cryptosystème LRPC . . . 101

1 Généralités . . . 101

2 Résultats sur le produit de deux sous-espaces . . . 102

3 Low Rank Parity Check Codes . . . 104

3.1 Contexte d'utilisation et définitions . . . 104
3.2 Cas particulier important : les codes DC-LRPC (Double Circulant Low Rank Parity Check) . . . 104
3.2.1 Matrices circulantes ; matrices coublement circulantes . . . 104
3.2.2 Codes DC-LRPC . . . 105
3.3 Création d'une matrice de décodage à coefficients dans le corps de base $\mathbb{F}_q$ à partir de la matrice de parité . . . 106

4 Algorithme de décodage pour les codes LRPC . . . 109

4.1 Idée générale . . . 109
4.2 Présentation de l'algorithme de décodage . . . 109
4.3 Preuve de la validité de l'algorithme . . . 110
4.4 Probabilité d'échec . . . 110
4.5 Complexité du décodage . . . 111
4.6 Bilan . . . 111
4.7 Exemple . . . 112

5 Application à la cryptographie : le cryptosystème LRPC . . . 113

5.1 Le cryptosystème LRPC . . . 113
5.2 Paramètres du cryptosystème LRPC . . . 114

6 Sécurité du cryptosystème LRPC . . . 115
6.1 Sécurité sémantique . . . 115
6.2 Sécurité pratique . . . 115
6.2.1 Attaque du message . . . 115
6.2.2 Attaque par clé contrefaite (Spurious key) . . . 115
6.2.3 Attaque sur la clé secrète par repliement du code . . . 115
6.2.4 Une première remarque . . . 116
6.2.5 Code replié . . . 116
6.2.6 Amélioration de l'attaque grâce à un repliement du code . . . 116
6.2.7 Protection contre cette attaque . . . 117

7 Exemples de paramètres et comparaison avec les MDPC . . . 117
7.1 Exemples de paramètres . . . 117
7.1.1 Premiers exemples . . . 117
7.1.2 Nouveaux paramètres résistants à l'attaque par repliement . . . 118
7.2 Comparaison avec le cryptosystème MDPC . . . 118

8 Implémentation et temps d'exécution . . . 118

9 Pistes de recherche . . . 119

10 Conclusion . . . 119

Appendice : Démonstrations des résultats du paragraphe 2 . . . 121

Conclusion et pistes de recherche . . . 125

1 Concernant les q-polynômes . . . 125

2 Concernant la cryptographie sur les codes correcteurs en métrique rang . . . 125

Bibliographie . . . 128

Index des notations

$q = p^s$	Puissance d'un nombre premier p
$\mathbb{F}_{q^m}$	Le corps fini à $q^m = (p^s)^m$ éléments unique à isomorphisme près ce corps est isomorphe à $(\mathbb{F}_q)^m$
$C_{p,m}$	Le polynôme de Conway de paramètres p et m
$\mathcal{C}[n,k,d]$	Code linéaire de longueur n, dimension k et distance minimale d sur $\mathbb{F}_{q^m}$
G	Matrice génératrice de $\mathcal{C}$
G'	Matrice génératrice de $\mathcal{C}$ sous forme systématique
H	Matrice de parité de $\mathcal{C}$
H'	Matrice de parité de $\mathcal{C}$ sous forme systématique
$\mathbf{x}$	Mot du code $\mathcal{C}$
$\mathbf{e}$	Vecteur erreur
$\mathbf{y} = \mathbf{x}G + \mathbf{e}$	Mot transmis
$\mathbf{s} = H\mathbf{y}^T$	Syndrome
$d_H(\mathbf{x}_1, \mathbf{x}_2)$	Distance de Hamming entre $\mathbf{x}_1$ et $\mathbf{x}_2$
$w_H(\mathbf{x})$	Poids de Hamming du mot $\mathbf{x}$
$d_R(\mathbf{x}_1, \mathbf{x}_2)$	Distance rang entre $\mathbf{x}_1$ et $\mathbf{x}_2$
$w_R(\mathbf{x})$	Poids rang de $\mathbf{x} \in \mathbb{F}_{q^m}$
$\mathbb{F}_{q^m}[X, \sigma, \delta]$	L'anneau des polynômes de Ore muni de l'automorphisme σ et de la σ-dérivation δ
θ	L'automorphisme de Frobenius $x \mapsto x^q$ sur $\mathbb{F}_{q^m}$
$\mathbb{F}_{q^m}[X^q, \theta]$	L'anneau des q-polynômes sur $\mathbb{F}_{q^m}$
A et B	Chapitres 3 et 4 : des polynômes de Ore $A = \sum_{i=0}^{d_A} a_i X^i$ et $B = \sum_{i=0}^{d_B} b_i X^i$ de degrés d_A et d_B ou des q-polynômes $A = \sum_{i=0}^{d_A} a_i X^{q^i}$ et $B = \sum_{i=0}^{d_B} b_i X^{q^i}$ de q-degrés d_A et d_B Chapitres 5 et 6 : Des matrices à coefficients dans $\mathbb{F}_{q^m}$
$\mathcal{M}_{k\times n}(\mathbb{F}_{q^m})$	Ensemble des matrices à k lignes et n colonnes à coefficients dans $\mathbb{F}_{q^m}$
$\begin{bmatrix} m \\ r \end{bmatrix}_q$	Le binôme de Gauss $\prod_{i=0}^{r-1} \frac{q^m - q^i}{q^r - q^i}$

Introduction

1 Contexte d'étude

La cryptographie désigne l'ensemble des techniques consistant à masquer le contenu d'un message de façon à ce que seul son destinataire soit capable d'y avoir accès. De telles techniques sont utilisées depuis l'antiquité mais ont pendant longtemps relevé davantage de l'artisanat que de la science. C'est dans la deuxième moitié du vingtième siècle que la cryptographie est devenue un thème de recherche scientifique à part entière, et qu'elle a pris de plus en plus d'importance avec le développement des télécommunications.

Les cryptosystèmes classiques sont basés sur des problèmes de théorie des nombres (les plus connus étant la factorisation de grands nombres et le logarithme discret). Ces cryptosystèmes offrent une bonne sécurité tout en nécessitant une implémentation peu coûteuse, de l'ordre du millier de bits. Cependant, des méthodes de plus en plus performantes existent pour résoudre ces problèmes, on dispose notamment d'algorithmes sous-exponentiels [BGJT] et d'algorithmes utilisables en cryptographe post-quantique [MR].

Il est donc pertinent d'étudier d'autres familles de cryptosystèmes ne présentant pas cette vulnérabilité. Il en existe plusieurs types. Nous nous intéressons ici aux cryptosystèmes basés sur les codes correcteurs, introduits par McEliece en 1978 dans [McEl], dont la sécurité repose sur l'indistinguabilité d'un code donné avec un code aléatoire, et sur le fait que le décodage d'un code (d'apparence) aléatoire est un problème difficile, que l'on ne sait pas résoudre en temps polynomial : le problème du décodage par syndrome pour la distance de Hamming est un problème fondamental en théorie de la complexité, beaucoup de travaux lui ont été consacrés depuis plus de 30 ans, depuis que la NP-complétude du problème a été prouvée par Berlekamp, McEliece et van Tilborg [BET]. Le problème du décodage est d'une importance fondamentale par ses applications en théorie de l'information et ses liens avec les treillis. En plus de leur sécurité, les cryptosystèmes basés sur les codes correcteurs ont aussi l'avantage de bénéficier d'opérations de chiffrement et de déchiffrement très rapides

Cependant, ces cryptosystèmes présentent aussi des inconvénients, qui font qu'ils sont peu utilisés en pratique : l'indistiguabilité avec un code aléatoire dépend de la famille de codes choisie, et certains codes ont une structure tellement forte qu'il est impossible de la masquer. Les codes de Reed-Solomon [RS] sont ainsi de bons candidats en théorie pour le cryptosystème de McEliece mais leur forte structure rend les cryptosystèmes très vulnérables [SiSh]. Le principal inconvénient des

cryptosystèmes de McEliece tels qu'ils avaient été introduits initialement reste cependant la taille des clés : pour un tel cryptosystème, la clé est une matrice, qui peut prendre beaucoup de place. Le cryptosystème de McEliece a été introduit initialement en utilisant les codes de Goppa [CFS] qui, étant indistinguables de codes aléatoires, offrent une bonne résistance aux attaques structurelles. Cependant les codes de Goppa ont une capacité de correction très faible et nécessitent des clés publiques de taille très importantes pour être viables.

Suite à la notion de distance de Hamming pour les codes correcteurs et la notion de distance euclidienne pour les réseaux, le concept de métrique rang a été introduit en 1951 par Loo-Keng Hua [Hua] comme une « distance arithmétique » pour des matrices à coefficients dans un corps $\mathbb{F}_q$. On rappelle que pour deux matrices M et $N \in \mathcal{M}_n(\mathbb{F}_q)$, la distance rang entre M et N notée $d_r(M,N)$ est définie par le rang de la matrice $M-N$ soit $d_r(M,N) = \mathrm{rk}(M-N)$. En 1978 Delsarte définit dans [Del] la notion de distance rang pour un ensemble de formes bilinéaires (que l'on peut voir comme des matrices rectangulaires). Il introduit une borne de type borne de Singleton pour ces codes et donnes la construction de *codes matriciels* optimaux. Un code matriciel sur $\mathbb{F}_q$ pour la métrique rang est défini comme l'ensemble des $\mathbb{F}_q$-combinaisons linéaires d'un ensemble $\mathcal{M}$ de matrices de dimension $m \times n$ sur $\mathbb{F}_q$. Ce sont des codes linéaires sur $\mathbb{F}_q$ et le nombre k de matrices indépendantes dans $\mathcal{M}$ est borné par nm.

En 1985 Gabidulin [Gab1], dans la suite des travaux de Delsarte, introduit les *codes en métrique rang en représentation vectorielle* (et non plus en représentation *matricielle* comme dans les travaux de Delsarte) Un code en métrique rang $\mathcal{C}[n,\, k]$ de longueur n et dimension k sur $\mathbb{F}_{q^m}$ en *représentation vectorielle* est défini comme un $\mathbb{F}_{q^m}$-sous-espace de dimension k de $(\mathbb{F}_{q^m})^n$. A tout vecteur $\mathbf{x} = (x_1, \ldots, x_n) \in (\mathbb{F}_{q^m})^n$ on associe une matrice $\mathrm{Mat}(\mathbf{x}) \in \mathcal{M}_{m\times n}(\mathbb{F}_q)$ définie de la manière suivante : soit $\mathcal{B}$ une base de $\mathbb{F}_{q^m}$ vu comme un $\mathbb{F}_q$-espace vectoriel, la colonne $i \in \{1 \ldots n\}$ de la matrice est égale à $(x_{1,i}, \ldots, x_{m,i})^T$, ce vecteur correspondant aux coordonnées de x_i écrites dans la base $\mathcal{B}$. Le poids rang de x est alors défini par $w_R(\mathbf{x}) = \mathrm{rk}(\mathrm{Mat}(\mathbf{x}))$ et la distance rang entre deux vecteurs quelconques $\mathbf{x}$ et $\mathbf{y}$ est définie par $d_R(\mathbf{x}, \mathbf{y}) = \mathrm{rk}(\mathrm{Mat}(\mathbf{x}) - \mathbf{y})$. On définit ainsi une nouvelle distance sur les codes linéaires, qui permet « d'étaler » les erreurs sur toutes les composantes du vecteur erreur tout en gardant un poids très faible. Les méthodes de décodage classiques en métrique de Hamming par ensemble d'information ne sont donc plus utilisables dans ce contexte. Les meilleurs algorithmes de décodage en métrique rang sont plus complexes qu'en métrique de Hamming, on peut donc espérer construire par ce biais des cryptosystèmes dont la clé pourrait être de taille plus petite.

Les codes en métrique rang en représentation vectorielle, que nous appellerons aussi *codes rang sur des extensions de corps*, peuvent être vus comme des codes correcteurs d'erreurs classiques sur $\mathbb{F}_{q^m}$, vu comme un espace métrique muni de la métrique rang à la place de la métrique de Hamming. On peut ainsi définir les notions de matrice génératrice et matrice de contrôle de parité. Tout code rang en représentation vectorielle est linéaire sur $\mathbb{F}_{q^m}$ et peut être vu comme un code matriciel défini avec des matrices $km \times n$ sur $\mathbb{F}_q$ mais l'inverse n'est pas vraie. Une matrice quelconque de cette forme n'admet pas en général de représentation vectorielle. La représentation vectorielle est intéressante car de tels codes sont plus faciles à décrire et à manipuler.

Dans [Gab1] Gabidulin redécouvre les codes rang optimaux de Delsarte qui s'avèrent être linéaires sur l'extension de corps. Ces codes sont des codes d'évaluation analogues aux codes de Reed-Solomon mais dans un contexte de métrique rang, où tout monôme de la forme x^α est remplacé par un monôme linéarisé de la forme x^{q^α} introduits par Ore en 1933 dans [Ore2]. Gabidulin a mis en avant les analogies avec les codes de Reed-Solomon de façon plus approfondie que dans l'article de Delsarte et a formalisé les problèmes de décodages pour ces codes à présent connus sous le nom de codes de Gabidulin :

Rank Syndrome Decoding Problem (RSD)

Données : Une matrice $H \in \mathcal{M}_{(n-k)\times n}(\mathbb{F}_{q^m})$, un syndrome $\mathbf{s} \in (\mathbb{F}_{q^m})^{n-k}$ et un entier w.

Question : Existe-t-il $\mathbf{x} \in (\mathbb{F}_{q^m})^n$ tel que $H\mathbf{x}^t = \mathbf{s}$ et $w_R(\mathbf{x}) \leqslant w$?

Rank Minimum Distance Problem (RMD)
Données : un code rang $\mathcal{C}[n,k]$, un entier w
Question : Existe-t-il $\mathbf{x} \in \mathcal{C}$ tel que $w_R(\mathbf{x}) \leqslant w$?

Remarque 1. Ces deux problèmes diffèrent fondamentalement du problème dénommé **MinRank problem** qui est aussi associé à la métrique rang mais dans un cas plus général.

Dans la foulée, Gabidulin introduit la notion de codes de Gabidulin, qui sont un équivalent des codes de Reed-Solomon en métrique rang, et propose un cryptosystème de type McEliece basé sur ces codes. Cependant, ils souffrent du même défaut que les codes de Reed-Solomon : une trop grande vulnérabilité aux attaques structurelles. Les cryptosystèmes ultérieurs utilisant la cryptographie en métrique rang présentent également cette vulnérabilité.

2 Travaux précédents

2.1 En théorie de la complexité

Etonnament, les complexités théoriques de la résolution algorithmique des problèmes RSD et RMD pour la métrique rang ne sont pas connues, alors que le problème et ses variations a été étudié de manière intensive pour la distance de Hamming ou les treillis. En particulier au-delà de la NP-complétude du problème de décodage par syndrome pour la distance de Hamming prouvé dans [Var], le problème de distance minimale pour Hamming a été prouvé NP-complet par Vardy dans [Var] ainsi que des variations de ce problème dans [QD].

Comme nous l'avons expliqué dans le paragraphe précédent il est possible de considérer le problème du décodage et celui de la distance minimale pour des codes *matriciels*. Ces problèmes peuvent être vus comme des généralisations des problèmes RSD et RMD. Par exemple dans le cas du problème de décodage pour les codes rang matriciels, nous connaissons un ensemble $\mathcal{M} = \{M_1, ..., M_k\}$ de matrices $n \times n$ sur $\mathbb{F}_q$, une matrice M sur $\mathbb{F}_q$ et un entier w. La question est de décider s'il existe une matrice M_0 sur $\mathbb{F}_q$ de rang $\leqslant w$ telle que $M - M_0$ puisse être exprimée comme une combinaison $\mathbb{F}_q$-linéaire de matrices de $\mathcal{M}$ (i.e. appartienne au code rang matriciel $\mathbb{F}_q$-linéaire engendré par les matrices de $\mathcal{M}$). Remarquons que le code est linéaire si l'on prend le petit corps $\mathbb{F}_q$ comme corps des scalaires mais nécessairement si l'on prend l'extension $\mathbb{F}_{q^m}$.

Ces deux problèmes (décodage et distance minimale pour des codes matriciels) sont apparus dans une forme légèrement plus générale, tous les deux (ce qui peut être source de confusion) sous le même nom de « MinRank Problem » dans la littérature. Dans [Co] Courtois remarque que ces deux problèmes sont NP-complets en observant qu'un code en métrique de Hamming dans $(\mathbb{F}_q)^n$ peut être « injecté » dans un code rang en représentation matricielle simplement en transformant un vecteur $\mathbf{x} \in (\mathbb{F}_q)^n$ en matrice diagonale avec les composantes de $\mathbf{x}$ écrites sur la diagonale, ce qui permet, à partir d'un code de Hamming de dimension k et distance minimale d, d'obtenir un code rang en représentation matricielle avec $\mathcal{M}$ en ensemble de k matrices correspondant aux vecteur de la base, et une distance rang minimale d.

Cette transformation permet d'obtenir la NP-complétude du problème de décodage pour les codes matriciels à partir de la NP-complétude du problème de décodage par syndrome pour les codes en métrique de Hamming. De même, la NP-complétude du problème de distance minimale pour les codes rang matriciels peut être obtenue à partir de la NP-complétude du problème de distance minimale pour la métrique de Hamming.

Cependant, dans le cas de représentation matricielles, la structure des matrices à coefficients dans $\mathbb{F}_q$ est plus simple que la structure pour les codes rang en représentation vectorielle qui sont linéaires si le corps des scalaires est l'extension $\mathbb{F}_{q^m}$ et plus seulement le « petit corps » de base $\mathbb{F}_q$. Le « MinRank Problem » apparaît comme une variation peut structurée des problèmes RSD et RMD. La remarque précédente de Courtois marche bien pour les codes rang matriciels $\mathbb{F}_q$-linéaires mais ne s'applique pas aux codes rang vectoriels $[n,k]$ $\mathbb{F}_{q^m}$-linéaires.

2.2 En cryptographie basée sur les codes correcteurs (depuis les travaux de Gabidulin)

La métrique rang (vectorielle) a été introduite par Gabidulin en 1985 dans [Gab1] en même temps que les codes de Gabidulin qui sont un équivalent des codes de Reed-Solomon pour la métrique rang. Depuis, les codes en métrique rang ont connus de nombreuses applications : en théorie des codes, citons notamment le space-time coding et le network coding, et aussi en cryptographie. Jusqu'à présent l'outil principal pour la cryptographie en métrique rang consistait à masquer la structure des codes de Gabidulin [GPT] de plusieurs façons et d'utiliser un cryptosystème de type McEliece ou Niederreiter avec ces codes. Entre temps la plupart des systèmes ont été attaqués avec succès, par des attaques structurelles utilisant la structure particulière des codes de Gabidulin ([Ove], [FL], [BL], [Loi2], [Gab2]). Une situation similaire existe dans le cas de Hamming, où tous les cryptosystèmes basés sur les codes de Reed-Solomon ont été brisés pour la même raison : les codes de Reed-Solomon sont si structurés que leur structure est difficile à masquer et que l'on parvient toujours à obtenir des renseignements sur la structure du code.

Depuis l'introduction de la cryptographie basée sur les codes par McEliece en 1978, les différents cryptosystèmes (dans le contexte de la métrique de Hamming) étaient basés sur le fait de masquer la structure d'une famille particulière de codes décodables tels les codes de Goppa, de Reed-Muller ou de Reed-Solomon. Ces codes sont très structurés ce qui permet des clés publiques de petite taille, cependant des attaques peuvent être trouvées sur cette structure, pour s'en prémunir il faut augmenter les paramètres des codes ce qui augmente la taille de la clé.

En 1996 et 1997, deux cryptosystèmes basés sur les treillis ont été proposés indépendamment : le cryptosystème NTRU [HPS] et le cryptosystème GGH [GGH] qui peuvent être vus comme des cryptosystèmes dans un schéma de McEliece mais pour la distance euclidienne. Remarquons que la cryptographie basée sur les treillis n'est rien de plus que la cryptographie basée sur les codes avec des codes q-aires mais avec la distance euclidienne plutôt que la distance de Hamming. Les deux cryptosystèmes NTRU et GGH sont basés sur la même idée : la connaissance d'une base *aléatoire* de vecteurs de poids faibles permet d'obtenir un algorithme de décodage adapté à la cryptographie. De plus le cryptosystème NTRU (qui peut être vu comme un cas optimisé du cryptosystème GGH [Loi3]) introduit pour la première fois l'idée d'utiliser des matrices doublement circulantes de façon à faire décroître la taille de la clé publique ; cette idée étant rendu possible grâce au caractère aléatoire de la petite base duale. Remarquons enfin que depuis 15 ans le cryptosystème NTRU n'a jamais connu d'attaques basées sur sa structure doublement circulante, en effet les meilleures attaques restent des attaques générales LLL sur les réseaux.

Dans un contexte classique de cryptographie en métrique de Hamming, Gaborit introduit en 2005 [Gbt] l'idée d'utiliser des codes quasi-cycliques pour faire décroître la taille de la clé publique, cependant l'idée d'ajouter une structure quasi-cyclique à une famille de codes déjà très structurée conduit à des ensemble extrêmement structurés et le système a été brisé [OTD]. Cette idée était alors utilisée avec d'autres familles de codes structurés quasi-cycliques (ou quasi-dyadiques) tels les codes de Goppa quasi-dyadiques [MB] ou les codes alternés quasi-cycliques [BCG], ces systèmes menant à des clés beaucoup plus petites, mais ils ont finalement été attaqués dans [FOPT] et bien que l'idée reste valide, la cryptanalyse de [FOPT] a montré que l'utilisation des codes quasi-cycliques ou quasi-dyadiques ne pouvait pas aboutir à des clés publiques sûres de quelques milliers de bits, mais plutôt quelques dizaines de milliers.

Plus récemment de nouvelles propositions ont été faites dans l'esprit du cryptosystème NTRU initial avec la métrique de Hamming, d'abord en utilisant des codes LDPC quasi-cycliques, puis plus récemment avec les codes MDPC ([MTSB], [Mis].) Barreto, Misoczki, Sendrier et Tillich proposent un cryptosystème basé sur un nouvelle famille de codes en métrique de Hamming appelés codes MDPC (Moderate Density Parity Check), qui sont une généralisation des codes LDPC introduits par Gallager dans [Gal]. Les codes MDPC semblent être distinguables seulement en trouvant des mots de poids faibles dans leur dual, les affranchissant ainsi d'une éventuelle vulnérabilité aux attaques structurelles. De plus, en utilisant une des matrices quasi-cycliques, ils obtiennent des clés de taille très compacte. Cette famille de codes permet d'obtenir le même types de caractéristiques que le cryptosystème NTRU : une clé très petite (4800 bits) et une sécurité basée sur le décodage par une matrice duale aléatoire avec un poids faible.

3 Notre contribution

Philippe Gaborit, Olivier Ruatta, Julien Schreck, Gilles Zémor et moi-même avons pour notre part travaillé dans le contexte de la métrique rang :

- Nous avons commencé par travailler autour de la notion de polynôme de Ore [Ore1] ainsi que le cas particulier des polynôme linéarisés ou q-polynômes [Ore2]. Les polynômes de Ore, à coefficients dans $\mathbb{F}_{q^m}$ (le corps fini de cardinal q^m) sont des polynômes munis d'une loi de multiplication non-commutative (si $a \in \mathbb{F}_{q^m}$ alors $Xa = \sigma(a)X + \delta(a)$ où $\sigma \in \mathrm{Aut}(\mathbb{F}_{q^m})$ et δ est une σ-dérivation). Les q-polynômes, quant à eux, sont des polynômes de la forme $\sum a_i X^{q^i}$ avec $a_i \in \mathbb{F}_{q^m}$. Un anneau de q-polynômes est muni de la loi d'addition habituelle, la loi de multiplication (non commutative) étant la loi de composition. Les q-polynômes sont des applications linéaires.

  Ces polynômes constituent un objet d'étude important en métrique rang, de par leur utilisation dans la construction des codes de Gabidulin et le cryptosystème associé qui utilise les q-polynômes.

  Nous formalisons et nous démontrons avec le vocabulaire des espaces vectoriels des résultats déjà connus sur la structure des racines d'un q-polynôme. Nous introduisons un nouvel algorithme de calcul du PGCD de deux q-polynômes, qui peut être adapté pour déterminer l'intersection de deux espaces vectoriels.

  Nous introduisons de nouveaux algorithmes de calcul des résultants et sous-résultants de deux polynômes de Ore et de calcul des sous-résultants de polynômes usuels. Ces algorithmes sont basés sur l'utilisation de la matrice de multiplication à droite, qui est une matrice de taille plus petite que la matrice de Sylvester utilisée habituellement pour le calcul des résultants et sous-résultants.

  Ces résultats sont présentés dans la deuxième partie de notre travail (la première étant consacrée à des généralités). Ils peuvent être réexploités indirectement dans le cryptosystème présenté par la suite bien que celui-ci ne soit pas basé sur les q-polynômes.

- L'autre partie de notre travail est consacrée à l'introduction d'une nouvelle famille de codes en métrique rang appelés codes LRPC (pour Low Rank Parity Check codes). Ces codes ont la particularité d'avoir une matrice de parité de poids rang faible (et peuvent donc être vus comme une généralisation des codes LDPC ou MDPC à la métrique rang).

  Plus précisément, les coefficients d'une matrice de parité des codes LRPC appartiennent à un sous-espace vectoriel de $\mathbb{F}_{q^m}$ (on rappelle que $\mathbb{F}_{q^m}$ peut être vu comme un espace vectoriel sur le corps des scalaires $\mathbb{F}_q$) dont la dimension est majorée par un « petit » nombre d. Ceci permet d'obtenir un algorithme de décodage très rapide que nous présentons, basé sur une multiplication matricielle dans le « petit corps » $\mathbb{F}_q$ et non plus l'extension $\mathbb{F}_{q^m}$.

  Nous présentons le cryptosystème LRPC, un cryptosystème de McEliece en métrique rang basé sur les codes LRPC. Ces codes sont très peu structurés et sont donc vraisemblablement résistants aux attaques structurelles. La matrice de parité peut être choisie doublement circulante (on parle alors de codes DC-LRPC) ce qui diminue considérablement la taille de la clé.

  Ainsi, le cryptosystème DC-LRPC cumule les avantages d'offrir une bonne sécurité en étant basé sur un problème difficile (comme tous les cryptosystèmes basés sur les codes correcteurs), d'être faiblement structurés, de disposer d'une clé de taille assez petite (quelques milliers de bits au plus) et d'un algorithme de décodage efficace.

  Cependant, une attaque possible a été trouvée contre le cryptosystème DC-LRPC mais cette attaque ne marche a priori que dans le cas où où le polynôme $X^{k-1} + X^{k-2} + \cdots + 1$ est scindable (k désignant la dimension du code)..

4 Plan

Cette thèse est partagée en trois parties, chacune étant partagée en deux chapitres :

Dans la **Première partie : Généralités**, nous rappelons des résultats fondamentaux dans le contexte où nous travaillons :

- Le **Chapitre I** répertorie les résultats les plus importantes en **théorie des codes** (dans le contexte de la métrique de Hamming). Nous rappelons notamment les définitions de code linéaire, matrice génératrice, matrice de parité qui se généralisent immédiatement à la métrique rang et seront couramment utilisés dans notre travail. Nous rappelons l'algorithme de **décodage par syndrome** et sa complexité. Nous redonnons des méthodes de **construction algorithmiques de matrices de parité** afin de préparer le terrain pour les algorithmes du chapitre VI.

  Nous détaillons l'utilisation des codes dans un contexte **cryptographique** via l'introduction du **cryptosystème de McEliece** (et le cryptosystème alternatif proposé par Niederreiter). Nous redonnons les avantages et les inconvénients de ce type de système et réexpliquons en quoi il a motivé les travaux ultérieurs en cryptographie basée sur la théorie des codes.

- Dans un court **Chapitre II** nous redonnons des résultats importants sur les **corps finis** qui constituent une structure mathématique fondamentale en théorie des codes, particulièrement en métrique rang où la bijection entre $(\mathbb{F}_{q^m})$, le corps fini à q^m éléments, et $(\mathbb{F}_q)^m$, espace vectoriel de dimension m sur le corps des scalaires $\mathbb{F}_q$, est utilisée pour définir la distance rang. Nous rappelons l'existence de cette bijection ainsi que d'autres résultats théoriques, et nous rappelons aussi comment construire explicitement de tels corps. Nous donnons des tableaux de correspondance pour des petits corps auxquels nous nous référerons par la suite pour pouvoir faire directement des calculs en métrique rang.

Dans la **Deuxième partie : polynômes de Ore, *q*-polynômes et applications**, nous rappelons des résultats théoriques sur les polynômes de Ore et le cas particulier, important dans la cryptographie en métrique rang, des q-polynômes. Nous reformalisons certains résultats déjà connus et nous donnons de nouveaux résultats sur les polynômes de Ore via l'utilisation de la matrice de multiplication.

- Dans le **Chapitre III** nous rappelons des résultats fondamentaux sur les polynômes de Ore [Ore1] et le cas particulier des q-polynômes [Ore2], important dans la théorie des codes en métrique rang. Les **polynômes de Ore** sont des polynômes de la forme $\sum a_i X^i$ où les a_i appartiennent au corps fini $\mathbb{F}_{q^m}$. Un anneau de polynômes de Ore est muni de la loi d'addition standard sur les polynômes. La multiplication à droite de X par $a \in \mathbb{F}_{q^m}$ est définie par $Xa = \sigma(a)X + \delta(a)$ où σ est un automorphisme de $\mathbb{F}_{q^m}$ (nous verrons que les automorphismes de $\mathbb{F}_{q^m}$ sont des itérés de l'**automorphisme de Frobenius** $\theta : x \mapsto x^q$) et δ une σ-dérivation. Nous déduisons de cette égalité la loi de multiplication (non commutative) de deux polynômes de Ore.

  La loi de multiplication choisie permet de munir ces anneaux de polynômes d'une **division euclidienne à droite** ce qui permet de définir des notions de plus grand diviseur commun à droite et plus petit multiple commun à droite pour ces polynômes.

  Nous détaillons le cas particulier des ***q*-polynômes** également introduits par Ore. Les q-polynômes sont des polynômes $\sum_{i=0}^{\deg} a_i X^{q^i}$ où les a_i appartiennent au corps fini $\mathbb{F}_{q^m}$. Le nombre deg est appelé le ***q*-degré** du q-polynôme. L'addition de deux q-polynômes est l'addition usuelle, la multiplication (non commutative) est égale à la loi de composition i.e. $(A \times B)(X) = A \circ B(X) = A(B(X))$ où A et B sont des q-polynômes.

Nous reformalisons certains résultats déjà donnés par Ore avec le vocabulaire des **espaces vectoriels**. Nous voyons que l'ensemble des racines d'un polynôme de Ore a une structure d'espace vectoriel sur $\mathbb{F}_q$, et réciproquement que pour tout espace vectoriel V sur $\mathbb{F}_q$ il existe un unique q-polynôme P_V de coefficient dominant 1 dont l'ensemble des racines est exactement V. Nous montrons que le polynôme associé à l'intersection de deux espaces vectoriels V et W est égal au PGCD à droite de P_V et P_W et que le polynôme associé à la somme $V + W$ est égal au PPCM à droite de P_V et P_W.

- Dans le **Chapitre IV**, nous commençons par rappeler les définitions des résultants et des sous-résultants de polynômes usuels et la méthode de calcul des résultants par la **matrice de multiplication** dont nous rappelons la définition.

  Nous utilisons la matrice de multiplication pour donner une nouvelle méthode de **calcul des sous-résultants** de polynômes usuels.

  Nous généralisons ces résultats aux polynômes de Ore. Nous introduisons pour cela la notion de **matrice de multiplication à droite**.

  Etant donnés deux polynômes de Ore A et B, la matrice de multiplication à droite par B modulo A est égale à la matrice de l'application linéaire qui à tout polynôme de Ore P de degré strictement inférieur à celui de A associe $PB \bmod A$. A partir de cette matrice nous définissons la notion de **résultant de polynômes de Ore** et nous en déduisons un nouvel **algorithme de recherche du PGCD à droite** pour deux polynômes de Ore.

  En nous inspirant des travaux de Li [Li], mais dans une forme plus simplifiée adaptée à notre travail, nous définissons la notion de **sous-résultants** pour des polynômes de Ore et nous donnons un nouvelle méthode de calcul des sous-résultants de polynômes de Ore à partir de la matrice de multiplication.

Dans la **Troisième partie : Métrique rang et cryptosystème LRPC**, nous redonnons les notions de base sur les codes correcteurs d'erreurs en métrique rang. Nous présentons les codes LRPC, une nouvelle famille de codes en métrique rang, et le cryptosystème LRPC, un cryptosystème de type McEliece basé sur ces codes.

- Dans le **Chapitre V**, nous redonnons les principaux résultats concernant la **métrique rang** pour un code linéaire, plus généralement un sous-espace vectoriel de $(\mathbb{F}_{q^m})^n$. On a établi l'isomorphisme $\mathbb{F}_{q^m} \cong (\mathbb{F}_q)^m$. Pour tout $\mathbf{x} = (x_1, ..., x_n)$ on écrit chaque $x_i \in \mathbb{F}_{q^m}$ comme un vecteur (colonne) de $(\mathbb{F}_q)^m$. On obtient ainsi une matrice à m ligne et n colonnes. On définit le **poids rang** de $\mathbf{x}$ comme le rang de cette matrice, et la **distance rang** entre deux vecteurs $\mathbf{x}$ et $\mathbf{y}$ comme le poids rang de $(\mathbf{x} - \mathbf{y})$.

  On montre que l'on définit bien ainsi une distance. On retrouve les paramètres d'un code linéaire pour cette distance. On montre que la métrique rang est bien adaptée à une utilisation en cryptographie. On redonne l'exemple des **codes de Gabidulin**, qui sont des codes d'évaluation de q-polynômes, et du cryptosystème GPT.

- Dans le **Chapitre VI**, nous définissons une nouvelle famille de codes en métrique rang, les **codes LRPC**. Les codes LRPC sont définis à partir de leur matrice de parité qui est une **matrice de poids rang faible** i.e. dont les composantes appartiennent à un sous-espace vectoriel de $\mathbb{F}_{q^m}$ de faible dimension d. Grâce à cette particularité nous sommes en mesure de donner un algorithme de décodage très rapide en nous ramenant à une multiplication matricielle dans $\mathbb{F}_q$.

  Nous définissons le **cryptosystème LRPC** qui est un cryptosystème de type McEliece basé sur ces codes offrant une bonne sécurité et, comme tous les cryptosystèmes basés sur la théorie des codes, une résistance aux attaques menées à l'aide d'ordinateurs quantiques. Les codes LRPC sont **faiblement structurés** ce qui rend ce cryptosystème résistant aux attaques structurelles. En prenant comme matrice de parité une matrice doublement circulante (on parle alors de codes **DC-LRPC**) on obtient une clé publique de petite taille.

Nous présentons une première attaque qui a été trouvée contre les codes DC-LRPC. Cette attaque utilise des **codes repliés** ce qui permet de chercher des éléments du dual dans un code de plus petite dimension. Nous expliquons comment contrer cette attaque par des paramètres bien choisis.

Première partie : Généralités

Chapitre I :
Notions essentielles en théorie des codes ; Applications à la cryptographie

Dans cette première partie nous réintroduisons les notions fondamentales de la théorie des codes correcteurs d'erreurs, qui représente le domaine d'étude de notre travail. Nous redonnons la *métrique de Hamming* qui permet de munir un code correcteur d'une structure d'espace métrique.

Nous travaillerons pour notre part essentiellement avec la *métrique rang* qui sera réintroduite dans un chapitre ultérieur. Nous rappelons dans le présent chapitre les définitions et les propriétés les plus importantes pour les codes munis de la métrique de Hamming ; nous redonnerons leurs équivalents en métrique rang dans le chapitre correspondant.

Nous redonnons également des notions essentielles en cryptographie. Nous rappelons comment nous pouvons utiliser une structure de code correcteur pour construire un *cryptosystème de McEliece*. Nous expliquons les attaques possibles contre un tel système lorsque l'on utilise la métrique de Hamming. Nous verrons que l'utilisation de la métrique rang permet de limiter ce type d'attaque.

Les démonstrations des résultats généraux de théorie des codes peuvent être trouvées par exemple dans [JH] ou [MS].

1 Théorie de l'information

L'ingénieur en télécommunications américain Claude Shannon pose les bases de la *théorie de l'information* dans un article publié à la fin des années 1940 [Sha] La théorie de l'information décrit les aspects les plus fondamentaux des systèmes de communication, à l'aide essentiellement de la théorie des probabilités.

1.1 Transmission d'un signal

La théorie des communications étudie les moyens de transmettre un message depuis une *source* jusqu'à un *destinataire*, via un *canal*.

La *source* peut être une voix (onde acoustique), une onde électromagnétique, une onde lumineuse (dans une fibre optique), un signal numérique (représenté en langage binaire)...

Le *canal* peut être une ligne téléphonique, une liaison radio, un support (bande magnétique, DVD...) Lors de la transmission par le canal, le signal est susceptible d'être dégradé ou modifié par un *bruit* (perturbations électriques parasitaires, rayures...).

Le *codage* du signal représente l'ensemble des opérations effectuées entre la sortie de la source et la transmission par le canal. On peut séparer le codage en deux parties : le *codage de source* et le *codage de canal*. Le codage de source consiste à représenter le signal en sortie de la source en un format adapté au canal, et ceci de la façon la plus économique possible. Le codage de canal consiste à modifier le signal de façon à ce qu'il soit transmis le plus fidèlement possible malgré le passage à travers le canal bruité.

C'est au codage de canal que l'on s'intéresse dans la théorie des codes correcteurs. Pour diminuer au maximum la probabilité de mauvaise transmission, l'idée fondamentale est d'ajouter de la *redondance* au message, du contenu supplémentaire qui n'ajoute aucune information, ne servant à rien d'autre que retrouver le message de départ.

Enfin, le *décodage* du signal consiste à restituer le plus fidèlement possible le signal de départ à partir du signal à la sortie du canal.

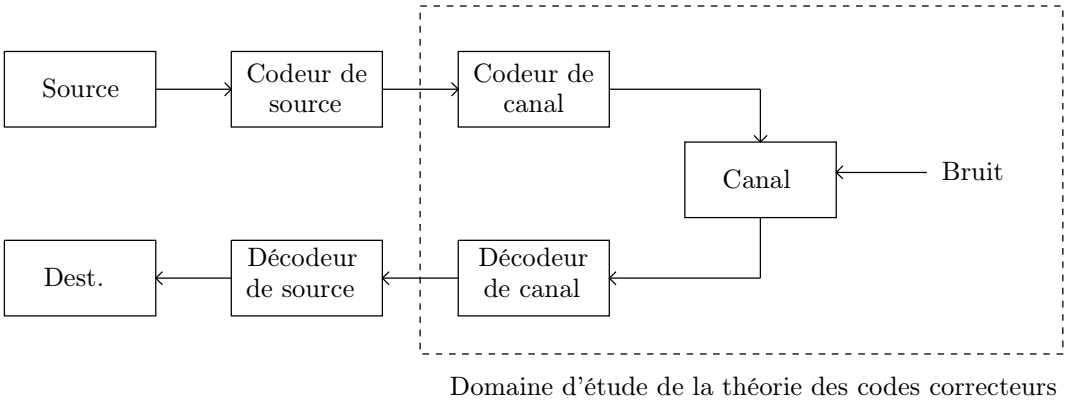


Figure 1. Système de communication avec codeurs de source et de canal séparés

1.2 Codage de canal

On se place dans le cas du *canal discret sans mémoire :* à l'entrée du canal, le signal est constitué d'une suite finie de symboles appartenant à un alphabet $\mathcal{A} = \{a_1, ..., a_K\}$. Une telle suite finie sera appelée un *mot*. A la sortie du canal, et suite aux perturbations éventuelles, le signal est une suite de symboles appartenant à un alphabet $\mathcal{B} = \{b_1, ..., b_J\}$ éventuellement égal à $\mathcal{A}$.

Ainsi, un canal peut être vu comme une variable aléatoire telle que chaque élément a_k de $\mathcal{A}$ entrant dans le canal peut donner à la sortie un des divers éléments de $\mathcal{B}$ avec une certaine probabilité dépendant uniquement de a_k (d'où la terminologie canal discret sans mémoire). Ainsi un canal discret sans mémoire est entièrement décrit par la donnée des probabilités conditionnelles $P(b_j|a_k)$ pour tous $k, j \in \{1...K\} \times \{1...J\}$.

Exemple 2. Nous nous placerons souvent dans le cas du *canal binaire symétrique* :

- Le signal à l'entrée est un train binaire. Chaque symbole entrant dans le canal est un bit égal à 0 ou 1 (alphabet d'entrée : $\mathcal{A} = \{0; 1\}$)
- De même, chaque symbole sortant dans le canal est égal à 0 ou 1 (alphabet de sortie : $\mathcal{B} = \mathcal{A} = \{0; 1\}$)
- A cause du bruit, chaque symbole est *mal transmis* (i.e. le symbole de sortie est différent du symbole d'entrée) avec une probabilité p et donc bien transmis avec une probabilité $(1 - p)$.

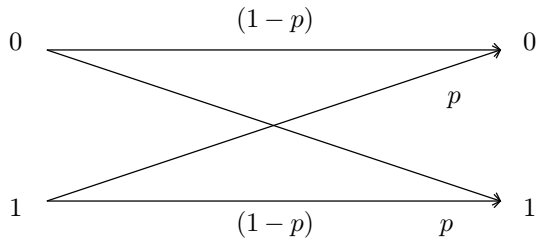


Figure 2. Canal binaire symétrique

Le modèle du canal binaire symétrique est simple à modéliser mathématiquement puisque indépendamment du symbole d'entrée, l'apparition d'une erreur revient à ajouter 1 dans le corps $\mathbb{F}_2$.

Exemple 3. Le canal $q-$aire symétrique est une généralisation du cas précédent :

- Les alphabets d'entrée et de sortie contiennent tous les deux q éléments ($\mathcal{A}=\mathcal{B}=\{a_1,...,a_q\}$)
- La probabilité qu'un symbole entrant soit bien transmis est $(1-p)$.
- Si un symbole entrant est mal transmis, les différentes possibilités de mauvaise transmission sont équiprobables (et donc de probabilité $\frac{p}{q-1}$).

Par exemple, le canal 3-aire symétrique peut se modéliser ainsi :

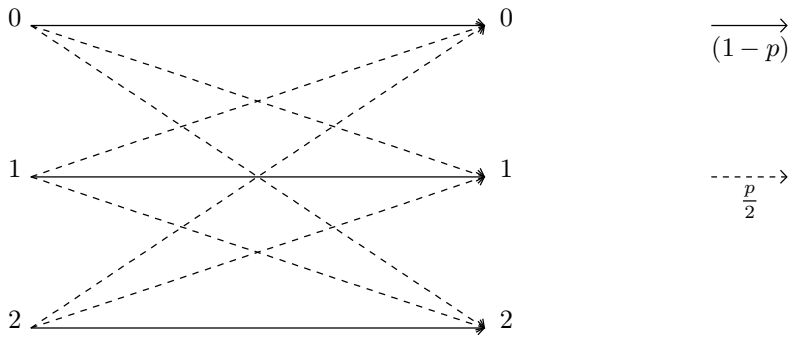


Figure 3. Canal 3-aire symétrique

2 Théorie des codes correcteurs d'erreurs

Dans ses travaux, Claude Shannon s'est intéressé à la probabilité de transmettre un message sans erreur à travers un canal bruité. Ces travaux nous renseignent sur la quantité minimale de redondance que doit contenir un code de canal, mais ne donnent pas vraiment de renseignements sur la façon de construire de tels codes qui peuvent être trop peu structurés.

On va donc chercher à définir des codes munis d'une structure mathématique, tels que le codage et le décodage puissent se faire par des opérations mathématiques ou à l'aide d'algorithmes. C'est l'objet de la *théorie des codes correcteurs d'erreurs*. Nous verrons que la structure d'*espace vectoriel sur un corps fini* est particulièrement bien adaptée à la construction de tels codes.

Si Shannon est considéré comme le père fondateur de la théorie de l'information, c'est à un autre mathématicien américain, Richard Hamming, que l'on doit les premiers travaux en théorie des codes. Après avoir travaillé pendant la deuxième guerre mondiale sur le projet Manhattan, Hamming entre en 1946 au département de mathématiques des laboratoires Bell, où il devient collègue de Claude Shannon :

> « [*At*] *Bell Labs I came into a very productive department. Bode was the department head at the time; Shannon was there ... I shared an office for a while with Shannon. At the same time he was doing information theory, I was doing coding theory. It is suspicious that the two of us did it at the same place and at the same time - it was in the atmosphere.* »

En 1947, Hamming utilise les calculateurs des laboratoires Bell pour lancer un long calcul qui doit être fait pendant un week-end. Le lundi matin, il découvre qu'une erreur s'est produite tôt dans les calculs, rendant tous ses résultats caducs. Il a alors l'idée d'ajouter à chaque bloc de bits un bit de contrôle de parité (*parity check bit*) afin de repérer les blocs de bits contenant une erreur. En ajoutant d'autres parity checks, il parvient à repérer la position des erreurs. Il formalise les bases de la théorie des codes en 1950 [Ham].

2.1 Code et principales caractéristiques d'un code

Définition 4. *Un* alphabet $\mathcal{A}$ *est un ensemble fini. On supposera sauf mention contraire que* $\mathcal{A}$ *est un corps fini.*

Un symbole *ou une* lettre *est un élément de l'alphabet.*

Un mot *noté* a *est une suite finie d'éléments de* $\mathcal{A}$*.*

L'alphabet pourra être le corps fini de Galois à deux éléments ($\mathcal{A} = \mathbb{F}_2 = \{0; 1\}$). C'est la modélisation mathématique naturelle pour un mot correspondant à une suite de bits. On rencontrera aussi les corps $\mathbb{F}_{q^m}$ où q est une puissance d'un nombre premier.

On tronçonne le train de lettres entrant dans le codeur canal en paquets de k lettres. A chaque mot de k lettres ainsi formé, l'opération de *codage canal* consiste à rajouter rajoute $(n-k)$ lettres dépendant uniquement des k lettres initiales.

Définition 5. *Un* codage (canal) *est une application de* $\mathcal{A}^k$ *dans* $\mathcal{A}^n$*. (*k *et* n *sont deux entiers naturels fixés avec* $k \leqslant n$*). On parle ici plus précisément de* codage systématique en bloc*, car on rajoute des blocs de* $(n-k)$ *à des blocs de* k *lettres. Les* $(n-k)$ *lettres supplémentaires sont appelées* lettres redondantes *ou* lettres de parité*.*

Un code *noté* $\mathcal{C}$ *est un sous-ensemble de* $\mathcal{A}^n$*.*

Le nombre n *est appelé la* longueur *du code.*

$M = |\mathcal{C}|$ *est la* taille *du code. On dit que* $\mathcal{C}$ *est de paramètres* (n, M)*.*

Le rendement *du code est égal à* $R = \frac{k}{n}$*. Il mesure la proportion de lettres utiles transmises par mot de code.*[1]

2.2 Structure d'espace métrique sur $\mathcal{C}$

On va voir que l'on peut considérer un code comme un espace métrique. On va alors chercher à construire des mots de code le plus « éloignés » possibles au sens la distance de cet espace métrique ; on verra en effet que plus les mots sont éloignés, mieux on arrive à décoder.

Définition 6. *Soit un code* $\mathcal{C}$ *de paramètres* (n, M) *sur l'alphabet* $\mathcal{A} = \mathbb{F}_{q^m}$*.*

Soient $\mathbf{a} = (a_1, ..., a_n)$ *et* $\mathbf{a}' = (a'_1, ..., a'_n)$ *deux mots du code.*

La distance de Hamming *entre* $\mathbf{a}$ *et* $\mathbf{a}'$ *notée* $d_H(\mathbf{a}, \mathbf{a}')$ *est égale au nombre de lettres différentes entre* $\mathbf{a}$ *et* $\mathbf{a}'$*.*

$$d_H(\mathbf{a}, \mathbf{a}') = \#\{(a_i, a'_i), i \in \{1...n\}, a_i \neq a'_i\}$$

Le poids de Hamming *de* $\mathbf{a}$ *noté* $w_H(\mathbf{a})$ *est égal au nombre de lettres du* a *non nulles.*

$$w_H(\mathbf{a}) = \#\{a_i, i \in \{1...n\}, a_i \neq 0\}$$

1. Le rendement se mesure normalement en bits utiles par lettre émise, et la formule est $R = \frac{k}{n}\log_2|\mathcal{A}|$. Comme chaque lettre correspond à $\log_2|\mathcal{A}|$ bits, on retrouve la formule $\frac{k}{n}$ en lettres utiles par lettre émise.

Remarque 7. La notion de lettre non nulle a un sens car on rappelle que l'alphabet est un corps fini donc muni d'un élément neutre pour l'addition.

La notation w est l'initiale du mot anglais weight (poids).

On peut définir le poids à partir de la distance ($w_H(\mathbf{a})$ est égal à la distance entre a et le mot nul) et réciproquement la distance à partir du poids ($d_H(\mathbf{a}, \mathbf{a}') = w_H(\mathbf{a} - \mathbf{a}')$ où la soustraction se fait composante par composante).

Exemple 8. Prenons $\mathcal{A} = \mathbb{F}_2$, $n = 6$, $\mathbf{a} = 010101$ et $\mathbf{a}' = 010110$, alors $w_H(\mathbf{a}) = w_H(\mathbf{a}') = 3$ et $d_H(\mathbf{a}, \mathbf{a}') = 2$

Proposition 9. *La distance de Hamming est une distance au sens classique du terme. Ainsi, un code $\mathcal{C}$ possède une structure d'espace métrique.*

Définition 10. *Soit un code $\mathcal{C}$ de paramètres (n, M) sur l'alphabet $\mathcal{A} = \mathbb{F}_{q^m}$.*

La distance minimale *de $\mathcal{C}$ notée d est égale à*

$$d = \min\{d_H(\mathbf{a}, \mathbf{b}); \mathbf{a}, \mathbf{b} \in \mathcal{C}, \mathbf{a} \neq \mathbf{b}\}$$

On rajoute d dans les paramètres du code. Ainsi les paramètres d'un code $\mathcal{C}$ seront notés (n, M, d).

Proposition 11. *Pour tout code $\mathcal{C}$, d existe et $d \in \mathbb{N}^*$.*

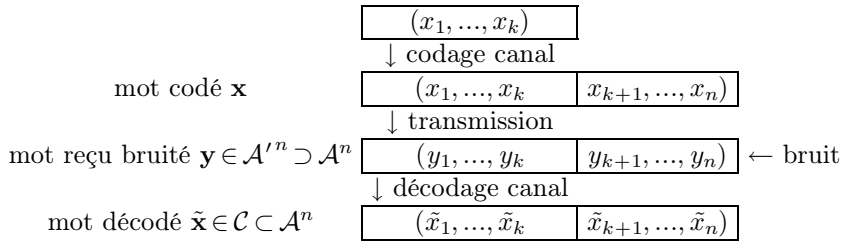
2.3 Décodage

Comme on l'a vu, lors du passage par le canal, certains symboles du mot sont susceptibles d'être mal transmis ou non transmis. Le *nombre d'erreurs* est le nombre de symboles mal transmis.

A la sortie du canal, le mot ainsi bruité, toujours de taille n, n'appartient plus nécessairement au code, mais à un ensemble $\mathcal{A'}^n$ contenant $\mathcal{A}^n$.

Le *décodage* consiste à associer à tout mot reçu (potentiellement bruité) un mot du code $\mathcal{C}$. Evidemment, le but est de retrouver ainsi le mot envoyé initialement. Nous allons voir que c'est possible si le taux d'erreur ne dépasse pas une certaine borne.

Soit un code $\mathcal{C}$ de paramètres (n, M) sur l'alphabet $\mathcal{A}$. Un *décodage* peut être vu comme une application de $\mathcal{A'}^n$ dans $\mathcal{C}$.



Si le mot est décodé correctement $\tilde{\mathbf{x}} = \mathbf{x}$.

Figure 4. Procédure de codage/transmission/décodage de paramètres $(n, M = |\mathcal{A}|^k)$. Certains symboles sont bien transmis ($y_i = x_i$), d'autres mal transmis ($y_i \neq x_i$)

Exemple 12. Le *code par bit de contrôle de parité de longueur n sur le corps* $\mathbb{F}_2$ est l'ensemble des éléments $(x_1, ..., x_n) \in \mathbb{F}_2^n$ où $x_1, ..., x_{n-1}$ sont des éléments quelconques de $\mathbb{F}_2$ et x_n vaut 0 (resp.1) s'il y a un nombre pair (resp. impair) de bits dans $x_1, ..., x_{n-1}$, c'est à dire plus précisément $x_n = \left(\sum_{i=1}^{n-1} x_i\right) \bmod 2$.

On se place à nouveau dans le cas du canal binaire symétrique étudié au chapitre précédent. On note p la probabilité de mauvaise transmission. On a vu qu'on pouvait toujours de ramener au cas où $p < \frac{1}{2}$ (sauf dans le cas particulier $p = \frac{1}{2}$ qui donne un code aléatoire).

Lemme 13. *Supposons que l'on reçoive le mot $\mathbf{y} = (y_1 ... y_n)$, alors le mot du code $\mathcal{C}$ le plus probablement émis est le mot $\mathbf{x} = (x_1 ... x_n)$ tel que la distance de Hamming entre $\mathbf{x}$ et $\mathbf{y}$ soit la plus petite possible.*

Autrement dit, à $\mathbf{y}$ fixé, le mot $\mathbf{x}$ maximisant $P(x|y)$ est le même que celui qui minimise $d(x, y)$.

On supposera donc que le mot envoyé est le mot le plus proche du mot reçu au sens de la distance de Hamming. On dit que l'on *décode au maximum de vraisemblance.*

Proposition 14. *Soit un code $\mathcal{C}$ de paramètres (n, M, d).*

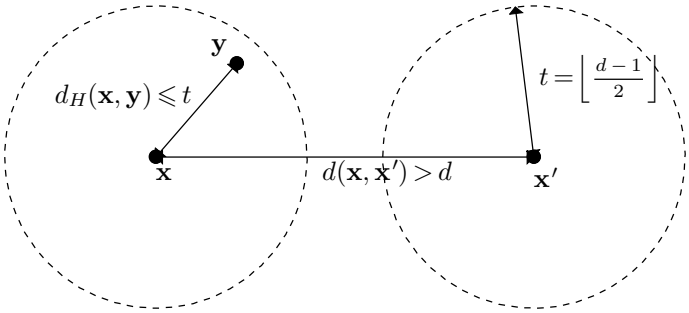
On suppose que l'on envoie un mot x et que l'on reçoit un mot $\mathbf{y}$.

Si le mot reçu $\mathbf{y}$ contient moins de $t = \left\lfloor \frac{d-1}{2} \right\rfloor$ erreurs, alors

- *Il existe un unique mot de $\mathcal{C}$ minimisant la distance de Hamming avec $\mathbf{y}$.*
- *Ce mot est égal au mot émis $\mathbf{y}$.*

Ainsi, si on décode au maximum de vraisemblance, le mot décodé est bien égal au mot envoyé initialement.

Démonstration. *Par définition de la distance minimale, les boules de rayon t centrées sur les mots du code sont disjointes. Si le mot reçu $\mathbf{y}$ contient moins de t erreurs, il existe donc une unique boule de rayon C centrée sur un mot du code contenant $\mathbf{y}$. Le centre de cette boule est le mot $\mathbf{x}$.*



□

On voit ainsi que la distance minimale est une notion fondamentale pour un code : plus elle est grande, plus le code va pouvoir corriger efficacement d'erreurs.

Définition 15. *Le nombre $t = \left\lfloor \frac{d-1}{2} \right\rfloor$ est appelé* capacité de correction *du code.*

On suppose que le mot émis après codage canal est $\mathbf{x} = (x_1, ..., x_n)$ et que le mot reçu est $\mathbf{y} = (y_1, ..., y_n)$. On peut écrire

$$\mathbf{y} = \mathbf{x} + \mathbf{e}$$

Le vecteur $\mathbf{e} = (e_1, ..., e_n) = \mathbf{y} - \mathbf{x}$ est appelé le *vecteur erreur*. Pour tout $i \in \{1...n\}$ dire que le symbole d'indice i est bien transmis revient à dire que $e_i = 0$, ainsi $y_i = x_i$. Au contraire, si $e_i \neq 0$ alors $y_i \neq x_i$ et le symbole est mal transmis.

Le poids du Hamming $w_H(\mathbf{e})$ du vecteur erreur est égal au nombre d'erreurs produites pendant la transmission. Ainsi, si $w_H(\mathbf{e}) < \left\lfloor \frac{d-1}{2} \right\rfloor$, le décodage au maximum de vraisemblance permet de retrouver le mot émis.

2.4 Codes linéaires

Les codes linéaires, qui représentent des sous-espaces vectoriels de $\mathcal{A}^n$, forment la quasi totalité des codes correcteurs.

On rappelle que l'alphabet $\mathcal{A}$ est un corps fini $\mathbb{F}_{q^m}$. L'ensemble $(\mathbb{F}_{q^m})^n$ est naturellement muni d'une structure d'espace vectoriel. Le corps $\mathbb{F}_{q^m}$ peut être pris comme corps des scalaires.

2.4.1 Généralités

Définition 16. *Un code $\mathcal{C}$ de longueur n sur l'alphabet $\mathcal{A} = \mathbb{F}_{q^m}$ est dit* linéaire *si $\mathcal{C}$ est un sous-espace vectoriel de $\mathcal{A}^n$.*

La dimension *du code notée k est égale à la dimension de $\mathcal{C}$ en tant qu'espace vectoriel.*

Pour un code linéaire $\mathcal{C}$ de longueur n et de dimension k, on note ses paramètres

$$[n, k, d]$$

où d représente la distance minimale du code telle que définie au paragraphe précédent.[2]

Proposition 17. *Si $\mathcal{C}$ est un code linéaire alors* $d = \min_{\mathbf{x} \in \mathcal{C}, \mathbf{x} \neq 0} w_H(\mathbf{x})$.

2.4.2 Bornes sur les paramètres d'un code

Nous redonnons deux bornes importantes sur les paramètres des codes linéaires :

La borne de Gilbert-Varshamov permet de déterminer des paramètres pour lesquels on est sûr qu'un code linéaire existe :

Proposition 18. *Borne de Gilbert-Varshamov :*

Il existe un code linéaire de longueur n, de dimension k et de distance minimale $\geqslant d$ sur l'alphabet $\mathbb{F}_q$ lorsque l'inégalité suivante est vérifiée :

$$q^{n-k} - 1 > \sum_{i=1}^{d-2} \binom{n-1}{i} (q-1)^i$$

De la proposition 17 on déduit la borne de Singleton :

Proposition 19. *Borne de Singleton :*

Notons $|\mathcal{C}|$ le cardinal du code $|\mathcal{C}|$, alors :

$$|\mathcal{C}| \leqslant q^{n-d+1}$$

Dans le cas d'un code linéaire $\mathcal{C}$ de paramètres $[n, k, d]$ on a $|\mathcal{C}| = q^k$ et ce qui précède implique

$$d \leqslant n - k + 1$$

2. Ainsi, lorsque l'on note les paramètres d'un code linéaire, on écrit sa dimension k au lieu de son cardinal M. Dans le cas d'un code linéaire $M = |\mathcal{A}|^k$.

Lorsque $d = n - k + 1$ on dit que le code est *parfait* ou *MDS* (Maximum Distance Separable).

2.4.3 Matrice génératrice, matrice de contrôle de parité

Définition 20. *Une* matrice génératrice *de $\mathcal{C}$, notée G, est une matrice à k lignes et n colonnes dont les vecteurs lignes forment une base de $\mathcal{C}$ en tant que sous-espace vectoriel de $\mathcal{A}^n$.*

Remarque 21. Un sous ensemble $\{j_1 ... j_k\}$ de $\{1...n\}$ tel que la matrice carrée formée des colonnes $(c_{j_1}, ..., c_{j_k})$ est inversible est appelé un *ensemble d'information.*

Si $\{1...k\}$ est un ensemble d'information on peut se ramener par des opérations sur les lignes à une matrice de la forme

$$G = \begin{array}{c} \\ k \text{ lignes} \end{array} \overbrace{\begin{pmatrix} 1 & 0 & \dots & 0 & \times & \dots & \times \\ 0 & 1 & & \vdots & \vdots & & \vdots \\ \vdots & & \ddots & 0 & & & \\ 0 & \vdots & 0 & 1 & \times & \dots & \times \end{pmatrix}}^{n \text{ colonnes}} = \left(\begin{array}{c|c} I_k & M \end{array} \right)$$

où M est une matrice de dimensions $k \times (n-k)$.

On dit qu'une telle matrice est sous *forme systématique*.

Exemple 22. Le code par bit de contrôle de parité de longueur n sur $\mathbb{F}_2$ est un code linéaire de paramètres $[n, n-1, 2]$. Il est MDS. Sa matrice génératrice est

$$\begin{pmatrix} 1 & 0 & \dots & 0 & 1 \\ 0 & 1 & & \vdots & 1 \\ \vdots & & \ddots & 0 & \vdots \\ 0 & \vdots & 0 & 1 & 1 \end{pmatrix}$$

Définition 23. *Soit un code linéaire $\mathcal{C}[n, k, d]$.*

Une matrice H de dimension $(n-k) \times n$ est une matrice de parité *pour $\mathcal{C}$ si elle vérifie*

$$\mathcal{C} = \ker(H)$$

(On fait l'analogie entre la matrice et l'application linéaire)

Autrement dit pour tout $\mathbf{x} \in (\mathbb{F}_{q^m})^n$ on a l'équivalence

$$\mathbf{x} \in \mathcal{C} \Leftrightarrow H\mathbf{x}^T = 0$$

où 0 désigne le vecteur colonne nul à $(n-k)$ composantes.

Proposition 24. *Expression de la matrice de parité :*

Considérons un code $\mathcal{C}[n, k, d]$ dont on donne la matrice génératrice G sous forme systématique $G = \left(\begin{array}{c|c} I_k & M \end{array} \right)$ où M est de dimension $k \times (n-k)$.

Alors la matrice

$$H = \left(\begin{array}{c|c} -M^T & I_{n-k} \end{array} \right)$$

est une matrice de parité de $\mathcal{C}$.

Exemple 25. Le *code de Hamming* $\mathcal{H}_m$ est un code linéaire dont on donne la matrice de parité. Cette matrice notée H_m est constituée des 2^m-1 vecteurs colonnes non nuls de $(\mathbb{F}_2)^m$. Par exemple pour $m=3$ on a

$$H_3=\begin{pmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{pmatrix}$$

On a choisi l'ordre des vecteurs colonnes de façon à écrire H_3 directement sous la forme $H_3 = (\ -{}^tM \ \mid \ I_3\ )$. Ceci permet d'écrire la matrice génératrice

$$G_3=\begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}=(\ I_4 \ \mid \ M\ )$$

Par construction de G_3 on voit que le code $\mathcal{H}_3$ est de longueur 7 et de dimension 4.

Dans la matrice M on trouve tous les éléments de $(\mathbb{F}_2)^3$ de poids supérieur ou égal à 2 de sorte que tout vecteur ligne de G_3 a un poids supérieur ou égal à 3. Si on additionne deux vecteurs lignes v_i et v_j distincts de G_3 le vecteur résultat a au moins 3 bits égaux à 1 : les bits $I_{4_{i,i}}$ et $I_{4_{j,j}}$ et comme M regroupe des vecteurs lignes distincts de $(\mathbb{F}_2)^3$ il existe au moins un indice k tel que $M_{i,k}\neq M_{j,k}$ de sorte que $M_{i,k}+M_{j,k}=1$. On a trouvé trois bits égaux à 1 donc la distance minimale d est supérieure ou égale à 3.

Comme il existe des mots de poids 3 on a finalement $d=3$.

Le code de Hamming $\mathcal{H}_3$ est un code sur $\mathbb{F}_2$ de paramètres $[7,4,3]$. Il est MDS.

On généralise ce résultat pour tout entier m :

Le code de Hamming $\mathcal{H}_m$ est un code sur $\mathbb{F}_2$ de paramètres $[2^m-1, 2^m-m-1, 3]$.

Remarquons que l'on peut utiliser la matrice de parité pour déterminer l'intersection de deux sous-espaces vectoriels de $\mathcal{A}^n$:

Proposition 26. *Soit $\mathcal{C}_1$ et $\mathcal{C}_2$ deux codes sur $\mathcal{A}^n$ (ce sont donc deux sous-espaces vectoriels quelconques de $\mathcal{A}^n$) de matrices de parité respectives H_1 et H_2. Alors la matrice $H=\begin{pmatrix} H_1 \\ H_2 \end{pmatrix}$définie par blocs est une matrice de parité de $\mathcal{C}_1\cap\mathcal{C}_2$.*

Démonstration. *Pour tout vecteur $\mathbf{v}=(v_1,\ldots,v_n)\in\mathcal{A}^n$ on a $Hv^t=\begin{pmatrix} H_1 \\ H_2 \end{pmatrix}v^t=\begin{pmatrix} H_1v^t \\ H_2v^t \end{pmatrix}$. Ce vecteur est nul si et seulement si $H_1\mathbf{v}^T$ et $H_2\mathbf{v}^T$ sont nuls tous les deux i.e. $v\in\mathrm{Ker}(H_1)=\mathcal{C}_1\cap\mathcal{C}_2=\mathrm{Ker}(H_2)$.* □

2.4.4 Code dual

Soient de vecteurs $\mathbf{v}=(v_1,\ldots,v_n)$ et $\mathbf{w}=(w_1,\ldots,w_n)$ de $\mathcal{A}^n$. On définit sur $\mathcal{A}^n$ la forme bilinéaire symétrique suivante :

$$\begin{array}{rccc} <.|.>: & \mathcal{A}^n\times\mathcal{A}^n & \to & \mathcal{A} \\ & (\mathbf{v},\mathbf{w}) & \mapsto & <\mathbf{v}|\mathbf{w}>=\displaystyle\sum_{j=1}^{n}v_jw_j \end{array}$$

On appellera *produit scalaire* sur $\mathcal{A}^n$ cette forme bilinéaire. On fait ainsi un léger abus de langage car le caractère défini n'est pas vérifié, il existe des vecteurs $\mathbf{v}\neq 0_{\mathcal{A}^n}$ tels que $<\mathbf{v}|\mathbf{v}>\geqslant 0$. De tels vecteur sont dits *isotropes*.

Généralisant la définition usuelle, on dit que $\mathbf{v}$ et $\mathbf{w}$ sont *orthogonaux* pour ce produit scalaire si $<\mathbf{v}|\mathbf{w}>=0$.

On en déduit la définition de *code dual :*

Définition 27. *Soit $\mathcal{C}$ un code (linéaire) de $\mathcal{A}^n$, le* code dual *de $\mathcal{C}$, noté $\mathcal{C}^\perp$, est l'ensemble des éléments de $\mathcal{A}^n$ orthogonaux à tous les éléments de $\mathcal{C}$.*

$$\mathcal{C}^\perp = \{\mathbf{w} \in \mathcal{A}^n ; \forall \mathbf{v} \in \mathcal{C}, <\mathbf{v}|\mathbf{w}>=0\}$$

On vérifie que $\mathcal{C}^\perp$ est aussi un code linéaire, plus précisément :

Proposition 28. *Soit $\mathcal{C}$ un code linéaire de paramètres $[k,n]$ de matrice génératrice G et de matrice de parité H, alors $\mathcal{C}^\perp$ est un code linéaire de paramètres $[n-k,n]$ de matrice génératrice H et de matrice de parité G. On a également $\mathcal{C}^{\perp\perp}=\mathcal{C}$.*

Ainsi, pour avoir une matrice génératrice de l'intersection de deux espaces vectoriels $\mathcal{C}_1$ et $\mathcal{C}_2$, on peut déterminer une matrice de parité de $\mathcal{C}_1$, une matrice de parité de $\mathcal{C}_2$, en déduire par concaténation une matrice de parité de $\mathcal{C}_1 \cap \mathcal{C}_2$, ce qui donne une matrice génératrice de $(\mathcal{C}_1 \cap \mathcal{C}_2)^\perp$, dont on reprend à nouveau une matrice de parité qui génère donc $\mathcal{C}_1 \cap \mathcal{C}_2$.

2.4.5 Exemples d'algorithmes de calcul pour la matrice de parité

Nous nous attardons un peu plus longuement sur les méthodes de calcul des matrices de parité qui jouent un rôle prépondérant dans les codes que nous introduisons au chapitre VI. Il y a deux cas où nous avons été amenés à calculer des matrices de parité :

D'une part, mentionnons le cas (fréquent) où $k=\frac{n}{2}$, pour lequel nous avons le résultat suivant :

Proposition 29. *Lorsque $k=\frac{n}{2}$ la matrice génératrice du code peut s'écrire par blocs $G=(A|B)$ où A et B sont des matrices carrées de dimension k.*

Si A et B sont inversibles (ce qui se produit dans le contexte où nous en avons besoin au chapitre VI) alors une matrice de parité du code est donnée par la matrice par blocs suivante :

$$H = \left(-{A^{-1}}^T | {B^{-1}}^T\right)$$

Démonstration. *La produit par bloc GH^T est égal à $(-AA^{-1}+BB^{-1})=(-I+I)=(0)$* □

L'autre situation où nous avons eu besoin, dans nos travaux, de calculer une matrice de parité, est une généralisation de la situation présentée dans la proposition (24). Dans la pratique, rien ne garantit que la matrice G puisse s'écrire sous forme systématique $G=(I|M)$. En effectuant des opérations sur les lignes de G (on reste dans une base de l'espace vectoriel engendré par les lignes de G) on peut écrire la matrice sous forme échelonnée de la façon suivante :

$$G = \begin{pmatrix}
0 & \dots & 1 & & 0 & & 0 & & \times & \dots & \times \\
0 & \dots & 0 & \dots & 1 & & \vdots & & \vdots & & \vdots \\
 & & \vdots & & 0 & \dots & 1 & \dots & \times & \dots & \times \\
 & & & & \vdots & & 0 & \ddots & & & \\
 & & & & & & \vdots & & 0 & \dots & 0 \\
 & & & & & & & & \vdots & & \vdots \\
 & & 0 & & 0 & & 0 & & 0 & \dots & 0
\end{pmatrix}$$

Quitte à enlever les dernières lignes de la matrice on peut supposer que G est de rang k.

Si l'on note m_j les vecteurs colonnes de G nous obtenons l'écriture suivante pour G après opérations sur les lignes :

$$G = \begin{array}{c} \\ \\ \\ \\ \end{array} \overset{\begin{array}{ccccccccccc} & & & & & & & & & & \\ & & & & & & & & & & \\ & & & & & & & & & & \\ & & & & & & & & & & \end{array}}{\begin{pmatrix} m_{11} & \dots & m_{1\,(c_1-1)} & 1_{c_1} & m_{1(c_1+1)} & \dots & m_{1(c_2-1)} & 0 & m_{1(c_2+1)} & \dots & 0 & \dots & m_{1(n-k)} \\ m_{21} & & \vdots & 0 & \vdots & & \vdots & 1_{c_2} & \vdots & & \vdots & & \vdots \\ \vdots & & & \vdots & & & & \vdots & & & 0 & & \\ m_{k1} & & m_{k\,(c_1-1)} & 0 & m_{k\,(c_1+1)} & & m_{k\,(c_2-1)} & 0 & m_{k(c_2+1)} & & 1_{c_k} & & m_{k(n-k)} \end{pmatrix}} \begin{array}{c} \\ \\ \\ \\ k \end{array}$$

Les k colonnes grisées d'indices $\{c_1, ..., c_k\}$ (ensemble d'information) sont les colonnes de la matrice identité I_k.

Nous en déduisons l'expression d'une matrice de parité :

Proposition 30. *Avec les notations précédentes pour la matrice G, on définit la matrice H suivante :*

$$H = \begin{array}{c} \\ \\ \\ \\ \\ \end{array} \overset{\begin{array}{ccccccccccc} & & & & & & & & & & \\ & & & & & & & & & & \\ & & & & & & & & & & \\ & & & & & & & & & & \\ & & & & & & & & & & \end{array}}{\begin{pmatrix} 1 & 0 & \dots & m_{11} & \dots & m_{21} & \dots & m_{31} & \dots & 0 \\ \vdots & 1 & & \vdots & & \vdots & & \vdots & & \vdots \\ & & \ddots & & & & & & & \\ & & & & & & & & & 0 \\ 0 & & & m_{1(n-k)} & & m_{2(n-k)} & & m_{3(n-k)} & & 1 \\ & & & c_1 & \dots & c_2 & \dots & c_k & & \end{pmatrix}} \begin{array}{c} \\ \\ \\ \\ \\ n-k \end{array}$$

La matrice H ainsi obtenue est une matrice de parité pour G.

Démonstration. *Si l'on effectue sur les indices la permutation circulaire* $\begin{pmatrix} c_1 & c_2 & & c_k & 1 & 2 & & n-k \\ \downarrow & \downarrow & \dots & \downarrow & \downarrow & \downarrow & \dots & \downarrow \\ 1 & 2 & & k & k+1 & k+2 & & n \end{pmatrix}$*, on voit que dans les coefficients du produit HG^T les termes qui interviennent dans la somme sont exactement les mêmes que dans la proposition (24) mais dans un ordre différent, ainsi H est bien une matrice de parité pour G.* □

On en déduit un l'algorithme suivant pour la génération d'une matrice de parité :

Entrée : La matrice G.

Sortie : Le rang de G, une matrice H de parité pour G.

- On écrit G sous forme échelonnée. Quitte à enlever des lignes on suppose que G est de rang k.
- Au fur et à mesure de la mise sous forme échelonnée on relève un ensemble d'information $\{c_1, ..., c_k\}$ pour G (par exemple en construisant itérativement l'entier $\sum_{i=1}^{k} 2^{c_k}$)
- On construit la matrice H de la façon suivante :
  - Les colonnes $c_1, ..., c_k$ de H sont formées des transposées des lignes $1, ..., k$ de G auxquelles on a enlevés les coefficients d'indice (i, c_i).
  - Les $(n-k)$ autres colonnes de H sont les colonnes de la matrice identité I_{n-k}.

2.5 Aspects algorithmiques du décodage ; exemples

Il est toujours possible de décoder un mot de code via un dictionnaire répertoriant tous les mots codés. Evidemment cette façon de procéder est très longue et dans la pratique on utilise la structure mathématique des codes pour trouver des algorithmes de décodage plus efficaces. Nous présentons rapidement deux algorithmes de décodage utilisables pour tout code linéaire ; d'autres algorithmes plus efficaces sont spécifiques au type de code utilisé, comme l'algorithme de Berlekamp-Welch pour les codes de Reed-Solomon.

2.5.1 Décodage d'un code linéaire par syndrome

On reprend les notations précédentes. On note $\mathbf{x} = (x_1, ..., x_n)$ le mot émis, $\mathbf{y} = (y_1, ..., y_n)$ le mot reçu et $\mathbf{e} = (e_1, ..., e_n)$ le vecteur erreur de sorte que

$$\begin{array}{ccc} \mathbf{y} & = & \mathbf{x} + \mathbf{e} \\ (y_1, ..., y_n) & = & (x_1 + e_1, ..., x_n + e_n) \end{array}$$

Définition 31. *Pour tout élément* $\mathbf{y} \in (\mathbb{F}_{q^m})^n$*, le* syndrome *de* $\mathbf{y}$ *noté* $S(\mathbf{y})$ *est le vecteur de* $(\mathbb{F}_{q^m})^{n-k}$ *défini par*

$$S: \begin{array}{ccc} (\mathbb{F}_{q^m})^n & \to & (\mathbb{F}_{q^m})^{n-k} \\ \mathbf{y} & \mapsto & S(\mathbf{y}) = H\,\mathbf{y}^T \end{array}$$

Remarque 32. Quitte à prendre la transposée nous supposerons que le syndrome est écrit en ligne.

On va définir à partir du syndrome une relation d'équivalence $\mathcal{R}$ telle que pour tous $\mathbf{y}, \mathbf{y}' \in (\mathbb{F}_{q^m})^n$ on a $\mathbf{y}\mathcal{R}\mathbf{y}'$ lorsqu'ils ont le même syndrome ($S(\mathbf{y}) = S(\mathbf{y}')$) ce qui revient à dire, l'opération de syndrome étant linéaire, que $S(\mathbf{y} - \mathbf{y}') = 0$, autrement dit $\mathbf{y} - \mathbf{y}' \in \mathcal{C}$ par définition de H.

Pour tout $\mathbf{y}$ notons $\mathcal{R}(\mathbf{y})$ la classe d'équivalence de $\mathbf{y}$ pour la relation $\mathcal{R}$.

On a vu que H est de rang $(n-k)$ donc tout élément de $(\mathbb{F}_{q^m})^{n-k}$ est un syndrome possible. Il y a donc $(q^m)^{n-k}$ classes d'équivalence pour cette relation.

Chaque classe d'équivalence comporte tous les éléments qui ont le même syndrome $\mathbf{s}$, c'est à dire tous les éléments de $\mathcal{C}$ auxquels on a ajouté $\mathbf{s}$. Autrement dit chaque classe d'équivalence comporte $(q^m)^k$ éléments.

Pour chacune des classes d'équivalence, l'élément de poids de Hamming minimal est appelé le *chef de classe*. Il n'est pas nécessairement unique.

Par définition de H on a pour tout mot x du code $S(x) = 0$. Ainsi pour tout mot reçu $\mathbf{y} = \mathbf{x} + \mathbf{e}$, $\mathbf{x} \in \mathcal{C}$, on a

$$S(\mathbf{y}) = H\,(\mathbf{x} + \mathbf{e}) = H\,\mathbf{x}^T + H\,\mathbf{e}^T = H\,\mathbf{e}^T = S(\mathbf{e})$$

Pour tout mot reçu $\mathbf{y}$ on calcule son syndrome $S(\mathbf{e})$. L'erreur $\mathbf{e}$ peut être égale à nimporte quel élément de la classe d'équivalence correspondante. Comme on décode au maximum de vraisemblance le vecteur erreur $\mathbf{e}$ le plus probable est le chef de classe qui est le vecteur erreur de poids minimal donnant $\mathbf{y}$ comme mot reçu.

On suppose finalement que le mot émis est

$$\mathbf{x} = \mathbf{y} - \mathbf{e}$$

où $\mathbf{e}$ est le chef de la classe $S(\mathbf{y})$.

2.5.2 Le problème du décodage par syndrome et sa complexité

Nous formalisons le problème du décodage par syndrome qui, dans une forme généralisée à la métrique rang, est un résultat fondamental dans notre travail :

Problème du décodage par syndrome (Syndrome Decoding problem, ou SD) :

Soit H une matrice de parité de dimension $(n-k)\times n$ sur $\mathbb{F}_{q^m}$ avec $k\leqslant n$ et soit $\mathbf{s}\in(\mathbb{F}_q)^{n-k}$ et un entier w :

Existe-t-il un élément $\mathbf{x}\in(\mathbb{F}_{q^m})^n$ tel que $w_H(\mathbf{x})\leqslant w$ et $H\mathbf{x}^T=\mathbf{s}$?

Si pour certaines valeurs ce problème peut être résolu en tamps polynomial, Berlekamp, McEliece et van Tilborg ont prouvé en 1978 dans [BET] que ce problème était NP-complet[3], ce qui légitime son utilisation dans un contexte cryptographique, nous parlerons notamment du cryptosystème de McEliece dans la partie suivante.

2.5.3 Décodage d'un code linéaire par ensemble d'information

Le décodage par ensemble d'information est basé sur le résultat suivant ([Ste], [BJMM]) :

Proposition 33. *Soit un code linéaire $\mathcal{C}[n,k,d]$ et $\{j_1,...,j_k\}$ un ensemble d'information. Soit $(y_1,...,y_k)$ un vecteur de $\mathcal{A}^k$. Il existe un unique mot $\mathbf{x}\in\mathcal{C}$ que $\forall i\in\{1...k\}, x_{j_i}=y_i$. Le calcul de $\mathbf{x}$ à partir de $\mathbf{y}$ se fait par inversion matricielle.*

On suppose que le mot transmis est $\mathbf{y}=\mathbf{x}+\mathbf{e}$ où $\mathbf{e}$ est une erreur de poids t. On choisit un ensemble d'information $\{j_1,...,j_k\}$. D'après ce qui précède il existe un unique mot de code $\tilde{\mathbf{x}}$ coincidant avec $\mathbf{y}$ sur l'ensemble d'information. Si $d(\tilde{\mathbf{x}},\mathbf{y})\leqslant t$ on décode $\mathbf{y}$ par $\tilde{\mathbf{x}}$, sinon on choisit un nouvel ensemble d'information.

Remarquons en particulier que si le vecteur erreur est nul sur l'ensemble d'information alors $\mathbf{y}$ coincide avec $\mathbf{x}$ sur cet ensemble et le décodage renvoie bien $\mathbf{x}$. On montre que ceci se produit avec une probabilité $\frac{\binom{n-t}{k}}{\binom{n}{k}}$. Si le poids de l'erreur est faible cette probabilité tend vers 1. Nous allons voir que ceci peut être exploité pour attaquer un cryptosystème de McEliece.

2.6 Codes cycliques

Soit q une puissance d'un nombre premier et m un entier, on considère $\mathbb{F}_{q^m}$ le corps fini à q^m éléments. On sait que $\mathbb{F}_{q^m}$ peut être vu comme l'anneau quotient $\mathbb{F}_{q^m}[X]/(P)$ où P est le polynôme primitif d'un élément α qui engendre $\mathbb{F}_{q^m}^*$. Des détails sur ces notions sont donnés au chapitre suivant.

Définition 34. *Un code linéaire $\mathcal{C}$ de longueur n sur le corps $\mathbb{F}_{q^m}$ est dit* cyclique *s'il est stable par permutation circulaire des lettres de chaque mot i.e.*

$$(a_0,a_1,...,a_{n-1})\in\mathcal{C}\Rightarrow(a_{n-1},a_0,...,a_{n-2})\in\mathcal{C}$$

A chque mot $\mathbf{a}=(a_0,a_1,...,a_{n-1})$ du code on peut associer le polynôme $a(X)$ de $\mathbb{F}_{q^m}[X]$ défini par

$$\mathbf{a}(X)=a_0+a_1X+\cdots+a_{n-1}X^{n-1}$$

de sorte que le code $\mathcal{C}$ peut être vu comme un sous espace vectoriel de $(\mathbb{F}_{q^m}[X])_{n-1}$.

Ainsi le code est cyclique si le polynôme $\tilde{a}(X)$ suivant est lui aussi associé à un mot du code :

$$\tilde{\mathbf{a}}(X)=a_{n-1}+a_0X+\cdots+a_{n-2}X^{n-1}$$

Or le polynome $\tilde{\mathbf{a}}(X)$ est égal au reste du polynôme $X\mathbf{a}(X)$ dans la division euclidienne par X^n-1, de sorte que $\mathbf{a}(X)=\tilde{\mathbf{a}}(X)$ si on les identifie à leurs images par la surjection canonique dans l'anneau quotient $\mathbb{F}_{q^m}[X]/(X^n-1)$.

3. Des rappels sur la difficulté des problèmes sont redonnés dans la partie suivante.

Ainsi, en identifiant chaque mot $\mathbf{a}$ avec le polynôme $\mathbf{a}(X)$, on obtient que $\mathcal{C}$ est cyclique s'il est stable par multiplication par X, soit, par linéarité, s'il est stable par la multiplication par tout polynôme. Autrement dit :

Proposition 35. *Un code cyclique de longueur n sur le corps $\mathbb{F}_{q^m}$ est un idéal de $\mathbb{F}_{q^m}[X]/(X^n - 1)$.*

Exemple 36. Le *code de Reed-Solomon* de longueur n et de dimension k sur $\mathbb{F}_{q^m}$ a été introduit dans [RS] comme étant l'ensemble

$$\mathcal{C} = \{(P(a_1), ..., P(a_n)), P \in \mathbb{F}_{q^m}[X], \deg(P) < k\}$$

où les $a_1, ..., a_n$ sont k éléments distincts de $\mathbb{F}_{q^m}$.

Les codes de Reed-Solomon sont MDS.

On montre que si α est un élément primitif de $\mathbb{F}_{q^m}$, ce code admet pour matrice génératrice une matrice de Vandermonde égale à

$$G = \begin{pmatrix} 1 & 1 & 1 & ... & 1 \\ 0 & \alpha & \alpha^2 & ... & \alpha^n \\ 0^2 & \alpha^2 & (\alpha^2)^2 & ... & (\alpha^n)^{k-1} \\ \vdots & \vdots & \vdots & & \vdots \\ 0^{k-1} & \alpha^{k-1} & (\alpha^2)^{k-1} & ... & (\alpha^n)^{k-1} \end{pmatrix}$$

Dans [BW] Berlekamp et Welch proposent un algorithme de décodage des codes de Reed-Solomon.

Dans [Sud] et [GS] Gurusami et Sudan proposent un algorithme de décodage en liste de ces codes. Cet algorithme est une généralisation de l'algorithme de Berlekamp-Welch.

3 Utilisation des codes correcteurs dans un contexte cryptographique

3.1 Notion de cryptosystème

3.1.1 Définition

La *cryptographie* désigne l'ensemble des techniques permettant de *chiffrer* des messages (via l'utilisation d'un *algorithme de chiffrement* nécessitant une *clé de chiffrement*), c'est à dire de les rendre incompréhensibles sans une action du destinataire légitime du message (habilité à le *déchiffrer* à l'aide d'une *clé de déchiffrement*) ou d'une tierce personne qui peut essayer *d'attaquer* le message, c'est à dire d'en trouver le contenu sans connaissance de la clé de déchiffrement.

Un *cryptosystème* peut être modélisé de la façon suivante :



Figure 5. Cryptosystème

La transmission d'un message sous forme chiffrée permet de répondre à des contraintes de *confidentialité* (seul le destinataire légitime doit pouvoir accéder au contenu du message), d'*authenticité* (assurer l'identité de la source) et d'*intégrité* (seules les personnes ayant accès au contenu du message sont habilitées à modifier son contenu).

En 1883 le cryptologue Auguste Kerkchoffs énonce le principe, en vigueur depuis, selon lequel les algorithmes de chiffrement et de déchiffrement peuvent être connus publiquement, seules les clés devant être tenues secrètes, notamment la clé de déchiffrement.

3.1.2 Cryptosystèmes symétriques, cryptosystèmes asymétriques

Les cryptosystèmes *symétriques* sont les plus anciens. Dans un cryptosystème symétrique, la clé de chiffrement et la clé de déchiffrement sont identiques. Pour qu'un système symétrique soit inconditionnellement sûr il faut que la taille de la clé soit égale à la taille du message. Ainsi, le cryptosystèmes symétriques ne sont dans la pratique jamais utilisés seuls. On les utilise dans des systèmes hybrides, pour transmettre des clés privées de cryptosystèmes asymétriques :

En 1976 Diffie et Hellman introduisent les algorithmes de chiffrement et de déchiffrement *asymétriques* : la clé de chiffrement est différente de la clé de déchiffrement, et la connaissance de la première ne permet pas de retrouver la seconde. La clé de chiffrement peut ainsi être connue de tous ; on parle de *clé publique* pour la clé de chiffrement et de *clé privée* pour la clé de déchiffrement. Les algorithmes asymétriques sont basés sur des problèmes complexes c'est à dire dont il n'existe pas de méthode suffisamment rapide pour les résoudre, les exemples les plus courants étant la factorisation de grands nombres ou la détermination de la réciproque de certaines fonctions dans un anneau quotient. Les algorithmes asymétriques sont plus lents et nécessitent des clés de grande taille mais ne nécessitent pas que l'émetteur et le destinataire partagent le secret de la clé.

3.1.3 Sécurité d'un cryptosystème

En essayant de manière exhaustive toutes les combinaisons possibles, il est théoriquement possible pour un attaquant de trouver la clé de déchiffrement, d'autant plus que pour garantir de bonnes performances aux algorithmes on essaie de limiter la taille des clés. Pour qu'un procédé de chiffrement soit sûr il est donc nécessaire qu'une attaque exhaustive ne puisse pas être faite dans un temps « raisonnable ». Nous allons préciser ce point dans le paragraphe suivant.

Nous considèrerons notamment lorsque nous présenterons les cryptosystème LRPC qu'un cryptosystème est sûr lorsqu'une attaque exhaustive nécessite un nombre d'opérations de l'ordre de 2^{80}. La puissance de calcul des ordinateurs évoluant avec le temps ce nombre est évidemment appelé à augmenter.

Cependant, il est toujours possible qu'un attaquant trouve un autre algorithme de *décryptage* plus efficace lui permettant de *décrypter* le message (c'est à dire d'en déterminer le contenu sans avoir au préalable la clé de déchiffrement). On considère qu'un cryptosystème est viable lorsque la complexité du décryptage croît exponentiellement en fonction de la taille des paramètres alors que la complexité du chiffrage et du déchiffrage croît polynomialement.

3.1.4 Notion de problème difficile

La sécurité des cryptosystèmes repose sur donc le fait que le décryptage est un problème « difficile » à résoudre. Cette notion de « difficulté » est reliée à la *classe de complexité* des problèmes étudiés. Rappelons les principales classes de complexité (voir [AR] pour des détails) :

- Classe P : Ce sont les problèmes qui peuvent être résolus algorithmiquement en temps polynomial. Ils sont considérés comme « faciles ».

- Classe NP : Cette classe est associée aux problèmes décisionnels (ceux auquels on doit répondre par oui ou non). Un problème de décision revient à décider si un mot fait partie d'un langage donné. On désigne souvent le problème de savoir si un mot appartient à un langage Q par la même lettre Q. Soit Q un problème de décision, il est dans la classe NP s'il existe un algorithme A tel que :

a) à chaque mot du langage Q (i.e. à chaque instance I tel que $I \in Q$, i.e. les mots I pour lesquels la réponse est « oui ») un certificat $C(I)$ est associé tel que si la paire $(I, C(I))$ est donnée en entrée de A, celui-ci reconnait que $I \in Q$;

b) si I est un mot qui n'appartient pas à Q, alors il n'existe aucun certificat tel que A reconnaisse I ;

c) l'algorithme A, s'arête en temps polynomial.

Pour être clair, la classe NP est la classe des problèmes de décision pour lesquels il est facile de vérifier qu'une solution est correcte. Ici, on ne parle pas de trouver une solution, mais uniquement de la vérifier.

- Classe NP-difficile (NP-hard) : Problèmes au moins aussi difficiles que tous les problèmes de la classe NP.

- Classe NP-complet (NP-complete) : Sous-classe de NP vérifiant que tous les problèmes NP-complets se ramènent à tout problème NP-complet via une réduction polynomiale. Les problèmes NP-complets sont aussi NP-difficiles.

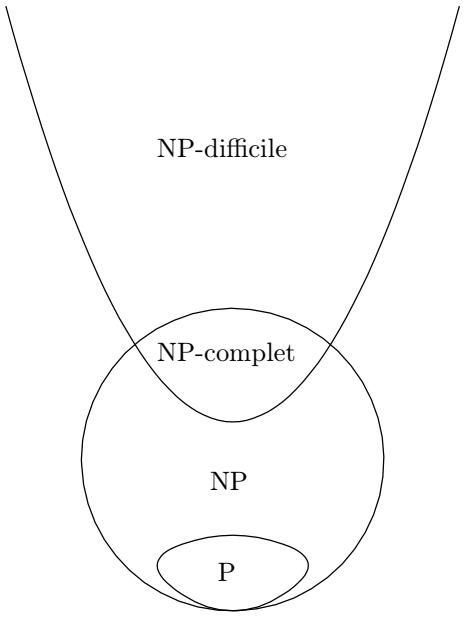


Figure 6. Représentation ensembliste des différentes classes de complexité en supposant P≠NP.

3.2 Application des codes correcteurs à la cryptographie : le cryptosystème de McEliece

3.2.1 Présentation du cryptosystème

Les cryptosystèmes de McEliece introduit dans [McEl] sont basés sur la fait que le décodage d'un *code aléatoire* (c'est à dire dont la matrice génératrice est générée aléatoirement) est un problème difficile. L'algorithme de déchiffrement va consister à décoder un mot sur un code dont la structure est masquée. Si ce code est indiscernable d'un code aléatoire, un attaquant ne dispose

donc pas d'algorithme efficace pour décoder le mot. La structure du code va faire office de clé de déchiffrement.

- **Génération de clés :**

  On choisit un code $\mathcal{C}$ sur $\mathbb{F}_{q^m}$ de matrice génératrice G. On suppose que l'on sait décoder t erreurs pour $\mathcal{C}$. On génère aléatoirement R, une matrice inversible de dimension $k \times k$ sur $\mathbb{F}_{q^m}$ et P, une matrice de permutation de dimension $n \times n$.

  La clé publique est

  $$G' = RGP$$

  La clé privée est constituée de R, G et P.

- **Chiffrement :**

  On veut transmettre le message $\mathbf{x}$. L'émetteur génère un vecteur erreur $\mathbf{e}$ aléatoire de poids t et envoie

  $$\mathbf{y} = \mathbf{x}G' + \mathbf{e}$$

- **Déchiffrement :**

  Connaissant la clé privée, on calcule $\mathbf{y}P^{-1} = \mathbf{x}RG + \mathbf{e}P^{-1}$ qui correspond un mot du code auquel on ajouté t erreurs ; on décode l'erreur ce qui permet de retrouver $\mathbf{x}RG$, puis le message initial pré-encodage $\mathbf{x}R$ et enfin $\mathbf{x}$ par multiplication par R^{-1}.

Figure 7. Cryptosystème de McEliece

En 1986 [Nie] Niederreiter propose un cryptosystème équivalent du point de vue de la sécurité et un peu plus efficace en termes de calculs :

- **Génération de clés :**

  On choisit un code $\mathcal{C}$ sur $\mathbb{F}_{q^m}$ de matrice de parité H. On suppose que l'on sait décoder t erreurs pour $\mathcal{C}$. On génère aléatoirement R, une matrice inversible de dimension $(n-k) \times (n-k)$ sur $\mathbb{F}_{q^m}$ et P, une matrice de permutation de dimension $n \times n$.

  La clé publique est

  $$H' = RHP$$

  La clé privée est constituée de R, H et P. Le destinataire doit aussi connaître un algorithme de décodage par syndrome pour $\mathcal{C}$.

- **Chiffrement :**

  On veut transmettre le message $\mathbf{x}$. On suppose que $w_H(\mathbf{x}) \leqslant t$ (x représente une erreur décodable), on envoie le syndrome de $\mathbf{x}$:

  $$\mathbf{y} = H'\mathbf{x}^T$$

- **Déchiffrement :**

  Connaissant la clé privée, on calcule $R^{-1}\mathbf{y} = HP\mathbf{x}^T$, puis on retrouve $P\mathbf{x}^T$ à l'aide du décodage par syndrome, et enfin on retrouve x en multipliant à gauche par P^{-1}.

Figure 8. Variante de Niederreiter

3.2.2 Sécurité du cryptosystème de McEliece

Un attaquant a deux approches pour tenter de décrypter le système : soit il essaie de décoder directement le mot du code (en tentant par exemple une attaque par ensemble d'information), ce qui est difficile si le code est indiscernable d'un code aléatoire, soit il essaie de trouver une décomposition valide de la clé publique (attaque structurelle), ce qui est difficile si la structure du code est difficile à déterminer.

La sécurité de ce cryptosystème dépend donc de la famille de codes utilisée. Les codes de Goppa [CFS] sont supposés indiscernables de codes aléatoires et représentent donc naturellement de bons candidats. Cependant la capacité de correction t des codes de Goppa est très basse et l'utilisation de la métrique de Hamming rend alors le système vulnérable aux attaques par ensemble d'information à moins de choisir une clé publique de grande taille importante ce qui est déjà, on va le voir, le principal point faible des cryptosystèmes de McEliece.

Les codes de Reed-Solomon, au contraire des codes de Goppa, sont optimaux, et constituent théoriquement de bons candidats pour le cryptosystème de McEliece : pour une clé publique de taille relativement faible, on peut introduire des erreurs de poids plus importants, ce qui augmente la résistance aux attaques par ensemble d'information. Cependant ces codes sont tellement structurés qu'il est possible d'effectuer une attaque structurelle en temps polynomial [SiSh] ce qui les rend inutilisables en pratique.

3.2.3 Avantages et inconvénients de la cryptographie basée sur les codes correcteurs

Le cryptosystème de McEliece est très peu utilisé en pratique pour plusieurs raisons, principalement à cause de sa clé publique qui est une matrice dont la taille est de $nk \log_2 (q^m)$. Cette taille est beaucoup trop importante par rapport aux tailles des clés publiques d'autres systèmes asymétriques : pour une sécurité de 2^{128} bits, par exemple, le cryptosystème classique RSA nécessite une clé publique (représentant un grand nombre) de quelques milliers de bits alors que l'on dépasse le million de bits pour un cryptosystème de McEliece.

Cependant le cryptosystème de McEliece présente plusieurs avantages. On peut notamment mentionner que les opérations de chiffrement et de déchiffrement, correspondant à des opérations d'encodage et de décodage, peuvent être faites de manière très rapide. Mais surtout, comme il est basé sur un problème difficile lié au codes linéaires, il est résistant aux attaques effectuées à l'aide d'ordinateurs quantiques permettant d'attaquer facilement des cryptosystèmes basés sur les nombres premiers ou le logarithme discret.

4 Conclusion

La cryptographie basée sur les codes correcteurs présente les deux avantages de permettre un chiffrement et un déchiffrement extrêmement rapide et d'être résistante à certaines attaques, contrairement aux cryptosystèmes basés sur des problèmes tels que la factorisation des grands nombres premiers. Nous ne disposons pas de la technologie pour lancer ces attaques mais si les ordinateurs quantiques deviennent une réalité nous disposons déjà de cryptosystèpes viables.

Le principal inconvénient des cryptosystèmes basés sur les codes correcteurs est la taille des clés : dans un tel cryptosystème, la clé est une matrice, dans le stockage des coefficients nécessite dans l'absolu une place très importante. C'est principalement pour cette raison que la cryptographie basée sur les codes correcteurs reste peu utilisée en pratique.

Au chapitre 5 nous présentons une nouvelle métrique : la métrique rang, qui, nous le verrons, permet d'avoir de « petites » erreurs qui ont malgré tout un impact sur toutes les composantes du mot et ont par exemple l'avantage de rendre les cryptosystèmes basés sur cette métrique résistants aux attaques par ensemble d'information. Nous redonnerons aussi d'autres résultats en cryptographie basée sur les codes correcteurs afin de mettre en avant ce qui a motivé notre travail.

Au chapitre 6 nous présentons un cryptosystème basé sur une nouvelle famille de codes en métrique rang, faiblement structurée, qui, en utilisant des matrices adaptées, peut être muni d'une clé dont la taille est de l'ordre de grandeur du millier de bits et pour laquelle nous disposons d'un algorithme de décodage efficace.

Chapitre II :
Construction des corps finis

La structure de corps fini est fondamentale dans notre travail, notamment dans l'étude des codes correcteurs en métrique rang. Aussi, il nous a paru pertinent de rappeler des résultats déjà connus sur l'existence et l'unicité corps finis et d'en détailler la construction effective afin d'avoir un contexte d'étude clair dans les chapitres ultérieurs.

Dans cette partie nous redonnons les résultats principaux sur la structure des corps finis, notamment que tout corps fini a pour cardinal une puissance d'un nombre premier , est unique à isomorphisme près , et que lorsque m_1 divise m_2 le corps à m_2 éléments peut être vu comme un espace vectoriel sur le corps à m_1 éléments.

Nous rappelons comment construire effectivement cette structure d'espace vectoriel, qui sera utilisée lors de l'implémentation des codes en métrique rang, notamment dans la dernière partie pour le décodage des codes LRPC. Nous donnons des exemples sur lesquels nous nous baserons pour les calculs en métrique rang.

Des détails et les démonstrations des résultats peuvent être trouvées par exemple dans [MS] ou [Zém].

1 Idée de base : quotient d'un anneau de polynômes par un polynôme irréductible

Soit $\mathbb{K}[X]$ l'anneau des polynômes en une indéterminée X à coefficients dans un corps $\mathbb{K}$. L'identité de Bézout est vérifiée dans cet anneau, pour deux polynômes A et $B \in \mathbb{K}[X]$ de degrés respectifs d_A et d_B, il existe deux polynômes U et V tels que $AU + BV = \mathrm{PGCD}(A, B)$. En particulier si A est un polynôme irréductible, nous avons pour tout polynôme B qui ne soit pas un multiple de A l'égalité $AU + BV = 1$.

Considérons l'anneau quotient $\mathbb{K}[X]/(A)$ constitué formellement des restes dans la division euclidienne des polynômes de $\mathbb{K}[X]$ dans la division euclidienne par A. Il s'agit d'un anneau que l'on peut aussi voir comme un espace vectoriel dont une base est $\{1, X, ..., X^{d_A-1}\}$ (en particulier le neutre pour la multiplication est dans $\mathbb{K}[X]/(A)$). L'égalité de Bézout précédente transposée dans cet anneau quotient devient $BV = 1$ pour tout B qui n'est pas un multiple de A. Ceci permet de trouver un inverse à tout élément B dans $\mathbb{K}[X]/(A)$. Nous construisons ainsi un corps à $|\mathbb{K}|^{d_A}$ éléments.

Nous allons rappeler rapidement dans la suite pourquoi nous pouvons ainsi construire tous les corps finis, dont le cardinal est nécessairement une puissance d'un nombre premier, deux corps de même cardinal étant isomorphes.

2 Cardinal d'un corps fini

Tout corps fini a pour caractéristique un nombre premier p. Le corps $\mathbb{F}_p$ est isomorphe à $\mathbb{Z}/p\mathbb{Z}$.

Si L et K sont deux corps avec $L \subset K$ alors K possède une structure de L-espace vectoriel de dimension finie $\dim_L(K)$. Ainsi, Si L contient q éléments, K contient $q^{\dim_L(K)}$ éléments. On en déduit :

- Tout corps fini K est une extension de son sous-corps premier $\mathbb{F}_p$ et son cardinal vaut donc $p^r, r \in \mathbb{N}$.
- Si $L \subset K$ avec $\#L = p^s$ et $\#K = p^r$ alors s divise r (en effet $p^r = (p^s)^{\dim_L(K)} = p^{s \dim_L(K)}$)

$$\mathbb{F}_p \subset L = \mathbb{F}_{p^s} \subset K = \mathbb{F}_{p^r} \text{ avec } s{=}rm$$

Exemple 37. Le corps $\mathbb{F}_{16} = \mathbb{F}_{2^4}$ peut être vu comme un $\mathbb{F}_2$-espace vectoriel ou un $\mathbb{F}_4$-espace vectoriel mais pas un $\mathbb{F}_8$-espace vectoriel.

Dans notre travail les notations standards pour un corps fini seront $\mathbb{F}_q$ ou $\mathbb{F}_{q^m}$ où q est une puissance d'un nombre premier p.

3 Groupe multiplicatif d'un corps fini

Le groupe multiplicatif d'un corps fini $K = \mathbb{F}_q$, noté $\mathbb{F}_q^*$, est cyclique d'ordre $q-1$.

On note α un générateur de ce groupe. On dit que α est un élément primitif de $\mathbb{F}_q$.

On peut ainsi décrire la totalité des éléments d'un corps fini :

$$\mathbb{F}_q = \{0, 1, \alpha, \alpha^2, ..., \alpha^{q-2}\}$$

Tout élément $x \in \mathbb{F}_q^*$ vérifie $x^{q-1} = 1$ soit $x^q = x$.

4 Unicité d'un corps fini à isomorphisme près

De ce qui précède on déduit :

Théorème 38. *Si s est un diviseur de r, il existe à isomorphisme près un seul sous-corps de $\mathbb{F}_{p^r}$ de cardinal $\mathbb{F}_{p^s}$. C'est l'ensemble des $x \in \mathbb{F}_{p^r}$ tels que $x^{p^s} = x$ soit $x^{p^s} - x = 0$*

En particulier pour $s = r$:

Théorème 39. *A isomorphisme près il existe un seul corps de cardinal p^r, c'est le corps de décomposition du polynôme $X^{p^r} - X$.*

5 Construction effective

Soient un corps L et un élément β appartenant à une extension de L. Le plus petit corps contenant L et β est appelé extension de L par β et noté $L(\beta)$.

On note I_β l'ensemble des polynômes de $L[X]$ ayant β comme racine. I_β est un idéal (non vide) de $L[X]$, principal car $L[X]$ est euclidien.

Le générateur de cet idéal (que l'on choisit de coefficient dominant 1) est appelé polynôme minimal de β sur L. Il est irréductible dans $L[X]$. On le note $\text{irr}(\beta, L, X)$.

Notons $L[\beta]$ l'ensemble obtenu par évaluation des polynômes de $L[X]$ en β. Comme $\text{irr}(\beta, L, \beta) = 0$ on obtient en prenant le reste dans la division euclidienne par $\text{irr}(\beta, L, X)$ l'isomorphisme de corps

$$L[\beta] \;\cong\; L[X]/I_\beta$$

$L[\beta]$ contient des combinaisons linéaires de puissances de β à coefficients dans L, ces éléments sont dans $L(\beta)$ car $L(\beta)$ est un corps donc $L[\beta] \subset L(\beta)$ et comme $L(\beta)$ est le plus petit corps contenant L et β on a

$$L(\beta) = L[\beta]$$

L'ensemble $L(\beta)$ est donc un L-espace vectoriel de dimension $\deg(\text{irr}, \beta, L, X)$ dont une base est $\left\{1, \beta, ..., \beta^{\deg(\text{irr},\beta,L,X)-1}\right\}$.

Il reste une question importante : comment choisir l'élément β et le polynôme minimal associé ? Dans le cas où l'on souhaite construire $\mathbb{F}_{p^m} = (\mathbb{F}_p)^m$ où p est un nombre premier, on choisit ce que l'on appelle un *polynôme primitif*, c'est l'objet du paragraphe suivant.

Dans le cas où l'on souhaite construire $\mathbb{F}_{q^m} = (\mathbb{F}_q)^m$ où $q = p^r$ n'est pas un nombre premier mais une puissance d'un nombre premier, le meilleur moyen pour obtenir un polynôme irréductible P sur $\mathbb{F}_q$ est de générer aléatoirement des polynômes, puis de tester leur irréductibilité. La probabilité qu'un polynome de degré n soit irréductible sur $\mathbb{F}_q$ vaut environ $\frac{1}{n}$ ce qui donne un nombre d'essais raisonnable[4] ; de plus les tests d'irréductibilité sur des polynômes à coefficients dans des corps finis peuvent être faits très rapidement, bien plus que les tests de primalité d'entiers par exemple. Une fois que l'on a un polynôme irréductible P on peut construire l'anneau quotient $\mathbb{F}_q[X]/(P)$.

Comme $\mathbb{F}_{q^m} = \mathbb{F}_{(p^r)^m} = \mathbb{F}_{p^{rm}}$, on peut aussi simplement construire $(\mathbb{F}_p)^{rm}$, puisqu'on a établi qu'il y avait de toutes façons isomorphisme entre $(\mathbb{F}_{p^r})^m$ et $(\mathbb{F}_p)^{rm}$.

Il existe des méthodes de construction beaucoup plus efficaces d'un point de vue de l'implémentation où les opérations dans le corps sont hard-codées, nous verrons ce point par la suite.

6 Cas particulier important : construction d'un corps par adjonction d'un élément primitif à son sous-corps premier

Dans cette partie on s'intéresse à la construction des corps comme extensions de leurs sous -corps premiers, soit avec les notations précédentes $s = 1$ et $r = m$. Afin de pouvoir faire directement certains calculs dans les paragraphes ultérieurs sans avoir à refaire la division euclidienne à chaque fois, nous donnons quelques exemples exhaustifs de corps finis.

Soit $K = \mathbb{F}_{p^r} = \mathbb{F}_{p^m}$ le corps fini à p^m éléments, $\mathbb{F}_p$ son sous-corps premier et α un élément primitif de $\mathbb{F}^*_{p^m}$.

On appelle polynôme primitif de K le polynôme minimal d'un élément primitif de K.

Par définition $\mathbb{F}_p(\alpha) = \mathbb{F}_p[\alpha] \subset K$, de plus les éléments de $K = \{0, 1, \alpha, \alpha^2, ..., \alpha^{p^m-2}\}$ sont dans $\mathbb{F}_p[\alpha]$ (0 correspondant au polynôme nul, les α^i à des monômes), on a $K \subset \mathbb{F}_p[\alpha]$ et finalement

$$K = \mathbb{F}_p[\alpha] = \mathbb{F}_p(\alpha)$$

Pour tous p et m il est toujours possible de trouver un polynôme primitif. Les plus répandus sont les polynômes de Conway. Des détails sur leur existence et leur construction peuvent être trouvés dans [Wer].

4. Le nombre de polynômes irréductibles unitaires est déduit dans ce cas de la formule de Möbius [Rol].

Les polynômes de Conway sont implémentés dans la plupart des logiciels de calcul formel gérant la manipulation des corps finis. Par exemple dans Magma la commande donnant le polynôme de Conway $C_{p,m}$ est ConwayPolynomial(p,m).

Nous donnons quelques exemples de polynômes de Conway dont nous nous servirons pour illustrer nos résultats :

	$p=2$	$p=3$
$r=1$	$X+1$	$X+1$
$r=2$	X^2+X+1	X^2+2X+2
$r=3$	X^3+X+1	X^3+2X+1
$r=4$	X^4+X+1	X^4+2X^3+2
$r=5$	X^5+X^2+1	X^5+2X+1
$r=6$	$X^6+X^4+X^3+X+1$	$X^6+2X^4+X^2+2X+2$

Une liste de polynômes de Conway pour de nombreuses valeurs de p est disponible par exemple sur [Lue]

On obtient finalement

$$\mathbb{F}_{p^m} \cong \mathbb{F}_p[X]/(P)$$

où P est un polynôme primitif de $\mathbb{F}_{p^m}$, par exemple le polynôme de Conway $C_{p,m}$.

Une base de $\mathbb{F}_{p^m}$ comme $\mathbb{F}_p$-espace vectoriel est donnée par

$$\{1, \alpha, ..., \alpha^{m-1}\}$$

A tout élément $\alpha^i \in \mathbb{F}_{p^m}^*$ on associe le reste dans la division euclidienne de X^i par P (à 0 on associe évidemment 0)

$$\begin{array}{ccc} \mathbb{F}_{q^m} = \mathbb{F}_{p^m} & \cong & \mathbb{F}_p[X]/(P) \\ \alpha^i & \mapsto & X^i \bmod P \end{array}$$

Par linéarité de la division euclidienne, on obtient bien ainsi un isomorphisme d'espaces vectoriels.

Nous donnons à présent quelques exemples de corps construits de cette façon. Nous nous référerons à ces exemples pour des calculs faits dans les chapitres ultérieurs.

Exemple 40. Construction de $\mathbb{F}_4 = \mathbb{F}_{2^2}$:

Le polynôme de Conway $C_{2,2}$ est égal à $X^2 + X + 1$. Soit α la classe d'équivalence de X dans $\mathbb{F}_2[X]/(C_{2,2})$. D'après ce qui précède, $\mathbb{F}_4 \cong \mathbb{F}_2(\alpha)$.

Les éléments de $\mathbb{F}_4$ sont $\{0, 1 = \alpha^0, \alpha, \alpha^2\}$. A chaque élément on associe son reste dans la division euclidienne formelle par $\alpha^2 + \alpha + 1$ que l'on exprime dans la base $\{1, \alpha\}$:

Elément de $\mathbb{F}_4$	Elément correspondant de $(\mathbb{F}_2)^2$ dans la base $\{1, \alpha\}$
0	$(0,0)$
1	$(1,0)$
α	$(0,1)$
α^2	$(1,1)$

Exemple 41. Construction de $\mathbb{F}_8 = \mathbb{F}_{2^3}$:

Le polynôme de Conway $C_{2,3}$ est égal à $X^3 + X + 1$. Soit α la classe d'équivalence de X dans $\mathbb{F}_2[X]/(C_{2,3})$. D'après ce qui précède, $\mathbb{F}_8 \cong \mathbb{F}_2(\alpha)$.

Les éléments de $\mathbb{F}_4$ sont $\{0, 1 = \alpha^0, \alpha, \alpha^2, ..., \alpha^6\}$. A chaque élément on associe son reste dans la division euclidienne formelle par $\alpha^3 + \alpha + 1$ que l'on exprime dans la base $\{1, \alpha, \alpha^2\}$:

$x \in \mathbb{F}_8$	$\cong(\mathbb{F}_2)^3$, base $\{1, \alpha, \alpha^2\}$
0	$(0, 0, 0)$
1	$(1, 0, 0)$
α	$(0, 1, 0)$
α^2	$(0, 0, 1)$

$x \in \mathbb{F}_8$	$\cong(\mathbb{F}_2)^3$, base $\{1, \alpha, \alpha^2\}$
α^3	$(1, 1, 0)$
α^4	$(0, 1, 1)$
α^5	$(1, 1, 1)$
α^6	$(1, 0, 1)$

Exemple 42. Construction de $\mathbb{F}_{16} = \mathbb{F}_{2^4}$ à l'aide du polynôme de Conway $C_{2,4} = X^4 + X + 1$. Un raisonnement analogue donne :

$x \in \mathbb{F}_{16}$	$\in(\mathbb{F}_2)^4$, base $\{1, \alpha, \alpha^2, \alpha^3\}$
0	$(0, 0, 0, 0)$
1	$(1, 0, 0, 0)$
α	$(0, 1, 0, 0)$
α^2	$(0, 0, 1, 0)$
α^3	$(0, 0, 0, 1)$
α^4	$(1, 1, 0, 0)$
α^5	$(0, 1, 1, 0)$
α^6	$(0, 0, 1, 1)$

$x \in \mathbb{F}_{16}$	$\in(\mathbb{F}_2)^4$, base $\{1, \alpha, \alpha^2, \alpha^3\}$
α^7	$(1, 1, 0, 1)$
α^8	$(1, 0, 1, 0)$
α^9	$(0, 1, 0, 1)$
α^{10}	$(1, 1, 1, 0)$
α^{11}	$(0, 1, 1, 1)$
α^{12}	$(1, 1, 1, 1)$
α^{13}	$(1, 0, 1, 1)$
α^{14}	$(1, 0, 0, 1)$

Exemple 43. On étudiera des exemples ou le corps de base n'est pas de cardinal premier. On a vu par exemple que $\mathbb{F}_{16}$ pouvait aussi être vu comme une extension du corps $\mathbb{F}_4$. On reprend donc le corps $\mathbb{F}_4 = \{0, 1, \beta, \beta^2 = 1 + \beta\}$ tel qu'il a été construit dans l'exemple 40. On montre que le polynôme $P(X) = X^2 + \beta X + 1$ est irréductible sur $\mathbb{F}_4$. On construit l'anneau quotient $\mathbb{F}_4[X]/(P)$. Soit γ la classe de X dans cet anneau quotient. Les éléments de $\mathbb{F}_4[X]/(P)$ sont écrits sous la forme (λ_1, λ_2) dans la base $\{1, \gamma\}$, où λ_1 et λ_2 sont des éléments de $\mathbb{F}_4$.

Le corps $\mathbb{F}_{16}$ ainsi construit ainsi est isomorphe à celui construit dans l'exemple précédent. En reprenant les notations des deux exemples on a $\beta = \alpha^5$ et $\gamma = \alpha^6$. On obtient la description suivante de $\mathbb{F}_{16}$:

$x \in \mathbb{F}_{16}$	$\in(\mathbb{F}_4)^2$, base $\{1, \gamma\}$
0	$(0, 0)$
1	$(1, 0)$
α	$(0, \beta^2)$
α^2	(β, β^2)
α^3	(β, β)
α^4	$(1, \beta^2)$
α^5	$(\beta, 0)$
α^6	$(0, 1)$

$x \in \mathbb{F}_{16}$	$\in(\mathbb{F}_4)^2$, base $\{1, \gamma\}$
α^7	$(\beta^2, 1)$
α^8	(β^2, β^2)
α^9	$(\beta, 1)$
α^{10}	$(\beta^2, 0)$
α^{11}	$(0, \beta)$
α^{12}	$(1, \beta)$
α^{13}	$(1, 1)$
α^{14}	(β^2, β)

Exemple 44. Prenons un exemple en caractéristique 3 : Construction de $\mathbb{F}_9 = \mathbb{F}_{3^2}$ à l'aide du polynôme de Conway $C_{3,2} = X^2 + 2X + 2$. Un raisonnement analogue donne :

$x \in \mathbb{F}_9$	$\cong(\mathbb{F}_3)^2$, base $\{1, \alpha\}$
0	$(0,0)$
1	$(1,0)$
α	$(0,1)$
α^2	$(1,1)$
α^3	$(1,2)$

$x \in \mathbb{F}_9$	$\cong(\mathbb{F}_3)^2$, base $\{1, \alpha\}$
α^4	$(2,0)$
α^5	$(0,2)$
α^6	$(2,2)$
α^7	$(2,1)$

7 Implémentation

Pour le cryptosystème LRPC vu au chapitre 6 nous avons eu besoin de programmer une structure de code en langage C. Nous avons pour cela utilisé la librairie MPFQ [mpfq] qui gère les structures de corps finis dans C. Cependant pour des tests avec la librairie bitvector, ne gérant pas les structures de corps finis, nous avons aussi été ramenés à redéfinir des structures de corps finis « à la main ».

Les éléments du corps $\mathbb{F}_{q^m}$ sont alors directement représentés comme des vecteurs (soit, dans C, des tableaux) de $(\mathbb{F}_q)^m$. Les coordonnées de ces vecteurs sont les coefficients des éléments de $\mathbb{F}_q(X)/(P)$ où $P = \sum_{i=0}^{m} c_i X^i$ est irréductible de degré m. Dans ce qui suit on fait l'analogie entre les vecteurs et les polynômes.

L'addition et la multiplication scalaire correspondent aux opérations correspondantes sur les vecteurs.

Pour tout vecteur $v = (v_0, ..., v_{m-1})$ correspondant à $\sum_{i=0}^{m-1} v_i X^i$, la multiplication par X (soit $(0, 1, 0, ..., 0)$) donne $\sum_{i=0}^{m-2} v_i X^{i+1} + (v_{m-1} X^m \bmod P)$ (seul le dernier monôme, de degré $\geqslant m$, doit être divisé par P). Ceci revient à effectuer le produit matriciel suivant :

$$Xv = \begin{pmatrix} 0 & 0 & \dots & 0 & -\frac{c_o}{c_m} \\ 1 & 0 & & 0 & -\frac{c_1}{c_m} \\ 0 & 1 & & \vdots & \vdots \\ \vdots & 0 & \ddots & & \\ & \vdots & & & \\ 0 & 0 & & 1 & -\frac{c_{m-1}}{c_m} \end{pmatrix} \times v$$

La matrice intervenant est appelée *matrice compagnon* du polynôme P.

Remarque 45. Les $m-1$ premières colonnes de la matrice précédente ont pour effet de décaler les $m-1$ première coordonnées du vecteur d'un rang vers la droite, ceci peut être fait sans nécessiter d'opérations dans $\mathbb{F}_q$. Seule la dernière colonne de la matrice correspond à des multiplications dans $\mathbb{F}_q$. Le nombre de multiplications est égal au nombre de coefficients non nuls dans le polynôme P. Selon les valeurs de q et m il peut être possible de trouver des polynômes irréductibles P ayant très peu de coefficients non nuls, par exemple pour $p = 2$ et $m = 41$ le polynôme de Conway $C_{2,41}$ est égal à $X^{41} + X^3 + 1$ et dans ce cas la multiplication matricielle précédente ne coûte que deux multiplications dans $\mathbb{F}_2$.

Pour effectuer le produit par le monôme X^k on peut répéter k fois la multiplication précédente. Pour effectuer le produit de deux polynômes on procède monôme par monôme et on ajoute successivement les résultats obtenus.

Si nous sommes dans le cas favorable mentionné dans la remarque précédente la multiplication par la matrice compagnon coûte une constante $\lambda \ll m$ multiplications dans $\mathbb{F}_q$. Ainsi dans ce cas la multiplication de deux éléments de $\mathbb{F}_{q^m}$ peut être faite en $O(\lambda m)$ opérations dans $\mathbb{F}_q$ ce qui est très performant même comparé à des algorithmes de type Karatsuba.

Remarque 46. Pour les corps de petit cardinal le plus rapide est d'implémenter directement la multiplication par des tableaux exhaustifs.

Deuxième partie : Polynômes de Ore, q-polynômes et applications

Chapitre III :
Polynômes de Ore et q-polynômes

Ce chapitre et le suivant reprennent des résultats parus dans l'article [GMR] écrit par Philippe Gaborit, Olivier Ruatta et moi-même.

Nous introduisons dans cette partie la notion de polynôme de Ore telle qu'elle a été introduite par Oystein Ore dans [Ore1]. Un anneau de polynômes de Ore est un anneau non commutatif de polynômes en une variable X définis sur un corps $\mathbb{K}$ que dans notre travail nous supposons fini ; cet anneau de polynômes est non commutatif dans le sens où pour $a \in \mathbb{K}$, $aX \neq Xa$. Nous présentons les propriétés de ces polynômes, nous voyons notamment qu'un anneau de polynômes de Ore est euclidien. Dans un chapitre ultérieur nous généraliserons la notion de résultant et de sous-résultants à des polynômes de Ore.

Nous étudions ensuite un cas particulier de polynômes de Ore appelés polynômes linéaires ou q-polynômes. Nous étudions la structure de l'ensemble des racines d'un polynôme linéaire. Nous formalisons des résultats déjà connus sur les racines de polynômes linéarisés avec le vocabulaire des espaces vectoriels.

Dans toute cette partie, q est une puissance d'un nombre premier, m un entier. Les polynômes de Ore peuvent être définis sur un corps de cardinal infini, cependant dans notre travail nous nous placerons toujours dans le cas où $\mathbb{K}$ (qui désignera le corps des coefficients des polynômes) est le corps fini à q^m éléments (défini à isomorphisme près). On rappelle que $\mathbb{F}_{q^m}$ peut être vu comme un $\mathbb{F}_q$-espace vectoriel de dimension m.

$$\mathbb{K} = \mathbb{F}_{q^m} \cong (\mathbb{F}_q)^m$$

1 Automorphismes des corps finis

Nous étudions dans cette partie les automorphismes de corps finis. Nous introduisons l'automorphisme de Frobenius et nous montrons que tout automorphisme de corps fini est un itéré de l'automorphisme de Frobenius. Les polynômes linéaires, introduits dans un chapitre ultérieur, pourront ainsi être considérés comme des combinaisons linéaires d'automorphismes de corps finis.

Définition 47. *Pour un corps fini* $\mathbb{F}_{q^m}$, l'automorphisme de Frobenius *noté* θ *s'il n'y a pas d'ambiguïté est l'application définie pour tout* $x \in \mathbb{F}_{q^m}$ *par*

$$\theta : x \mapsto x^q$$

Proposition 48. *L'automorphisme de Frobenius est un isomorphisme sur $\mathbb{F}_{q^m}$ vu comme un $\mathbb{F}_q$-espace vectoriel, en particulier θ est $\mathbb{F}_q$-linéaire i.e. pour tous x_1 et $x_2 \in \mathbb{F}_{q^m}$ et $\lambda \in \mathbb{F}_q$, $\theta(\lambda x_1 + x_2) = \lambda\theta(x_1) + \theta(x_2)$.*

Démonstration. *Remarquons d'abord que $\forall i \in \{0...m-1\}$ on a $\theta(x_1 + x_2) = (x_1 + x_2)^{q^i} = x_1^{q^i} + x_2^{q^i} = \theta(x_1) + \theta(x_2)$ (en développant avec la formule du binôme, un facteur q apparaît devant tous les autres termes qui ainsi disparaissent car on est en caractéristique p et q est un multiple de p).*

D'autre part $\theta(\lambda x_1) = (\lambda x_1)^{q^i} = \lambda^{q^i} x_1^{q^i} = \lambda x_1^{q^i} = \lambda\theta(x_1)$ (en effet $\lambda \in \mathbb{F}_q$ donc $\lambda^q = \lambda$).

Les deux points précédents montrent que θ est une application linéaire.

De plus θ est clairement injective ($x^q = 0 \Leftrightarrow x = 0$) donc bijective par égalité de dimensions. □

Pour vérifier que l'automorphisme de Frobenius est bien un automorphisme de corps il reste à vérifier que $\theta(1) = 1$ et que pour tous $x, y \in \mathbb{F}_{q^m}$ $\theta(xy) = \theta(x)\theta(y)$. Ces deux points sont immédiats.

Etablissons maintenant que tout automorphisme de corps fini est un itéré du Frobenius :

Proposition 49. *Soit σ un automorphisme de $\mathbb{F}_{q^m}$, alors il existe $k \in \{0, ..., m-1\}$ tel que σ s'obtient en itérant k fois l'automorphisme de Frobenius, autrement dit :*

$$\text{Pour tout } \sigma \in \mathrm{Aut}(\mathbb{F}_{q^m}) \text{ il existe un unique } k \in \{0, ..., m-1\} \text{ tel que } \sigma = \theta^k : x \mapsto x^{q^k}$$

Démonstration. *Avec les mêmes notations que précédemment, soit α un élément primitif de $\mathbb{F}_{q^m}^*$ et P son polynôme minimal. Par construction les racines de P sont $\alpha, \alpha^q, ..., \alpha^{q^{m-1}}$ les images de α par les itérés successifs du Frobenius, en effet par linéarité $P(\theta^i(\alpha)) = \theta^i(P(\alpha)) = 0 \forall i \in \{0, ..., m-1\}$ ce qui permet de trouver la totalité des m racines de P.*

Soit $\sigma \in \mathrm{Aut}(\mathbb{F}_{q^m})$ un automorphisme de $\mathbb{F}_{q^m}$, alors $P(\sigma(\alpha)) = \sigma(P(\alpha)) = 0$ et $\sigma(\alpha)$ est donc une racine de P donc il existe un unique $k \in \{0, ..., m-1\}$ tel que $\sigma(\alpha) = \theta^k(\alpha)$. Comme α est un élément primitif on généralise cette formule à α^2 (σ et θ sont des un automorphismes de corps donc $\sigma(\alpha^2) = \sigma(\alpha)\sigma(\alpha) = \theta^k(\alpha)\theta^k(\alpha) = \theta^k(\alpha^2)$), puis α^3, α^4... plus généralement tout $x \in \mathbb{F}_{q^m}^*$; de plus comme σ est un automorphisme, $\sigma(0) = 0 = \theta^k(0)$ ce qui achève de montrer la formule pour tout $x \in \mathbb{F}_{q^m}$. □

2 Polynômes de Ore

2.1 Définition

Pour une description complète de ces polynômes le lecteur pourra se référer à [Ore1]

Définition 50. *Un* anneau de polynômes de Ore en une variable X, *noté* $\mathbb{K}[X, \sigma, \delta]$ *ou plus simplement* $\mathbb{K}[X, \sigma]$ *si* $\delta \equiv 0$, *est l'ensemble*

$$\left\{ A(X) = \sum_{i=0}^{d_A} a_i X^i, a_i \in \mathbb{K}, d_A \in \mathbb{N} \right\}$$

que l'on munit des opérations suivantes :

- *L'addition de deux polynômes est l'addition usuelle.*
- *La multiplication de deux constantes est la multiplication usuelle dans $\mathbb{K}$.*

- *Pour tout* $a \in \mathbb{K}$ *on a*

$$Xa = \sigma(a)X + \delta(a) \tag{1}$$

On souhaite que cet ensemble soit effectivement muni d'une structure d'anneau. Certaines des propriétés d'un anneau sont évidentes pour $\mathbb{K}[X, \sigma, \delta]$, d'autres non, et vont imposer des propriétés sur σ et δ.

Par exemple, pour avoir un anneau, il est nécessaire que la multiplication soit distributive par rapport à l'addition ce qui impose

$$\begin{array}{rcl} X(a+b) & = & Xa + Xb \\ \sigma(a+b)X + \delta(a+b) & = & \sigma(a)X + \delta(a) + \sigma(b)X + \delta(b) \end{array}$$

Par égalité des coefficients on souhaite donc choisir donc σ et δ de façon à avoir

$$\sigma(a+b) = \sigma(a) + \sigma(b) \tag{2}$$

$$\delta(a+b) = \delta(a) + \delta(b) \tag{3}$$

On souhaite aussi que la multiplication soit associative, ce qui impose

$$\begin{array}{rcl} X(ab) & = & (Xa)b \\ \sigma(ab)X + \delta(ab) & = & (\sigma(a)X + \delta(a))b \\ \sigma(ab)X + \delta(ab) & = & \sigma(a)Xb + \delta(a)b \\ \sigma(ab)X + \delta(ab) & = & \sigma(a)\sigma(b)X + \sigma(a)\delta(b) + \delta(a)b \end{array}$$

On souhaite donc choisir donc σ et δ de façon à avoir

$$\sigma(ab) = \sigma(a)\sigma(b) \tag{4}$$

$$\delta(ab) = \sigma(a)\delta(b) + \delta(a)b \tag{5}$$

D'après (2) et (4), σ doit être un morphisme de l'anneau $\mathbb{K}$.

Une application δ vérifiant (3) et (5) est appelée une *σ-dérivation.*

En choisissant pour σ une morphisme de corps et pour δ une σ-dérivation, on munit effectivement $\mathbb{K}[X, \sigma, \delta]$ d'une structure d'anneau. Par contre il est évident que $\mathbb{K}[X, \sigma, \delta]$ n'est pas un anneau commutatif puisqu'on voit immédiatement que $Xa \neq aX$.

Remarque 51. Lorsque $\delta \equiv 0$, on parle d'anneau de *polynômes tordus* (Skew polynomials).

Si $\sigma = \mathrm{Id}$ on retrouve les opérateurs différentiels.

Si $\sigma = \mathrm{Id}$ et $\delta \equiv 0$ on retrouve l'anneau des polynômes commutatifs $\mathbb{K}[X]$.

Lorsque $\delta \equiv 0$ et σ est l'automorphisme de Frobenius $x \mapsto x^q$, on parle de polynômes linéarisés ou q-polynômes. Ce cas particulier, important dans notre travail, fera l'objet d'une étude spécifique.

2.2 Sous-corps des éléments commutatifs pour la multiplication

Comme σ un morphisme de corps, l'ensemble $\sigma(\mathbb{K}) = \{\sigma(a), a \in \mathbb{K}\}$ est un sous-corps de $\mathbb{K}$. De plus, l'ensemble des éléments a de $\mathbb{K}$ tels que $\sigma(a) = a$ est un sous-corps de $\mathbb{K}$ appelé *corps des invariants de* $\mathbb{K}$.

D'autre part, on vérifie immédiatement que l'ensemble des éléments a de $\mathbb{K}$ tels que $\delta(a)=0$ est un sous-corps de $\mathbb{K}$ appelé *corps des constantes de* $\mathbb{K}$.

Proposition 52. *L'ensemble des éléments a de $\mathbb{K}$ tels que $Xa=aX$ est le plus grand sous-corps de $\mathbb{K}$ inclus à la fois dans le corps des invariants et le corps des constantes.*

2.3 Exemples de calculs

Lemme 53. *Si on note $a^{[n]}=\sigma^n(a)$ (i.e. σ composée n fois) et $a^{(n)}=\delta^n(a)$ (i.e. δ composée n fois) alors on a*

$$X^n\,a=S_{n,0}(a)X^n+S_{n,1}(a)\,X^{n-1}+\cdots+S_{n,n}(a)$$

avec

$$S_{n,i}(a)=\big(a^{[n-i]}\big)^{(i)}+\sigma\Big(\big((a)^{[n-i-1]}\big)^{(i)}\Big)+\cdots+\big((a)^{(i)}\big)^{[n-i]}$$

autrement dit $S_{n,i}(a)$ représente la somme de la totalité des $\binom{n}{i}$ éléments obtenus lorsqu'on utilise $(n-i)$ fois l'opérateur σ et i fois l'opérateur δ sur a, dans n'importe quel ordre.

On a notamment $S_{n,0}(a)=a^{[n]}$ and $S_{n,n}(a)=a^{(n)}$

Démonstration. *On obtient ce résultat en itérant la formule $Xa=\sigma(a)X+\delta(a)$ ce qui donne par exemple sur les premiers termes :*

$X^2a=X\,(\sigma(a)X+\delta(a))=X\sigma(a)X+X\delta(a)=\big(a^{[2]}X+\delta(\sigma(a))\big)X+\sigma(\delta(a))X+a^{(2)}=a^{[2]}X^2+(\delta(\sigma(a))+\sigma(\delta(a)))\,X+a^{(2)}$

$X^3\,a=XX^2a=X\big(a^{[2]}X^2+(\delta(\sigma(a))+\sigma(\delta(a)))\,X+a^{(2)}\big)a=a^{[3]}\,X^3+(\sigma(\delta(\sigma(a)))+(\delta(a))^{[2]})\,X^2+(\sigma(a^{(2)})+\sigma(a)^{(2)}+\delta(\sigma(\delta(a))))X+a^{[3]}$ □

On en déduit la formule suivante [Ore1]:

Proposition 54. *Soit deux polynômes de Ore $A=\sum_{i=0}^{d_A}a_iX^i$ et $B=\sum_{i=0}^{d_B}b_iX^i$, alors*

$$A(X)B(X)=\sum_{i=0}^{d_A+d_B}c_iX^i$$

où les coefficients c_i sont définis par

$$c_i=\sum_{j=0}^{i}b_j\sum_{k=0}^{i-j}S_{d_B-j,i-j-k}(a_k)$$

2.4 Degré d'un polynôme de Ore

Définition 55. *Soit $A=\sum_{i=0}^{d_A}a_iX^i\in\mathbb{K}[X,\sigma,\delta]$. Comme dans le cas commutatif, le plus grand i tel que $a_i\neq 0$ est appelé le* degré *du polynôme et noté* $\deg(P)$.

Par convention on choisit $\deg(0)=-\infty$.

On supposera sauf exception que le coefficient de X^{d_A} est non nul de sorte que $\deg(A) = d_A$.

Proposition 56. *Soit deux polynômes de Ore P_1 et P_2, alors*

1. $\deg(A+B) \leqslant \max(\deg(A), \deg(B))$
2. $\deg(AB) = \deg(A)\deg(B)$

Démonstration. *1. est immédiat, 2. vient de la proposition précédente : le coefficient du monôme $X^{d_A+d_B}$ est égal à $a_{d_A} b_{d_B}^{[d_B]}$ et il est donc non nul car σ est un automorphisme.* □

2.5 Division euclidienne à droite

Nous allons voir qu'un anneau de polynômes de Ore est euclidien. Cependant, le fait que la multiplication ne soit pas commutative impose de définir une division euclidienne à droite et une division euclidienne à gauche.

Nous allons voire que la division euclidienne à droite est toujours possible, alors que la division euclidienne à gauche n'est pas toujours possible.

Par la suite, lorsque ce ne sera pas précisé, nous utiliserons toujours la division euclidienne à droite.

Soient $A = \sum_{i=0}^{d_A} a_i X^i$ et $B = \sum_{i=0}^{d_B} b_i X^i$ deux polynômes avec les mêmes notations que dans la proposition 54. On suppose $d_B \leqslant d_A$.

D'après cette même proposition le terme dominant du polynôme $X^{d_A - d_B} B$ est $b_{d_B}^{[d_A - d_B]} X^{d_B}$. Le terme dominant du polynôme $a_{d_A}\left(b_{d_B}^{[d_A-d_B]}\right)^{-1} X^{d_A-d_B} B$ est donc $a_{d_A} X^{d_A}$, soit exactement le même que A. On en déduit que le polynôme

$$A - a_{d_A}\left(b_{d_B}^{[d_A-d_B]}\right)^{-1} X^{d_A-d_B} B \tag{6}$$

est de degré strictement inférieur à A.

Si on note $Q_1 = a_{d_A}\left(b_{d_B}^{[d_A-d_B]}\right)^{-1} X^{d_A-d_B}$ et $R_1 = A - QB$ on a donc la relation

$$A = Q_1 B + R_1 \text{ avec } \deg(R_1) < \deg(A) \tag{7}$$

On va itérer ce calcul pour construire un algorithme d'Euclide sur $\mathbb{K}[X, \sigma, \delta]$: on refait la même manipulation avec R_1 et trouver Q_2 et R_2 tels que

$$R_1 = Q_2 B + R_2 \text{ avec } \deg(R_2) < \deg(R_1)$$

On va ainsi construire une suite finie de polynômes $(Q_k)_{k\geqslant 1}$ et $(R_k)_{k\geqslant 0}$ (en notant $R_0 = A$), les R_k étant de degrés strictement décroissants, tels que $R_k = Q_{k+1} B + R_{k+1}$. Il va donc exister un indice l tel que $\deg(R_l) < \deg(B)$.

On a donc $A = Q_1 B + R_1 = Q_1 B + (Q_2 B + R_2) = \ldots = (Q_1 + \ldots Q_l) B + R_l$. En notant $Q = Q_1 + \ldots + Q_l$ et $R = R_l$ on obtient :

Proposition 57. *Pour tous polynômes $A, B \in \mathbb{K}[X, \sigma, \delta]$ avec $\deg(A) \geqslant \deg(B)$, il existe un unique polynôme Q et un unique polynôme R tel que*

$$A = QB + R \text{ et } \deg(R) < \deg(B)$$

Autrement dit, $\mathbb{K}[X, \sigma, \delta]$ *peut être muni d'une* division euclidienne à droite.

Des exemples seront donnés ultérieurement dans le cas particulier des polynômes linéaires.

On retrouve les notions habituelles lorsqu'on travaille sur un anneau euclidien. Les résultats suivants ne sont pas démontrés car il s'agit de résultats connus sur les anneaux euclidiens :

Définition 58. *Si* $R = 0$ *i.e.* $A = QB$ *on dit que* B *est un* diviseur à droite de A.

On généralise aux polynômes de Ore la notion de plus grand diviseur commun et plus petit multiple commun :

Proposition 59. *Pour deux polynômes de Ore quelconques* A *et* B*, il existe un unique polynôme de Ore de degré maximal qui divise* A *et* B *à droite. Ce polynôme est appelé* plus grand diviseur commun *(à droite) et on le notera* $\mathrm{RGCD}(A, B)$.

Comme dans le cas commutatif, le RGCD *correspond au dernier reste non nul dans la suite des divisions euclidiennes.*

Définition 60. *Si* $\mathrm{RGCD}(A, B) = 1$ *on dit que* A *et* B *sont* premiers entre eux à droite.

Théorème 61. *(identité de Bézout)*

Si un polynôme de Ore D *est divisible à droite par* $\mathrm{RGCD}(A, B)$ *(i.e. il existe* $\tilde{D}$ *tel que* $D = \tilde{D}\,\mathrm{RGCD}(A, B)$*) alors il existe deux polynômes de Ore* P_1 *et* P_2 *tels que*

$$P_1 A + P_2 B = D\big(= \tilde{D}\,\mathrm{RGCD}(A, B)\big)$$

Proposition 62. *Pour deux polynômes de Ore quelconques* A *et* B*, il existe un unique polynôme de Ore de degré minimal qui est divisible par* A *et* B *à droite. Ce polynôme est appelé* union à droite *ou* plus petit multiple commun à droite *et on le notera* $\mathrm{RLCM}(A, B)$.

Le degré de ce polynôme est égal à $d_A + d_B - d_{\mathrm{RGCD}}$.

2.6 Division euclidienne à gauche

Etablir une division euclidienne à gauche dans un anneau de polynômes de Ore revient à trouver, pour deux polynômes de Ore quelconques A et B avec $\deg(A) \geqslant \deg(B)$, deux polynômes de Ore Q et R avec

$$A = BQ + R \text{ et } \deg(R) < \deg(B)$$

On souhaite refaire la même construction que dans (6) et (7) mais en multipliant B à droite. On souhaite donc trouver un monôme $\lambda X^{d-d'}$ tel que le terme dominant de $B\lambda X^{d-d'}$ soit égal à $a_d X^d$ afin d'annuler le terme dominant de A.

Le terme dominant de $B\lambda X^{d_A - d_B} = \sum_{i=0}^{d_B} b_i X^i \lambda X^{d_A - d_B}$ est $b_{d_B} X^{d_B} \lambda X^{d_A - d_B} = b_{d_B} \lambda^{[d_B]} X^{d_A}$.

On cherche donc à établir l'égalité $a_{d_A} = b_{d_B} \lambda^{[d_B]}$ soit

$$\sigma^{d_B}(\lambda) = a_{d_A} b_{d_B}^{-1}$$

On souhaite pouvoir faire cette manipulation pour deux polynômes quelconques donc a_{d_A} et b_{d_B} représentent potentiellement deux éléments quelconques de $\mathbb{K}$.

Il y a un cas où il est clair que l'on peut trouver λ vérifiant l'équation précédente, c'est lorsque σ est un automorphisme de $\mathbb{K}$. Dans ce cas σ^{d_B} est aussi un automorphisme et il suffit de prendre $\lambda = \sigma^{-d_B}(a_{d_A} b_{d_B})$.

Proposition 63. *Lorsque σ est un automorphisme de $\mathbb{K}$, l'anneau $\mathbb{K}[X, \sigma, \delta]$ peut être muni d'une division euclidienne à gauche.*

Nous ne détaillerons pas davantage les explications puisque de toutes façons nous supposerons sauf exception que la division euclidienne est à droite.

2.7 Evaluation d'un élément par un polynôme de Ore

Le morphisme d'évaluation habituel pour un polynôme consistant pour un élément $a \in \mathbb{F}_{q^m}$ fixé à remplacer X par a dans l'expression du q-polynôme n'est plus un morphisme d'anneau dans le cas commutatif. Par exemple, si on note $\theta : x \mapsto x^2$ le Frobenius nous avons sur $\mathbb{F}_4[X, \theta, 0] = \mathbb{F}_2(\alpha)[X, \theta, 0]$ l'égalité

$$X^2 + 1 = (X + \alpha)(X + \alpha^2)$$

En remplaçant X par 1 on obtient $0 = 1$.

On est conduit à adopter une nouvelle définition pour l'évaluation d'un élément de $\mathbb{F}_{q^m}$ par un polynôme de Ore :

Définition 64. *Avec les notations précédentes, soit P un polynôme de Ore sur $\mathbb{F}_{q^m}[X, \sigma, \delta]$ et un élément $a \in \mathbb{F}_{q^m}$,* l'évaluation de a par P *est égale au reste dans la division euclidienne de P par $(X - a)$.*

$$P(a) \overset{déf}{=} P \bmod (X - a)$$

Les propriétés de la division euclidienne assurent qu'on obtient bien ainsi un morphisme d'anneau.

Remarque 65. Dans le cas de polynômes dans un anneau commutatif, cette définition est équivalente à la définition naturelle consistant à remplacer X par a dans $P(X)$.

2.8 Racine d'un polynôme de Ore

Avec la définition précédente nous obtenons :

Définition 66. *On dit que a est une racine de P si P est divisible par $(X - a)$.*

3 Anneaux de q-polynômes (ou polynômes linéarisés)

Les polynômes linéaires sont un cas particulier de polynômes de Ore, qui leur a consacré une étude spécifique dans [Ore2].

3.1 Définitions et premières propriétés

Définition 67. *On rappelle que q est une puissance d'un nombre premier et $m \in \mathbb{N}^*$.*

On considère l'ensemble des polynômes de la forme

$$A(X) = a_{d_A} X^{q^{d_A}} + a_{d_A - 1} X^{q^{d_A - 1}} + \cdots + a_0 X = \sum_{i=0}^{d_A} a_i X^{q^i} \text{ avec } a_i \in \mathbb{F}_{q^m}$$

Un tel polynôme est appelé polynôme linéarisé *ou* q-polynôme[5] *(sur* $\mathbb{F}_{q^m}$*). Les éléments* X^{q^i} *sont appelés des* q*-monômes.*

Pour $i \in \{1...d\}$ *notons*[6] $X^{[i]} = X^{q^i}$*. Ainsi la notation d'un* q*-polynôme devient*

$$A(X) = a_{d_A} X^{[d_A]} + a_{d_A-1} X^{[d_A-1]} + \cdots + a_0 X^{[0]} = \sum_{i=0}^{d_A} a_i X^{[i]}$$

Le nombre d_A *est appelé* q-degré *du* q*-polynôme. On note*

$$d_A = \deg_q(A)$$

Exemple 68. On note $\mathbb{F}_8 = \{0, 1, \alpha, ..., \alpha^6\}$. Le polynôme $X^8 + \alpha X^2 + X = X^{2^3} + \alpha X^{2^1} + X^{2^0}$ est un 2-polynôme sur $\mathbb{F}_8$, son 2-degré est 3.

Remarque 69. Si on note $\theta : x \mapsto x^q$ l'automorphisme de Frobenius dans $\mathbb{F}_q$ alors un q-polynôme peut également s'écrire

$$A(X) = a_{d_A} \theta^{d_A}(X) + a_{d_A-1} \theta^{d_A-1}(X) + \cdots + a_0 X = \sum_{i=0}^{d_A} a_i \theta^i(X)$$

Remarquons enfin que la linéarité de l'automorphisme de Frobenius que nous avons établie au chapitre précédent permet de considérer un polynôme linéaire comme un polynôme d'endomorphismes et écrire

$$A = a_{d_A} \theta^{d_A} + a_{d_A-1} \theta^{d_A-1} + \cdots + a_0 Id$$

Ainsi le terme de polynôme linéarisé est justifié : un p-polynôme sur $\mathbb{F}_{q^m}$ est une application linéaire[7] sur $\mathbb{F}_{q^m}$ vu comme un $\mathbb{F}_q$-espace vectoriel i.e. $A(\lambda x_1 + x_2) = \lambda A(x_1) + A(x_2)$ pour $\lambda \in \mathbb{F}_q$ et $x_1, x_2 \in \mathbb{F}_{q^m}$.

Il est clair que la somme de deux q-polynômes est encore un q-polynôme, et qu'un q-polynôme multiplié par un élément de $\mathbb{F}_q$ est encore un q-polynôme. Ainsi, L'ensemble des q-polynômes sur $\mathbb{F}_{q^m}$ est un $\mathbb{F}_q$-espace vectoriel.

Un q-polynôme élevé à la puissance q est toujours un q-polynôme (plus précisément $A(X)^q = \sum_{i=0}^{d_A} a_i X^{[i+1]}$, en développant les autres termes disparaissent car un facteur q correspondant à une puissance de p apparaît et on est en caractéristique p).

3.2 Produit de q-polynômes

On vient de voir que l'ensemble des q-polynômes était stable par combinaison linéaire et par l'automorphisme de Frobenius. Cependant, cet ensemble n'est pas stable par multiplication puisque le produit de deux q-monômes X^{q^i} et X^{q^j} est égal à $X^{q^i+q^j}$ et ne peut en général pas s'écrire comme un q-monôme X^{q^k}.

5. Prenons la précaution de rappeler que l'exponentiation n'est pas associative ; les monômes sont bien $X^{q^i} = X^{(q^i)}$ et non pas $(X^q)^i = X^{qi}$.

6. Nous voyons que cette notation est cohérente avec la notation $a^{[i]} = \sigma^i(a)$ du chapitre précédent car ici σ va désigner l'automorphisme de Frobenius.

7. Attention aux ensembles auxquels appartiennent les différents éléments, l'égalité est par exemple fausse pour $\lambda \in \mathbb{F}_{q^m}$.

Afin d'obtenir une loi de multiplication interne, on va donc définir le produit de deux q-polynômes comme étant leur composée, au sens de la composition des applications :

Définition 70. *Soient deux q-polynômes* $A(X)=\sum_{i=0}^{d_A} a_i X^{[i]}$ *et* $B(X)=\sum_{i=0}^{d_B} b_i X^{[i]}$*, alors*

$$A(X)\times B(X) \overset{déf}{=\!=} A\circ B(X)=A(B(X))$$

Avec cette loi de multiplication, le produit de deux q-polynômes est encore un q-polynôme, en effet on remarque que pour deux q-monômes $a_iX^{[i]}=a_i\theta^i(X)$ et $b_jX^{[j]}=b_j\theta^j(X)$ on a

$$a_iX^{[i]}\times b_jX^{[j]}=a_i\theta^i(b_j\theta^jX)=a_i\, b_j^{[i]}\, X^{[i+j]}$$

On en déduit par linéarité l'expression du produit de deux q-polynômes :

$$A(X)\times B(X)=A(B(X))=\sum_{i=0}^{d_A} a_i\theta^i\left(\sum_{j=0}^{d_B} b_j\theta^j(X)\right)=\sum_{i=0}^{d_A}\sum_{j=0}^{d_B} a_i b_j^{[i]} X^{[i+j]}$$

Avec cette loi de multiplication, le produit de deux q-polynômes est donc bien un q-polynôme (remarquons aussi que $\deg_q(A\times B)=\deg_q(A)+\deg_q(B)$).

L'exemple du produit de deux q-monômes montre que cette loi de multiplication n'est évidemment pas commutative. Remarquons notamment, ce qui servira au paragraphe suivant, que

$$X^{[1]}\times aX^{[0]}=a^{[1]}X^{[1]}=\theta(a)X^{[1]} \tag{8}$$

L'anneau des polynômes de Ore sur $\mathbb{F}_{q^m}$ muni de la loi d'addition usuelle et de cette loi de multiplication sera noté

$$\mathbb{F}_{q^m}[X^q,\theta]$$

Exemple 71. On se place sur le corps de caractéristique 2 $\mathbb{F}_8=\{0,1,\alpha,...,\alpha^6\}\cong\mathbb{F}_2[X]/(C_{2,3})$ tel que nous l'avons établi dans un chapitre précédent. On considère les deux 2-polynômes $A(X)=X^4+\alpha X^2+X$ et $B(X)=\alpha^2X^2+X$. On obtient[8] :

$A(X)B(X)=A(B(X))=(\alpha^2X^2+X)^4+\alpha(\alpha^2X^2+X)^2+\alpha^2X^2+X=\alpha^8X^8+X^4+\alpha(\alpha^4X^4+X^2)+\alpha^2X^2+X=\alpha X+X^4+\alpha^5X^4+\alpha X^2+\alpha^2X^2+X=(1+\alpha^5)X^4+(\alpha+\alpha^2)X^2+(1+\alpha)X=\alpha^4X^4+\alpha^4X^2+\alpha^3X$

$B(X)A(X)=B(A(X))=\alpha^2(X^4+\alpha X^2+X)^2+X^4+\alpha X^2+X=\alpha^2(X^8+\alpha^2X^4+X^2)+X^4+\alpha X^2+X=(1+\alpha^4)X^4+(\alpha+\alpha^2)X^2+(1+\alpha^2)X=\alpha^5X^4+\alpha^4X^2+\alpha^6X$

3.3 Les q-polynômes vus comme des cas particuliers des polynômes de Ore

3.3.1 Bijection entre un anneau de Ore et un anneau de q-polynômes

Nous allons voir dans cette partie qu'à isomorphisme près les q-polynômes peuvent être considérés comme des cas particuliers des polynômes de Ore tels qu'introduits au paragraphe précédent dont on reprend les notations. On considère l'anneau suivant :

$$\mathbb{F}_{q^m}[X,\theta,0]$$

8. Les opérations dans $\mathbb{F}_8$ ont été faites en utilisant la méthode donnée dans le chapitre sur les corps finis.

C'est un anneau des polynômes de Ore sur $\mathbb{F}_{q^m}$; on a pris comme morphisme $\sigma = \theta$ (l'automorphisme de Frobenius) et comme σ-dérivation la dérivation triviale $\delta \equiv 0$.

Dans ce cas le produit Xa devient

$$Xa = \sigma(a)X + \delta(a) = \theta(a)X$$

D'après la formule (8), il existe donc une bijection entre l'ensemble des polynômes linéaires sur $\mathbb{F}_{q^m}$ et l'anneau des polynômes de Ore $\mathbb{F}_{q^m}[X, \theta, 0]$. Cette bijection est la correspondance $X^{[i]} \mapsto X^i$.

$$\begin{array}{ccc} \mathbb{F}_{q^m}[X^q, \theta] & \cong & \mathbb{F}_q[X, \theta] \\ A(X) = \sum_{i=0}^{d_A} a_i X^{[i]} & \overset{\text{bijection}}{\leftrightarrow} & \sum_{i=0}^{d_A} a_i X^{[i]} \end{array} \tag{9}$$

3.3.2 Conséquences

Tous les résultats établis sur les polynômes de Ore vont donc s'adapter aux q-polynômes, en particulier :

Proposition 72. *L'anneau $\mathbb{F}_{q^m}[X^q, \theta]$ est un anneau euclidien à droite, c'est à dire que pour deux q-polynômes A et B (on suppose $\deg_q(B) < \deg_q(A)$) il existe deux q-polynômes uniques Q et R avec $\deg_q(R) < \deg_q(B)$ tels que*

$$A = QB + R$$

Si $R = 0$ on dit que B divise A à droite.

Les notions de plus grand diviseur commun à droite (RGCD(A, B)) et d'union à droite (RLCM(A, B)) vues au paragraphe précédent, se généralisent aux q-polynômes.

Le plus grand diviseur commun à droite de A et B est égal au dernier reste non nul dans la suite des divisions euclidiennes de A par B puis les restes successifs.

Il existe deux q-polynômes tels que $P_1 A + P_2 B = \text{RGCD}(A, B)$

Si $\text{RGCD}(A, B) = 1$ on dit que A et B sont premiers entre eux à droite.

Remarque 73. Lorsque B divise A à droite alors B divise aussi A au sens classique du terme. En effet, B divise A à droite signifie qu'il existe Q tel que $A = QB$ donc avec la loi de multiplication sur $\mathbb{F}_{q^m}[X^q, \theta]$ signifie $A = \sum_{i=0}^{\deg(Q)} q_i \times B^{q^i}$ en notant q_i les coefficients de Q (à ne pas confondre avec q le cardinal de $\mathbb{F}_q$) ; la multiplication $q_i \times B^{q^i}$ est ici la multiplication normale. Le monôme de plus petit degré dans un q-polynôme étant $X^{q^0} = X$, on peut écrire l'égalité suivante dans l'anneau commutatif $\mathbb{F}_q[X]$:

$$A = \left(\sum_{i=0}^{\deg(Q)} q_i B^{q^i - 1} \right) \times B$$

On voit donc que si B divise A à droite dans $\mathbb{F}_{q^m}[X^q, \theta]$ alors B divise A dans $\mathbb{F}_q[X]$.

3.4 Complexité des opérations arithmétiques dans $\mathbb{F}_{q^m}[X^q, \theta]$

Il est simple de voir que l'isomorphisme (9) peut être calculé en temps linéaire ainsi que son inverse. La complexité des opérations de base est donc la même pour les polynômes de Ore et les q-polynômes.

Nous considérons la complexité en nombres d'opérations dans $\mathbb{F}_q$ en admettant que nous ayons une représentation adaptée de $\mathbb{F}_{q^m}$. La taille d'un éléments de $\mathbb{F}_{q^m}$ est en $O(m)$ et l'addition est également linéaire. La multiplication dans $\mathbb{F}_{q^m}$ peut être obtenue en $M(m)$ opérations dans $\mathbb{F}_q$ avec $M(m) \in O(m^2)$ par un algorithme naïf et $M(m) \in O(m \log(m) \log(\log(m)))$ avec l'algorithme de Schönhage et Strassen [ScSt] ou Schönhage [Sc] ou Cantor et Kaltoffen [CK] ; nous avons vu dans le chapitre 2 que pour certaines valeurs de m on pouvait descendre jusqu'à $M(m) = O(m)$ en hard-codant la multiplication.

Pour un automorphisme σ de $\mathbb{F}_{p^m}$ on peut calculer $\sigma(a), ..., \sigma^{m-1}(a)$ pour $a \in \mathbb{F}_{q^m}$ en utilisant les résultats de Von zur Gaten et Shoup [GS].

Le coût d'une multiplication matricielle dans un corps $\mathbb{K}$ vaut $\mathrm{MM}(n)$ opérations dans $\mathbb{K}$ avec $\mathrm{MM}(n) \in O(n^3)$ par multiplication naïve et $\mathrm{MM}(n) \in O(n^{2.376\cdots})$ par l'algorithme de Coppersmith et Winograd. Le coût de résolution d'un système linéaire dans $\mathbb{K}$ est identique.

Les résultats suivants peuvent être trouvés dans [Gie] :

Lemme 74. *Soient A et $B \in \mathbb{F}_{q^m}[X, \sigma]$, de degrés bornés par d, on peut calculer $A+B$ en $\mathcal{O}(dm)$ opérations dans $\mathbb{F}_q$ et AB with $\mathcal{O}(d^2 M(m) + dm M(m) \log(m))$ opérations dans $\mathbb{F}_q$.*

Lemme 75. *Soient A et $B \in \mathbb{F}_{q^m}[X, \sigma]$, avec $\deg(A) = d_A$ et $\deg(B) = 0 \leqslant d_B \leqslant d_A$, on peut calculer Q et $R \in \mathbb{F}_{q^m}[X, \sigma]$ tels que $A = BQ + R$, $\deg(R) < \deg(B)$ en $\mathcal{O}(d_B(d_A - d_B)m + d_B m^2)$ opérations dans $\mathbb{F}_q$.*

Lemme 76. *Soient A et $B \in \mathbb{F}_{q^m}[X, \sigma]$, avec $\deg(A) = d_A$ et $\deg(B) = 0 \leqslant d_B \leqslant d_A$, on peut par l'algorithme d'Euclide étendu calculer $\mathrm{RGCD}(A, B)$ avec une complexité de $\mathcal{O}(d^2 M(m) m \log(m))$ opérations dans $\mathbb{F}_q$.*

3.5 Racines de q-polynômes

Soit $A \in \mathbb{F}_{q^m}[X^q, \theta]$ un q-polynôme à coefficient dans $\mathbb{F}_{q^m}$.

Nous allons étudier dans cette partie la structure de l'ensemble des racines d'un q-polynôme. Il s'agit ici de racine au sens « classique » du terme c'est à dire de solution de l'équation $A(x) = 0$, autrement dit, A étant linéaire, l'ensemble des racines (dans $\mathbb{F}_{q^m}$) de A est le noyau de A.

De même l'opération d'évaluation d'un q-polynôme est définie au sens classique.

3.5.1 Structure de l'ensemble des racines d'un q-polynôme

On a vu qu'un q-polynôme A est une application linéaire. L'ensemble des racines de A forme le noyau de cette application linéaire. On en déduit :

Proposition 77. *Soit $A \in \mathbb{F}_{q^m}[X^q, \theta]$ un q-polynôme de q-degré d_A. On suppose que le coefficient de $X^{[0]}$ est non nul (ce qui est le cas dans le contexte qui nous intéresse).*

L'ensemble des racines $\mathrm{Ker}(A)$ de A est un $\mathbb{F}_q$-sous-espace vectoriel de $\mathbb{F}_{q^m}$ de dimension d_A.

Pour déterminer explicitement une base de cet espace vectoriel, on procède de la façon suivante [Ber] :

Soit A un q-polynôme de q-degré d_A sur $\mathbb{F}_{q^m}$ vu comme un $\mathbb{F}_q$-espace vectoriel ayant pour base $(\beta_1, ..., \beta_m)$. On évalue $P(\beta_1), ..., P(\beta_m)$. On cherche à résoudre

$$\lambda_1 \beta_1 + \cdots + \lambda_m \beta_m \in \mathrm{Ker}(A) \, ; \lambda_i \in \mathbb{F}_q$$

Ce qui revient à résoudre le système linéaire

$$\lambda_1 A(\beta_1) + \cdots + \lambda_m A(\beta_m) = 0 ; \lambda_i \in \mathbb{F}_q$$

Ce système sous-déterminé a m inconnues (les λ_i) et $d_A \leqslant m$ équations données par l'évaluation de chaque monôme de A. En résolvant ce système on trouve donc d_A éléments indépendants formant une base du noyau :

Proposition 78. *Pour tout q-polynôme A de q-degré d_A sur $\mathbb{F}_{q^m}$ il est possible de trouver une base des racines de A en $m d_A$ opérations.*

Démonstration. *L'évaluation de A sur chaque $(\beta_i)_{1 \leqslant i \leqslant m}$ se fait en d_A opérations, ce qui fait un total de $m d_A$ opérations.*

La résolution du système linéaire se fait en d_A^3 opérations mais il s'agit d'opérations dans $\mathbb{F}_q$ et leur coût est donc négligeable par rapport aux opérations dans $\mathbb{F}_{q^m}$ □

Remarque 79. Les résultats de ce paragraphe ont été énoncés dans le cas où le coefficient de $X^{[0]}$ est non nul. Plus généralement la dimension de l'ensemble des racines est $d_A - d_{\min}$ où $d_{\min}$ désigne le q-degré du plus petit coefficient non nul de A. Le lecteur est invité à consulter le Théorème 1 de [CLU] pour plus de détails.

3.5.2 Polynôme associé à un espace vectoriel

Intéressons nous à présent au problème inverse : on va voir que tout espace vectoriel sur $\mathbb{F}_q$ correspond à l'ensemble des racines d'un q-polynôme. Pour expliciter la construction de ce polynôme on part du résultat suivant présent dans [Ore2] :

Lemme 80. *Le q-polynôme A admet α pour racine si et seulement si A est divisible à droite par $X^q - \alpha^{q-1} X$.*

Etant donné un sous-espace vectoriel de $\mathbb{F}_{q^m}$, on va se servir de ce résultat pour construire un q-polynôme dont l'ensemble des racines est exactement cet espace vectoriel :

Proposition 81. *Soit V un sous-espace vectoriel de $\mathbb{F}_{q^m} \cong (\mathbb{F}_q)^m$ de dimension k. Il existe un unique q-polynôme de $\mathbb{F}_{q^m}[X^q, \theta]$ noté P_V, unitaire de q-degré k, tel que V est l'ensemble des racines de P_V.*

Démonstration. Ore donne une construction explicite dans [Ore2]. Soit $(\alpha_1, ..., \alpha_k)$ une base de V. On construit itérativement une suite $(P_i)_i$ de polynômes définie par :

Lorsque $i = 1$ on pose $P_1(X) = X^q - \alpha_1^{q-1} X$. Il est clair que $P_1(\alpha_1) = 0$.

Pour $i \geqslant 2$ on pose $P_i(X) = (X^q - P_{i-1}(\alpha_i)^{q-1} X) \times P_{i-1}(X)$, ce qui correspond avec la loi de multiplication choisie à

$$P_i(X) = (\theta^q - P_{i-1}(\alpha_i)^{q-1} \mathrm{Id}) \circ P_{i-1}(X)$$

On a $P_{i-1}(\alpha_1) = ... = P_{i-1}(\alpha_{i-1}) = 0$; comme on compose à gauche par des applications linéaires on obtient $P_i(\alpha_1) = ... = P_i(\alpha_{i-1}) = 0$

Pour vérifier que l'ensemble des racines de P_i est $\mathrm{Vect}\{\alpha_1, ..., \alpha_i\}$ il reste à vérifier

$P_i(\alpha_i) = (\theta^q - P_{i-1}(\alpha_i{}^{q-1} \mathrm{Id}) \circ P_{i-1}(\alpha_i) = P_{i-1}(\alpha_i)^q - P_{i-1}(\alpha_i)^{q-1} P_{i-1}(\alpha_i) = 0$

Le polynôme cherché est égal au terme de rang k dans cette suite :

$$P_V(X) = P_k(X) = (X^q - P_{k-1}(\alpha_k)^{q-1} X) \times (X^q - P_{k-2}(\alpha_{k-1})^{q-1} X) \times ... \times (X^q - \alpha_1^{q-1} X) \quad (10)$$

□

Remarque 82. Dans [CLU] les auteurs montrent comment calculer un tel polynôme sur un corps plus petit.

Dans [Ore2] Ore donne une autre expression possible pour $P_V(X)$, qui n'est pas utilisée en pratique car sa complexité est moins bonne que la méthode précédente, mais dont nous aurons besoin pour la preuve de certains résultats :

$$P_V(X) = \frac{\begin{vmatrix} \alpha_1 & \alpha_2 & ... & \alpha_k & X \\ \alpha_1^q & \alpha_2^q & & & \\ \vdots & & \ddots & & \\ \alpha_1^{q^k} & & & \alpha_k^{q^k} & X^{q^k} \end{vmatrix}}{\begin{vmatrix} \alpha_1 & \alpha_2 & ... & \alpha_k \\ \alpha_1^q & \alpha_2^q & & \\ \vdots & & \ddots & \\ \alpha_1^{q^{k-1}} & & & \alpha_k^{q^{k-1}} \end{vmatrix}} \overset{\text{déf}}{=} \frac{\Delta(\alpha_1, ..., \alpha_k, X)}{\Delta(\alpha_1, ..., \alpha_k)} \quad (11)$$

Le fait que $\{\alpha_1, ..., \alpha_k\}$ forme une famille libre garantit que $\Delta(\alpha_1, ..., \alpha_k) \neq 0$.

On vérifie que ce calcul de déterminant permet bien d'obtenir un q-polynôme répondant aux contraintes posées : un développement par rapport à la dernière colonne permet d'établir que le déterminant est bien une combinaison linéaire de X, ..., X^{q^k} ce qui correspond à l'expression formelle d'un q-polynôme. Le terme de X^{q^k} est $\Delta(\alpha_1, ..., \alpha_k)$

Cette expression permet de mettre en avant le fait que le calcul de $P_V(X)$ est bien indépendant du choix de la base, en particulier du choix de l'ordre des vecteurs dans la base : un changement de base revient à effectuer des combinaisons linéaires sur les colonnes du déterminant (en exprimant la nouvelle base en fonction de $\{\alpha_1, ..., \alpha_k\}$). Ces combinaisons linéaires laissent le déterminant inchangé ou reviennent à le multiplier par un scalaire mais dans ce cas la multiplication se faisant dans $\Delta(\alpha_1, ..., \alpha_k, X)$ et $\Delta(\alpha_1, ..., \alpha_k)$ l'expression globale de $P_V(X)$ reste inchangée.

Exemple 83. On se place sur le corps de caractéristique 2 $\mathbb{F}_8 = \{0, 1, \alpha, ..., \alpha^6\} \cong \mathbb{F}_2[X]/(C_{2,3})$ tel que nous l'avons établi dans un chapitre précédent. On souhaite construire un 2-polynôme de 2-degré 2 dont les racines sont par exemple le $\mathbb{F}_2$-vectoriel $V = \text{Vect}\{1, \alpha^3\}$.

On obtient

$$P_V(X) = \frac{\begin{vmatrix} 1 & \alpha^3 & X \\ 1 & \alpha^6 & X^2 \\ 1 & \alpha^{12} & X^4 \end{vmatrix}}{\begin{vmatrix} 1 & \alpha^3 \\ 1 & \alpha^6 \end{vmatrix}} = \frac{\begin{vmatrix} 1 & \alpha^3 & X \\ 1 & \alpha^6 & X^2 \\ 1 & \alpha^5 & X^4 \end{vmatrix}}{\begin{vmatrix} 1 & \alpha^3 \\ 1 & \alpha^6 \end{vmatrix}} = \frac{(\alpha^3 + \alpha^6)X^4 + (\alpha^3 + \alpha^5)X^2 + (\alpha^5 + \alpha^6)X}{\alpha^3 + \alpha^6}$$

La construction de $\mathbb{F}_8$ établie au chapitre précédent donne $\alpha^3 + \alpha^6 = \alpha^4$, $\alpha^3 + \alpha^5 = \alpha^2$ et $\alpha^5 + \alpha^6 = \alpha$. Ainsi le quotient précédent vaut $P_V(X) = \frac{\alpha^4 X^4 + \alpha^2 X^2 + \alpha X}{\alpha^4} = X^4 + \alpha^{-2} X^2 + \alpha^{-3} X = X^4 + \alpha^5 X^2 + \alpha^4 X$.

On peut vérifier avec cette formule que $P_V(1) = P_V(\alpha^3) = 0$. L'ensemble des racines de P_V est $\text{Vect}\{1, \alpha^3\} = \{0, 1, \alpha^3, 1 + \alpha^3 = \alpha\}$.

3.5.3 Complexité

La formule précédente fait intervenir un calcul de déterminant. Les méthodes usuelles de calcul de déterminant de taille $k \times k$ ont une complexité plus que quadratique, par exemple $O(k^{2.376})$ avec l'algorithme de Coppersmith-Winograd. La construction itérative présentée dans la formule (10) donne une meilleure complexité :

Lemme 84. *Soit V un sous-espace vectoriel de $\mathbb{F}_{q^m}$ de dimension d_V. Si l'on connaît une base de V, la construction du polynôme P_V peut se faire en $\mathcal{O}(d_V^2 + d_V \log(m))$ opérations.*

Démonstration. A chaque appel récursif nous devons évaluer un q-polynôme et additionner $X^q\ P_{i-1}(X)$ avec $P_{i-1}(\alpha_k)^{q-1} X P_{i-1}(X)$, le q-degré de $P_{i-1}(X)$ étant de degré $i-1$. Ainsi chaque appel récursif coûte $i + \log(m)$ et le coût est proportionnel à $\sum_{i=0}^{d_V} (i + \log(m)) \in \mathcal{O}(d_V^2 + d_V \log(m))$. □

3.5.4 Structure des racines du RGCD, du RLCM de deux q-polynômes

Dans cette section nous reformulons des résultats présents dans [Ore2] avec le vocabulaire des espaces vectoriels :

Proposition 85. *Soit V et W deux sous-espaces de $\mathbb{F}_{q^m}$, alors le q-polynôme associé à l'intersection de V et W est égal au plus grand diviseur commun à droite de P_V et P_W :*

$$P_{V \cap W} = \mathrm{RGCD}(P_V, P_W)$$

Démonstration. *Soit $\dim(V) = k_1$, $\dim(W) = k_2$ et $\dim(V \cap W) = d$.*

Soit $(\alpha_1, \dots \alpha_d)$ une base de $V \cap W$.

Soit $(\alpha_1, \dots \alpha_d, v_{d+1}, \dots v_{k_1})$ une base de V.

Soit $(\alpha_1, \dots \alpha_d, w_{d+1}, \dots w_{k_2})$ une base de W.

D'après la démonstration de la proposition 81 on connaît l'expression de $P_{V \cap W}$:

$P_{V \cap W}(X) = (X^q - P_{d-1}(\alpha_d)^{q-1} X) \times (X^q - P_{d-2}(\alpha_{d-1})^{q-1} X) \times \dots \times (X^q - \alpha_1^{q-1} X)$

$P_V(X) = (X^q - P_{k_1-1}(v_{k_1})^{q-1} X) \times \dots \times (X^q - P_{V \cap W}(v_{d+1})^{q-1} X) \times P_{V \cap W}(X)$

Ceci prouve que $P_{V \cap W}$ est un diviseur à droite pour P_V. De même $P_{V \cap W}$ est un diviseur pour P_W. Ainsi $P_{V \cap W}$ est un diviseur à droite pour P_V et P_W.

Soit un q-polynôme $\tilde{P}$ tel que $\mathrm{RGCD}(P_V, P_W) = \tilde{P} \times P_{V \cap W}$. Les racines de $\tilde{P}$ seraient dans V et W, or tous les éléments communs à V et W ont déjà été dénombrés donc la seule possibilité est $\tilde{P} = \mathrm{Id}$ (l'espace vectoriel associé étant $\{0\}$) ainsi $P_{V \cap W}$ est le plus grand diviseur commun à droite de P_V et P_W. □

***Corollaire* 86.** *Si $W \subset V$ alors P_W divise P_V.*

Démonstration. *Ici $V \cap W = W$ donc $\mathrm{RGCD}(P_V; P_W) = P_{V \cap W} = P_W$ donc P_W est effectivement un diviseur de P_V.* □

***Corollaire* 87.** *W et V sont en somme directe $\Leftrightarrow P_V$ et P_W sont premiers entre eux à droite (au sens des q-polynômes i.e. $\mathrm{RGCD}(P_V, P_W) = x$)*

Proposition 88. *Soit V et W deux sous-espaces de $\mathbb{F}_{q^m}$, alors le q-polynôme associé à la somme de V et W est égal à l'union à droite de P_V et P_W :*

$$P_{V+W} = \mathrm{RLCM}(P_V, P_W)$$

Démonstration. *On reprend les notations de la proposition 81. Soit* $\dim(V) = k_1$, $\dim(W) = k_2$ *et* $\dim(V \cap W) = d$.

Soit $(\alpha_1, \dots \alpha_d)$ *une base de* $V \cap W$, $(\alpha_1, \dots \alpha_d, v_{d+1}, \dots v_{k_1})$ *une base de* V *et* $(\alpha_1, \dots \alpha_d, w_{d+1}, \dots w_{k_2})$ *une base de* W.

En utilisant la démonstration de la proposition 81, et en prenant en compte que d'après la formule 11 on sait que quel que soit l'ordre des racines choisi dans la base précédente on obtiendra la même expression pour P_{V+W}, on va calculer P_{V+W} de deux manières :

Soit $(\alpha_1, \dots \alpha_d, v_{d+1}, \dots, v_{k_1}, w_{d+1}, \dots w_{k_2})$ *une base de* $V+W$, *alors*

$$P_{V+W}(X) = (X^q - P_{k_2+k_1-1}(w_{k_2})^{q-1} X) \times \dots \times (X^q - P_{k_1-1}(v_{k_1})^{q-1} X) \times \dots \times (X^q - \alpha_1^{q-1} X)$$

$$P_{V+W}(X) = (X^q - P_{k_2+k_1-1}(w_{k_2})^{q-1} X) \times \dots \times P_V(X)$$

Cette expression montre que P_{V+W} est divisible à droite par P_V.

Si on choisit comme ordre pour les éléments de la base $(\alpha_1, \dots \alpha_d, w_{d+1}, \dots, w_{k_2}, v_{d+1}, \dots v_{k_1})$ *alors un raisonnement analogue montre que P_{V+W} est divisible à droite par P_W.*

Ainsi P_{V+W} est un q-polynôme divisible à droite par P_V et P_W.

Remarquons pour conclure que $\deg_q(P_{V+W}) = d + k_1 + k_2$ *; or, comme* $\mathrm{RLCM}(P_V, P_W)$ *doit par définition avoir pour racines tous les éléments de* $(\alpha_1, \dots \alpha_d, v_{d+1}, \dots, v_{k_1}, w_{d+1}, \dots w_{k_2})$, *on a* $\deg_q(\mathrm{RLCM}(P_V, P_W)) \geqslant d + k_1 + k_2$ *donc nécessairement* $\mathrm{RLCM}(P_V, P_W) = P_{V+W}$.

Remarquons que nous avons en même temps redémontré le résultat $\deg_q(\mathrm{RLCM}(P_V, P_W)) = \deg_q(P_V) + \deg_q(P_W) - \deg_q(\mathrm{RGCD}(P_V, P_W))$. □

3.6 Application : utilisation des *q*-polynômes pour déterminer l'intersection de deux sous-espaces vectoriels

On déduit de ce qui précède un nouvel algorithme de recherche de l'intersection de deux sous-espaces vectoriels de $\mathbb{F}_{q^m}$:

Algorithme 1

Entrée : Une base de V et une base de W, deux sous-espaces vectoriels de $\mathbb{F}_{q^m}$
Sortie : Une base de $V \cap W$

- Construire P_V, le q-polynôme dont V est une base des racines, en suivant la méthode de la démonstration de la proposition 81.
- Construire de même P_W.
- Calculer $\mathrm{RGCD}(P_V; P_W)$. Ceci peut être fait par l'algorithme d'Euclide, nous présentons une nouvelle façon de le faire via l'utilisation de la matrice de multiplication à droite dans le chapitre suivant.
- Déterminer une base de l'ensemble des racines de $P_{V \cap W} = \mathrm{RGCD}(P_V; P_W)$ de la façon vue dans [Ber] et la proposition 78.

Cet algorithme, d'une complexité comparable aux algorithmes existants, a l'intérêt d'être clair et simple à mettre en oeuvre, l'intersection d'espaces vectoriels se faisant par le biais d'un calcul de PGCD.

Cet algorithme pourra être intégré à l'algorithme de décodage des codes LRPC vu au paragraphe 4.2 du chapitre VI, dans lequel on est amené à déterminer l'intersection d'espaces vectoriels.

Chapitre IV :
Matrice de multiplication à droite pour des q-polynômes et applications

Les travaux que nous avons effectués sur les codes en métrique rang nous ont fréquemment amenés à manipuler des espaces vectoriels, nous verrons par exemple que l'algorithme de décodage des codes LRPC nécessite de calculer l'intersection de d espaces vectoriels pour une erreur de poids rang d (ces notions seront introduites dans le prochain chapitre). Or nous avons vu lors de l'étude des q-polynômes que tout espace vectoriel pouvait être vu comme l'ensemble des racines d'un q-polynôme (et réciproquement) et nous avons formalisé le lien entre les opérations sur les espaces vectoriels et celles sur les q-polynômes, nous avons vu en particulier que si on a deux espaces vectoriels V et W sur un corps fini et P_V, P_W les q-polynômes associés, alors $V \cap W$ est l'espace vectoriel associé au RGCD de P_V et P_W (autrement dit $P_{V\cap W} = \mathrm{RGCD}(P_V, P_W)$).

Nous avons donc été amenés à étudier de façon plus approfondie l'arithmétique des q-polynômes, et plus généralement des polynômes de Ore.

On rappelle la définition d'un anneau de polynômes de Ore :

Soit $\mathbb{K}$ un corps, σ un automorphisme de $\mathbb{K}$ et δ une σ-dérivation (c'est à dire que pour tous a et $b \in \mathbb{K}$ on a $\delta(ab) = \sigma(a)\delta(b) + \delta(a)b$).

On définit l'anneau des polynômes $\mathbb{K}[X, \sigma, \delta]$ de la façon suivante :

- L'addition de deux polynômes est l'addition usuelle.
- La multiplication de deux constantes est la multiplication usuelle dans $\mathbb{K}$.
- Pour tout $a \in \mathbb{K}$ on a

$$Xa = \sigma(a)X + \delta(a)$$

L'anneau $\mathbb{K}[X, \sigma, \delta]$ n'est pas commutatif mais peut être muni d'une division euclidienne à droite, c'est à dire que pour tous polynômes $A, B \in \mathbb{K}[X, \sigma, \delta]$, il existe $Q, R \in \mathbb{K}[X, \sigma, \delta]$, $\deg(R) < \deg(A)$, tels que $A = QB + R$. En tant qu'anneau euclidien, il est en particulier principal.

Dans la suite, si nous parlons de division euclidienne sans plus de précisions, il s'agira toujours de la division euclidienne à droite dans $\mathbb{K}[X, \sigma, \delta]$.

Dans ce chapitre nous commençons par des rappels sur les notions de résultants et de sous-résultants dans un anneau commutatif de polynômes ainsi qu'une méthode de recherche du PGCD à partir des sous-résultants. Nous donnons aussi une nouvelle méthode de calcul des sous-résultants à partir d'une matrice extraite de la matrice de multiplication.

Nous généralisons la notion de matrice de multiplication aux polynômes de Ore que nous avons vus précédemment. L'introduction de la matrice de multiplication à droite de deux polynômes de Ore nous permet de construire un algorithme de calcul du RGCD.

Nous détaillons ensuite la notion de résultants et sous-résultants de deux polynôme de Ore. Cette notion a été étudiée par Chardin dans [Cha] dans le cas des opérateurs différentiels (c'est à dire, avec les notation précédentes, $\sigma = \mathrm{Id}$). Ces résultats ont été généralisés par Li dans [Li]. Nous reprenons le formalisme de Li que nous simplifions pour l'adapter à notre travail. Nous proposons une nouvelle méthode de calcul des résultants et sous-résultants de polynômes de Ore via l'utilisation de la matrice de multiplication à droite. Nous travaillons ainsi avec une matrice de plus petite taille que la matrice de Sylvester utilisée habituellement.

Remarque 89. Dans beaucoup d'ouvrages les résultants et les sous-résultants sont présentés « en ligne » c'est à dire que les lignes de la matrice à partir de laquelle ils sont calculés correspondent aux coefficients des polynômes qui interviennent. Pour notre part nous présenterons les coefficients des polynômes dans des vecteurs colonnes afin de présenter directement ces matrices comme des matrices d'applications linéaires, les vecteurs colonnes étant les images de la base.

1 Résultants et sous-résultants sur un anneau commutatif

Dans toute cette partie on se place dans $\mathbb{K}[X]$ anneau de polynômes commutatifs à une variable X à coefficients dans un corps $\mathbb{K}$.

$A = \sum_{i=0}^{d_A} a_i X^i$ et $B = \sum_{i=0}^{d_B} b_i X^i$ sont deux polynômes de degrés d_A et d_B. On suppose $d_A \leqslant d_B$.

Nous rappelons des résultats classiques sur les résultants et les sous-résultants [AJ].

1.1 Résultant de deux polynômes

1.1.1 Définition à partir de la matrice de Sylvester et propriétés

On rappelle que la matrice de Sylvester de A et B est la matrice de dimension l'expression de la matrice de Sylvester. Le résultant de A et B est égal au déterminant de cette matrice :

$$\mathrm{Res}(A,B) = \det\left(\mathrm{Sylv}(A,B)\right) = \begin{array}{c} \\ \\ d_A + d_B \text{ lignes} \\ \\ \\ \end{array} \begin{array}{c} \\ 1 \\ x \\ \\ \vdots \\ \\ X^{d_A+d_B-1} \end{array} \begin{array}{c} \overbrace{\phantom{A \ \ldots \ X^{d_B-1}A}}^{d_B \text{ colonnes}} \quad \overbrace{\phantom{B \ \ldots \ X^{d_A-1}B}}^{d_A \text{ colonnes}} \\ A \ \ldots \ X^{d_B-1}A \quad B \ \ldots \ X^{d_A-1}B \end{array} \left| \begin{array}{cccccc} a_0 & & (0) & b_0 & & (0) \\ a_1 & a_0 & & b_1 & b_0 & \\ \vdots & a_1 & a_0 & \vdots & b_1 & b_0 \\ a_p & \vdots & a_1 & b_q & \vdots & b_1 \\ & a_p & \vdots & & b_q & \vdots \\ (0) & & a_p & (0) & & b_q \end{array} \right|$$

L'application la plus importante du calcul du résultant est la détermination de la coprimalité de A et B :

Théorème 90. $\mathrm{Res}(A,B) = 0 \Leftrightarrow A$ *et* B *ont un facteur commun non trivial dans* $\mathbb{K}[X]$.

Corollaire 91. $\mathrm{Res}(A,B) \neq 0 \Leftrightarrow A$ *et* B *sont premiers entre eux dans* $\mathbb{K}[X]$.

1.1.2 Matrice de multiplication et application au calcul du résultant

Avant de nous intéresser au calcul des sous-résultants, commençons par rappeler la formule qui permet de calculer le résultant de A et B noté $\mathrm{Res}(A,B)$ à partir de la matrice de multiplication par B dans $\mathbb{K}[X]/(f)$. Cette formule est donnée par exemple dans [AJ]

On introduit la matrice de multiplication par B dans $\mathbb{K}[X]/(A)$:

Définition 92. *La matrice de multiplication par B dans $\mathbb{K}[X]/(A)$, notée M_B, est la matrice carrée de dimension $d_A \times d_A$ dont les vecteurs colonnes sont les coefficients des polynômes $B \bmod A ; XB \bmod A ; ... ; X^{d_A-1}B \bmod A$ dans la base $1, X..., X^{d_A-1}$.*

$$M_B = \begin{pmatrix} B \bmod A & XB \bmod A & \dots & X^{d_A-1}B \bmod A \end{pmatrix}$$

Théorème 93. *A une constante multiplicative non nulle près,*

$$a_{d_A}^{d_B} \det M_B = \mathrm{Res}(A,B) \tag{12}$$

Comme $a_{d_A} \neq 0$ ceci revient à

$$\det(M_B) = a_{d_A}^{-d_B} \mathrm{Res}(A,B)$$

Démonstration. *L'idée de la construction consiste, par des combinaison linéaires sur les d_A dernières colonnes, à faire apparaître un bloc correspondant à M_B dans la matrice de Sylvester :*

$$\begin{array}{c|cccccc} & 1 & \dots & X^{d_B-1} & B \bmod A & \dots & X^{d_A-1}B \bmod A \\ \hline 1 & a_0 & & (0) & & & \\ X & a_1 & \ddots & & & M_B & \\ \vdots & \vdots & & a_0 & & & \\ X^{d_A-1} & & & \vdots & & & \\ & & & & & & \\ X^{d_A} & a_{d_A} & & a_1 & 0 & \dots & 0 \\ \vdots & & \ddots & \vdots & \vdots & (0) & \vdots \\ X^{d_A+d_B-1} & (0) & & a_{d_A} & 0 & \dots & 0 \end{array}$$

On sera amené à généraliser cette démonstration pour l'adapter au cas des sous-résultants ; on donnera les détails à ce moment-là. □

Remarquons notamment que si le coefficient dominant de A vaut 1, le déterminant de M_B est exactement égal au résultant de A et B.

Corollaire 94. *On retrouve le résultat principal des résultants :* $\det(M_B) = 0 \Leftrightarrow A$ *et B ont un facteur commun dans* $\mathbb{K}[X]$.

On partira de cette formule pour généraliser la définition du résultant de deux polynômes sur des anneaux de polynômes de Ore.

1.2 Suites des sous-résultants

Dans cette partie nous nous proposons de généraliser la formule précédente afin de calculer les coefficients sous-résultants à partir des coefficients apparaissant dans la matrice de multiplication. Commençons par rappeler la définition des coefficients sous-résultants, des polynômes sous-résultants et leurs propriétés :

Soit $k \in \{0...\inf(d_A, d_B)\}$ et Ψ_k l'application linéaire ainsi définie :

Définition 95.

$$\Psi_k : \begin{cases} \mathbb{K}[X]_{d_B-k-1} \times \mathbb{K}[X]_{d_A-k-1} & \to & \mathbb{K}[X]_{d_B+d_A-k-1} \\ (U, V) & \mapsto & AU+BV \end{cases}$$

Ecrivons Mat_{Ψ_k} la matrice associée à Ψ_k :

$$\mathrm{Mat}_{\Psi_k} = \begin{array}{c} \\ \\ d_A+d_B-k \text{ lignes} \\ \\ \\ \end{array} \begin{array}{c} \\ 1 \\ x \\ \\ \vdots \\ X^{d_A+d_B-k-1} \end{array} \begin{array}{c} \overbrace{\phantom{A \ \ldots \ AX^{d_B-k-1} \ B \ \ldots \ BX^{d_A-k-1}}}^{d_A+d_B-2k \text{ colonnes}} \\ A \ \ldots \ AX^{d_B-k-1} \ B \ \ldots \ BX^{d_A-k-1} \\ \begin{pmatrix} a_0 & & (0) & b_0 & & (0) \\ a_1 & \ddots & & b_1 & \ddots & \\ \vdots & & a_0 & \vdots & & b_0 \\ a_{d_A} & & a_1 & b_{d_B} & & b_1 \\ & \ddots & \vdots & & \ddots & \vdots \\ (0) & & a_{d_A} & (0) & & b_{d_B} \end{pmatrix} \end{array} \tag{13}$$

Remarque 96. Lorsque $k=0$, $\mathrm{Mat}_{\Psi_0} = \mathrm{Sylv}(A, B)$

Définition 97. *On considère la matrice carrée de dimension $(d_A + d_B - 2k)$ formée de la $(i+1)$ème ligne et des (d_A+d_B-2k-1) dernières lignes de Mat_{Ψ_k}.*

On note $\mathrm{sr}_{k,i}$ le déterminant de cette matrice :

$$\mathrm{sr}_{k,i}(A,B) = \det \begin{array}{c} \\ \\ d_A+d_B-2k \text{ lignes} \\ \\ \\ \end{array} \begin{array}{c} \\ X^i \\ X^k \\ \vdots \\ X^{d_A+d_B-k-1} \end{array} \begin{array}{c} \overbrace{\phantom{1 \ \ldots \ X^{d_B-k-1} \ 1 \ \ldots \ X^{d_A-k-1}}}^{d_A+d_B-2k \text{ colonnes}} \\ 1 \ \ldots \ X^{d_B-k-1} \ 1 \ \ldots \ X^{d_A-k-1} \\ \begin{pmatrix} a_i & & (0) & b_i & & (0) \\ a_k & & & b_k & & \\ & & & & & \\ & \ddots & \vdots & & \ddots & \vdots \\ (0) & & a_{d_A} & (0) & & b_{d_B} \end{pmatrix} \end{array}$$

Le nombre $\mathrm{sr}_{k,i}(A,B)$ est appelé (k,i)ème coefficient sous-résultat associé à A et B.

Définition 98. *Le k-ème polynôme sous-résultant associé à A et B, noté $\mathrm{Sr}_k(A,\ B)$, est le polynôme défini par*

$$\mathrm{Sr}_k(A,B) = \sum_{i=0}^{k} \mathrm{sr}_{k,i}(A,B) X^i$$

Proposition 99. *Nous avons les propriétés suivantes :*

1. *Lorsque $i > k$, $\mathrm{sr}_{k,i}(A,B) = 0$ puisqu'il y a deux lignes identiques dans la matrice donc*

$$\sum_{i=0}^{d_A+d_B-k-1} \mathrm{sr}_{k,i}(A,B) X^i = \sum_{i=0}^{k} \mathrm{sr}_{k,i}(A,B) X^i$$

   est effectivement un polynôme de degré k.

2. $\mathrm{sr}_{0,0}(A,B) = \mathrm{Res}(A,B)$.

3. *Lorsque* $d_B < d_A$ *on obtient* $\mathrm{Sr}_{d_B}(A,\,B) = 0$ *(*$\mathrm{Sr}_k(A,\,B)$ *est défini dans ce qui suit) et* $\mathrm{sr}_{d_B,d_B}(A,B) = (b_{d_B})^{d_A-d_B}$.

Définition 100. *Le kème polynôme sous-résultant associé à A et B, noté* $\mathrm{Sr}_k(A,\,B)$*, est le polynôme défini par*

$$\mathrm{Sr}_k(A,B) = \sum_{i=0}^{k} \mathrm{sr}_{k,i}(A,B)X^i$$

Comme nous l'avons vu dans la remarque précédente le degré de ce polynôme est effectivement k.

Remarque 101. Attention à ne pas confondre les notations Sr_k et $\mathrm{sr}_{k,i}$.

1.3 Lien entre sous-résultants et PGCD

Lemme 102. *Pour tout* $k \in \{0...\inf(d_A,d_B)\}$ *on a* $\mathrm{Sr}_k(A,B) \in \mathrm{Im}(\Psi_k)$.

Démonstration. *Notons* $(c_{i,j})_{(i,j)\in\{1...d_A+d_B-2k\}\times\{0,d_A+d_B-k-1\}}$ *les coefficients de la matrice* $\mathrm{Mat}(\Psi_k)$.

On effectue un développement de $\mathrm{sr}_{k,j}(A,B)$ *par rapport à la première ligne, qui correspond, on le rappelle, à la ligne i de la matrice* Ψ_k *:*

$$\mathrm{sr}_{k,i}(A,B) = \sum_{j=1}^{d_A+d_B-2k} (-1)^{1+i} c_{i,j} m_{i,k}$$

où $m_{i,k}$ *est le déterminant de la matrice formée par les* $(d_A + d_B - 2k - 1)$ *dernières lignes de* $\mathrm{Mat}(\Psi_k)$ *et toutes les colonnes sauf la ième.*

Le polynôme $\mathrm{Sr}_k(A,B)$ *vaut*

$$\mathrm{Sr}_k(A,B) = \sum_{i=0}^{d_A+d_B-k-1} \mathrm{sr}_{i,k}(A,B)X^i = \sum_{i=0}^{d_A+d_B-k-1} \sum_{j=1}^{d_A+d_B-2k} (-1)^{j+1} c_{i,j} m_{k,j} X^i$$

$$= \sum_{j=1}^{d_A+d_B-2k} (-1)^{j+1} m_{k,j} \sum_{i=0}^{d_A+d_B-k-1} c_{i,j}X^i = \sum_{j=1}^{d_A+d_B-2k} (-1)^{j+1} m_{k,j} P_j$$

avec $P_j = \sum_{i=0}^{d_A+d_B-k-1} c_{i,j}X^i$.

Rappelons que les $c_{i,j}$ *sont les coefficients de* $\mathrm{Mat}(\Psi_k)$*. Ainsi à j fixé nous avons* $P_j = X^{j-1}A$ *pour* $1 \leqslant j \leqslant d_B - k$ *et* $P_j = X^{j-1-d_B+k}B$ *pour* $d_B - k + 1 \leqslant j \leqslant d_A + d_B - 2k$ *:*

$$\begin{array}{c} \\ \\ 1 \\ x \\ \\ \vdots \\ x^{d_A+d_B-k-1} \end{array} \begin{array}{c} \begin{array}{cccccccc} & & & j & & & & \\ & 1 & \ldots & \downarrow & X^{d_B-k-1} & 1 & \ldots & X^{d_A-k-1} \end{array} \\ \begin{pmatrix} a_0 & 0X^0 & & (0) & b_0 & & (0) \\ a_1 & 0X^1 & \ddots & & b_1 & \ddots & \\ \vdots & \vdots & & a_0 & \vdots & & b_0 \\ a_{d_A} & a_0X^{j-1} & & a_1 & b_q & & b_1 \\ & \vdots & \ddots & \vdots & & \ddots & \vdots \\ & a_{d_A}X^{d_A+j-1} & & & & & \\ (0) & (0) & & a_{d_A} & (0) & & b_{d_B} \end{pmatrix} \end{array}$$

Ainsi les P_j appartiennent à $\mathrm{Im}(\Psi_k)$. Comme $\mathrm{Im}(\Psi_k)$ est stable par combinaison linéaire, $\mathrm{Sr}_k(A, B) \in \mathrm{Im}(\Psi_k)$. □

Proposition 103. *Soit* $D = \mathrm{PGCD}(A, B), d = \deg(D)$ *et* $k \in \{0...d\}$.

On a :

1. $k < d \Leftrightarrow \Psi_k$ *est non injective* $\Leftrightarrow \mathrm{Sr}_k(A, B) = 0 \Leftrightarrow \mathrm{sr}_{k,k}(A, B) = 0$

2. $k = d \Rightarrow \mathrm{sr}_{k,k}(A, B) \neq 0$ *et* $\mathrm{Sr}_k(A, B) = \mathrm{PGCD}(A, B)$

Démonstration.

1. $k < d \Rightarrow \Psi_k$ *non injective : il existe* $(A_1, B_1) \neq (0; 0)$ *tels que* $A = A_1 D$ *et* $B = B_1 D$. *Comme* $d > k$ *alors* $\deg(A_1) = d_A - d \leqslant d_A - k - 1$ *et* $\deg(B_1) = d_B - d \leqslant d_B - k - 1$, *de sorte que* $(-B_1, A_1)$ *appartient à* $\mathbb{K}[x]_{d_B - k - 1} \times \mathbb{K}[x]_{d_A - k - 1}$, *l'ensemble de départ de* Ψ_k. *On a de plus* $\Psi_k(-B_1; A_1) = -B_1 A + A_1 B = -B_1 A_1 D + A_1 B_1 D = 0$. *Ainsi* $(-B_1; A_1) \in \mathrm{Ker}(\Psi_k)$ *et donc* Ψ_k *n'est pas injective.*

   Ψ_k *non injective* $\Rightarrow \mathrm{Sr}_k(A, B) = 0$ *: comme* Ψ_k *n'est pas injective, les colonnes de* M_k *sont linéairement dépendantes donc les « sous-colonnes » à* k *coefficients le sont aussi, ainsi les déterminants des matrices carrées* $k \times k$ *extraites de* M_k *sont nuls, en particulier les coefficients sous-résultants sont nuls donc* $\mathrm{Sr}_k(A, B)$ *est nul.*

   $\mathrm{Sr}_k(A, B) = 0 \Rightarrow \mathrm{sr}_{k,k}(A, B) = 0$ *: immédiat car* $\mathrm{sr}_{k,k}(A, B)$ *est le coefficient dominant de* $\mathrm{Sr}_k(A, B)$.

   $\mathrm{sr}_{k,k}(A, B) = 0 \Rightarrow k < d$ *: c'est la contraposée de 2.*

2. *Si* $k = d$, *par le théorème de Bézout il existe* U *et* V *dans* $\mathbb{K}[X]$ *tels que* $\deg(U) \leqslant d_B - d - 1, \deg(V) \leqslant d_A - d - 1$ *et* $D = AU + BV$ *donc* U *et* V *sont dans l'ensemble de départ de* Ψ_d *donc* $\Psi_d(U, V) = D \in \mathrm{Im}(\Psi_d)$.

   D'autre part $\mathrm{Im}(\Psi_d)$, *l'ensemble des polynômes* $AU + BV$, *est un idéal de* $\mathbb{K}[X]$, *principal car* $\mathbb{K}[X]$ *est euclidien donc principal.*

   Il est clair que tout diviseur commun à A *et* B *divise tout élément de* $\mathrm{Im}(\Psi_d)$, *donc* D *divise tout élément de* Ψ_d, *donc l'idéal* $\mathrm{Im}(\Psi_d)$ *est engendré par* D.

   Par le lemme précédent, $\mathrm{Sr}_d(A, B) \in \mathrm{Im}(\Psi_d)$ *donc* $\mathrm{Sr}_d(A, B)$ *est un multiple de* D. *Comme on a vu que* $\deg(\mathrm{Sr}_d(A, B)) \leqslant d$ *alors* $\mathrm{Sr}_d(A, B) = \lambda D, \lambda \in \mathbb{K}$. *Comme* $\mathrm{sr}_{d,d}(A, B) \neq 0, \lambda \neq 0$, *donc à une constante multiplicative non nulle près,* $\mathrm{Sr}_d(A, B) = \mathrm{PGCD}(A, B)$. □

2 Calcul des coefficients sous-résultants à partir d'une matrice extraite de la matrice de multiplication

Dans cette partie nous généralisons au calcul des sous-résultants le théorème 93. Nous proposons une nouvelle manière de calculer les sous-résultants grâce à des matrices extraites de la matrice de multiplication.

2.1 Polynôme déterminantal et application au calcul des sous-résultants

Pour calculer les sous-résultants à partir de la matrice de multiplication nous serons amenés à faire des opérations sur les colonnes ; nous introduisons la notion de polynôme déterminantal, une autre façon de présenter le calcul des sous-résultants, qui permet de vérifier facilement que les opérations sur les colonnes ne changent pas la valeur des polynômes sous-résultants (à une constante multiplicative près)

Définition 104. *Soit $M = (m_{i,j})$ une matrice de dimensions $r \times c$ à coefficients dans $\mathbb{K}$. On suppose $r \geqslant c$.*

On définit la matrice M_i, $0 \leqslant i \leqslant c-r$ comme la matrice $c \times c$ formée des $(c-1)$ dernières lignes de M et de la $(i+1)$-ème.

$$M_i = \begin{array}{c} \\ \text{ligne } i \rightarrow \\ \\ \uparrow \\ \text{dernières } (c-1) \text{ lignes} \\ \downarrow \end{array} \begin{pmatrix} m_{1,1} & \dots & m_{1,c} \\ \vdots & & \vdots \\ m_{i,1} & & m_{i,c} \\ \vdots & & \vdots \\ m_{r-(c-2),1} & & m_{r-(c-2),c} \\ \vdots & & \vdots \\ m_{r,1} & \dots & m_{r,c} \end{pmatrix}$$

On définit le polynôme déterminantal de M, noté $\mathrm{DetPol}(M)$*, comme*

$$\mathrm{DetPol}(M) = \sum_{i=0}^{c-r} \det(M_i) X^i$$

Avec cette notation il est clair que le $k$ème polynôme sous-résultant associé à A et B est égal au polynôme déterminantal de la matrice (13).

Observons que l'on peut faire facilement des combinaisons linéaires sur les colonnes de la matrice intervenant dans un polynôme déterminantal. Comme de telles opérations reviennent à multiplier la matrice à droite par une matrice inversible U nous avons plus précisément

Lemme 105. *Pour toute matrice inversible U de dimension $c \times c$ nous avons*

$$\mathrm{DetPol}(MU) = \mathrm{Det}(U)\mathrm{DetPol}(M)$$

Démonstration. *Faire des opérations sur les colonnes dans la matrice M revient à faire les mêmes opérations sur les colonnes dans les matrices carrées M_i. Ainsi nous avons*

$$\mathrm{DetPol}(MU) = \sum_{i=0}^{c-r} \det(M_i U) X^i = \sum_{i=0}^{c-r} \det(M_i) \det(U) X^i = \mathrm{Det}(U)\mathrm{DetPol}(M)$$

□

2.2 Utilisation de la matrice de multiplication

Théorème 106. *Rappelons que M_B désigne la matrice de multiplication par B dans $\mathbb{K}[X]/(A)$.*

Notons $M_{B,k,i}$ la matrice carrée formée

- *des $d_A - k$ premières colonnes de M_B ;*
- *de la $i$ème ligne de M_B ;*
- *des $d_A - k - 1$ dernières lignes de M_B.*

Alors on a

$$\mathrm{sr}_{k,i} = \det(U_k) a_{d_A}^{d_B - k} \det(M_{B,k,i})$$

Où U_k est une matrice inversible dépendant uniquement de k. On explique dans la démonstration à quoi correspond U_k.

Démonstration. *Rappelons l'écriture de la matrice de Ψ_k :*

$$\begin{array}{c} \\ \\ \\ \mathrm{Mat}_{\Psi_k} = \\ \\ \\ \end{array} \begin{array}{c} \\ \\ \\ d_A+d_B-k \text{ lignes} \\ \\ \\ \end{array} \begin{array}{c} \\ 1 \\ x \\ \\ \vdots \\ \\ x^{d_A+d_B-k-1} \end{array} \begin{array}{c} \overbrace{\phantom{1 \ \ldots \ X^{d_B-k-1} \ 1 \ \ldots \ X^{d_A-k-1}}}^{d_A+d_B-2k \text{ colonnes}} \\ 1 \ \ldots \ X^{d_B-k-1} \ 1 \ \ldots \ X^{d_A-k-1} \\ \begin{pmatrix} a_0 & & (0) & b_0 & & (0) \\ a_1 & \ddots & & b_1 & \ddots & \\ \vdots & & a_0 & \vdots & & b_0 \\ a_{d_A} & & a_1 & b_{d_B} & & b_1 \\ & \ddots & \vdots & & \ddots & \vdots \\ (0) & & a_{d_A} & (0) & & b_{d_B} \end{pmatrix} \end{array}$$

Les $d_B - k$ premières colonnes de la matrice sont formées des coefficients des polynômes $A, XA, ...X^{d_B-k-1}A$ dans la base canonique $(1, ..., X^{d_A+d_B-k-1})$.

Les $d_A - k$ dernières colonnes de la matrice sont formées des coefficients des polynômes $B, XB, ...X^{d_A-k-1}B$ dans la base canonique $(1, ..., X^{d_A+d_B-k-1})$.

Pour tout $l \in \{0...d_A-k-1\}$, effectuons la division euclidienne de X^lB par A :

$$X^lB = Q_lA + R_l \text{ avec } \deg(R_l) \leqslant \deg(A) - 1 = d_A - 1$$

Remarquons que pour tout $l \in \{0...d_A-k-1\}$ on a

$$\deg(Q_l) = \deg(X^lB) - \deg(A) \leqslant l + d_B - d_A \leqslant d_B - k - 1$$

Ceci implique que Q^lA peut être exprimé comme une combinaison linéaire des polynômes A, XA, $...X^{d_B-k-1}A$, autrement dit Q^lA peut être exprimé comme une combinaison linéaire des $d_B - k$ premières colonnes de la matrice de Ψ_k.

Dans les $d_A - k$ dernières colonnes de la matrice de Ψ_k on va remplacer les coefficients des polynômes x^lB par les coefficients des polynômes $R_l = x^lB - Q_lA$, ce qui correspond à des combinaisons linéaires de la $(d_B - k + l)$ème colonne et des $(d_B - k)$ premières, et ne change donc pas la valeur du résultant à une constante multiplicative près, et cette constante multiplicative, qui correspond au déterminant de la matrice inversible U_k correspondant aux opérations sur les colonnes utilisées pour faire faire apparaître la matrice K, est la même pour tous les $\mathrm{sr}_{k,0}$, ..., $\mathrm{sr}_{k,k}$ comme on l'a vu dans le paragraphe précédent.

Comme $\deg(R_l) < \deg(A) = d_A$ on a $r_l \in \mathrm{Vect}(1, X, ...X^{d_A-1})$

En notant K la matrice des coefficients des r_l on peut donc écrire

$$\begin{array}{c} \\ 1 \\ X \\ \vdots \\ X^{d_A-1} \\ \\ X^{d_A} \\ \vdots \\ X^{d_A+d_B-k-1} \end{array} \left| \begin{array}{ccc|ccc} 1 \ \ldots & X^{d_B-k-1} & & 1 \ \ldots & X^{d_A-k-1} & \\ a_0 & & (0) & & & \\ a_1 & \ddots & & & K & \\ \vdots & & a_0 & & & \\ & & \vdots & & & \\ & & & & & \\ a_{d_A} & & a_1 & 0 \ \ldots & 0 & \\ & \ddots & \vdots & \vdots \ (0) & \vdots & \\ (0) & & a_{d_A} & 0 \ \ldots & 0 & \end{array} \right| \tag{14}$$

La matrice K étant constituée des restes de $B, XB, ..., X^{d_A-k-1}B$ dans la division euclidienne par A, elle est en fait constituée des $d_A - k$ premières colonnes de la matrice M_B.

Rappelons que $\mathrm{sr}_{k,i}$ *correspond au déterminant de la matrice obtenue à partir de la matrice précédente en prenant les* d_A+d_B-2k-1 *dernières lignes et la* i*ème.*

D'après les règles de calcul de déterminants par blocs, seules les (d_A-k-1)*èmes lignes de* K *et la* i*ème vont intervenir car après combinaison linéaire on a fait apparaître un bloc nul sur les* (d_B-k) *dernières lignes et les* (d_A-k) *dernières colonnes de* $\mathrm{Mat}(\Psi_k)$ *:*

$$\begin{array}{c|c|c|c|c}
 & 1 \;\dots\; X^{d_B-k-1} & 1 \;\dots\; X^{d_A-k-1} & & \\
\begin{matrix}1\\X\\\vdots\\X^{d_A-1}\end{matrix} & \begin{matrix}a_0 & & (0)\\ a_1 & \ddots & \\ \vdots & & a_0\\ & & \vdots\end{matrix} & K & \begin{matrix}\uparrow\\ d_A\\ \\ \\ \downarrow\end{matrix} & \begin{matrix} & & i\\ & \uparrow & \uparrow\\ d_A+d_B-2k-1 & & d_A-k-1\\ & & \downarrow\end{matrix}\\
\begin{matrix}X^{d_A}\\\vdots\\X^{d_A+d_B-k-1}\end{matrix} & \begin{matrix}a_{d_A} & & \\ & \ddots & \vdots\\ (0) & & a_{d_A}\end{matrix} & \begin{matrix}0 & \dots & 0\\ \vdots & (0) & \vdots\\ 0 & \dots & 0\end{matrix} & \begin{matrix}\uparrow\\ d_B-k\\ \downarrow\end{matrix} & \downarrow
\end{array}$$

D'après les règles de calcul de déterminants par blocs on obtient bien

$$\mathrm{sr}_{k,i} = \det(U_k)\, a_{d_A}^{d_B-k} \det(M_{B,k,i}) \tag{15}$$

où U_k *est la matrice inversible correspondant aux opérations sur les colonnes utilisées pour faire apparaître* K *(donc* $\det(U_k)\neq 0$*)*

et où $M_{B,k,i}$ *est la matrice carrée formée*

- *des* d_A-k *premières colonnes de* M_B *;*
- *de la* i*ème ligne de* M_B *;*
- *des* d_A-k-1 *dernières lignes de* M_B*.* □

Corollaire 107. *A une constante multiplicative non nulle près,*

$$\mathrm{Sr}_k(A,B) = \sum_{i=0}^{k} \det(M_{B,k,i}) X^i$$

3 Résultant de deux polynômes de Ore

La notion de résultant (et de sous-résultant) de polynômes de Ore a été étudiée par Chardin [Cha] dans le cas des opérateurs différentiels ($\sigma=\mathrm{Id}$) puis par Li [Li] dans un cadre général. Nous redéfinisssons ces notions à partir de la matrice de multiplication à droite.

Après avoir rappelé les règles de calcul sur les polynômes dans un anneau de Ore, nous donnerons une définition du résultant de deux polynômes dans un tel anneau. Nous montrons que les propriétés fondamentales des résultants sont conservées.

3.1 Utilisation de la matrice de multiplication à droite

3.1.1 Idée générale

On peut définir, tout comme dans la cas commutatif, l'application

$$\begin{array}{rccc}
\Psi\colon & \mathbb{K}[X,\sigma,\delta] & \to & \mathbb{K}[X,\sigma,\delta]\\
 & (U,V) & \mapsto & UA+VB
\end{array}$$

L'absence de commutativité rend une écriture de Mat_Ψ plus difficile à expliciter. Par exemple, si $\delta \equiv 0$, on a pour tout entier k, $X^k A = X^k \sum_{i=0}^{d_A} a_i X^i = \sum_{i=0}^{d_A} X^k a_i X^i = \sum_{i=0}^{d_A} \sigma^k(a_i) X^{i+k}$ et la matrice devient

$$
\begin{array}{cc}
 & \overbrace{\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad}^{d_A+d_B \text{ colonnes}} \\
 & \begin{array}{ccccccc} A & & \ldots & X^{d_B-1}A & B & \ldots & X^{d_A-1}B \end{array} \\
\begin{array}{c} \\ 1 \\ x \\ \\ \vdots \\ \\ X^{d_A+d_B-1} \end{array} &
\left|\begin{array}{ccccccc}
a_0 & & & (0) & b_0 & & (0) \\
a_1 & \sigma(a_0) & & & b_1 & \sigma(b_0) & \\
\vdots & \sigma(a_1) & \sigma^{d_B-1}(a_0) & \vdots & \vdots & \sigma(b_1) & \sigma^{d_A-1}(b_0) \\
a_{d_A} & \vdots & \sigma^{d_B-1}(a_1) & b_{d_B} & \vdots & \sigma(b_{d_B}) & \sigma^{d_A-1}(b_1) \\
 & \sigma(a_{d_A}) & \vdots & & \sigma(b_{d_B}) & & \vdots \\
(0) & & \sigma^{d_B-1}(a_{d_A}) & (0) & & & \sigma^{d_A-1}(b_{d_B})
\end{array}\right|
\end{array}
\begin{array}{l} \\ \\ \\ d_A+d_B \text{ lignes} \\ \\ \\ \end{array}
$$

D'autres exemples seront posés plus tard avec $\delta \not\equiv 0$.

On ne va pas utiliser cette formule explicite, mais s'intéresser à la définition formelle de la matrice, comme matrice formée des vecteurs colonnes $(A, XA, \ldots X^{d_B-1}A, B, XB, \ldots X^{d_A-1}B)$.

Comme $\mathbb{K}[X, \sigma, \delta]$ est euclidien, on peut immédiatement généraliser au cas non commutatif le théorème (93) : les quotients des $X^k B$ dans la division euclidienne à droite par A peuvent être exprimés comme des combinaisons linéaires des colonnes A, XA, $\ldots X^{d_B-1}A$ (remarquons que les X^k sont multipliés par A à droite, ainsi il s'agit bien d'une division à droite). Ainsi, par combinaison linéaire des colonnes, on peut changer les $X^k B$ en $X^k B \bmod A$, et faire apparaître une matrice de la même forme que (14)

Ainsi, la partie inférieure gauche de la matrice est triangulaire à coefficients diagonaux non nuls, le déterminant de cette matrice est donc égal à celui de M_B à une constante multiplicative non nulle près.

On va utiliser cette idée pour définir le résultant de deux polynômes de Ore.

Dans l'exemple donné ici on avait $\delta \equiv 0$ mais on va voir que les résultats se généralisent pour δ quelconque.

3.1.2 Matrice de multiplication à droite et nouvelle définition du résultant

On note à nouveau $d_A = \deg(A)$ et $d_B = \deg(B)$. On suppose $d_A \leqslant d_B$.

Notons $\mathbb{K}[X, \sigma, \delta]/(A)$ l'ensemble des restes des polynômes de $\mathbb{K}[X, \sigma, \delta]$ dans la division à droite par A. C'est un $\mathbb{K}$-espace vectoriel de dimension d_A dont une base est $1, X \ldots X^{d_A-1}$.

Notons M_B la matrice dont les vecteurs colonnes sont formés des coefficients des restes des $X^i B$ pour $i \in \{0 \ldots p-1\}$ dans la division à droite par A dans la base $1, X \ldots X^{d_A-1}$.

$$M_B = \left(\; B \bmod A \;\middle|\; XB \bmod A \;\middle|\; \ldots \;\middle|\; X^{d_A-1}B \bmod A \; \right)$$

M_B est la matrice associée à l'application linéaire $\mathcal{M}_B$ de multiplication à droite par B dans $\mathbb{K}[X, \sigma, \delta]/(A)$:

$$\mathcal{M}_B : \left\{ \begin{array}{ccc} \mathbb{K}[X, \sigma, \delta]/(A) & \to & \mathbb{K}[X, \sigma, \delta]/(A) \\ P & \mapsto & PB \bmod A \end{array} \right.$$

On va définir le résultant de deux polynômes de Ore comme le déterminant de cette matrice :

Définition 108. *Le résultant des deux polynômes de Ore A et B, à nouveau noté* $\mathrm{Res}(A, B)$ *est défini par*

$$\mathrm{Res}(A, B) = \det(M_B)$$

Remarque 109. Cette formule permet de retrouver le résultant dans le cas commutatif à une constante multiplicative non nulle près.

3.2 Propriétés du résultant de deux polynômes de Ore

Le plus grand diviseur commun de A et B pour la division euclidienne à droite sera noté $\mathrm{RGCD}(A, B)$.

Proposition 110. *L'application linéaire $\mathcal{M}_B$ est de rang maximal d_A si et seulement si A et B sont premiers entre eux à droite (c'est à dire si $\mathrm{RGCD}(A, B) = 1$).*

Démonstration. Si $\mathcal{M}_B$ est de rang maximal, c'est un isomorphisme donc $1 \in \mathrm{Im}(\mathcal{M}_B)$. Ainsi il existe $P \in \mathbb{K}[X, \sigma, \delta]/(A)$ tel que $PB = 1$. On peut considérer que $P \in \mathbb{K}[X, \sigma, \delta]$ avec $d_P < d_A$. Ceci signifie qu'il existe $L \in \mathbb{K}[X, \sigma, \delta]$ tel que $PB = 1 + LA$ soit $PB - LA = 1$. Une telle relation de Bézout[9] implique que $\mathrm{RGCD}(A, B) = 1$.

Réciproquement, si $\mathrm{RGCD}(A, B) = 1$, la relation de Bézout précédente existe c'est à dire $PB - LA = 1$ avec $d_P < d_A$, autrement dit $PB \bmod A = 1$. Comme[10] $d_P < d_A$, P est dans l'ensemble de départ de $\mathcal{M}_B$ et $\mathcal{M}_B(P) = 1$. En multipliant à gauche par X^k, $k \leqslant d_A - 1$, on a donc $X^k(PB - LA) = X^k 1 = X^k \in \mathrm{Im}(\mathcal{M}_B)$. Donc $\mathrm{Im}(\mathcal{M}_B)$ contient une suite de polynômes de degrés échelonnés $\{0...d_P - 1\}$, donc $\mathrm{Im}(\mathcal{M}_B) = \mathbb{K}[X, \sigma, \delta]/(A)$ et $\mathcal{M}_B$ est de rang maximal. □

Corollaire 111. *Les polynômes A et B sont premiers entre eux à droite si et seulement si $\det(M_B) \neq 0$ soit $\mathrm{Res}(A, B) \neq 0$.*

Proposition 112. *Pour tous $A, B, C \in \mathbb{K}[X, \sigma, \delta]$, on a $\mathrm{Res}(A, BC) = \mathrm{Res}(A, B)\mathrm{Res}(A, C)$.*

Démonstration. $\mathrm{Res}(A, BC) = \det(M_{BC}) = \det(M_C M_B) = \det(M_C)\det(M_B) = \mathrm{Res}(A, B)\mathrm{Res}(A, C)$. □

La proposition suivante est une généralisation de la proposition 110. Nous utiliserons cette proposition pour introduire un nouvel algorithme de recherche du PGCD de deux polynômes de Ore.

Proposition 113. *Soient $A, B \in \mathbb{K}[X, \sigma, \delta]$ et $D = \mathrm{RGCD}(A, B)$, alors $\deg(D) = d_A - \mathrm{rk}(\mathcal{M}_B)$ où $\mathrm{rk}(\mathcal{M}_B)$ désigne le rang de l'application linéaire $\mathcal{M}_B$.*

Plus précisément, $\mathrm{Im}(\mathcal{M}_B)$ est un idéal à gauche de $\mathbb{K}[X, \sigma, \delta]/(A)$ engendré par D (on utilise ici le terme d'idéal à gauche par commodité mais il faut bien prendre garde au fait que $\mathbb{K}[X, \sigma, \delta]/(A)$ n'est pas nécessairement un anneau ; cependant les degrés des polynômes qui interviennent sont choisis de façon à ce que les opérations effectuées soient des opérations dans $\mathbb{K}[X, \sigma, \delta]$).

Démonstration. On généralise la preuve de la proposition 110 : si $\mathrm{RGCD}(A, B) = D$, la relation de Bézout est vérifiée : il existe P et L dans $\mathbb{K}[X, \sigma, \delta]$ tels que $PB + LA = D$ avec notamment $d_P < d_A - d_D$, autrement dit $PB \bmod A = D$. Comme $d_P < d_A - d_D$, P est dans l'ensemble de départ de $\mathcal{M}_B$ et $\mathcal{M}_B(P) = D$. En multipliant à gauche par X^k, $k \leqslant d_A - 1$, on a donc $X^k(PB + LA) = X^k D \in \mathrm{Im}(\mathcal{M}_B)$.

9. L'ensemble $\mathbb{K}[X, \sigma, \delta]/(A)$ n'est pas un anneau euclidien a priori mais on s'est ramené à des polynômes de $\mathbb{K}[X, \sigma, \delta]$ qui, lui, est un anneau euclidien ce qui garantit l'existence d'une relation de Bézout.

10. Ce point est important est sera utilisé souvent dans ce paragraphe. Comme on travaille avec des polynômes de degré strictement inférieur à d_A, on n'a pas besoin de quotienter et les opérations dans $\mathbb{K}[X, \sigma, \delta]/(A)$ se ramènent en fait à des opérations dans l'anneau $\mathbb{K}[X, \sigma, \delta]$.

Par définition du RGCD, il est impossible d'avoir dans $\mathrm{Im}(\mathcal{M}_B)$ un polynôme de degré strictement inférieur à d_D donc dim $(\mathrm{Im}(\mathcal{M}_B)) \leqslant d_A - d_D - 1$. Or on vient de voir que la famille $\{D, XD, ..., X^{d_A-d_D-1}D\}$ est dans $\mathrm{Im}(\mathcal{M}_B)$, et cette famille est libre car les polynômes sont de degrés échelonnés. On a donc exactement dim $(\mathrm{Im}(\mathcal{M}_B)) = d_A - d_D - 1$ et $\mathrm{Im}(\mathcal{M}_B) = <D, XD, ..., X^{d_A-d_D-1}D>$. □

Corollaire 114. *D est l'unique polynôme de degré minimal dans* $\mathrm{Im}(\mathcal{M}_B)$.

4 Application : algorithme de recherche du PGCD de deux polynômes de Ore

De la proposition précédente on déduit l'algorithme suivant :

4.1 Algorithme de recherche du PGCD

Algorithme 2

Entrée : A et B de degrés d_A et d_B.
Sortie : RGCD(A, B)

- $B^{<0>} \leftarrow B \bmod A$
- pour i de 1 à $(d-1)$ faire
  $B^{<i>} \leftarrow XB^{<i-1>} \bmod A$
- fin pour
- Construire $M_B = (B^{<0>}|B^{<1>}|...|B^{<d-1>})$
- Ecrire M_B sous forme échelonnée.
- Les vecteurs colonnes de la matrice représentent les coefficients de polynômes de $\mathbb{K}[X, \sigma, \delta]/(A)$. D'après la proposition 113 et son corollaire, la colonne associée au polynôme de plus petit degré donne les coefficients de D.

Coefficients de D ↓

$$M_B = \begin{pmatrix} 0 & \dots & 0 & \times & \times & \dots & \times \\ \vdots & & \vdots & \vdots & & & \\ & & & \times & & & \\ & & & 0 & \vdots & & \\ & & & \vdots & \times & \dots & \\ & & & & 0 & & \vdots \\ & & & & \vdots & & \times \\ & & & & & & 0 \\ & & & & & & \vdots \\ 0 & \dots & 0 & 0 & 0 & \dots & 0 \end{pmatrix}$$

4.2 Etude de la complexité

La complexité de l'algorithme sera comptabilisée en nombre d'opérations dans $\mathbb{K}$.

Lemme 115. *Il est possible de calculer $B \bmod A$ avec une complexité de $O(d_A(d_B - d_A))$ opérations.*

Remarque 116. Ce coût est quasiment linéaire en d_B lorsque le degré A est petit ou au contraire proche de celui de B. Comme $d_A \leqslant d_B$ on peut majorer ce coût par $O(\frac{d_B^2}{2})$.

Reprenons chaque étape de l'algorithme :

- $B^{<0>} \leftarrow B \bmod A$

  D'après le lemme précédent le coût de cette opération est en $O(d_A(d_B - d_A))$.

- pour i de 1 à $(d_A - 1)$ faire $B^{<i>} \leftarrow XB^{<i-1>} \bmod A$

  Seul le monôme dominant de $XB^{<i-1>}$ est éventuellement de degré d_A et doit donc être divisé à droite par A. Cela nécessite $O(d_A)$ opérations dans $\mathbb{K}$; en effet, si on reprend la méthode du lemme précédent, comme le monôme que l'on divise est de degré d_A, une seule itération suffit pour obtenir un reste de degré $\leqslant p-1$.

  Comme on répète cette opération $(d_A - 1)$ fois on a une complexité de $O(d_A(d_A - 1))$

- Ecrire M_B sous forme échelonnée : pour écrire M_B sous cette forme on va utiliser la méthode du pivot de Gauss. On rappelle que chaque étape du pivot de Gauss peut être considérée comme une multiplication matricielle, et qu'il y a en tout d_A étapes. Si on suppose que le coût d'une multiplication matricielle est en $O(d_A^{\mu})$, le coût de l'écriture de M_B sous forme échelonnée est donc de $O(d_A^{1+\mu})$.

Comme $\mu \geqslant 1$, finalement la complexité de l'algorithme est en $O(d_A^{1+\mu} + d_A(d_B - d_A))$.

5 Sous-résultants de deux polynômes de Ore

Dans cette partie nous définissons les sous-résultants de deux polynômes de Ore en nous basant sur les opérateurs de Ore tels qu'ils sont introduits par exemple dans la thèse de Ziming Li [Li]. Dans un souci d'adapter ses résultats à notre travail, le formalisme de certaines notions sera cependant simplifié. En effet, Li manipule des polynômes de Ore à coefficients dans un anneau, nous utilisons des polynômes de Ore à coefficients dans un corps. Li introduit également la notion de modules de polynômes de Ore afin de pouvoir utiliser les polynômes de Ore pour modéliser des problèmes d'équations différentielles linéaires non homogènes. Notre motivation étant de trouver des méthodes de recherche de PGCD, nous continuerons pour notre part à travailler dans des anneaux de polynômes de Ore comme nous le faisons jusqu'à présent. Le fait de se placer dans un tel ensemble (anneau de polynômes de Ore à coefficients dans un corps) est plus restrictif mais permet de simplifier notablement l'énoncé et la démonstration de certains résultats que nous redonnerons dans ce contexte.

On se place sur l'anneau des polynômes $\mathbb{K}[X]$.

5.1 Opérateurs de Ore

Définition 117. *On appelle opérateur de Ore toute application* $\Theta\colon \mathbb{K}[X] \to \mathbb{K}[X]$ *vérifiant les propriétés suivantes :*

- $\Theta(P+Q) = \Theta(P) + \Theta(Q)$ *pour tous* $P, Q \in \mathbb{K}[X]$.
- $\Theta(X^n) = X^{n+1}$
- $\deg(\Theta(A)) = \deg A + 1$ *pour tout* $A \in \mathbb{K}[X]$.
- *(règle multiplicative) Il existe deux applications* σ *et* δ *de* $\mathbb{K}$ *dans lui-même telles que pour tout* $a \in \mathbb{K}$ *et* $A \in \mathbb{K}[x]$,

$$\Theta(aA) = \sigma(a)\Theta(A) + \delta(a)A \tag{16}$$

On définit $\Theta^k(A) = \Theta \circ \Theta \ldots \circ \Theta(A)$ (k fois).

Le lemme suivant sera réutilisé dans les parties suivantes pour déterminer de façon explicite les coefficients dominants de certains polynômes, qui donneront les coefficients diagonaux de certaines matrices :

Lemme 118. *Si* $A = \sum_{i=0}^{d_A} a_i X^i$ *alors* $\deg(\Theta^k(A)) = \deg(A) + k$ *et le coefficient dominant de* $\Theta^k(A)$ *est* $\sigma^k(a_{d_A})$.

Démonstration. *Notons* $\tilde{A} = a_{d_A}^{-1} \sum_{i=0}^{d_A} a_i X^i$. $\tilde{A}$ *est un polynôme de coefficient dominant 1 et on a* $A = a_{d_A}\tilde{A}$.

$\Theta^k(A) = \Theta^k(a_{d_A}\tilde{A}) = \Theta^{k-1}(\Theta(a_{d_A}\tilde{A})) = \Theta^{k-1}(\sigma(a_{d_A})\Theta(\tilde{A}) + \delta(a_{d_A})\tilde{A}) = \Theta^{k-1}(\sigma(a_{d_A})\Theta(\tilde{A})) + \Theta^{k-1}(\delta(a_{d_A})\tilde{A}))$

Comme $\deg(\Theta^{k-1}(\sigma(a_{d_A})\Theta(\tilde{A})) = \deg(\tilde{A}) + k$ *et* $\deg(\Theta^{k-1}(\delta(a_{d_A})\tilde{A}) = \deg(\tilde{A}) + k - 1$, *le terme dominant est à chercher dans* $\Theta^{k-1}(\sigma(a_{d_A})\Theta(\tilde{A})) = \Theta^{k-2}(\Theta(\sigma(a_{d_A})\Theta(\tilde{A}))) = \Theta^{k-2}(\sigma(\sigma(a_{d_A}))\Theta^2(\tilde{A}) + \delta(\sigma(a_{d_A})\Theta(\tilde{A}))$

En répétant k fois ce procédé de manière récursive on voit que le coefficient dominant de $\Theta^k(A)$ *est égal au coefficient dominant de* $\sigma^k(a_{d_A})\Theta^k(\tilde{A})$. *Comme le coefficient dominant de* $\tilde{A}$ *est* 1, *c'est également le cas pour* $\Theta^k(\tilde{A})$ *(par linéarité de* Θ*).*

Ainsi le coefficient dominant de $\sigma^k(a_{d_A})\Theta^k(\tilde{A})$ *donc de* $\Theta^k(A)$ *est bien égal à* $\sigma^k(a_{d_A}) \times 1 = \sigma^k(a_{d_A})$. □

La proposition suivante permet de faire le lien avec ce qui été introduit dans la partie précédente :

Proposition 119. *Soit* Θ *un opérateur de Ore défini comme dans ce qui précède, alors*

- σ *est injective*
- $\sigma(a+b) = \sigma(a) + \sigma(b)$ *pour* $a, b \in \mathbb{K}$
- $\sigma(ab) = \sigma(a)\sigma(b)$ *pour* $a, b \in \mathbb{K}$ *(ce qui implique entre autres* $\sigma(1) = 1$*)*
- $\delta(a+b) = \delta(a) + \delta(b)$ *pour* $a, b \in \mathbb{K}$
- $\delta(ab) = \sigma(a)\delta(b) + \delta(a)b$

  Réciproquement, si σ *et* δ *satisfont aux propriétés précédentes, alors il existe un unique opérateur de Ore satisfaisant à la règle multiplicative de la définition.*

  Dans ce cas σ *est appelé conjugaison et* δ *est appelé pseudo-dérivation.*

Démonstration. *Soient* $a, b \in \mathbb{K}$. *D'après la règle 16, on a*

$$\Theta((a+b)X) = \sigma(a+b)X^2 + \delta(a+b)X$$

mais aussi, comme Θ *est linéaire,*

$$\Theta((a+b)X) = \Theta(aX + bX) = \Theta(aX) + \Theta(bX) = (\sigma(a) + \sigma(b))X^2 + (\delta(a) + \delta(b))X$$

Ce qui prouve par identification que $\sigma(a+b) = \sigma(a) + \sigma(b)$ *et* $\delta(a+b) = \delta(a) + \delta(b)$.

En prenant $b = 0$ *on obtient* $\Theta(aX) = \sigma(a)X^2 + \delta(a)X$. *Comme il est nécessaire que* $\deg(\Theta(aX)) = \deg(aX) + 1$, *si* $a \neq 0$ *alors* $\sigma(a) \neq 0$. *Ceci, avec la règle* $\sigma(a+b) = \sigma(a) + \sigma(b)$, *prouve que* σ *est injective.*

En reprenant la règle 16 on peut écrire

$$\Theta((ab)X) = \sigma(ab)X^2 + \delta(ab)X$$

et

$$\Theta((ab)X) = \Theta(a(bX)) = (\sigma(a)\sigma(b))X^2 + (\sigma(a)\delta(b) + \delta(a)b)X$$

ce qui par identification des coefficients donne $\sigma(ab) = \sigma(a)\sigma(b)$ *et* $\delta(ab) = \sigma(a)\delta(b) + \delta(a)b$.

Réciproquement, supposons que σ *et* δ *satisfont aux conditions précédentes. Supposons qu'il existe un opérateur de Ore* Θ *associé à* σ *et* δ*, alors nécessairement* Θ *doit satisfaire 16 et pour tout monôme* aX^n *on a* $\Theta(aX^n) = \sigma(a)\Theta(X^n) + \delta(a)X^n$ *et* $\Theta(X^n) = X^{n+1}$.

On pose ainsi

$$\Theta(aX^n) = \sigma(a)X^{n+1} + \delta(a)X^n \tag{17}$$

Ceci montre l'existence de $\Theta : aX^n \mapsto \sigma(a)X^{n+1} + \delta(a)X^n$*, qui est bien un opérateur de Ore, et son unicité, car* Θ *est défini de manière unique sur les éléments d'une base de* $\mathbb{K}[X]$. □

Remarque 120. Dans ce cas on a $\delta(1) = 0$ (en effet $\delta(b) = \delta(1b) = \sigma(1)\delta(b) + \delta(1)b = \delta(b) + \delta(1)b$ donc pour $b \neq 0$ on obtient bien $\delta(1) = 0$).

5.2 Sous-résultants de deux polynômes de Ore

Définition 121. *On garde les notations précédentes. Pour* $0 \leqslant k \leqslant d_A$*, on définit le* k*-ème polynôme sous-résultant comme*

$$\mathrm{Sr}_k(A, B) = \mathrm{DetPol}(\mathrm{Mat}(A, \Theta(A), ...\Theta^{d_B - k - 1}(A), B, \Theta(B), ..., \Theta^{d_A - k - 1}(B)))$$

où $\mathrm{Mat}(...)$ *désigne la matrice dont les vecteurs colonnes sont les coefficients des polynômes qui interviennent.*

Les coefficients d'un polynôme sous-résultant sont, comme dans le cas commutatif, appelés les coefficients sous-résultants. Le coefficient associé au monôme X^i *sera comme dans le cas commutatif noté* $\mathrm{sr}_{k,i}$*. Ainsi on retrouve la même notation :*

$$\mathrm{Sr}_k(A, B) = \sum_{i=0}^{k} \mathrm{sr}_{k,i}(A, B)X^i$$

Exemple 122. On considère deux polynômes de Ore $A = a_3X^3 + a_2X^2 + a_1X + a_0$ et $B = b_2X^2 + b_1X + b_0$.

La formule 17 et la linéarité de Θ permettent d'établir les résultats suivants :

$\Theta(A) = \sigma(a_3)X^4 + (\delta(a_3) + \sigma(a_2))X^3 + (\delta(a_2) + \sigma(a_1))X^2 + (\delta(a_1) + \sigma(a_0))X + \delta(a_0)$

$\Theta(B) = \sigma(b_2)X^3 + (\delta(b_2) + \sigma(b_1))X^2 + (\delta(b_1) + \sigma(b_0))X + \delta(b_0)$

$\Theta^2(B) = \sigma^2(b_2)X^4$

$+(\delta(\sigma(b_2)) + \sigma(\delta(b_2)) + \sigma^2(b_1))X^3$

$+(\delta^2(b_2) + \delta(\sigma(b_1)) + \sigma(\delta(b_1)) + \sigma^2(b_0))X^2$

$+(\delta^2(b_1) + \delta(\sigma(b_0)) + \sigma(\delta(b_0)))X$

$+\delta^2(b_0)$

Ainsi nous avons

$$\mathrm{Sr}_2(A,B) = \mathrm{DetPol}(B) = \mathrm{DetPol}\begin{pmatrix} b_0 \\ b_1 \\ b_2 \end{pmatrix} = b_0 + b_1 X + b_2 X^2 = B$$

$$\mathrm{Sr}_1(A,B) = \mathrm{DetPol}(A, B, \Theta(B))$$

$$= \mathrm{DetPol}\begin{pmatrix} a_0 & b_0 & \delta(b_0) \\ a_1 & b_1 & \delta(b_1)+\sigma(b_0) \\ a_2 & b_2 & \delta(b_2)+\sigma(b_1) \\ a_3 & 0 & \sigma(b_2) \end{pmatrix}$$

$$= \begin{vmatrix} a_0 & b_0 & \delta(b_0) \\ a_2 & b_2 & \delta(b_2)+\sigma(b_1) \\ a_3 & 0 & \sigma(b_2) \end{vmatrix} + \begin{vmatrix} a_1 & b_1 & \delta(b_1)+\sigma(b_0) \\ a_2 & b_2 & \delta(b_2)+\sigma(b_1) \\ a_3 & 0 & \sigma(b_2) \end{vmatrix} X$$

$$\mathrm{Sr}_0(A,B) = \mathrm{DetPol}(A, \Theta(A), B, \Theta(B), \Theta^2(B))$$

$$= \begin{vmatrix} a_0 & \delta(a_0) & b_0 & \delta(b_0) & \delta^2(b_0) \\ a_1 & \delta(a_1)+\sigma(a_0) & b_1 & \delta(b_1)+\sigma(b_0) & \delta^2(b_1)+\delta(\sigma(b_0))+\sigma(\delta(b_0)) \\ a_2 & \delta(a_2)+\sigma(a_1) & b_2 & \delta(b_2)+\sigma(b_1) & \delta^2(b_2)+\delta(\sigma(b_1))+\sigma(\delta(b_1))+\sigma^2(b_0) \\ a_3 & \delta(a_3)+\sigma(a_2) & 0 & \sigma(b_2) & \delta(\sigma(b_2))+\sigma(\delta(b_2))+\sigma^2(b_1) \\ 0 & \sigma(a_3) & 0 & 0 & \sigma^2(b_2) \end{vmatrix}$$

6 Calcul des coefficients sous-résultants de deux polynômes de Ore à partir d'une matrice de multiplication

Soit $\mathbb{K}$ un corps, σ un automorphisme de $\mathbb{K}$ et δ une σ-dérivation.

On définit l'anneau des polynômes de Ore $\mathbb{K}[X, \sigma, \delta]$. Pour toute constante a on rappelle que $Xa = \sigma(a)X + \delta(a)$.

6.1 Liens entre les deux points de vue sur les polynômes de Ore

On considère l'application $\Theta : P \mapsto XP$ pour tout polynôme de Ore P.

Proposition 123. *L'application Θ ainsi définie est un opérateur de Ore associé à la conjugaison σ et la pseudo-dérivation δ.*

Démonstration. *Pour tous* $P, Q \in \mathbb{K}[X], \Theta(P+Q) = X(P+Q) = XP + XQ = \Theta(P) + \Theta(Q)$

$\Theta(X^n) = XX^n = X^{n+1}$

Il est évident que $\deg(\Theta(P)) = \deg(XP) = \deg(P) + 1$

Pour $a \in \mathbb{K}$, *on a* $\Theta(aP) = XaP = (Xa)P = (\sigma(a)X + \delta(a))P = \sigma(a)XP + \delta(a)P = \sigma(a)\Theta(P) + \delta(a)P$.

Les quatre points de la définition sont vérifiés, Θ *est bien un opérateur de Ore.* □

6.2 Généralisation de la formule vue dans le cas commutatif

Dans cette partie on va généraliser au cas des polynômes de Ore la formule (15).

D'après ce qu'on a vu dans la partie précédente, le k-ème polynôme sous-résultant associé à A et B est ici défini par

$$\mathrm{Sr}_k(A,B) = \mathrm{DetPol}(A, \Theta(A), ..., \Theta^{d_B-k-1}(A), B, \Theta(B), ..., \Theta^{d_A-k-1}(B))$$

$$\mathrm{Sr}_k(A,B) = \mathrm{DetPol}(A, XA, ..., X^{d_B-k-1}A, B, XB, ..., X^{d_A-k-1}B)$$

$$\mathrm{Sr}_k(A,B) = \sum_{i=0}^{k} \mathrm{sr}_{k,i}(A,B) X^i$$

Notons M la matrice dont les vecteurs colonnes sont les coefficients de $(A, XA, ..., X^{d_B-k-1}A, B, XB, ..., X^{d_A-k-1}B)$ dans la base canonique de $\mathbb{K}[X]_{d_A+d_B-k-1}$:

$$M = d_A + d_B - k \text{ lignes} \quad \begin{array}{c} \\ 1 \\ X \\ \vdots \\ \\ X^{d_A+d_B-k-1} \end{array} \quad \begin{array}{c} \overset{d_A+d_B-2k \text{ colonnes}}{\overbrace{}} \\ A \quad ... \quad X^{d_B-k-1}A \quad B \quad ... \quad X^{d_A-k-1}B \end{array}$$

$$\begin{pmatrix} a_0 & ... & & \times & b_0 & ... & & \times \\ a_1 & & & & b_1 & & & \\ \vdots & & & & \vdots & & & \\ a_{d_A} & & & \vdots & b_{d_B} & & & \vdots \\ & \sigma(a_{d_A}) & & & & \sigma(b_{d_B}) & & \\ & & \ddots & & & & \ddots & \\ (0) & & \sigma^{d_B-k-1}(a_{d_A}) & & & & \sigma^{d_A-k-1}(b_{d_B}) & \end{pmatrix}$$

Par définition de DetPol, $\mathrm{sr}_{k,i}(A,B)$ est le déterminant de la matrice extraite de M constituée des d_A+d_B-2k-1 dernières lignes de M et de la $(i+1)$ème.

$$\mathrm{sr}_{k,i}(A,B) =$$

$$d_A + d_B - 2k \text{ lignes} \quad \begin{array}{c} X^i \\ X^k \\ X^{k+1} \\ \vdots \\ \\ X^{d_A+d_B-k-1} \end{array} \quad \left| \begin{array}{c} \overset{d_A+d_B-2k \text{ colonnes}}{\overbrace{}} \\ 1 \quad ... \quad X^{d_B-k-1} \quad 1 \quad ... \quad X^{d_A-k-1} \end{array} \right.$$

$$\left| \begin{array}{cccccccc} a_i & ... & & \times & b_i & ... & & \times \\ a_k & ... & & \times & b_k & ... & & \times \\ a_{k+1} & & & & b_{k+1} & & & \\ \vdots & & & & \vdots & & & \\ a_{d_A} & & & \vdots & b_{d_B} & & & \vdots \\ & \sigma(a_{d_A}) & & & & \sigma(b_{d_B}) & & \\ & & \ddots & & & & \ddots & \\ (0) & & \sigma^{d_B-k-1}(a_{d_A}) & & & & \sigma^{d_A-k-1}(b_{d_B}) & \end{array} \right|$$

Nous en déduisons la formule suivante généralisant celle vue dans le cas commutatif :

Théorème 124. *Rappelons que M_B est la matrice de multiplication par B dans $\mathbb{K}[X,\sigma,\delta]/(A)$.*

Notons $M_{B,k,i}$ la matrice formée

- *des $d_A - k$ premières colonnes de M_B*
- *de la $i$ème ligne de M_B*
- *des $A - k - 1$ dernières lignes de M_B*

Le coefficient sous résultant $\mathrm{sr}_{k,i}(A,B)$ *vaut*

$$\mathrm{sr}_{k,i}(A,B) = \prod_{j=0}^{d_B-k} \sigma^j(a_{d_A}) \times \det\left(M_{B,k,i}\right)$$

Démonstration. *La démonstration donnée dans le cas commutatif fait intervenir des opérations sur les colonnes de la matrice pour faire apparaître les restes des* X^jB *dans la division euclidienne par* A*. Cette opération se généralise au cas non commutatif où l'anneau* $\mathbb{K}|X,\sigma,\delta|$ *est lui aussi muni d'une division euclidienne.*

On ne va pas rappeler cette démonstration, seulement les étapes importantes et les changements par rapport au cas commutatif.

En notant K *la matrice des coefficients des restes des* $X^jB(0 \leqslant j \leqslant p-k-1)$ *dans la division euclidienne à droite par* A *on peut écrire*

$$\begin{array}{c|cccccccc}
 & 1 & \dots & X^{d_B-k-1} & 1 & \dots & X^{d_A-k-1} & & \\
1 & a_0 & \dots & & \times & & & & \\
\vdots & a_1 & \dots & & \times & & K & & \\
 & \vdots & & & & & & & \\
X^{d_A-1} & & & & & & & & \\
X^{d_A} & a_{d_A} & & & \vdots & & 0 & \dots & 0 \\
\vdots & & \sigma(a_{d_A}) & & & & \vdots & & \vdots \\
 & & & \ddots & & & & & \\
X^{d_A+d_B-k-1} & (0) & & & \sigma^{d_B-k-1}(a_{d_A}) & & 0 & \dots & 0
\end{array}$$

La matrice K *est constituée des restes de* $B, XB, ..., X^{d_A-k-1}B$ *dans la division euclidienne par* A *ce qui revient à dire qu'elle est constituée des* d_A-k *premières colonnes de la matrice* M_B*.*

Rappelons que $\mathrm{sr}_{k,i}$ *correspond au déterminant de la matrice obtenue à partir de la matrice précédente en prenant les* d_A+d_B-2k-1 *dernières lignes et la* i*ème.*

D'après les règles de calcul de déterminants par blocs, seules les (d_A-k-1)*èmes lignes de* K *et la* i*-ème vont intervenir car après combinaison linéaire on a fait apparaître un bloc nul sur les* (d_B-k) *dernières lignes et les* (d_A-k) *dernières colonnes :*

D'après les règles de calcul de déterminants par blocs on obtient bien

$$\mathrm{sr}_{k,i}(A,B) = \prod_{j=0}^{d_B-k} \sigma^j(a_{d_A}) \times \det\left(M_{B,k,i}\right)$$

□

Le seul véritable changement par rapport au cas commutatif est l'apparition du terme $\prod_{j=0}^{d_B-k} \sigma^j(a_{d_A})$ au lieu d'une simple exponentiation de a_{d_A}. Cependant comme $a_{d_A} \neq 0$ et que σ est un automorphisme, tous les $\sigma^j(a_{d_A})$ sont non nuls donc $\prod_{j=0}^{d_B-k} \sigma^j(a_{d_A}) \neq 0$ et on retrouve bien le corollaire important observé dans le cas commutatif à savoir

Corollaire 125. $\mathrm{sr}_{k,i}(A,B) \neq 0 \Leftrightarrow \det\left(M_{B,k,i}\right) \neq 0$

Troisième partie :
Métrique rang et cryptosystème LRPC

Chapitre V :
Codes correcteurs en métrique rang, applications à la cryptographie

La métrique rang a été introduit en 1985 dans [Gab1] comme une alternative à la métrique de Hamming. Peu adaptée à une utilisation concrète en théorie des codes car les canaux de transmission ne permettent pas de modéliser efficacement les erreurs dans cette métrique, elle est en revanche bien adaptée à une utilisation cryptographique.

Avant d'introduire le cryptosystème LRPC dans le prochain chapitre, nous rappelons ici les résultats fondamentaux sur la métrique rang. Nous expliquons comment les premiers cryptosystèmes basés sur des codes très structurés ont pu être attaqués ce qui a motivé l'introduction de cryptosystèmes basés sur des codes moins structurés. Nous redonnons le problème du décodage par syndrome en métrique rang et les algorithmes utilisés pour résoudre ce problème.

La plupart des résultats généraux de ce chapitre ont été tirés de [Loi] et [Loi3].

1 Définitions

On reprend les notations du chapitre 1. On considère un code linéaire $\mathcal{C}$ de longueur n dont l'alphabet est le corps fini $\mathbb{F}_{q^m}$. Les mots du code $\mathcal{C}$ sont des vecteurs de $(\mathbb{F}_{q^m})^n$ que l'on choisit par convention de représenter en ligne.

On a vu au chapitre 2 que tout élément de $\mathbb{F}_{q^m}$ pouvait être exprimé comme un vecteur de $(\mathbb{F}_q)^m$. Ainsi, si on choisit de représenter ces vecteurs en colonne, chaque mot de $\mathcal{C}$ peut être représenté par une matrice de $(\mathbb{F}_q)_{m\times n}$ de la manière suivante :

Soit $B=(\beta_1,...,\beta_m)$ une base de $\mathbb{F}_{q^m}$ comme $\mathbb{F}_q$-espace vectoriel. Soit $x=(x_1,...,x_n)\in\mathcal{C}\subset(\mathbb{F}_{q^m})^n$ un mot du code $\mathcal{C}$. On va associer à ce mot la matrice à m lignes et n colonnes dont la colonne i est égale au vecteur ${}^t(x_{1,i},x_{2,i},...,x_{m,i})$ représentant les coefficients du vecteur x_i dans la base $(\beta_1,...,\beta_m)$. La matrice ainsi associée à x sera notée $\mathrm{Mat}(x)$ où simplement X s'il n'y a pas d'ambiguïté.

$$\begin{array}{ccccc} & (\mathbb{F}_{p^m})^n & \to & & (\mathbb{F}_p)_{m\times n} \\ & & & & (\ x_1,...,x_n\ ) \\ \mathrm{Mat}: & x=(x_1,...,x_n) & \mapsto & X= & \begin{pmatrix} x_{1,1} & ... & x_{1,n} \\ \vdots & & \vdots \\ x_{m,1} & ... & x_{m,n} \end{pmatrix} \begin{matrix} \beta_1 \\ \vdots \\ \beta_m \end{matrix} \end{array}$$

Cette façon de représenter les éléments de $(\mathbb{F}_{q^m})^n$ va nous permettre de définir une nouvelle métrique sur $(\mathbb{F}_{q^m})^n$:

Définition 126. *Métrique rang :*

Soit $x=(x_1,...,x_n)$ un élément de $(\mathbb{F}_{q^m})^n$, le poids rang *de x, noté $w_r(x)$, est égal au rang de la matrice X :*

$$w_R(x)=\mathrm{rk}(X) \tag{18}$$

Soit $y=(y_1,...,y_n)$ un autre élément élément de $(\mathbb{F}_{p^m})^n$, la distance rang *de x à y, notée $d_r(x,y)$, est égale au rang de la matrice correspondant au vecteur $(x-y)$:*

$$d_R(x,y)=\mathrm{rk}(X-Y) \tag{19}$$

On vérifie que l'on définit bien ainsi une distance sur $(\mathbb{F}_{q^m})^n$:

Proposition 127. *La distance rang est une distance au sens classique du terme sur $(\mathbb{F}_{q^m})^n$. Ainsi, l'espace vectoriel $(\mathbb{F}_{q^m})^n$ (ou plus généralement tout code linéaire $\mathcal{C}$ sur $\mathbb{F}_{q^m}$) peut être muni d'une structure d'espace métrique grâce à cette distance.*

Démonstration. *Il est immédiat que $d_r(x,y)=d_r(y,x)$ (symétrie)*

$d_r(x,y)=0\Leftrightarrow \mathrm{Mat}(x-y)$ est de rang $0\Leftrightarrow x-y=0_{m\times n}\Leftrightarrow x=y$ (séparation)

Soit un troisième vecteur z alors $d_r(x,z)=\mathrm{rk}(X-Z)=\mathrm{rk}(X-Y+Y-Z)$. Supposons ce qui ne restreint pas la généralité que $z=0$. Notons $y=(y_1,\ ...,\ y_n)$. On veut montrer que $d_r(x,0)\leqslant d_r(x,y)+d_r(y,0)$ soit $\mathrm{rk}(X)\leqslant \mathrm{rk}(X-Y)+\mathrm{rk}(Y)$. Si $\mathrm{rk}(Y)\geqslant \mathrm{rk}(X)$ le résultat est immédiat, dans le cas contraire si on note r le rang de $\mathrm{rk}(Y)$ alors il existe une base dans laquelle y peut s'écrire $(y_1,...,y_r,0,...,0)$. Si on calcule $x-y$ dans cette base les $(n-r)$ dernières colonnes sont égales aux $(n-r)$ dernières colonnes de x, ainsi le rang du vecteur $x-y$ est au moins égal au rang du vecteur obtenu en ne prenant que les $(n-r)$ dernières colonnes de x, et ce rang est minoré par $\mathrm{rk}(X)-r$. On obtient finalement

$$\mathrm{rk}(X-Y)+\mathrm{rk}(Y)\geqslant \mathrm{rk}(X)-r+r\geqslant \mathrm{rk}(X)$$

Ce qui montre l'inégalité triangulaire. □

Exemple 128. On choisit $p=2, m=4$ et $n=5$. On se place donc sur $\mathbb{F}_{16}\cong\mathbb{F}_2[X]/C_{2,4}$ tel qu'il a été introduit au chapitre 2. On considère le vecteur $x=(1,0,\alpha^4,\alpha^{11},\alpha^{12})\in(\mathbb{F}_{16})^5$. En écrivant les coordonnées du vecteur en colonne dans $\mathbb{F}_2[X]/C_{2,4}$ nous pouvons identifier ce vecteur avec la matrice à m=4 lignes, $n=5$ colonnes et à coefficients dans $\mathbb{F}_p=\mathbb{F}_2$:

$$\mathrm{Mat}(1,0,\alpha^4,\alpha^{11},\alpha^{12})=\begin{pmatrix}1&0&1&0&1\\0&0&1&1&1\\0&0&0&1&1\\0&0&0&1&1\end{pmatrix}$$

Dans cette matrice les colonnes c_1, c_3 et c_4 sont indépendantes, par contre $c_5=c_1+c_3$ donc elle est de rang 3 et le poids rang de x vaut $w_R(x)=3$.

On peut remarquer que dans cet exemple le poids de Hamming est supérieur au poids rang puisque $w_H(x)=4$. C'est une propriété générale :

Proposition 129. *Comparaison entre la métrique rang et la métrique de Hamming :*

Notons $w_H(x)$ le poids de Hamming d'un mot x et $d_H(x,y)$ la distance de Hamming entre x et y, alors

$$w_R(x) \leqslant w_H(x)$$

On en déduit immédiatement :

$$d_R(x,y) \leqslant d_H(x,y)$$

Démonstration. *Le rang de la matrice* $\mathrm{Mat}(x)$ *est majoré par le nombre de colonnes non nulles de cette matrice qui correspond au poids de Hamming.* □

Remarquons que si $m = 1$ alors pour tout vecteur x on a $\mathrm{Mat}(x) = x$, par contre en général $w_r(x) < w_H(x)$, par exemple sur $(\mathbb{F}_{2^1})^n$ le vecteur $(1, ..., 1)$ a pour poids rang 1 et pour poids de Hamming n. Cet exemple met en avant le fait que si l'alphabet du code est $(\mathbb{F}_{q^1})$ le poids rang d'un mot est soit 0 soit 1 ; ainsi, l'utilisation de la distance rang devient pertinente lorsque l'alphabet est $\mathbb{F}_{q^m}$ avec $m > 1$.

2 Codes en métrique rang

2.1 Définitions

On définit dans cette partie la notion de code en métrique rang. On retrouve beaucoup de définitions et de propriétés du chapitre 1 qui se généralisent à la métrique rang :

Définition 130. *Un* code en métrique rang de longueur n *sur le corps* $\mathbb{F}_{q^m}$*, noté* $\mathcal{C}$*, est un sous-ensemble de* $(\mathbb{F}_{q^m})^n$*, où cet espace, vu comme un espace métrique, est muni de la distance rang.*

S'il n'y a pas d'ambiguïté on parlera de code rang *ou même simplement de* code *puisque dans ce chapitre et le suivant nous n'utiliserons que des codes en métrique rang.*

Un tel code est dit linéaire *si c'est un sous-espace vectoriel de* $(\mathbb{F}_{q^m})^n$*.*

La distance minimale *d'un code rang est* $d \overset{déf}{=} \min_{x \neq y \in \mathcal{C}} d_R(x,\, y)$*. Dans le cas d'un code linéaire ce nombre est égal à* $d = \min_{x \in \mathcal{C}} w_R(x)$*.*

Si un code rang linéaire $\mathcal{C}$ *a pour longueur* n*, dimension* k *et distance minimale* d*, on dira que* $\mathcal{C}$ *est un code* $[n, k, d]_r$*. On n'aura pas toujours besoin de mentionner la distance minimale, dans ce cas on dira que* $\mathcal{C}$ *est un code* $[n, k]_r$*.*

Tout comme pour les codes linéaires en métrique de Hamming, on retrouve pour les codes rang linéaire les notions de matrice génératrice*, de* matrice de parité *et de* code dual *qui sont définies de manière identique.*

2.2 Bornes sur les codes en métrique rang

On retrouve les équivalents des bornes en métrique de Hamming.

2.2.1 Borne de Singleton

La borne de Singleton donne une borne sur la capacité de correction :

Proposition 131. *Borne de Singleton :*

Soit $A_{q^m}(n,d)$ le nombre maximal de mots dans un code rang sur $\mathbb{F}_{q^m}$ de longueur n et distance minimale d, alors

$$|\mathcal{C}| \leqslant q^{\min(m(n-d+1),n(m-d+1))}$$

Dans le cas des codes linéaires avec $n \leqslant m$ cette inégalité équivaut à

$$d \leqslant n-k+1$$

Si $n > m$ on peut réécrire

$$d \leqslant 1 + \left\lfloor \frac{(n-k)m}{n} \right\rfloor$$

Définition 132. *Si $n \leqslant m$ et $|\mathcal{C}| = q^{m(n-d+1)}$, soit $d = n-k+1$ pour un code linéaire, alors on dit que le code est* MRD *(pour* Maximum Rank Distance*). C'est l'équivalent des codes MDS pour la distance rang.*

2.2.2 Borne de Gilbert-Varshamov

La borne de Gilbert-Varshamov permet d'assurer l'existence de codes rang pour certains paramètres :

Le nombre d'éléments $S(m,q,t)$ dans une *sphère* de rayon t dans $(\mathbb{F}_{q^m})^n$, est égal au nombre de matrices q-aires de rang t. Pour $t=0$, $S_0=1$, puis ([Loi3])

$$S(n,m,q,t) = \prod_{j=0}^{t-1} \frac{(q^n-q^j)(q^m-q^j)}{q^t-q^j}$$

Remarque 133. La quantité $\prod_{j=0}^{t-1} \frac{q^m-q^j}{q^t-q^j}$, appelée binôme de Gauss et notée $\begin{bmatrix} m \\ t \end{bmatrix}_q$, est égale au nombre d'espaces vectoriels de dimension t dans $\mathbb{F}_{q^m}$. C'est une généralisation du binôme de Newton.

A partir de cette formule nous déduisons le volume d'une *boule* $B(n,\, m,\, q,\, d)$ de rayon d dans $(\mathbb{F}_{q^m})^n$:

$$B(n,m,q,d) = \sum_{i=0}^{d} S(n,m,q,i)$$

Proposition 134. *Il est possible de construire un code rang linéaire $\mathcal{C}[n,k,d]_r$ lorsque*

$$B(n,m,q,d) < q^{m(n-k)}$$

La borne de Gilbert-Varshamov *d'un code rang linéaire sur $\mathbb{F}_{q^m}$, notée $GVR(n,k,m,q)$ est le plus petit entier $\tilde{d}$ tel que $B\big(n,m,q,\tilde{d}\big) \geqslant q^{m(n-k)}$.*

La borne de Gilbert-Varshamov pour un code rang $\mathcal{C}$ ayant H pour matrice duale correspond au plus petit poids rang pour lequel, pour tout syndrome s, il existe en moyenne un mot x de poids rang r tel que $Hx^T = s$. Pour donner une idée du comportement de cette borne, on sait [Loi3] qu'asymptotiquement, lorsque $m=n$, on a

$$\frac{GVR(n,k,m,q)}{n} \sim 1 - \sqrt{\frac{k}{n}}$$

3 Décodage en métrique rang

3.1 Problèmes liés au décodage en métrique rang

3.1.1 Problème du décodage par syndrome en métrique rang

On généralise à la métrique rang le problème du décodage par syndrome en métrique de Hamming :

Problème du décodage par syndrome en métrique rang (Rank Syndrome Decoding problem, ou RSD) :

Soit H une matrice de parité de dimension $(n-k)\times n$ sur $\mathbb{F}_{q^m}$ avec $k\leqslant n$ et soit $\mathbf{s}\in(\mathbb{F}_{q^m})^{n-k}$ et un entier r :

Existe-t-il un élément $\mathbf{x}\in(\mathbb{F}_{q^m})^n$ tel que $w_r(\mathbf{x})\leqslant r$ et $H\mathbf{x}^T=\mathbf{s}$?

Dans ce cas il n'est pas prouvé que le problème est NP-difficile mais ce problème est très proche du problème de décodage par syndrome qui est NP-difficile, de plus le problème peut être vu comme une version structurée du problème MinRank qui est aussi NP-difficile (le problème RSD peut être attaqué comme un problème MinRank mais dans la pratique l'attaque ne marche pas car il y a trop d'inconnues), de plus le problème a été étudié pendant plus de 20 ans les meilleures attaques sont exponentielles, on pense donc que le problème RSD est NP-difficile.

3.1.2 L'algorithme d'Ourivski-Johanson

La première attaque non triviale a été proposée par Chabaud et Stern [CS] en 1996, ensuite en 2002 Ourivski et Johansson [OJ] ont amélioré l'attaque précédente et ont proposé une nouvelle attaque.

L'algorithme d'Ourivski-Johanson est basé sur la proposition suivante :

Proposition 135. *Soit $\mathcal{C}[n,k,d]$ un code rang linéaire de matrice génératrice G. On considère un mot reçu $\mathbf{y}=\mathbf{x}+\mathbf{e}$ avec $x\in\mathcal{C}$ et $w_R(\mathbf{e})\leqslant\frac{d-1}{2}$.*

Soit $\tilde{\mathcal{C}}$ le code $[n,k+1]$ engendré par la matrice $\tilde{G}=G\mathbf{y}$. Les vecteurs non nuls de rang minimum dans $\tilde{\mathcal{C}}$ sont les multiples de $\mathbf{e}$ de la forme $\alpha\mathbf{e},\alpha\in\mathbb{F}_{q^m}$.

On va donc chercher un mot de poids faible $\tilde{\mathbf{e}}=\alpha\mathbf{e}$ dans $\tilde{\mathcal{C}}$ puis retrouver $\mathbf{e}$ grâce à l'égalité $H\tilde{\mathbf{e}}^T=\alpha H\mathbf{y}^T$.

On suppose pour cela, quitte à effectuer une permutation, que $\{1,\ k+1\}$ forme un ensemble d'information de $\tilde{\mathcal{C}}$. On considère $\widetilde{G'}=(I_{k+1}|M)$ et $\widetilde{H'}=(-M^T|I_{n-k-1})$ des matrices génératrice et de parité de $\tilde{\mathcal{C}}$ sous forme systématique.

Soit $\mathbf{x}=(x_1,\ ...,\ x_n)\in\mathcal{C}$ un mot de longueur n poids rang $r\leqslant\frac{d-1}{2}$. Il existe un vecteur $\mathbf{b}=(b_1,...,b_r)\in\mathcal{M}_{1\times r}(\mathbb{F}_{q^m})$ dont les composantes sont linéairement indépendantes[11] sur $\mathbb{F}_q$ et une matrice $A\in\mathcal{M}_{r\times n}(\mathbb{F}_q)$ tels que $\mathbf{x}=\mathbf{b}A$. En notant A_1 (resp. A_2) la matrice formée des $\frac{n}{2}$ premières (resp. dernières) colonnes de A on obtient $\mathbf{x}=\mathbf{b}(A_1|A_2)$. Remarquons que comme $\widetilde{G'}=(I_{k+1}|M)$ on a $\mathbf{b}A_1M=\mathbf{b}A_2$. On est donc amené à résoudre le système

$$\mathbf{b}(A_2-A_1M)=0$$

Pour résoudre ce système on peut procéder de deux façons :

- Par énumération des bases : On énumère les familles libres $(b_1,...,b_r)$. Par symétrie le nombre de familles à énumérer est majoré par $q^{(m-r)(r-1)}$. Pour chaque famille on tente de résoudre le système précédent à $mr(n-k-1)$ équations et nr inconnues.

11. $\mathbf{b}$ représente une base des coordonnées de $\mathbf{x}$. Pour chaque x_i on a $x_i\in<b_1,...,b_r>$

S'il existe un mot de code $\mathbf{x}$ tel que $d_R(\mathbf{x}, \mathbf{y}) \leqslant r$ on peut donc le trouver en un nombre d'opérations de l'ordre de

$$O\big((k+r)^3 q^{(r-1)(m-r)+2}\big)$$

- Par énumération des coordonnées : On cherche une matrice $A_2 - A_1 M$ de rang $\leqslant r-1$. On doit tester le rang de $q^{r(k+1)}$ matrices. Une fois qu'on a trouvé une telle matrice on résout le système précédent. Cette méthode donne une complexité de

$$O\big((k+r)^3 r^3 q^{(r-1)(k+1)}\big)$$

3.1.3 Un nouvel algorithme

On peut se dire intuitivement qu'un code plus long est plus facile à décoder. Pourtant, les deux attaques ne tenaient pas compte de la valeur de n dans l'exposant. Très récemment les deux attaques précédentes ont été généralisées par Gaborit, Ruatta et Schreck dans [GRS] (et utilisées pour briser des réparations du cryptosystème GPT) et une nouvelle configuration algébrique a été proposée.

On suppose que les coefficients de l'erreur $\mathbf{e}$ sont dans un espace E de dimension $\leqslant r$ (on dit que E est le support de $\mathbf{e}$). L'idée de cet algorithme est de chercher un espace vectoriel $E' \supset E$ de dimension $r' \geqslant r$. L'équation du syndrome donne $n-k$ équations dans $\mathbb{F}_{q^m}$ donc $(n-k)m$ équations dans $\mathbb{F}_q$. On peut résoudre ce système si $nr' \leqslant (n-k)m$ donc $r' \leqslant \left\lfloor \frac{m(n-k)}{n} \right\rfloor = m + \left\lfloor \frac{-km}{n} \right\rfloor$.

La probabilité d'avoir $E \subset E'$ est égale au nombre de sous-espaces vectoriels de E' de dimension r divisé par le nombre total de $\mathbb{F}_q$-sous-espaces vectoriels de dimension r de $\mathbb{F}_{q^m}$ soit

$$P(E \subset E') = \frac{\begin{bmatrix} r' \\ r \end{bmatrix}_q}{\begin{bmatrix} m \\ r \end{bmatrix}_q} = q^{-r(m-r')}$$

En prenant $r' = m + \left\lfloor \frac{-km}{n} \right\rfloor$ ceci donne $q^{r\left\lceil \frac{km}{n} \right\rceil}$.

Comme nous l'avons vu dans l'algorithme d'Ourivski-Johanson, il est plus efficace de chercher un multiple de $\mathbf{e}$ dans le code $\tilde{\mathcal{C}}[n, k+1]$. Comme il existe des multiples de l'erreur dont 1 est une coordonnée (par exemple $e_1^{-1}\mathbf{e}$) on peut chercher comme support d'un multiple de l'erreur un espace E' de dimension r' dont on sûr qu'il contient 1, ainsi on va pouvoir chercher dans E' des espaces de dimension $r-1$.

Dans ce contexte on a donc nr' inconnues et $m(n-k-1)$ équations. Un calcul analogue au précédent donne

$$r' = m + \left\lfloor \frac{-(k+1)m}{n} \right\rfloor \text{ et } P(E \subset E') = \frac{\begin{bmatrix} r' \\ r-1 \end{bmatrix}_q}{\begin{bmatrix} m \\ r-1 \end{bmatrix}_q} = q^{-(r-1)(m-r')} = q^{(r-1)\left\lceil \frac{(k+1)m}{n} \right\rceil}$$

En prenant en compte le coût des opérations algébriques, les nouvelles complexités pour les attaques les mieux connues sont à présent dans notre contexte

$$(n-k)^3 m^3 q^{(r-1)\left\lceil \frac{(k+1)m}{n} \right\rceil}$$

pour la généralisation des attaques précédentes incluant m dans le coefficient exponentiel et $q^{r\left\lceil \frac{r(k+1)-(n+1)}{r} \right\rceil}$, une borne inférieure pour une attaque hybride utilisant la base de Groebner de [GRS]. Remarquons qu'il y a plusieurs autres attaques avec les bases de Groebner (Kipnis-Shamir ou des attaques par des mineurs) mais elles ne sont pas significatives dans le contexte où nous travaillons (voir [GRS] pour des détails), sauf les attaques de [LP] qui peuvent être plus efficaces que celles de [GRS] dans le cas où q est élevé.

Remarque 136. Dans le calcul précédent on supposait qu'on connaissait un vecteur $1 \in E'$. Si on connaît a vecteurs de E' la complexité baisse significativement car on obtient un exposant $r-a$ au lieu de r ou $r-1$. Cette remarque est également valable pour l'algorithme d'Ourivski-Johanson.

3.2 Importance de la complexité algorithmique du décodage des codes en métrique rang

Les meilleurs algorithmes de décodage pour des codes génériques de Hamming de taux fixé ont une complexité temporelle exponentielle en la longueur n et sont décrits par une matrice génératrice ou une matrice de parité nécessitant $\Omega(n^2)$ bits. Un code rang constitué par exemple de matrices $n \times n$ à coefficients dans $\mathbb{F}_2$ sans autre structure que la $\mathbb{F}_2$-linéarité nécessite de même d'être décrit par une matrice génératrice ou une matrice de parité de longueur n^2 et nécessite par conséquent $\Omega(n^4)$ bits. D'un autre côté, la meilleure complexité connue pour un algorithme de décodage de code rang sans structure particulière est exponentielle en n^2 ce qui donne empiriquement un comportement similaire au niveau de la complexité en fonction du nombre de bits d'entrée comparé aux codes en métrique de Hamming.

Cependant, lorsqu'il est pourvu de cette linéarité supplémentaire sur $\mathbb{F}_{q^m}$, le code peut être décrit par une matrice génératrice de longueur n dans le corps à 2^n éléments ce qui veut dire qu'il peut être décrit en $\Omega(n^3)$ bits au lieu des $\Omega(n^4)$ bits mentionnés précédemment. On suppose que cette linéarité supplémentaire ne nécessite pas d'être introduite au prix de procédés de décodage significativement plus rapides. Pour des applications cryptographiques on essaie souvent de se baser sur le fait qu'il est difficile de décoder une sous-classe de codes qui ont une description compacte, par exemple les codes quasi-cycliques, mais aucun résultat théorique de complexité algorithmique n'est connu pour ces sous-classes. Dans [GZ] Gaborit et Zémor donnent une représentation compacte des codes tout en conservant une estimation théorique de la complexité.

La cryptographie basée sur la métrique rang appartient à la classe plus large des cryptosystèmes post-quantiques, qui constituent une classe alternative de cryptosystèmes *a priori* résistants à des attaques par un ordinateur quantique performant. En effet, pour la cryptographie post-quantique, la sécurité du cryptosystème est d'habitude reliée à un problème NP-difficile. Evidemment, la notion de NP-difficulté étant une réduction au pire cas, c'est une estimation grossière pour la cryptographie mais pour avoir une idée fiable de la difficulté intrinsèque d'un problème c'est mieux que de n'avoir pas de résultat théorique du tout comme le montrent les récentes avancées sur le problème du logarithme discret dans des petites caractéristiques [BGJT]

Le premier cryptosystème basé sur la métrique rang a été proposé en 1991 par Gabidulin, Paramonov et Tretjakov (le cryptosystème GPT [GPT] qui est une adaptation du cryptosystème de McEliece pour la métrique rang et les codes de Gabidulin).

L'une des particularités des problèmes basés sur la métrique rang, comparés aux réseaux ou à la métrique de Hamming, est que la complexité en pratique des meilleures attaques pour les problèmes basés sur la métrique rang [GRS] croît très vite comparé aux problèmes analogues en métrique de Hamming (attaques par ensemble d'information [BJMM]). Comme nous l'avons mentionné précédemment, de telles attaques ont un terme quadratique (lié aux paramètres du code rang) dans leur coefficient exponentiel, alors que pour les problèmes en distance de Hamming (et d'une certaine façon aussi pour les attaques heuristiques LLL pour la cryptographie basée sur les treillis), les meilleures attaques en pratique ont un terme exponentiel dont l'exposant est seulement linéaire en les paramètres du code. Ceci conduit à des codes rang ayant une complexité en $\exp\left(\Omega\left(N^{2/3}\right)\right)$, au lieu de $\exp\left(\Omega\left(N^{1/2}\right)\right)$ pour les codes de Hamming, où N est la taille des éléments en entrée i.e. le nombre de symboles q-aires nécessaires pour décrire le code.

Dans la pratique cela signifie qu'il est possible d'obtenir des paramètres sûrs (e.g. les tailles des clés) de seulement quelques milliers de bits pour les structures cryptographiques basées sur la difficulté du décodage de codes génériques en métrique rang alors qu'au moins 100000 bits sont nécessaires pour les codes en distance de Hamming ou les treillis. Des instances génériques de codes rang sont utilisées par exemple pour de l'authentification sans divulgation d'information dans [GSZ] et des instances faiblement structurées sont utilisées pour le cryptosystème LRPC, similaire au cryptosystème NTRU [HPS] pour les treillis ou au récent cryptosystème MDPC [MTSB] pour les codes, que nous allons présenter (voir [GMRZ] et chapitre suivant). On peut aussi l'utiliser pour une méthode de signature détaillée dans [GRSZ] que nous mentionnerons ici car elle est basée sur une méthode de décodage analogue à celle du cryptosystème LRPC.

Evidemment, avec les codes en métrique de Hamming et les treillis, il est possible de faire décroître la taille des paramètres jusqu'à quelques milliers de bits avec une structure additionnelle ([BCGO], [LPR]), mais alors on se ramène à des problèmes de décodage pour des classes spéciales de codes dont on ne connaît pas la complexité et on perd les propriétés de réduction aux problèmes difficiles. Remarquons en particulier que le schéma d'identification de [GRSZ] est le *seul* cryptosystème avec une clé publique de seulement quelques milliers de bits dont la sécurité est liée au fait de résoudre une instance générale d'un problème (décoder des codes rang génériques) avec une réduction de sécurité à un problème NP-difficile.

Mentionnons aussi que les codes rang ont également trouvé de nombreuses applications en théorie des codes, particulièrement en space-time coding et en network coding, deux domaines qui ont connu des développements importants depuis leur parution ([TSC], [LYC]). Les codes de Gabidulin, et une variante, les codes de Koetter-Kschichang [KK], sont prédominants dans ces domaines et leur décodage optimal est devenu un challenge ([GX], [GNW], [MV])

Remarquons enfin que puisque les codes utilisés pour des applications (cryptographiques ou autres) en métrique rang ont tendance à être des codes rang dans le sens présenté ici, c'est à dire linéaires sur l'extension $\mathbb{F}_{q^m}$, les problèmes de décodage et de distance minimale sont plus pertinents que les mêmes problèmes pour la classe plus vague des codes matriciels, dont on s'est référé à la NP-complétude de nombreuses fois.

4 Un exemple fondamental : les codes de Gabidulin

Dans cette partie nous étudions les codes de Gabidulin introduits dans [Gab1]. Ces codes sont un équivalent des codes de Reed-Solomon pour la métrique rang.

Pour la construction de ces codes nous aurons fréquemment besoin de notions vues au chapitre 3 concernant l'automorphisme de Frobenius et les q-polynômes. Nous rappelons rapidement les principales notations :

On note $\theta : x \mapsto x^q$ l'automorphisme de Frobenius sur $\mathbb{F}_{q^m}$ vu comme un $\mathbb{F}_q$-espace vectoriel. Pour tout $g_j \in \mathbb{F}_{q^m} \cong (\mathbb{F}_q)^m$, on note $g_j^{[i]} = \theta^i(g_j)$ la puissance $i$ème de l'automorphisme de Frobenius évalué en g_j, tel qu'introduit dans le chapitre 3.

L'anneau des q-polynômes (ou polynômes linéarisés) en une indéterminée X, noté, $\mathbb{F}_{q^m}[X, \theta]$, est l'ensemble des polynômes

$$A(X) = \sum_{i=0}^{d_A} a_i X^{q^i} \text{ avec } a_i \in \mathbb{F}_{q^m}$$

Cet anneau est muni de l'addition usuelle de polynômes, la multiplication (non commutative) de deux q-polynômes est égale à leur composition i.e. $A \times B(X) = A \circ B(X) = A(B(X))$.

4.1 Définition

Définition 137. *Soit* $\mathbf{g} = (g_1, ..., g_n)$ *un vecteur de* $(\mathbb{F}_{q^m})^n$. *On considère chaque* g_j *comme un vecteur de* $(\mathbb{F}_q)^m$ *comme expliqué précédemment. On suppose que la famille* $\{g_1, ..., g_n\}$ *est libre dans* $(\mathbb{F}_q)^m$ *(ce qui n'est possible que si* $n \leqslant m$, *ce que l'on suppose donc aussi).*

On considère la matrice

$$\mathbf{G} = \begin{pmatrix} g_1 & \dots & g_n \\ \vdots & & \vdots \\ g_1^{[k-1]} & \dots & g_n^{[k-1]} \end{pmatrix}$$

Le code de Gabidulin sur $\mathbb{F}_{q^m}$ *de dimension* k *et de vecteur générateur* $\mathbf{g}$ *est le code rang linéaire de matrice génératrice* $\mathbf{G}$*. Ce code est noté* $\mathrm{Gab}_k(\mathbf{g})$*.*

4.2 Propriétés

Nous rappelons quelques propriétés des codes de Gabidulin :

Le code de Gabidulin $\mathrm{Gab}_k(\mathbf{g})$ est un code d'évalution de q-polynômes de q-degré strictement inférieur à k sur le vecteur g :

$$\mathrm{Gab}_k(\mathbf{g}) = \left\{ (A(g_1), ..., A(g_n)); A(X) = \sum_{i=0}^{k-1} a_i X^{[i]}, a_i \in \mathbb{F}_{q^m} \right\}$$

Remarque 138. Les codes de Gabidulin sont une généralisation à la métrique rang des codes de Reed-Solomon en métrique de Hamming. On a vu au chapitre 1 que les codes de Reed-Solomon étaient des codes d'évaluation de polynômes.

Les codes de Gabidulin sont MRD.

La matrice suivante est une matrice génératrice de $\mathrm{Gab}_k(\mathbf{g})$ sous forme systématique :

$$\mathbf{G}_{\mathrm{syst}} = \begin{pmatrix} 1 & 0 & \dots & 0 & P_1(g_{k+1}) & \dots & P_1(g_n) \\ 0 & 1 & \ddots & 0 & P_2(g_{k+1}) & \dots & P_2(g_n) \\ \vdots & \ddots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & P_k(g_{k+1}) & \dots & P_k(g_n) \end{pmatrix}$$

où pour $i = 1...k$, P_i est l'unique q-polynôme de q-degré k vérifiant pour tout $j \leqslant k$ $P_i(g_j) = \delta_{i,j}$. L'unicité de ce q-polynôme vient de l'indépendance linéaire des g_j et de la proposition 81 du chapitre 3.

Une matrice de parité de $\mathrm{Gab}_k(\mathbf{g})$ est donnée par

$$\mathbf{H} = \begin{pmatrix} h_1 & \dots & h_n \\ \vdots & \ddots & \vdots \\ h_1^{[d-2]} & \dots & h_n^{[d-2]} \end{pmatrix}$$

avec $\mathbf{h} = \left(\lambda_1^{[d]}, ..., \lambda_n^{[d]} \right)$ et les λ_i solutions de l'équation

$$\sum_{i=1}^{n} \lambda_i g_i^{[j]} = 0$$

4.3 Décodage des codes de Gabidulin

L'algorithme de décodage par syndrome présenté au chapitre I se généralise au cas des codes de Gabidulin et plus généralement aux codes en métrique rang.

Remarquant que les codes de Gabidulin sont des codes d'évaluation de polynômes linéaires au même titre que les codes de Reed-Solomon sont des codes d'évaluation de polynômes, Loidreau propose dans [Loi] un algorithme de décodage pour les codes de Gabidulin, d'une complexité quadratique, qui s'inspire de l'algorithme de Berlekamp-Welch pour les codes de Reed-Solomon reconstruisant le polynôme de positions d'erreurs du mot reçu.

Les similarités entre les codes de Gabidulin et les codes de Reed-Solomon sont flagrantes à ce point qu'il est naturel de chercher pour les codes de Gabidulin un algorithme de décodage en liste du type de celui proposé par Sudan dans [GS] et qui permet d' « augmenter » la capacité de correction d'un code en autorisant à décoder au-delà de ladite capacité de correction. Une telle avancée trouverait immédiatement des applications en cryptographie basée sur les codes correcteurs où on a vu que la sécurité était liée à la capacité de correction des codes utilisés.

A l'heure actuelle ce problème n'a toujours pas été résolu. L'algorithme de Sudan fait intervenir des polynômes multivariés dont il n'existe pas d'équivalent dans les q-polynômes. La non-commutativité de la loi de multiplication sur les q-polynômes rend a priori une généralisation de l'algorithme de Sudan difficile à concevoir.

5 Cryptographie en métrique rang

Nous présentons rapidement dans cette partie le cryptosystème original GPT présenté dans [GPT] et expliquons en quoi sa forte structure le rend vulnérable. Nous renvoyons le lecteur au chapitre I pour des rappels sur la construction d'un cryptosystème de McEliece.

Des rappels sur les travaux ultérieurs en cryptographie basée sur les codes correcteurs, en métrique rang ou en métrique de Hamming, ont été donnés dans l'introduction de la thèse.

5.1 Introduction

L'usage de la métrique rang pour détecter ou décoder des erreurs paraît peu pertinent car les erreurs commises sur les canaux réels de communication sont rarement modélisables en métrique rang. La métrique rang est cependant adaptée au *network coding* ([TSC], [LYC]).

En cryptographie basée sur les codes, la métrique rang semble intéressante car elle permet de neutraliser les attaques par ensemble d'information. Précisons ce point :

Il est aisé de voir que pour des vecteurs quelconques $\mathbf{x}$ et $\mathbf{y} \in (\mathbb{F}_{q^m})^n$ on a $d_R(\mathbf{x}, \mathbf{y}) \leqslant d_H(\mathbf{x}, \mathbf{y})$ et $w_R(\mathbf{x}) \leqslant w_H(\mathbf{x})$, en effet toute composante nulle dans $\mathbf{x}$ est également de rang 0 et « ne compte pas » non plus pour le poids rang.

Prenons par exemple le vecteur $\mathbf{e} = (\alpha, ..., \alpha) \in (\mathbb{F}_{q^m})^n$ où α est un élément quelconque non nul de $\mathbb{F}_{q^m}$. Le poids de Hamming de $\mathbf{e}$ vaut $w_H(\mathbf{e}) = n$ alors que le poids rang de $\mathbf{e}$ vaut $w_R(\mathbf{e}) = 1$.

Il existe ainsi des vecteurs non nuls dont le poids rang est très faible alors que toutes leurs composantes sont non nulles. Si $\mathbf{e}$ représente le vecteur erreur lors d'une transmission, il va donc modifier toutes les composantes du mot émis ce qui empêche de retrouver ce mot par des attaques par ensemble d'information. Ainsi, les meilleurs algorithmes de décodage sont beaucoup moins performants qu'en métrique de Hamming ce qui permet d'envisager de réduire la taille de la clé publique.

Partant de cette observation, Gabidulin, Paramonow et Tretjakov proposent dans [GPT] le cryptosystème GPT, un cryptosystème en métrique rang utilisant les codes de Gabidulin. Cependant, la forte structure des codes de Gabidulin rend ce cryptosystème facilement attaquable.

5.2 Le cryptosystème GPT

La première version du cryptosystème GPT [Gab1] a été attaquée dans [Gib]. Gabidulin et Ourivski rajoutent dans [GO] une *matrice de distorsion* pour parer l'attaque. C'est cette version que nous présentons puisqu'elle englobe le cas cas initial.

- **Génération de clés :**

  On choisit un code de Gabidulin $\mathcal{C}[n,k,d]_r$ sur $\mathbb{F}_{q^m}$ de matrice génératrice G. On suppose que l'on sait décoder t erreurs pour $\mathcal{C}$, pour les codes de Gabidulin on a $t=\left\lfloor \frac{n-k}{2} \right\rfloor$. On génère aléatoirement R, une matrice inversible de dimension $k\times k$ sur $\mathbb{F}_{q^m}$, Z, une matrice à k lignes et t_1 colonnes à coefficients dans $\mathbb{F}_{q^m}$ appelée matrice de distorsion, et T, une isométrie linéaire de dimension $(n+t_1)\times(n+t_1)$.

  La clé publique est

  $$G' = R(\ G\ \mid\ Z\ )T$$

  La matrice concaténée $(\ G\ \mid\ Z\ )$ a donc k lignes et $n+t_1$ colonnes.

  La clé privée est constituée de R, G, Z et T.

- **Chiffrement :**

  On veut transmettre le message $\mathbf{x}$. L'émetteur génère un vecteur erreur $\mathbf{e}$ aléatoire de poids t et envoie

  $$\mathbf{y} = \mathbf{x}G' + \mathbf{e}$$

- **Déchiffrement :**

  Connaissant la clé privée, on calcule $\mathbf{y}T^{-1} = \mathbf{x}R\ (\ G\ \mid\ Z\ ) + \mathbf{e}T^{-1}$, il supprime les t_1 dernières coordonnées de $\mathbf{y}T^{-1}$ puis décode dans $\mathcal{C}$.

Figure 9. Cryptosystème GPT

La matrice de distorsion Z a été rajoutée pour se présmunir d'une première attaque proposée dans [Gib].

D'autres variantes du cryptosystème GPT ont été introduites par la suite, citons par exemple [BL] et [OGHA].

5.3 Attaque structurelle

Nous rappelons rapidement dans cette section l'attaque structurelle présentée par Overbeck dans [Ove]. Cette attaque structurelle est basée sur le fait qu'en faisant agir l'automorphisme de Frobenius sur un code de Gabidulin, on retrouve « beaucoup » d'éléments communs avec le code lui-même. Comme s'agit d'une propriété inhérente aux codes de Gabidulin, il est envisageable d'utiliser cette attaque pour d'autres cryptosystèmes basés sur ces codes.

Pour réaliser cette attaque on construit la matrice par blocs $\widetilde{G}'$ suivante définie par

$$\widetilde{G}' = \begin{pmatrix} R & 0 & \dots & 0 \\ 0 & R^{[1]} & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \dots & 0 & R^{[n-k-1]} \end{pmatrix} \begin{pmatrix} G & Z \\ G^{[1]} & Z^{[1]} \\ \vdots & \vdots \\ G^{[n-k-1]} & Z^{[n-k-1]} \end{pmatrix} T \overset{\text{notation}}{=} \tilde{R}(\ \tilde{G}\ \mid\ \tilde{Z}\ )T$$

où la matrice $R^{[i]}$ est obtenue en appliquant la puissance i-ème de l'automorphisme de Frobenius $\theta^i : x \mapsto x^{q^i}$ à tous les coefficients de R (on a une définition analogue pour $G^{[i]}$ et $R^{[i]}$).

Remarquons que la matrice T étant à coefficients dans $\mathbb{F}_q$ on aurait $T^{[i]} = T$.

On montre que le rang de $\tilde{G}$ est égal à $n-1$ et que le rang de $\tilde{Z}$ est égal à t_1 avec une forte probabilité si la matrice Z est aléatoire et si $t_1 \ll k(n-k)$. Ainsi le rang de $(\ \tilde{G}\ |\ \tilde{Z}\ )$ est égal à $n+t_1-1$ avec une forte probabilité.

Lorsque ce rang est exactement égal à $n+t_1-1$ On va construire une attaque basée sur le résultat suivant :

Lemme 139. *Si le noyau de $\widetilde{G'}$ est de dimension 1 alors :*

- *Il existe $\mathbf{h} = (h_1, ..., h_n) \in (\mathbb{F}_q)^n$, de rang n, tel que $\mathrm{Ker}(\widetilde{G'}) = \{T^{-1}(\alpha \mathbf{h}|0)^T; \alpha \in \mathbb{F}_{q^m}\}$.*
- *Si $\mathbf{y} \in \mathrm{Ker}(\widetilde{G'})$ alors pour toute matrice carrée $Q \in \mathcal{M}_{n+t_1}(\mathbb{F}_q)$ vérifiant $Q\mathbf{y} = (\ \mathbf{x}\ |\ 0\ )^T$ alors Q est inversible et vérifie*

$$TQ^{-1} = \begin{pmatrix} A & B \\ 0 & D \end{pmatrix}$$

où A est de dimension $n \times n$ inversible et D de dimension $t_1 \times t_1$ inversible. Il est possible de trouver une telle matrice Q en temps polynomial.

La matrice $\tilde{R}(\ \tilde{G}\ |\ \tilde{Z}\ )T$ a $n+t_1$ colonnes. Lorsque son rang est de dimension exactement égale à $n+t_1-1$ le rang du noyau vaut 1 et on est donc dans les conditions d'application de la proposition précédente.

On peut alors attaquer le système en construisant une matrice Q vérifiant les propriétés du lemme précédent. La matrice $G'Q^{-1}$ est égale à $R(\ GA\ |\ Z'\ )$. La matrice GA est une matrice génératrice du code $\mathrm{Gab}_k(\mathbf{g}A)$. Gabidulin montre dans [Gab2] comment déterminer le support de $\mathrm{Gab}_k(\mathbf{g}A)$. A partir de là il est possible de déterminer GA, puis R et Z'.

Loidreau montre dans [Loi] comment se prémunir contre cette attaque : elle n'est réalisable que si le rang de $(\ \tilde{G}\ |\ \tilde{Z}\ )$ est exactement $n+t_1-1$. En choisissant une matrice de distorsion Z de faible rang on montre que la dimension du noyau de $\widetilde{G'}$ augmente ce qui rend l'attaque précédente inapplicable.

Cependant un tel choix implique que $t_1 > n-k$ ce qui a pour conséquence d'augmenter la taille de la clé publique. Loidreau propose notamment un cryptosystème basé sur une clé publique de 18432 bits ce qui représente une taille très faible comparés aux cryptosystèmes de McEliece plus anciens mais reste nettement supérieur aux taille des clés des cryptosystème basés sur la factorisation de grands nombres ou les courbes elliptiques.

6 Conclusion

La métrique rang semble être une alternative intéressante à la métrique de Hamming pour la cryptographie basée sur les codes correcteurs, car elle a la propriété de neutraliser les attaques par ensemble d'information.

Les premiers travaux allant dans ce sens ont été faits dans [GPT]. Le cryptosystème GPT présenté dans cet article est cependant vulnérable aux attaques structurelles, à moins d'augmenter considérablement la taille de la clé publique, ce qui rend alors le cryptosystème peu compétitif par rapport aux cryptosystèmes « classiques » dont les clés publiques restent beaucoup plus petites. Des cryptosystèmes alternatifs présentés ultérieurement mais toujours très structurés, ont pu être attaqués également et la communauté regarde ces cryptosystèmes avec circonspection d'autant plus que les propriétés calculatoires intrinsèques à la métrique rang ne sont pas encore bien connues.

Dans le prochain chapitre nous présentons un nouveau cryptosystème basé sur une nouvelle famille de codes très peu structurés ce qui le prémunit de ce type d'attaques.

Chapitre VI :
Codes LRPC ; cryptosystème LRPC

Cette partie est basée sur l'article [GMRZ] écrit par Philippe Gaborit, Olivier Ruatta, Gilles Zémor et moi-même.

Dans cette partie nous introduisons une nouvelle famille de codes en métrique rang : les codes Low Rank Parity Check (LRPC) pour lesquels nous proposons un algorithme de décodage probabiliste performant. Cette famille de codes peut être vue comme l'équivalent des codes LDPC pour la métrique rang. Nous proposons ensuite ces codes pour une utilisation cryptographique dans le schéma de chiffrement de McEliece. Nous proposons un algorithme de décodage probabiliste performant. Nous présentons une attaque possible dans le cas particulier des codes DC-LRPC et nous expliquons pourquoi cette attaque ne marche pas si l'on choisit des paramètres adéquats pour le code.

A la différence des précédents algorithmes de chiffrement basés sur la métrique rang - particulièrement ceux basés sur les codes de Gabidulin - les codes que nous utilisons sont très peu structurés. Notre cryptosystème peut être vu comme un équivalent du cryptosystème NTRU [HPS] (et aussi du récent cryptosystème MDPC [MTSB]) dans le contexte de la métrique rang. Globalement notre système aboutit à la création d'une clé publique d'une très petite taille de 1517 bits pour une sécurité de 2^{80}, de plus notre système est est plus de 100 fois plus rapide que [MTSB] (en terme de nombre d'opérations) avec une complexité de déchiffrement dont l'ordre de grandeur est 2^{17} opérations dans le corps de base et une probabilité d'échec que l'on peut rendre arbitrairement petite.

Nous nous référons à l'introduction du présent document pour des éléments sur l'historique de la cryptographie en métrique et la motivation de la création du cryptosystème LRPC.

1 Généralités

Soit $q = p^r$ une puissance d'un nombre premier p, un entier m et V_n un $\mathbb{F}_{q^m}$-espace vectoriel de dimension n. Soit $\beta = \{\beta_1, ..., \beta_m\}$ une base de $\mathbb{F}_{q^m} \cong (\mathbb{F}_q)^m$ vu comme un $\mathbb{F}_q$-espace vectoriel.

Soit $\mathcal{F}_i$ l'application de $\mathbb{F}_{q^m}$ dans $\mathbb{F}_q$ qui à tout élément v vu comme un vecteur de $(\mathbb{F}_q)^m$ associe sa i-ème coordonnée dans la base β.

$$\begin{array}{rccc} \mathcal{F}_i : & \mathbb{F}_{q^m} & \to & \mathbb{F}_q \\ & v = v_1\beta_1 + \cdots + v_m\beta_m & \mapsto & \mathcal{F}_i(v) = v_i \end{array}$$

Comme nous l'avons vu au chapitre précédent, à tout vecteur $x = (x_1, ..., x_n) \in V_n$, on associe la matrice $X \in \mathcal{M}_{m,n}(\mathbb{F}_q)$ dans laquelle $X_{i,j} = \mathcal{F}_i(x_j)$. Chaque composante $x_j \in \mathbb{F}_{q^m}$ est vue comme un vecteur (colonne) de $(\mathbb{F}_q)^m$ écrit dans la base β.

$$X = \begin{array}{c} \begin{pmatrix} x_1 & \dots & x_n \end{pmatrix} \\ \begin{pmatrix} x_{11} & \dots & x_{1n} \\ \vdots & & \vdots \\ x_{m1} & \dots & x_{mn} \end{pmatrix} \end{array} \begin{array}{c} \\ \beta_1 \\ \vdots \\ \beta_m \end{array}$$

Le poids rang de x noté $w_r(x)$ est le rang de X ; la distance rang de deux éléments x et y de $(\mathbb{F}_{q^m})^n$ notée $d_r(x, y)$ est le rang de $X - Y$.

Un code rang $\mathcal{C}$ de longueur n et dimension k sur $\mathbb{F}_{q^m}$ est un sous-espace de dimension k de $\mathbb{F}_{q^m}$ muni de la métrique rang. La distance minimale du code correspond au rang du vecteur non nul de rang minimal.

Nous nous référons au chapitre précédent pour plus de détails sur la distance rang.

Définition 140. *Soit $x = (x_1, ..., x_n) \in (\mathbb{F}_{q^m})^n$ un vecteur de rang r. Notons E le $\mathbb{F}_q$-sous espace vectoriel de $\mathbb{F}_{q^m}$ engendré par $x_1, ..., x_n$. L'espace vectoriel E est appelé* support *de x.*

Remarque 141. L'espace E est de dimension r. On note $\{E_1, ..., E_r\}$ une base de E.

Remarque 142. La notion de support d'un mot de code pour la distance de Hamming et celle introduite dans la définition précédente sont différentes même si elles ont un point commun : dans les deux cas, étant donné un syndrome de poids faible associé à x, une fois que le support est connu, dans les deux cas, il reste seulement à résoudre un système linéaire.

Remarque 143. Pour un code quelconque, l'action du groupe général linéaire ne change pas le poids des mots. Dans le cas des codes rang, cette action ne change pas non plus les supports des mots. Remarquons aussi que dans le cas de la distance de Hamming sur $\mathbb{F}_q$ le fait d'augmenter la valeur de q permet d'augmenter la distance minimale d'un code mais ne change pas le type du support (qui reste un coefficient binomial de même longueur) alors qu'en métrique rang, augmenter q augmente considérablement la taille du support (i.e. le nombre de bases que l'on peut trouver dans le Gaussien binomial).

Notation 144. *Dans tout ce qui suit $\mathcal{C}$ désigne un code rang $[n, k]_r$ sur le corps $\mathbb{F}_{q^m}$.*

Sauf mention contraire on prendra $k = \frac{n}{2}$, autrement dit $n - k = k$.

On prend une matrice génératrice de $\mathcal{C}$ qui sera notée G et une matrice de parité notée H.

2 Résultats sur le produit de deux sous-espaces

Avant d'introduire les codes LRPC nous avons besoin d'introduire des résultats sur le produit de deux sous-espaces. Afin de ne pas distraire l'attention du lecteur sur des points techniques de démonstration, lesdites démonstrations seront données en fin de chapitre.

Définition 145. *Soient A et B deux $\mathbb{F}_q$-sous-espaces vectoriels de $\mathbb{F}_{q^m}$ de dimensions respectives α et β et $\{A_1, ..., A_\alpha\}$ et $\{B_1, ..., B_\beta\}$ deux bases respectives de ces espaces (les A_i et B_j étant des éléments de $\mathbb{F}_{q^m}$).*

Nous notons $\langle A.B\rangle$ l'espace produit *engendré par* $\{a.b, a \in A, b \in B\}$.

La famille $\{A_1.B_1, ..., A_1.B_\beta, ..., A_\alpha.B_1, ..., A_\alpha.B_\beta\}$ est génératrice pour $\langle A.B\rangle$ et la dimension de $\langle A.B\rangle$ est bornée par $\alpha\beta$.

Pour définir notre algorithme de décodage probabiliste, nous avons besoin de déterminer la probabilité que la dimension ne soit pas maximale (i.e. $\alpha\beta < m$) lorsque α et β sont relativement petits. Nous supposons $\alpha\beta < m$ et nous allons déterminer la dimension de l'espace produit $\langle A.B\rangle$.

Soient A et B deux $\mathbb{F}_q$-sous-espaces aléatoires de $\mathbb{F}_{q^m}$ de dimensions respectives α et β. Nous supposons $\alpha\beta < m$.

Nous allons nous baser sur le constat suivant :

Lemme 146. *Soient A' et B deux sous-espaces de $\mathbb{F}_{q^m}$ de dimensions α' et β telles que* $\dim \langle A'B\rangle = \alpha'\beta$. *Soit $A = A' + \langle a\rangle$ où a est un élément aléatoire uniformément choisi de $(\mathbb{F}_q)^m$, alors*

$$\text{Prob}(\dim \langle AB\rangle < \alpha'\beta + \beta) \leqslant \frac{q^{\alpha'\beta+\beta}}{q^m}$$

Proposition 147. *Soit B un sous-espace fixé, supposons que nous construisons un sous-espace aléatoire A en choisissant α vecteurs aléatoires de $\mathbb{F}_{q^m}$ indépendamment (indépendamment dans le sens probabiliste du terme) et soit A le sous-espace engendré par ces α vecteurs indépendants, alors*

$$\text{Prob}(\dim \langle AB\rangle = \alpha\beta) \geqslant 1 - \alpha\frac{q^{\alpha\beta}}{q^m}$$

Soit B un sous-espace fixé de $\mathbb{F}_{q^m}$ contenant 1 et soit $\langle B^2\rangle$ le sous-espace engendré par tous les éléments de B. Soit $\beta_2 = \dim \langle B^2\rangle$. Soit A un sous-espace aléatoire de $\mathbb{F}_{q^m}$ de dimension α. D'après la proposition précédente nous avons

$$\text{Prob}(\dim \langle AB^2\rangle = \alpha\beta_2) \geqslant 1 - \alpha\frac{q^{\alpha\beta_2}}{q^m}$$

Remarque 148. Nous avons $\beta_2 \leqslant \frac{\beta(\beta+1)}{2}$

Lemme 149. *Supposons que* $\dim \langle AB^2\rangle = \alpha\beta_2$. *Soit $e \in \langle AB\rangle$ avec $e \notin A$. Supposons $eB \subset \langle AB\rangle$, alors il existe $x \in B, x \notin \mathbb{F}_q$, tel que $xB \subset B$.*

Proposition 150. *Supposons que m est premier. Soient A et B deux sous-espaces de dimensions respectives α et β. Soit (b_i) une base de B et $S = \langle AB\rangle$, alors*

$$\text{Prob}\left(\bigcap_i b_i^{-1}S = A\right) \geqslant 1 - \alpha\frac{q^{\alpha\frac{\beta(\beta+1)}{2}}}{q^m}$$

Proposition 151. *Soit B un sous-espace de dimension β contenant 1 tel que* $\dim(B + Bb^{-1}) = 2\beta - 1$ *pour un certain $b \in B$. Soit A un sous-espace choisi aléatoirement de dimension α, alors*

$$\text{Prob}(\langle AB\rangle \cap \langle AB\rangle b^{-1} = A) \geqslant 1 - \alpha\frac{q^{\alpha(2\beta-1)}}{q^m}$$

Remarque 152. Il est intéressant de remarquer que, dans la pratique, la probabilité qu'une borne supérieure soit donnée dans les propositions 150 et 151 décroît beaucoup plus vite vers 0. En effet lorsque le degré de l'extension m augmente de 1 (pour $m \geqslant rd$) la probabilité que $\bigcap_i b_i^{-1} S \neq A$ est divisée par un facteur d'au moins q^{r-1}. Ceci signifie qu'en pratique la borne supérieur précédente est assez mauvaise et que l'on peut considérer que dès que m est supérieur à rd de 8 ou plus, la probabilité est bien en dessous de 2^{-30}. Ce sera le cas lorsque nous choisirons les paramètres dans la dernière section.

3 Low Rank Parity Check Codes

L'idée à la base de ces codes consiste à généraliser l'approche classique des codes LDPC en métrique de Hamming à la métrique rang. Il y a une analogie naturelle entre matrices avec faibles densité et matrices avec faible rang.

3.1 Contexte d'utilisation et définitions

Définition 153. *Un code avec matrice de parité de faible rang, en anglais* Low Rank Parity Check code *(LRPC), de rang d, longueur n et dimension k sur le corps $\mathbb{F}_{q^m}$ est un code rang dont la matrice de parité H de dimension $(n-k) \times n$ est telle que le sous-espace de $\mathbb{F}_{q^m}$ engendré par les coefficients $(h_{ij})_{1 \leqslant i \leqslant n-k; 1 \leqslant j \leqslant n}$ de H a pour dimension au plus d. Nous appellerons cette dimension le poids (rang) de H.*

Notons F le sous-espace de $\mathbb{F}_{q^m}$ engendré par les coefficients de H et $\{F_1, ..., F_d\}$ une base de F.

Dans la pratique cela signifie que pour tout $1 \leqslant i \leqslant n-k; 1 \leqslant j \leqslant n$, il existe des $(h_{ijk})_{1 \leqslant k \leqslant d} \in \mathbb{F}_q$ tels que

$$h_{ij} = \sum_{k=1}^{d} h_{ijk} F_k$$

Nous pouvons définir une sous-classe spéciale des codes LRPC :

Définition 154. *Un code quasi-cyclique avec matrice de parité de faible rang (*Quasi-Cyclic Low Rank Parity Check, ou QC-LRPC*) de rang d est un code quasi-cyclique ayant une matrice de parité H quasi-cyclique de faible poids d.*

3.2 Cas particulier important : les codes DC-LRPC (Double Circulant Low Rank Parity Check)

3.2.1 Matrices circulantes ; matrices coublement circulantes

Définition 155. *Soit $A = (a_{i,j})_{1 \leqslant i \leqslant k, 1 \leqslant j \leqslant n} \in \mathcal{M}_{k \times n}(\mathbb{F}_{q^m})$. On dit que A est une matrice circulante si pour tout $i, j \geqslant 2$ on a $a_{i,j} = a_{i-1,j-1}$ et $a_{i,1} = a_{i-1,n}$. Si A est carrée circulante de dimension $k \times k$ on peut donc écrire :*

$$A = \begin{pmatrix} a_0 & a_1 & \dots & a_{k-1} \\ a_{k-1} & a_0 & \dots & a_{k-2} \\ & & \ddots & \\ a_1 & a_2 & \dots & a_0 \end{pmatrix}$$

En posant le calcul il est facile de voir que la somme et le produit de deux matrices circulantes sont des matrices circulantes. La transposée d'une matrice circulante est elle aussi une matrice circulante.

L'ensemble des matrices circulantes de $\mathcal{M}_k(\mathbb{F}_{q^m})$ que nous noterons est isomorphe à l'anneau quotient de polynômes $\mathbb{F}_{q^m}[X]/(X^k-1)$, plus précisément nous avons :

$$\begin{array}{rcl} \mathbb{F}_{q^m}[X]/(X^k-1) & \cong & \{A \in \mathcal{M}_k(\mathbb{F}_{q^m}), A \text{ circulante}\} \\ A_{\text{pol}} = \sum_{j=1}^{k-1} a_j X^j & \leftrightarrow & A_{\text{mat}} = \begin{pmatrix} a_0 & a_1 & \dots & a_{k-1} \\ a_{k-1} & a_0 & \dots & a_{k-2} \\ & & \ddots & \\ a_1 & a_2 & \dots & a_0 \end{pmatrix} \end{array}$$

Nous ferons par la suite l'analogie entre la matrice et le polynôme. Nous notons A_{mat} une matrice circulante et A_{pol} le polynôme associé. S'il n'y a pas d'ambiguïté nous noterons simplement A la matrice.

On en déduit en particulier que la matrice A_{mat} est inversible si et seulement si le polynôme associé est premier avec X^k-1 et dans ce cas A_{mat}^{-1} est encore une matrice circulante.

Nous présentons pour finir le résultat suivant utilisé dans le paragraphe 6.2. On peut en trouver une démonstration dans [Hau] :

Proposition 156. *Le rang de A_{mat} est égal à* $\deg(\text{PGCD}(A_{\text{pol}}, X^k-1))$.

Corollaire 157. *Soient deux matrices circulantes A et B. Soit un polynôme D_{pol} qui divise X^k-1, alors le rang de la matrice $(DA_{\text{mat}}|DB_{\text{mat}})$ est majoré par $(k-\deg(D_{\text{pol}}))$.*

Remarque 158. La matrice $(DA_{\text{mat}}|DB_{\text{mat}})$ est la concaténation de deux matrices circulantes. On dit qu'une telle matrice est doublement circulante (voir paragraphe suivant).

3.2.2 Codes DC-LRPC

Nous portons un intérêt particulier au cas des codes LRPC doublement circulants (DC-LRPC) de rang d, qui sont des codes dont la matrice de parité H est une matrice doublement circulante de poids d.

On rappelle qu'une matrice doublement circulante est la concaténation de deux matrices circulantes, plus précisément dans notre cas une matrice doublement circulante peut s'écrire

$$H = (\ A\ \mid\ B\ ) = \begin{pmatrix} a_0 & a_1 & \dots & a_{k-1} & b_0 & b_1 & \dots & b_{k-1} \\ a_{k-1} & a_0 & \dots & a_{k-2} & b_{k-1} & b_0 & \dots & b_{k-2} \\ & & \ddots & & & & \ddots & \\ a_1 & a_2 & \dots & a_0 & b_1 & b_2 & \dots & b_0 \end{pmatrix}$$

avec $k = \frac{n}{2}$, A et $B \in \mathcal{M}_k(\mathbb{F}_{q^m})$ et comme précédemment $\dim < a_0, \dots, a_{k-1}, b_0, \dots, b_{k-1} > = d$.

Il y a une forte probabilité pour que les matrices B et notamment A soient inversibles, dans ce cas on peut mettre la matrice H sous forme systématique en effectuant les opérations sur les lignes correspondant à la multiplication à gauche par A^{-1} :

$$H' = (\ I_k\ \mid A^{-1}B\ )$$

Dans ce cas une matrice génératrice du code sous forme systématique est $(\ -A^{-1}B^T\ \mid\ I_k\ )$, dans les exemples que l'on prend on est en caractéristique 2 donc cette matrice peut s'écrire :

$$G' = (\ A^{-1}B^T\ \mid\ I_k\ )$$

Nous verrons ultérieurement que G' correspond à la clé publique du cryptosystème DC-LRPC D'après ce que nous avons établie au paragraphe précédent la matrice $A^{-1}B^T$ est circulante. Ainsi, pour transmettre G', il n'y a pas besoin de transmettre tous les coefficients mais seulement ceux de la première ligne de $A^{-1}B^T$. Ceci permet un gain significatif sur la taille de la clé publique que nous détaillerons dans le paragraphe correspondant.

3.3 Création d'une matrice de décodage à coefficients dans le corps de base $\mathbb{F}_q$ à partir de la matrice de parité

Dans ce qui suit nous nous intéresserons aussi à une matrice particulière A_H^r qui permet de décoder les codes LRPC d'une manière très efficace :

Dans cette section nous voyons comment nous pouvons écrire les équations du syndrome dans $\mathbb{F}_{q^m}$ comme des équations dans le corps de base $\mathbb{F}_q$ afin d'obtenir un algorithme de décodage que nous détaillerons dans la section suivante. Nous introduisons notamment la matrice A_H^r qui correspond à la matrice H « dépliée » dans $\mathbb{F}_q$. Nous utiliserons cette matrice pour le décodage.

Notons H la matrice de parité telle qu'introduite au paragraphe précédent. Notons $\mathbf{e} = (e_1, ..., e_n) \in (\mathbb{F}_{q^m})^n$ le vecteur erreur. On suppose que la dimension du sous-espace vectoriel de $\mathbb{F}_{q^m}$ engendré par $\{e_1, ..., e_n\}$ est égale à r c'est à dire qu'il existe $\{E_1, ..., E_r\} \in \mathbb{F}_{q^m}$ tels que pour tout $1 \leqslant i \leqslant n$, il existe des scalaires $e_{iu} \in \mathbb{F}_q$ tels que $e_i = \sum_{u=1}^{r} e_{iu} E_u$. Ainsi l'espace $E = <E_1, ..., E_r>$ est le support de l'erreur.

L'équation du syndrome dans $\mathbb{F}_{q^m}$ s'écrit $H\mathbf{e}^T = \mathbf{s}$ soit

$$\begin{pmatrix} h_{11} & h_{12} & \dots & h_{1n} \\ h_{21} & \ddots & & \\ \vdots & & & \\ h_{(n-k)1} & \dots & & h_{(n-k)n} \end{pmatrix} \begin{pmatrix} e_1 \\ e_2 \\ \vdots \\ e_n \end{pmatrix} = \begin{pmatrix} s_1 \\ s_2 \\ \vdots \\ s_{n-k} \end{pmatrix} \tag{20}$$

On considère l'espace $\langle F.E \rangle = <F_v E_u, 1 \leqslant v \leqslant d, 1 \leqslant u \leqslant r>$. Supposons que la dimension de cet espace est exactement dr.

En exprimant les coordonnées de H dans la base des F_l, les coordonnées de e dans la base des E_u et les coordonnées de s dans la base des $F_l E_u$, nous allons réécrire l'équation (20) comme un système linéaire dans $\mathbb{F}_q$:

Choisissons une unique composante du syndrome, par exemple s_1, qui vérifie l'équation

$$\begin{pmatrix} h_{11} & h_{12} & \dots & h_{1n} \end{pmatrix} \begin{pmatrix} e_1 \\ e_2 \\ \vdots \\ e_n \end{pmatrix} = \begin{pmatrix} s_1 \end{pmatrix} \tag{21}$$

On peut décomposer s_1 dans la base $\{F_v E_u, 1 \leqslant v \leqslant d, 1 \leqslant u \leqslant r\}$ et ainsi écrire $s_1 = \sum_{j=1}^{n} h_{1j} e_j = \sum_{j=1}^{n} \sum_{v=1}^{d} \sum_{u=1}^{r} h_{1jv} e_{ju} F_v E_u$ où les $h_{ijv} e_{ju}$ sont dans $\mathbb{F}_q$.

On décompose les éléments $(h_{ij})_{1 \leqslant j \leqslant n}$ en vecteurs colonnes à d composantes dans la base des F_v et les $(e_j)_{1 \leqslant j \leqslant n}$ en vecteurs lignes dans la base des E_u, de sorte que l'équation (21) devient

$$\begin{pmatrix} h_{111} & h_{121} & \dots & h_{1n1} \\ h_{112} & h_{122} & \dots & h_{1n2} \\ \vdots & \vdots & & \vdots \\ h_{11d} & h_{12d} & \dots & h_{1nd} \end{pmatrix} \begin{pmatrix} e_{11} & e_{12} & \dots & e_{1r} \\ e_{21} & e_{22} & & e_{2r} \\ \vdots & \vdots & \ddots & \vdots \\ e_{n1} & e_{n2} & \dots & e_{nr} \end{pmatrix} = \begin{pmatrix} s_{111} & s_{112} & \dots & s_{11r} \\ s_{121} & s_{122} & \dots & s_{12r} \\ \vdots & \vdots & & \vdots \\ s_{1d1} & s_{12d} & \dots & s_{1dr} \end{pmatrix} \tag{22}$$

où la matrice de droite correspond aux coordonnées du syndrome dans la base des F_vE_u c'est à dire avec les notations précédentes $s_{1vu} = \sum_j h_{1jv} e_{ju}$

Afin d'obtenir un unique système linéaire on va écrire les vecteurs colonnes de la matrice des (e_{ju}) comme un seul vecteur colonne. Pour cela, dans la matrice de gauche, à la place de chaque coefficient h_{1jv} on écrit une matrice diagonale de dimension $r \times r$ dont les coefficients valent tous h_{1jv} : l'équation (22) équivaut à

$$(22) \Leftrightarrow \begin{pmatrix} h_{111} & 0 & 0 & h_{121} & 0 & 0 & \dots & h_{1n1} & 0 & 0 \\ 0 & \ddots & 0 & 0 & \ddots & 0 & & 0 & \ddots & 0 \\ 0 & 0 & h_{111} & 0 & 0 & h_{121} & & 0 & 0 & h_{1n1} \\ h_{112} & 0 & 0 & h_{122} & 0 & 0 & & h_{1n2} & 0 & 0 \\ 0 & \ddots & 0 & 0 & \ddots & 0 & & 0 & \ddots & 0 \\ 0 & 0 & h_{112} & 0 & 0 & h_{122} & & 0 & 0 & h_{1n2} \\ \vdots & & & & & & & & & \\ h_{11d} & 0 & 0 & h_{12d} & 0 & 0 & & h_{1nd} & 0 & 0 \\ 0 & \ddots & 0 & 0 & \ddots & 0 & & 0 & \ddots & 0 \\ 0 & 0 & h_{11d} & 0 & 0 & h_{12d} & & 0 & 0 & h_{1nd} \end{pmatrix} \begin{pmatrix} e_{11} \\ e_{12} \\ \vdots \\ e_{1r} \\ e_{21} \\ e_{22} \\ \vdots \\ e_{2r} \\ \vdots \\ e_{n1} \\ e_{n2} \\ \vdots \\ e_{nr} \end{pmatrix} = \begin{pmatrix} s_{111} \\ s_{121} \\ \vdots \\ s_{1d1} \\ s_{112} \\ s_{122} \\ \vdots \\ s_{12d} \\ \vdots \\ s_{11r} \\ s_{12r} \\ \vdots \\ s_{1dr} \end{pmatrix}$$

On obtient ainsi un système linéaire dans $\mathbb{F}_q$ à nr inconnues (les e_{ju}) et à dr équations.

La ligne d'indice $u + vr$ correspond à la coordonnée de s_1 suivant F_vE_u.

Ce système a été obtenu pour une seule composante du syndrome. En procédant de la même façon pour chacune des $(n-k)$ composantes on obtient en tout $(n-k) \times dr$ équations linéaires ayant pour inconnues les (e_{ju}). Ceci donne le système suivant dans $\mathbb{F}_q$. La matrice de gauche est une matrice définie par blocs, la matrice A_{ijv} étant égale à la matrice diagonale dont les coefficients valent h_{ijv} :

$$\begin{pmatrix} H_{111} & \dots & H_{1n1} \\ \vdots & & \vdots \\ H_{11d} & \dots & H_{1nd} \\ H_{211} & \dots & H_{2n1} \\ \vdots & & \vdots \\ H_{21d} & \dots & H_{2nd} \\ \vdots & & \vdots \\ H_{(n-k)11} & \dots & H_{(n-k)n1} \\ \vdots & & \vdots \\ H_{(n-k)1d} & \dots & H_{(n-k)nd} \end{pmatrix} \begin{pmatrix} e_{11} \\ e_{12} \\ \vdots \\ e_{1r} \\ e_{21} \\ e_{22} \\ \vdots \\ e_{2r} \\ \vdots \\ e_{n1} \\ e_{n2} \\ \vdots \\ e_{nr} \end{pmatrix} = \begin{pmatrix} s_{111} \\ \vdots \\ s_{1dr} \\ s_{211} \\ \vdots \\ s_{2dr} \\ \vdots \\ s_{(n-k)11} \\ \vdots \\ s_{(n-k)dr} \end{pmatrix} \quad \text{avec } H_{ijv} = h_{ijv} \times \mathrm{Id}_r$$

Notons A_H^r la matrice par blocs ainsi définie. Plus formellement nous avons :

Définition 159. *Description de A_H^r :*

La matrice A_H^r est une matrice de dimensions $(n-k)rd \times nr$ à coefficients dans $\mathbb{F}_q$. Les coefficients notés $a_{\tilde{i},\tilde{j}}$, $1 \leqslant \tilde{i} \leqslant (n-k)rd, 1 \leqslant \tilde{j} \leqslant nr$, sont donnés à partir de ceux de H de la manière suivante : pour $1 \leqslant i \leqslant (n-k)$, pour $1 \leqslant j \leqslant n$, pour $1 \leqslant l \leqslant d$, pour $1 \leqslant u \leqslant r$:

$$a_{u+(v-1)r+(i-1)rd, j+(u-1)r} = h_{ijv}$$

Les autres coefficients de A_H^r valent 0.

On note A_H une matrice inversible de dimension $nr \times nr$ extraite de A_H^r, et $D_H = A_H^{-1}$ une matrice de décodage de H.

Notons que la matrice A_H^r ne dépend pas de l'erreur reçue ni même de son support, uniquement de son poids rang. Ainsi, si on connaît une borne du poids rang des erreurs reçues, les matrices A_H^r, A_H et surtout D_H peuvent être générées a priori et utilisées directement dans le décodage ce qui permet de baisser significativement sa complexité.

Définition 160. *Définition de la matrice A_H^r :*

Soit $H = (h_{ij})$ une matrice de parité $(n-k) \times n$ de faible poids d sur $\mathbb{F}_{q^m}$ telle que les h_{ij} appartiennent à $F = <F_1, ..., F_d>$ de sorte que pour $1 \leqslant i \leqslant n-k$; $1 \leqslant j \leqslant n$ on ait l'égalité $h_{ij} = \sum_{k=1}^{d} h_{ijk} F_k$ avec $h_{ijk} \in \mathbb{F}_q$.

Supposons que nous ayons une erreur e de faible rang, notons r ce rang. Soit E le support de e tel que défini précédemment et $\{E_1, ..., E_r\}$ une base de E.

Nous allons construire une matrice A_H^r dépendant de F et r mais pas directement de E. En fait, la construction suivante peut être vue comme un « dépliage » de la matrice H dans le petit corps $\mathbb{F}_q$ dans une base « symbolique » de l'espace produit $\langle E.F \rangle$ tel qu'introduit au paragraphe précédent. La matrice en question, de dimension $(n-k)rd \times nr$, est une matrice $A_H^r = (a_{ij})$ dont les coefficients sont définis de la manière suivante : on commence par mettre tous les a_{ij} égaux à 0 puis on écrit :

$$a_{wr+v+1+urd, j+vn} = h_{ijw}$$

$$\text{pour } 0 \leqslant u \leqslant n-k-1, 0 \leqslant v \leqslant r-1, 1 \leqslant j \leqslant n \text{ et } 0 \leqslant w \leqslant d-1$$

La matrice A_H^r correspond à une réécriture formelle du système $H.\mathbf{e}^T = \mathbf{s}$ où $e = (e_1, ..., e_n)$ avec $e_i \in E = <E_1, ..., E_r>$ et $e_i = \sum_{j=1}^{r} e_{ij} E_j$. Remarquons que la matrice A_H^r ne dépend pas du sous-espace F dans lequel elle est écrit symboliquement.

Tous les s_i peuvent être écrits dans la base formelle $\{F_1E_1, F_1E_2..., F_1E_r, F_2E_1, ..., F_dE_r\}$ sur $\mathbb{F}_q$. La matrice A_H^r vérifie alors :

$$A_H^r.(e_{11}, e_{21}, ..., e_{n1}, e_{12}, e_{22}, ..., e_{n2}, ..., e_{r1}, ..., e_{rn})^T = (s_1, ..., s_{n-k})^T$$

où chaque s_i est écrit dans la base formelle de l'espace produit $\{F_1E_1, F_1E_2..., F_1E_r, F_2E_1, ..., F_dE_r\}$ avec les coordonnées dans le même ordre , en conséquence $(s_1, ..., s_{n-k})$ est considéré comme un vecteur de longueur $(n-k)rd$ à coefficients dans $\mathbb{F}_q$.

Par exemple, la première ligne de A_H^r représente l'impact du vecteur erreur e sur $(h_{11}, ..., h_{1n})$, la première ligne de H, sur l'élément F_1E_1 de la base symbolique. Puisque les e_{ij} sont ordonnées selon l'ordre précédemment donné, la première ligne de A_H^r joue seulement sur la projection de h_{1j} sur l'élément F_1 de la base, en conséquence les h_{1j1} sont dans $\mathbb{F}_q$. A présent, puisque l'on considère pour la première ligne l'impact sur F_1E_1, les premières n coordonnées a_{1j} de la première ligne de A_H^r sont $a_{1j} = h_{1j1}$ pour $1 \leqslant j \leqslant n$ et les coordonnées restantes sont $a_{1j} = 0$ pour $(n+1) \leqslant j \leqslant nr$ puisqu'elles font intervenir $E_2, E_3, ...$

Remarquons que si l'on prend des valeurs aux hasard des coordonnées pour un faible rang H, il est facile de trouver des matrices A_H^r de rang maximal nr.

Définition 161. *Dans la suite nous notons A_H une matrice inversible $nr \times nr$ extraite de A_H^r et nous notons $D_H = A_H^{-1}$ une matrice de décodage de H.*

Remarque 162. La matrice D_H permet de retrouver directement les nr valeurs e_{ij} à partir de nr positions des s_i écrits dans la base de l'espace produit par une simple multiplication.

4 Algorithme de décodage pour les codes LRPC

4.1 Idée générale

L'idée générale de l'algorithme est d'utiliser le fait que le poids de la matrice de parité est faible ; l'espace $<s_1, ..., s_{n-k}>$ engendré par les coordonnées du syndrome permet de retrouver tout l'espace produit $P = \langle E.F \rangle$ du support de l'erreur et la petite base connue de H. Le fait de connaître *la totalité* de l'espace P et de l'espace F permet de retrouver l'espace E. Par suite, connaissant le support E de l'erreur e, il est très simple de retrouver la valeur exacte de chaque coordonnée en résolvant un système linéaire. Cette approche ressemble beaucoup à la méthode de décodage des codes BCH par exemple, où l'on retrouve le polynôme localisateur d'erreur, qui donne le support de l'erreur, puis la valeur des coordonnées de l'erreur.

4.2 Présentation de l'algorithme de décodage

Considérons un code $\mathcal{C}[n,k]_r$ sur $\mathbb{F}_{q^m}$ de faible rang d, notons G une matrice génératrice $k \times n$ et H et une matrice de parité $(n-k) \times n$, telles que toutes les coordonnées h_{ij} de H appartiennent à un espace F de rang d dont une base est $\{F_1, ..., F_d\}$, et supposons que comme dans le paragraphe précédent H est choisie de sorte qu'il existe une matrice de décodage inversible associée D_H.

Supposons que le mot reçu est $y = xG + e$ avec x et e dans $(\mathbb{F}_{q^m})^n$ et où $e = (e_1, ..., e_n)$ est le vecteur erreur de rang r dont le support est $E = \{E_1, ..., E_r\}$ ce qui veut dire que chaque composante e_i peut être écrite $e_i = \sum_{j=1}^{r} e_{ij} E_j$ où les coefficients e_{ij} sont dans $\mathbb{F}_q$.

Nous présentons ci-après un algorithme général de décodage ; nous étudierons ensuite la validité de cet algorithme, la probabilité d'échec et la complexité. A la fin du paragraphe nous donnons des exemples de paramètres pour lesquels l'algorithme marche.

Algorithme 3

1. **Génération de l'espace syndrome :**
   Calculer le vecteur syndrome $H.y^t = s = (s_1, ..., s_{n-k})$ et l'espace syndrome $S = <s_1, ..., s_{n-k}>$
2. **Détermination du support de l'erreur :**
   Définir $S_i = F_i^{-1} S$, le sous-espace où tous les générateurs de S sont multipliés par F_i^{-1}. Générer le support de l'erreur $E = S_1 \cap S_2 \cap ... \cap S_d$. Pour cela déterminer $S_1 \cap S_2$ et calculer sa dimension. Si cette dimension est égale à celle de E alors $E = S_1 \cap S_2$, sinon calculer $(S_1 \cap S_2) \cap S_3$ et ainsi de suite.
   Générer ensuite une base $\{E_1, ..., E_r\}$ de E.
3. **Retrouver le vecteur erreur e :**
   Ecrire $e_i, 1 \leqslant i \leqslant n$, dans le support de l'erreur c'est à dire $e_i = \sum_{j=1}^{r} e_{ij} E_j$, résoudre le système $H.e^t = s$ où les équations $H.e^t$ et les coordonnées du syndrome s_i sont écrites comme des éléments de l'espace produit $P = \langle E.F \rangle$ dans la base $\{F_1E_1, ..., F_1E_r, ..., F_dE_1, ..., F_dE_r\}$. Le système a nr inconnues (les e_{ij}) et $(n-k)\,rd$ équations à partir du syndrome.
4. **Retrouver le message x :**
   Retrouver x à partir de $xG = y - e$

Remarque 163. Pour l'étape 2 on montre qu'en fait le calcul de $S_1 \cap S_2 \cap ... \cap S_d$ est rarement nécessaire. Il y a une forte probabilité pour avoir $E = S_1 \cap S_2$ ce qui diminue significativement la complexité de l'algorithme. Le lecteur est invité à consulter [GRSZ] pour plus de détails.

Remarque 164. Pour déterminer l'intersection des espaces vectoriels on peut utiliser l'algorithme du paragraphe 3.6 du chapitre III.

4.3 Preuve de la validité de l'algorithme

Nous prouvons la validité de l'algorithme dans le cas idéal où $\dim(\langle E.F\rangle) = rd$, $\dim(S) = rd$ et $\dim(S_1 \cap S_2 \cap ... \cap S_d) = r$, nous verrons dans le prochain sous-paragraphe que c'est en fait le cas général.

1. **Génération de l'espace syndrome :**

   Cette étape est évidente.

2. **Détermination du support de l'erreur :**

   Nous voulons prouver que $E \subset S_1 \cap S_2 \cap ... \cap S_d$. Nous avons défini $S_i = F_i^{-1}S = \{F_i^{-1}x, x \in S\}$, par suite puisque par hypothèse S est exactement l'espace produit $\langle E.F\rangle = \{a.b | a \in E, b \in F\}$, nous avons $F_i.E_j \in S, \forall 1 \leqslant j \leqslant r$, donc $E_j \in S_i$ et par conséquent $E \subset S_i$, donc E est inclus dans $S_1 \cap S_2 \cap ... \cap S_d$; par hypothèse $\dim(S_1 \cap S_2 \cap ... \cap S_d) = \dim(E)$ et finalement par égalité de dimensions,

   $$E = S_1 \cap S_2 \cap ... \cap S_d$$

3. **Retrouver le vecteur erreur e :**

   Une fois que le support E de l'erreur est connu, on peut écrire $x = \sum_{1 \leqslant i \leqslant n, 1 \leqslant j \leqslant r} e_{ij} E_j$ pour $e_{ij} \in \mathbb{F}_q$, puis résoudre le système linéaire $H.x^t = s$ où les nr inconnues sont les e_{ij}. Ce système a donc nr inconnues dans $\mathbb{F}_q$ et $(n-k)rd$ équations dans $\mathbb{F}_q$ provenant des $(n-k)$ équations du syndrome dans $\mathbb{F}_{q^m}$. Le paramètre r est choisi tel que $r \geqslant \frac{(n-k)m}{n}$. Remarquons de plus que l'on peut considérer l'espace produit $\langle E.F\rangle$ pour un F défini formellement de sorte que les équations du système sont uniquement liées à la matrice H. Alors H peut être choisie de sorte qu'une matrice de décodage D_H existe et permette de résoudre le système linéaire par une simple multiplication par D_H de l'ensemble des s_i écrits dans la base de l'espace produit.

4. **Retrouver le message x :**

   Cette étape est évidente.

4.4 Probabilité d'échec

Nous étudions à présent les différents possibilités d'échec, il y a trois cas à considérer : le cas où $\dim(\langle E.F\rangle) = rd$ correspond à la proposition 147 du paragraphe 2, le cas $E = S_1 \cap S_2 \cap ... \cap S_d$ correspond à la proposition 150 du même paragraphe. Dans les deux cas la probabilité peut être rendu exponentiellement petite selon les paramètres, surtout si l'on prend en compte qu'en pratique les bornes supérieures données sont très élevées comparées aux résultats expérimentaux.

Le dernier cas est $\dim(S) = rd$. Nous avons la proposition suivante :

Proposition 165. *La probabilité que les $(n-k)$ syndromes n'engendrent pas l'espace produit $P = \langle E.F\rangle$ est inférieure à $q^{-(1+(n-k)-rd)}$.*

Démonstration. *Par construction tous les s_i appartiennent à l'espace produit P et puisque l'erreur est prise aléatoirement les s_i peuvent être vus comme des éléments aléatoires de P, par suite si l'on considère un ensemble de $(n-k)$ vecteurs aléatoires d'un espace de dimension rd (avec $n-k \geqslant rd$) la probabilité que cet ensemble n'engendre pas l'espace entier est environ de $q^{-(1+(n-k)-rd)}$ c'est à dire la probabilité qu'une matrice de dimension $rd \times (n-k) = rd \times (rd+(n-k)-rd)$ aléatoire sur $\mathbb{F}_q$ ne soit pas inversible.* □

Par conséquent, ce qui précède montre qu'en fonction des paramètres la probabilités d'échec de l'algorithme précédent peut être rendue arbitrairement petite et que la probabilité principale que nous devons considérer est la probabilité donnée dans la proposition 165 qui n'est pas une borne supérieure mais ce qui se passe dans la pratique.

4.5 Complexité du décodage

Détaillons la complexité des différentes étapes de l'algorithme :

1. **Génération de l'espace syndrome :**

   La complexité de cette étape est négligeable par rapport aux autres.

2. **Détermination du support de l'erreur :**

   Le coût de cette étape est le coût de l'intersection d'espaces vectoriels ce qui coûte $4r^2d^2m$ opérations dans le corps de base. Il s'agit d'une estimation dans le pire des cas : comme il y a une forte probabilité d'avoir $E = S_1 \cap S_2$ on peut en fait se ramener au cas $d = 2$ ce qui donne $16r^2m$ opérations.

   Remarquons que cette opération peut se faire en utilisant les q-polynômes en mettant en oeuvre les outils que nous avons présentés dans les deux chapitres de la deuxième partie, voir notamment le paragraphe 3.6 du chapitre 3.

3. **Retrouver le vecteur erreur e :**

   Le coût de cette étape est le coût de la résolution du système $H.e^t = s$ où le support E de l'erreur est connu ; si l'on procède naïvement il y a nr inconnues et la méthode du pivot de Gauss donne une complexité en $O(n^3r^3)$ pour l'inversion matricielle, mais comme nous l'avons remarqué dans le paragraphe précédent nous pouvons ici utiliser la matrice D_H (qui est générée une fois pour toutes et stockée) et retrouver les e_{ij} en multipliant par D_H les nr positions de $s_1, ..., s_{n-k}$ (écrits comme des éléments de $\mathbb{F}_q$ dans la base de $\langle E.F \rangle$ ordonnée comme nous l'avons expliqué précédemment). Nous n'avons plus alors besoin de procéder à une inversion mais simplement une multiplication matricielle dont le coût est en $O(n^2r^2)$.

4. **Retrouver le message x :**

   La complexité de cette étape est négligeable par rapport aux autres.

Nous obtenons finalement une complexité, en nombre d'opérations dans le corps de base, de l'ordre de

$$O(4r^2d^2m + n^2r^2)$$

Il y a une forte probabilité pour que l'on puisse se ramener au cas $d = 2$ ce qui donne alors un nombre d'opérations de

$$O(r^2(16m + r^2))$$

4.6 Bilan

Si nous résumons tous les résultats de ce paragraphe nous obtenons :

Théorème 166. *Soit H une matrice de parité de dimension $(n-k) \times n$ d'un code LRPC avec un rang faible $d \geqslant 2$ sur $\mathbb{F}_{q^m}$, alors l'algorithme que nous avons donné permet de décoder un vecteur erreur e de rang faible r tel que $rd \leqslant n-k$ avec une probabilité de*

$$q^{-(n-k+1-rd)}$$

et une complexité en nombre d'opérations dans $\mathbb{F}_q$ de l'ordre de

$$O(r^2(4d^2m + n^2))$$

Cette complexité est une estimation dans le pire des cas et il y a une forte probabilité pour que la complexité soit en fait de l'ordre de $O(r^2(16m + r^2))$.

Démonstration. *Ce théorème vient directement des résultats précédents.* □

Remarque 167. En termes de pure capacité de décodage, les codes LRPC sont moins intéressants que les codes de Gabidulin puisque la capacité de décodage atteint difficilement $\frac{n-k}{2}$ et qu'en plus l'algorithme est probabiliste, cependant ils sont parfaitement adaptés à une utilisation cryptographique comme nous allons le voir dans ce qui suit.

4.7 Exemple

Etudions un exemple avec des petites valeurs des paramètres pour expliciter la construction de la matrice A_H^r et le fonctionnement de l'algorithme de décodage.

On choisit un code sur $\mathbb{F}_{2^{11}} \cong \mathbb{F}_2[\alpha] = \{0, 1, \alpha, ..., \alpha^9\} \cong \mathbb{F}_2[X]/(P)$ où on choisit comme polynôme primitif le polynôme de Conway $P = C_{2,11} = X^{11} + X^2 + 1$.

On se place sur un code de longueur $n = 6$ et de dimension $k = 3$. On suppose que l'erreur appartient à un espace de dimension 1 ($r = 1$) engendré par $E_1 = \alpha$. On suppose que les coefficients de la matrice H appartiennent à un espace de dimension 2 ($d = 2$) engendré par $F_1 = 1$ et $F_2 = \alpha^2$.

Supposons que la matrice H est donnée par

$$H = \begin{pmatrix} 1 & \alpha^2 & 1 & 1+\alpha^2 & 0 & 0 \\ 0 & \alpha^2 & 1 & \alpha^2 & 1 & 1+\alpha^2 \\ 1 & 0 & \alpha^2 & 0 & 0 & 1+\alpha^2 \end{pmatrix}$$

et que l'on reçoive un mot $y = x + e$ où x est un mot du code et e est un vecteur erreur de rang 1 égal à

$$e = (\ 0 \quad \alpha \quad 0 \quad 0 \quad 0 \quad \alpha\ )$$

1. Détermination de l'espace syndrome :

   Le syndrome vaut

   $$s = Hy^t = He^t = \begin{pmatrix} \alpha^3 \\ \alpha \\ \alpha + \alpha^3 \end{pmatrix}$$

   Comme α et α^3 sont linéairement indépendants sur $\mathbb{F}_2$, l'espace S engendré par les coordonnées du syndrome vaut $S = <\alpha, \alpha^3>$.

2. Détermination du support de l'erreur :

   L'espace S_1 vaut $S_1 = <1^{-1}\alpha, 1^{-1}\alpha^3> = <\alpha, \alpha^3>$

   L'espace S_2 vaut $S_2 = <(\alpha^2)^{-1}\alpha, (\alpha^2)^{-1}\alpha^3> = <\alpha^{-1}, \alpha>$

   L'élément α^{-1} n'appartenant pas[12] à S_1 il est clair que $S_1 \cap S_2 = <\alpha> = E$.

12. Sinon nous aurions une relation du type $\alpha^{-1} = \lambda\alpha + \mu\alpha^3 \Leftrightarrow \mu\alpha^4 + \lambda\alpha^2 - 1 = 0$ ce qui est absurde car la famille $1, \alpha^2, \alpha^4$ est libre dans $\mathbb{F}_{2^{11}}$.

3. Détermination de l'erreur en écrivant les coordonnées dans le corps de base :

   Décomposons les coordonnées du syndrome s dans la base $\{F_1E_1, F_2E_1\} = \{\alpha, \alpha^3\}$. L'élément obtenu est noté $s_{\mathbb{F}_2}$:

$$s_{\mathbb{F}_2} = \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 1 \end{pmatrix}$$

   Décomposons chaque coefficient de H en vecteur colonne dans la base $\{F_1, F_2\} = \{1, \alpha^2\}$ afin d'obtenir la matrice A_H^r:

$$A_H^r = \begin{pmatrix} 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}$$

   On a choisi la matrice H de façon à ce que A_H^r soit carrée inversible, ainsi $A_H^r = A_H$ et

$$D_H = A_H^{-1} = \begin{pmatrix} 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 \end{pmatrix}$$

   Pour retrouver le vecteur erreur on calcule

$$D_H \times s_{\mathbb{F}_2} = \begin{pmatrix} 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} = e^t$$

   On retrouve bien les coordonnées du vecteur erreur écrites dans la base $\{E_1\}$.

5 Application à la cryptographie : le cryptosystème LRPC

Dans ce qui suit nous proposons un nouveau cryptosystème dans l'esprit du cryptosystème NTRU ou plus récemment du cryptosystème MDPC.

5.1 Le cryptosystème LRPC

Pour ce nouveau cryptosystème nous utilisons le schéma cryptographique de McEliece. Le schéma de Niederreiter pourrait aussi être utilisé mais nous connaissons moins bien la sécurité sémantique dans cette configuration.

Soit $\mathcal{C}$ un code LRPC ou DC-LRPC dont on connaît une matrice de parité H. Nous supposons que $k = \frac{n}{2}$ de sorte que H est de dimension $\frac{n}{2} \times n$. Nous supposons aussi que H est de rang d et que le code corrige des erreurs de rang r. Nous dissimulons la matrice H à l'aide d'une matrice aléatoire inversible R, dans le cas des codes DC-LRPC cette matrice est une matrice circulante aléatoire de dimension $\frac{n}{2} \times \frac{n}{2}$. Nous présentons le cryptosystème LRPC :

Cryptosystème LRPC

1. **Création de la clé**

   Choisir au hasard $\mathcal{C}$ un code LRPC sur $\mathbb{F}_{q^m}$ de faible rang d ; le code est déterminé à partir de la matrice de parité H dont les coefficients sont des éléments du support $F = <F_1, ..., F_d>$. On note G une matrice génératrice de $\mathcal{C}$ et D_H une matrice de décodage corrigeant les erreurs de poids $\leqslant r$, comme introduite dans le paragraphe précédent.

   Déterminer une matrice R à coefficients dans $\mathbb{F}_{q^m}$ inversible de dimension $(n-k)\times(n-k)$.

   - **Clé secrète :** La matrice de rang faible H, la matrice masquante R.
   - **Clé publique :** La matrice $G' = RG$.

2. **Chiffrement**

   Convertir le vecteur information v en un mot de code x, choisir une erreur aléatoire e de rang r sur $\mathbb{F}_{q^m}$. Le chiffrement de v est $c = xG' + e$.

3. **Déchiffrement**

   Calculer le syndrome $s = H.c^t$ et retrouver le vecteur erreur e, puis xR, puis x.

Remarque 168. Dans le cas des codes DC-LRPC, nous avons vu au paragraphe 3.2.2 que la matrice G' pouvait être écrite $G' = \left(A^{-1}B^T \middle| I_{\frac{n}{2}}\right)$ où A et B sont deux matrices circulantes de faible rang d pour le même espace F. Dans ce cas la seule connaissance d'une ligne de $A^{-1}B^T$ permet de reconstruire G' par cyclicité. Ainsi, pour générer la matrice, on n'a pas besoin de $nk\log_2(q^m)$ bits pour stocker les coefficients mais seulement[13] de $\frac{n}{2}\log_2(q^m)$ bits. Ce point est important car c'est en procédant ainsi que l'on arrive à faire significativement baisser la taille de la clé publique ce qui est, on le rappelle, le principal inconvénient des cryptosystèmes basés sur les codes.

5.2 Paramètres du cryptosystème LRPC

1. Taille en bits de la clé publique : $(n-k)(n-k)m\log(q)$ pour les LRPC ; $\frac{nm}{2}\log(q)$ pour les DC-LRPC
2. Taille en bits de la clé secrète : Un vecteur aléatoire peut être utilisé pour retrouver les différents paramètres.
3. Taille du message : $(n-k)m\log(q)$ pour les LRPC ; $\frac{nm}{2}\log(q)$ pour les DC-LRPC
4. Taux de chiffrement : $\frac{r(m+n)}{(n-k)m}$ pour les LRPC ; $\frac{2r(m+n)}{nm}$ pour les DC-LRPC
5. Complexité du chiffrement : $(n-k)(n-k)mr$ opérations dans $\mathbb{F}_q$ pour les LRPC ; $\frac{n^2mr}{4}$ opérations dans $\mathbb{F}_q$ pour les les DC-LRPC
6. Complexité du déchiffrement : $r^2(n^2+4d^2m)$ opérations dans $\mathbb{F}_q$.
7. Complexité de la meilleure attaque usuelle : $O\left((n-k)^3m^3q^{(r-1)\left\lfloor\frac{(k+1)m}{n}\right\rfloor}\right)$ pour une attaque du support ; $q^{r\left\lceil\frac{(k+1)-(n+1)}{r}\right\rceil}$ pour la borne inférieure d'une attaque algébrique (base de Groebner), et les résultats heuristiques de [LP].

13. On rappelle que $k = \frac{n}{2}$.

Remarque 169. Dans le cas des DC-LRPC, puisque la matrice est doublement circulante, la complexité du chiffrement peut être optimisée.

6 Sécurité du cryptosystème LRPC

6.1 Sécurité sémantique

La sécurité de notre système est basée sur le problème suivant :

Le problème LRPC : Etant donné une matrice G' correspondant à la clé publique il est difficile de retrouver un vecteur de faible poids rang d dans le code dual.

Ce problème est une adaptation des problèmes NTRU et MDPC mais dans le cas de la métrique rang. Les matrices utilisées ne sont clairement pas aléatoires, mais les attaques sur la structure sont infructueuses sur NTRU depuis l'introduction de ce cryptosystème. Il en est de même pour le cryptosystème MDPC.

En termes de sécurité sémantique, l'approche développée pour le cryptosystème MDPC dans [MTSB] sur l'indistinguabilité par rapport aux codes aléatoires peut être adaptée au contexte de la métrique rang, de plus la conversion CCA-2 de Kobara et Imai [KI] peut aussi être adaptée à la métrique rang mais cette discussion irait beaucoup trop loin par rapport au présent sujet.

Concernant l'échec du déchiffrement, il est aussi possible d'utiliser l'approche de Fujisaki et Okamoto [FO] qui permet qu'aucune information ne soit donnée en cas d'échec du déchiffrement (la même approche ayant été proposée pour NTRU et MDPC).

6.2 Sécurité pratique

Passons en revue les différentes attaques :

6.2.1 Attaque du message

Attaquer le message : dans ce cas l'attaquant essaie de retrouver directement le message $\mathbf{x}$ en essayant de retrouver $\mathbf{e}$ de rang r avec des attaques classiques sur le code.

6.2.2 Attaque par clé contrefaite (Spurious key)

Tout comme dans le cas NTRU (voir [HPS]) cette attaque consiste à trouver des mots de poids rang faible mais légèrement plus grand que d et de les utiliser pour décoder. Bien que théoriquement possible cette attaque ne marche pas dans la pratique puisque le fait que H contienne des mots de poids faible implique en effet que beaucoup de mots de poids $2d$ existent. Nous ne rentrerons pas dans les détails ici, signalons juste que comme pour les codes MDPC [MTSB], lorsque le poids augmente la complexité des attaques augmente plus vite que le nombre de vecteurs de faible poids, ainsi cette attaque ne marche pas en pratique, pas plus que pour NTRU ou MDPC.

6.2.3 Attaque sur la clé secrète par repliement du code

Nous rappelons (voir chapitre 5, paragraphes 3.1.2 et 3.1.3, notamment la remarque 136) l'existence d'un algorithme générique de décodage en métrique rang de complexité exponentielle permettant de retrouver le support d'un vecteur de $\mathbb{F}_{q^m}$ de poids rang d si on connaît a éléments de ce support (vus comme des vecteurs de $\mathbb{F}_q$) avec une probabilité de

$$q^{-(d-a)\left\lceil \frac{km}{n} \right\rceil} \tag{23}$$

Nous présentons une attaque détaillée par Adrien Hauteville dans [Hau], qui utilise la structure doublement circulante des codes DC-LRPC. Nous reprenons les notations du paragraphe 3.2.

6.2.4 Une première remarque

Rappelons que les coefficients de la matrice parité H sont dans un support F de poids rang d. Si on fait des combinaisons linéaires des lignes de H, les coefficients de la matrice obtenue après combinaison linéaire sont toujours dans F.

Si on additionne toutes les lignes de H, grâce à la structure doublement circulante (chaque vecteur colonne de la matrice contient les mêmes coefficients simplement dans un ordre différent) on obtient une ligne de la forme $(\beta, ..., \beta, \gamma, ..., \gamma)$ où β et γ apparaissent $\frac{n}{2}$ fois chacun. Ainsi on connaît deux éléments β et γ de F.

Dans le cas des DC-LRPC, à partir de la clé publique $G' = \left(A^{-1}B^T \middle| I_{\frac{n}{2}}\right)$ on peut retrouver une matrice de parité sous forme systématique $H' = \left(I_{\frac{n}{2}} | A^{-1}B\right)$. En additionnant les lignes de H' on obtient $(1, ..., 1, \alpha, ..., \alpha)$. On obtient donc deux éléments 1 et α d'un multiple $\beta^{-1}F$ de F.

De la façon présentée dans le paragraphe 3.1.3 du chapitre 5 on cherche un espace vectoriel de dimension d' contenant $\beta^{-1}F$. on a nd' inconnues et mk équations donc $d' = \left\lfloor \frac{km}{n} \right\rfloor = \left\lfloor \frac{m}{2} \right\rfloor$.

D'après l'équation (23) un algorithme probabiliste permet de retrouver $\beta^{-1}F$ puis F avec une probabilité de

$$\frac{\begin{pmatrix} d' \\ d-2 \end{pmatrix}_q}{\begin{pmatrix} m \\ d-2 \end{pmatrix}_q} = q^{-(d-2)(m-d')} = q^{-(d-2)\left\lceil \frac{km}{n} \right\rceil}$$

6.2.5 Code replié

Nous introduisons la notion de *code replié*. Nous nous référons au paragraphe 3.2 du présent chapitre pour les résultats théoriques et les notations :

Définition 170. *Soit $\mathcal{C}\left[n, k = \frac{n}{2}\right]$ un code rang doublement circulant sur $\mathbb{F}_{q^m}$ dont $G = (G_1|G_2)$ est une matrice génératrice doublement circulante et H une matrice de parité. On suppose comme précédemment que les coefficients de H sont dans un espace F de dimension r.*

Soit $D \in \mathbb{F}_q[X]$ un diviseur de $X^k - 1$. Le code replié de $\mathcal{C}$ par D *noté $\mathcal{C}_{\text{rep},D}$ ou $\mathcal{C}_{\text{rep}}$ s'il n'y a pas d'ambiguïté est le code engendré par $DG = (DG_1|DG_2)$.*

On a vu au paragraphe 3.2 que la dimension de ce code est majorée par $k - \deg(D)$.

Si la matrice G_1 est inversible alors la matrice $G' = \left(I_k | G_1^{-1}G_2\right)$ est une matrice génératrice de $\mathcal{C}$ sous forme systématique, dans ce cas $DG' = \left(D | DG_1^{-1}G_2\right)$ est une matrice génératrice de $\mathcal{C}_{\text{rep}}$ et la dimension de $\mathcal{C}_{\text{rep}}$ vaut alors exactement $k - \deg(D)$.

6.2.6 Amélioration de l'attaque grâce à un repliement du code

Comme $D \in \mathbb{F}_q[X]$, le fait de replier un code revient à faire des combinaisons linéaires des lignes de la matrice génératrice. Ceci veut dire que le support du code replié est inclus dans le support du code non replié. Comme ici le support F du code dual $\mathcal{C}^\perp$ est d'une petite dimension d, en repliant ce code on va pouvoir garder le même support. Ainsi, au lieu de chercher F à partir du code de départ on va le chercher à partir d'un code replié de plus petite dimension, la recherche sera donc plus rapide.

Pour replier un code on doit trouver un diviseur de $X^k - 1$. Il en existe toujours au moins deux, le polynôme $X - 1$ et le polynôme $\sum_{i=0}^{k-1} X^i$. Choisir $D = X - 1$ ne donne pas un gain important et choisir $D = \sum_{i=0}^{k-1} X^i$ donne un code replié de dimension 1 ce qui ne donne pas de renseignement supplémentaire par rapport à la première attaque du paragraphe 6.2.4.

Pour que l'attaque soit efficace il est donc a priori nécessaire qu'il existe des diviseurs de $\sum_{i=0}^{k-1} X^i$ de degré assez élevé ce qui n'est pas toujours possible suivant les valeurs de k.

Les calculs sur les gains de complexité donnant les résultats de l'exemple suivant sont détaillés dans [Hau]. L'estimation de la complexité donnée par Adrien Hauteville, en utilisant l'algorithme d'Ourivski-Johanson et en prenant en compte le fait que la première attaque permet de diminuer de 2 la taille du support, est égale à

$$O\left(m^3\left(d-2+\left\lceil\frac{2(k+d-2)}{m-2}\right\rceil\right)^3 q^{(d-2)\left(k+\left\lceil\frac{2(k+d-2)}{m-2}\right\rceil\right)+2}\right)$$

Exemple 171. Prenons les exemples qui seront présentés dans le paragraphe suivant :

- Pour $n=74, k=37, m=41, q=2, d=4, r=4$ il n'est pas possible de replier le code car le polynôme $X^{37}-1$ se factorise seulement en $(X-1)\sum_{i=0}^{36} X^i$. Dans ce cas l'attaque n'est pas efficace.

- Pour $n=94, k=47, m=47, q=2, d=5, r=5$, le polynôme $X^{47}-1$ se factorise en $(X-1)PQ$ où P et Q sont de degré 23. En choisissant $D=(X-1)P$ ou $D=(X-1)Q$ on peut donc se ramener à un code replié de dimension $k'=23$ au lieu de 47. L'estimation de la complexité donne une sécurité de 2^{101} ce qui fait baisser significativement la sécurité du cryptosystème, cependant le cryptosystème reste sûr.

- Pour $n=68, k=34, m=23, q=2^4, d=4, r=4$, le polynôme se factorise en facteurs de degré 2. Il est possible de replier le code jusqu'à avoir un code de dimension $k'=4$ ce qui donne $\left\lceil\frac{2(k'+d-2)}{m-2}\right\rceil=1$ et fait significativement baisser la complexité. Dans ce cas l'évaluation de la complexité donne un nombre d'opérations en 2^{66} ce qui est en dessous de la sécurité de 2^{80}, le système ne peut plus être considéré comme sûr.

6.2.7 Protection contre cette attaque

Une attaque a été trouvée dans le cas des codes DC-LRPC, cependant pour que cette attaque soit efficace il faut que le polynôme X^k-1 admette un diviseur de petit degré (à part $X-1$)ce qui n'est pas toujours possible.

Dans [GRSZ2] de nouveaux paramètres des codes sont présentés pour lesquels les codes gardent une sécurité et une probabilité d'échec du déchiffrement quasiment identiques (meilleurs dans certains cas) mais pour lesquels l'attaque par repliement n'est plus valable. Ceci se fait fait au prix d'une légère augmentation de la taille de la clé mais celle-ci reste très raisonnable, quelques centaines de bits seulement au maximum.

Les nouveaux paramètres sont donnés dans le paragraphe suivant.

7 Exemples de paramètres et comparaison avec les MDPC

7.1 Exemples de paramètres

7.1.1 Premiers exemples

Nous donnons trois exemples de paramètres pour le cas du DC-LRPC : un exemple avec une sécurité de 2^{80} qui optimise la taille de la clé publique à 1500 bits[14] avec une probabilité d'échec de déchiffrement de 2^{-22}, un exemple avec une sécurité de 2^{128}, et enfin un exemple avec une probabilité d'échec de déchiffrement de 2^{-80}.

14. La taille de la clé est dans le cas des DC-LRPC, on le rappelle, égale à $k\log_2(q^m)$

Dans le tableau suivant Echec signifie probabilité d'échec du déchiffrement, Coût dcf. est le coût du déchiffrement en nombre d'opérations dans $\mathbb{F}_q$, et les deux dernières colonnes donnent le coût des meilleurs attaques connues : l'attaque du support et l'attaque algébrique [GRS].

Nous donnons des paramètres pour différents niveaux de sécurité mais aussi pour différentes probabilités d'échec de déchiffrement, en particulier on voit que l'on peut atteindre 2^{-80} facilement en doublant la taille de la clé. Remarquons que les paramètres sont très versatiles. Bien qu'aucune attaque spéciale ne soit connue pour des nombres non premiers nous préférons en général utiliser tout de même des nombres premiers. De plus nous avons besoin de la primalité de m pour prouver certains résultats théoriques.

n	k	m	q	d	r	Echec($\log_2$)	Clé publique(bits)	Coût dcf.($\log_2$)	Att.supp.($\log_2$)	Att.alg.($\log_2$)
74	37	41	2	4	4	-22	1517	17	84	82
94	47	47	2	5	5	-23	2209	19	128	145
68	34	23	2^4	4	4	-80	3128	17	153	100

7.1.2 Nouveaux paramètres résistants à l'attaque par repliement

Suite à l'attaque par repliement vue au chapitre précédent, de nouveaux paramètres ont été donnés dans [GRSZ2] pour empêcher cette attaque tout en gardant des tailles de clés quasiment identiques :

n	k	m	q	d	r	Echec($\log_2$)	Clé publique(bits)	Sécurité ($\log_2$, meilleure attaque)
82	41	41	2	5	4	-22	1681	80
106	53	53	2	6	5	-24	2809	128
74	37	23	2^4	4	4	-88	3404	110

7.2 Comparaison avec le cryptosystème MDPC

Le système que nous proposons présente trois avantages par rapport au cryptosystème MDPC :

- La taille de la clé peut être trois fois plus petite ;
- Le système est beaucoup plus rapide (au moins 100 fois plus, la complexité de MDPC est en $\lambda w^2 r$ pour w proche de 90 et r autour de 5000) ;
- On a plus de contrôle sur la probabilité d'échec de déchiffrement, en doublant notamment la taille des paramètres on peut la faire décroître jusqu'à 2^{-80} au lieu de 2^{-23}.

8 Implémentation et temps d'exécution

Nous avons implémenté le cryptosystème LRPC sur un ordinateur dont le processeur est cadnecé à 1.2 GHz (sur un processeur plus récent et plus puissant, les temps d'exécution seraient sans doute meilleurs). Pour les opérations sur les corps finis nous avons utilisé la librairie MPFQ [mpfq] qui permet de faire très vite les opérations sur les corps finis. Cependant la librairie MPFQ ne gère pas directement les extensions de corps intermédiaires (par exemple $\mathbb{F}_{16}$ vu comme une extension de $\mathbb{F}_4$ au lieu de $\mathbb{F}_2$) et les temps de calcul sur de tels corps ne sont donc pas optimisés. Aussi, nous présentons seulement les temps de calcul pour les exemples ou le corps « de base » est le corps premier $\mathbb{F}_2$:

Temps en millisecondes :

n	k	m	q	d	r	Encodage	Calcul du syndrome	Décodage	$x R^{-1}$	Total
82	41	41	2	5	4	0.1	0.2	0.4	0.1	0.8
106	53	53	2	6	5	0.2	0.5	0.4	0.2	1.1

Les temps d'exécution sont plus rapides que pour le cryptosystème MDPC et peuvent vraisemblablement être encore améliorés par une optimisation des programmes.

Remarque 172. Pour implémenter le code nous avons utilisé les résultats du chapitre I sur les matrices de parité.

9 Pistes de recherche

L'implémentation du cryptosystème est a priori encore nettement améliorable en ce qui concerne les temps d'exécution. Nous avons réalisé l'implémentation sans chercher à optimiser la vitesse d'exécution et nous pouvons encore gagner du temps notamment sur l'étape du décodage proprement dit. A cette étape les opérations se font dans le corps de base $\mathbb{F}_q$ et non plus dans l'extension $\mathbb{F}_{q^m}$ et devraient donc théoriquement être beaucoup plus rapides, alors qu'actuellement c'est l'étape la plus longue comme on le voit dans le tableau précédent. L'étape la plus coûteuse est normalement le calcul du syndrome qui correspond à une multiplication matrice/vecteur dans l'extension $\mathbb{F}_{q^m}$. Si ces estimations sont justes on peut donc arriver à un temps total d'exécution de l'ordre de la milliseconde, voire moins.

Une piste de recherche intéressante à explorer est l'implémentation de ce cryptosystème avec une librairie plus légère : la librairie MPFQ, si elle permet de faire très rapidement les opérations sur les corps finis, est relativement volumineuse (plusieurs milliers voire dizaines de milliers de lignes) et nécessite l'installation de plusieurs outils additionnels pour la génération des codes et la compilation. Nous avons effectué une ébauche d'implémentation avec la librairie *bitvector* qui comme son nom le suggère permet de manipuler efficacement des vecteurs de bits. Nous pouvons utiliser ces bitvectors pour représenter des éléments de $\mathbb{F}_{2^m}$ de la façon que nous avons vue au chapitre 2. L'avantage de cette librairie est qu'elle est très légère (quelques centaines de lignes) et ne nécessite pas l'installation de programmes tiers pour fonctionner ce qui permettrait notamment d'embarquer ce cryptosystème sur des composants disposant d'assez peu de mémoire comme des cartes à puce par exemple.

Les premiers tests effectués avec la librairie bitvector avec le corps $\mathbb{F}_{2^{41}}$ donnent un temps d'exécution de l'ordre de $O(0.01s)$. Ces résultats sont encourageants (les programmes ont pour l'instant été faits simplement pour les tests, sans souci d'optimisation), mais la vitesse d'exécution doit encore pouvoir être significativement améliorée jusqu'à être comparable à celle des programmes utilisant la librairie MPFQ.

Une attaque a été explicitée dans le cas des codes DC-LRPC : l'attaque par code replié. Cette attaque n'est a priori efficace que lorsque le polynôme $\sum_{i=0}^{k-1} X^i$ admet des diviseurs de degré élevé. Si cette attaque est amenée à évoluer, ou si d'autres attaques sont trouvées, il faut voir dans quelle mesure on peut s'en protéger en modifiant les paramètres du code.

Signalons d'autre part que suite à nos travaux sur le cryptosystème LRPC, un algorithme de signature utilisant les mêmes techniques a été mis au point par Gaborit, Ruatta, Schrek et Zémor dans [GRSZ].

10 Conclusion

Dans ce chapitre, tout comme dans l'article récent sur le cryptosystème MDPC [MTSB], nous avons généralisé l'approche du cryptosystème NTRU [HPS] dans le contexte de codage mais avec la métrique rang. Pour ce faire nous avons introduit un nouveau type de codes, les codes LRPC, pour lesquels nous proposons un algorithme de décodage efficace. Dans l'ensemble, tout comme c'est souvent le cas pour la métrique rang, les résultats obtenus sont bons comparés aux cryptosystèmes en distance de Hamming puisque les attaques connues augmentent en difficulté.

De plus, alors que les cryptosystèmes en métrique rang ont connu de nombreuses attaques basées sur le recouvrement de la structure des codes de Gabidulin, le cryptosystème que nous proposons est le premier cryptosystème en métrique rang aléatoire, faiblement structuré, et qui n'est pas basé sur les codes de Gabidulin.

Une attaque possible a été trouvée dans le cas des codes DC-LRPC, mais cette attaque n'est a priori efficace que lorsque $\sum_{i=0}^{k-1} X^i$ admet des diviseurs de degré élevé et n'invalide pas la sécurité des deux premiers cryptosystèmes présentés dans 7.1 même si elle baisse tout de même la sécurité d'un facteur 2^{27} dans un cas.

De nouveaux paramètres pour les codes sont proposés dans [GRSZ2] qui permettent de protéger les codes contre l'attaque par repliement tout en conservant une sécurité et une probabilité d'échec de décodage quasiment identiques. Cette protection supplémentaire nécessite une légère augmentation de la taille de la clé publique, très peu pénalisante puisque l'augmentation la plus importante sur les paramètres est une augmentation de 600 bits, faisant passer la taille de la clé de 2209 bits à 2809 bits.

Ce cryptosystème devra bien sûr être étudié de manière plus approfondie mais il semble ouvrir des opportunités intéressantes.

Appendice : Démonstrations des résultats du paragraphe 2

Lemme 173. *Soient A' et B deux sous-espaces de $\mathbb{F}_{q^m}$ de dimensions α' et β telles que $\dim\langle A'B\rangle = \alpha'\beta$. Soit $A = A' + \langle a\rangle$ où a est un élément aléatoire uniformément choisi de $\mathbb{F}_{q^m}$, alors*

$$\text{Prob}(\dim\langle AB\rangle < \alpha'\beta + \beta) \leqslant \frac{q^{\alpha'\beta+\beta}}{q^m}$$

Démonstration. *Nous avons $\dim\langle AB\rangle < \alpha'\beta + \beta$ si et seulement si $aB \cap \langle A'B\rangle \neq \{0\}$. Par suite,*

$$\text{Prob}(\langle A'B\rangle \cap aB \neq \{0\}) \leqslant \sum_{b\in B, b\neq 0} \text{Prob}(ab \in \langle A'B\rangle) \tag{24}$$

$$\leqslant (|B|-1)\frac{q^{\alpha'\beta}}{q^m} \tag{25}$$

puisque pour tout $a \neq 0$ fixé, ab est distribué uniformément dans $\mathbb{F}_{q^m}$. Comme $|B|-1 \leqslant |B| = q^\beta$ nous obtenons le résultat voulu. □

Proposition 174. *Soit B un sous-espace fixé, supposons que nous construisons un sous-espace aléatoire A en choisissant α vecteurs aléatoires de $\mathbb{F}_{q^m}$ indépendamment (indépendamment dans le sens probabiliste du terme) et soit A le sous-espace engendré par ces α vecteurs indépendants, alors*

$$\text{Prob}(\dim\langle AB\rangle = \alpha\beta) \geqslant 1 - \alpha\frac{q^{\alpha\beta}}{q^m}$$

Démonstration. *On applique le lemme précédent α fois en commençant avec un sous-espace aléatoire $A' \subset A$ de dimension 1, et on ajoute un nouvel élément à A' jusqu'à ce qu'on obtienne A.* □

Soit B un sous-espace fixé de $\mathbb{F}_{q^m}$ contenant 1 et soit $\langle B^2\rangle$ le sous-espace engendré par tous les éléments de B. Soit $\beta_2 = \dim\langle B^2\rangle$. Soit A un sous-espace aléatoire de $\mathbb{F}_{q^m}$ de dimension α. D'après la proposition précédente nous avons

$$\text{Prob}(\dim\langle AB^2\rangle = \alpha\beta_2) \geqslant 1 - \alpha\frac{q^{\alpha\beta_2}}{q^m}$$

Remarque 175. Nous avons $\beta_2 \leqslant \frac{\beta(\beta+1)}{2}$

Lemme 176. *Supposons que $\dim\langle AB^2\rangle = \alpha\beta_2$. Soit $e \in \langle AB\rangle$ avec $e \notin A$. Supposons $eB \subset \langle AB\rangle$, alors il existe $x \in B, x \notin \mathbb{F}_q$, tel que $xB \subset B$.*

Démonstration. *Soit (a_i) une base de A. On a*

$$e = \sum_i \lambda_i a_i b_i$$

avec $\lambda_i \in \mathbb{F}_q$ pour tout i et $b_j \notin \mathbb{F}_q$ et $\lambda_j \neq 0$ pour un certain j, autrement $e \in A$ contrairement à ce que nous avons supposé. Soit b un élément de B. Par hypothèse $eb \in \langle AB\rangle$ ce qui signifie

$$\sum_i \lambda_i a_i b_i b = \sum_i \mu_i a_i b_i'$$

avec $b'_i \in B$. Le fait que $\langle AB^2 \rangle$ soit de dimension maximale implique que

$$\lambda_j a_j b_j b = \mu_j a_j b'_j$$

d'où nous déduisons $b_j b \in B$. Puisque ceci est vrai pour n'importe quel $b \in B$, nous obtenons $b_j B \subset B$. □

Proposition 177. *Supposons que m est premier. Soient A et B deux sous-espaces de dimensions respectives α et β. Soit (b_i) une base de B et $S = \langle AB \rangle$, alors*

$$\text{Prob}\left(\bigcap_i b_i^{-1} S = A \right) \geqslant 1 - \alpha \frac{q^{\alpha \frac{\beta(\beta+1)}{2}}}{q^m}$$

Démonstration. *Si ce n'est pas le cas il existe un sous-espace $E \supsetneq A$ tel que $\langle EB \rangle = \langle AB \rangle$. D'après la remarque 175 , les conditions du lemme 176 sont vérifiées. Mais alors il existe $x \notin \mathbb{F}_q$ tel que $xB \subset B$. Ceci implique que $\mathbb{F}_q(x)B \subset B$, mais le fait que m soit premier implique qu'il n'y a pas d'extension intermédiaire entre $\mathbb{F}_q$ et $\mathbb{F}_{q^m}$ donc $\mathbb{F}_{q^m} \subset B$ ce qui est une contradiction.* □

On peut atteindre une meilleure borne : montrons qu'avec une bonne probabilité, lorsque A et B sont choisis aléatoirement avec une dimension suffisamment petite, alors nous avons le résultat suivant avec une probabilité proche de 1 :

$$\bigcap_{b \in B} \langle AB \rangle b^{-1} = A$$

Sans perte de généralité nous pouvons supposer que $1 \in B$. Nous allons montrer que un $b \in B$ aléatoire nous avons

$$\langle AB \rangle \cap \langle AB \rangle b^{-1} = A$$

avec une probabilité proche de 1.

Nous avons besoin de deux lemmes intermédiaires :

Lemme 178. *Soit B_1 un sous-espace de dimension β_1. Soit b un élément aléatoire de $(\mathbb{F}_q)^m$ uniformément distribué, alors*

$$\text{Prob}(b \in B_1 + B_1 b^{-1}) \leqslant \frac{2q^{2\beta_1}}{q^m}$$

Démonstration. *Ceci peut se produire seulement si b est une racine d'une équation de la forme*

$$x^2 - b_1 x - b'_1 = 0$$

Il y a au plus $|B_1|^2 = q^{2\beta_1}$ équations de cette forme et chacune d'elles a au plus deux racines.

Soit B_1 un espace vectoriel quelconque contenant 1 et de dimension β_1. Soit b un élément aléatoire uniformément distribué dans $(\mathbb{F}_q)^m$, posons $B = B_1 + <b>$ Notons $\beta = \dim B = \beta_1 + 1$ (avec une probabilité $1 - \frac{q^{\beta_1}}{q^m}$). Puisque b^{-1} est lui aussi uniformément distribué, nous avons $B_1 \cap B_1 b^{-1} \neq \{0\}$ avec une probabilité d'au plus

$$(|B_1| - 1)\frac{|B_1|}{q^m} \leqslant \frac{q^{2\beta_1}}{q^m}$$

Par conséquent, avec une probabilité d'au moins

$$1 - \frac{q^{2\beta_1}}{q^m}$$

nous avons

$$\dim(B_1 + B_1 b^{-1}) = 2\beta_1$$

□

A partir de ce lemme on obtient :

Lemme 179. $$\text{Prob}(\dim(B + Bb^{-1}) = 2\beta - 1) \geqslant 1 - \frac{3q^{2\beta 1}}{q^m}$$

Proposition 180. *Soit B un sous-espace de dimension β contenant 1 tel que $\dim(B + Bb^{-1}) = 2\beta - 1$ pour un certain $b \in B$. Soit A un sous-espace choisi aléatoirement de dimension α, alors*

$$\text{Prob}(\langle AB\rangle \cap \langle AB\rangle b^{-1} = A) \geqslant 1 - \alpha\frac{q^{\alpha(2\beta-1)}}{q^m}$$

Démonstration. *D'après la proposition 174 nous obtenons qu'avec une probabilité d'au moins*

$$1 - \alpha\frac{q^{\alpha(2\beta-1)}}{q^m}$$

nous avons

$$\langle AB\rangle \cap \langle AB\rangle b^{-1} = A$$

Nous pouvons aussi établir, dans les conditions des lemmes précédents,

$$\dim\langle A(B + Bb^{-1})\rangle = \alpha(2\beta - 1) = 2\alpha\beta - \alpha$$

D'autre part remarquons que

$$\dim\langle A(B + Bb^{-1})\rangle = \dim\langle AB\rangle + \dim\langle ABb^{-1}\rangle - \dim(\langle AB\rangle \cap \langle ABb^{-1}\rangle)$$

$$= 2\alpha\beta - \dim(\langle AB\rangle \cap \langle ABb^{-1}\rangle)$$

d'où l'on déduit

$$\dim(\langle AB\rangle \cap \langle ABb^{-1}\rangle) = \alpha$$

Mais ceci prouve le résultat puisque $A \subset \langle AB\rangle \cap \langle ABb^{-1}\rangle$ et $\dim A = \alpha$. □

Conclusion et pistes de recherche

Nous avons travaillé sur des domaines liés au codes correcteurs d'erreurs en métrique rang. Deux domaines se sont dégagés, qui ont été présentés dans les deux grandes parties de cette thèse : premièrement, l'étude des q-polynômes d'un point de vue principalement théorique ; deuxièmement, la conception et l'implémentation du cryptosystème LRPC, un cryptosystème basé sur des codes rang faiblement structurés définis par une matrice de parité de poids faible. Nous présentons dans cette partie une synthèse de nos travaux et une ouverture à des pistes de recherche.

1 Concernant les q-polynômes

Les q-polynômes sont connus depuis longtemps mais certains résultats déjà connus les concernant, notamment par rapport à la structure d'espaces vectoriels de leurs racines et aux opérations sur ces espaces vectoriels, n'avaient jamais été formalisés. C'est ce que nous avons fait au chapitre 3 avec le formalisme de l'algèbre linéaire sur des corps finis. Nous avons vu qu'à tout espace vectoriel V sur un corps fini on peut associer un q-polynôme P_V dont l'ensemble des racines est exactement V ; le q-polynôme associé à l'intersection de deux espaces vectoriels $V \cap W$ est le PGCD à droite de V et W, et le q-polynôme associé à $V + W$ est le PPCM à droite de V et W.

Au chapitre 4 nous avons présenté de nouveaux résultats concernant les q-polynômes. Nous avons vu comment calculer le PGCD à droite de deux q-polynômes via l'utilisation de la matrice de multiplication et comment cette même matrice nou permettait de calculer les sous-résultants de deux q-polynômes, les sous-résultants étant définis en adaptant à notre travail la définition de Li [Li].

En combinant ces résultats nous avons décrit une nouvelle façon de déterminer l'intersection de deux espaces vectoriels V et W : on calcule P_V et P_W, puis leur PGCD à droite (par exemple avec la matrice de multiplication) puis l'espace vectoriel des racines de ce PGCD ; cet espace est $V \cap W$. La valeur de cet algorithme est essentiellement théorique puisque sa complexité est comparable aux algorithme existants.

En raison notamment de la non-commutativité de leur loi de multiplication, les q-polynômes restent assez peu étudiés. De nombreuses techniques de calcul sur les polynômes classiques n'ont pas d'équivalent sur les q-polynômes, par exemple il n'existe pas d'équivalent d'algorithme de Karatsuba pour effecture une multiplication rapide.

Les q-polynômes ont également une utilisation pratique en théorie des codes, dans la création des codes de Gabidulin et leur utilisation pour le cryptosystème GPT. Cependant nous avons vu que la forte structure de ces codes rendait le cryptosystème très vulnérables aux attaques structurelles.

2 Concernant la cryptographie sur les codes correcteurs en métrique rang

Par rapport aux cryptosystèmes usuels basés par exemple sur la factorisation de grands entiers ou les courbes elliptiques, les cryptosystème basés sur les codes correcteurs présentent l'avantage d'être basés sur la résolution de problèmes NP-difficiles ce qui les rend résistants aux attaques menés par des ordinateurs quantiques. Les cyryptosystèmes basés sur des codes correcteurs ont également l'avantage de proposer des opérations de chiffrement et de déchiffrement très rapide.

Cependant ces cryptosystèmes ont notamment deux défauts qui fait qu'ils sont peu utilisés en pratique : leur principal défaut est la taille très importante de leur clé publique qui est une matrice ; ils peuvent aussi avoir le défaut, selon la famille de codes choisie, d'être vulnérable aux attaques structurelles. En général, les moyens mis en oeuvre pour diminuer cette vulnérabilité reviennent d'une façon ou d'une autre à augmenter la taille de la clé, qui est déjà très importante.

En se basant sur des travaux antérieurs, Gabidulin introduit dans [Gab1] la métrique rang qui est peu adaptée à la théorie des codes mais semble bien convenir pour des applications cryptographiques car les cryptosystèmes en métrique sont immunisés aux attaques par ensemble d'information. Gabidulin introduit les codes de Gabidulin, une famille de codes dont les mots sont des évaluations de q-polynômes, et le cryptosystème GPT [GPT], un cryptosystème de type McEliece basé sur les codes de Gabidulin. Cependant il souffre de la même vulnérabilité aux attaques structurelles et est attaqué par Overbeck dans [Ove]. Par la suite, les améliorations et variantes ultérieures apportées au cryptosystème GPT ont toujours pu finalement être attaquées à cause de cette forte structure. Il y a donc des raisons de penser que des améliorations ultérieures seront aussi attaquées, et donc d'être assez pessimiste quant à la viabilité des cryptosystèmes basés sur les codes de Gabidulin et plus généralement sur des codes trop structurés.

Nous avons développé un cryptosystème qui conserve les avantages des cryptosystèmes en métrique rang tout en corrigeant leurs défauts : nous avons présenté le cryptosystème LRPC, que l'on peut voir comme une généralisation à la métrique rang du récent cryptosystème MDPC [MTSB]. Le cryptosystème LRPC est un cryptosystème de McEliece basés sur les codes LRPC, une famille de codes en métrique rang faiblement structurés et donc a priori résistants aux attaques structurelles. Ces codes sont définis par leur matrice de parité qui est de poids rang faible ce qui permet, en se ramenant à des opérations dans le corps de base, d'avoir un algorithme de décodage efficace. En choisissant pour la matrice de parité une matrice doublement circulante (on parle dans ce cas de codes DC-LRPC) on peut générer toute la matrice à partir d'une seule ligne ce qui permet de diminuer considérablement la taille de la clé, nous avons notamment présenté une clé de 1517 bits avec une sécurité de plus de 2^{80}.

L'algorithme de déchiffrement proposé a le défaut d'être probabiliste, cependant nous avons vu que la probabilité d'échec de déchiffrement peut être rendue arbitrairement basse tout en gardant une clé de taille raisonnable ; nous avons donné un exemple de clé de 3128 bits avec une probabilité d'échec de déchiffrement de 2^{-80}.

Une attaque a été proposée dans le cas des codes DC-LRPC en se basant sur la notion de code replié. Cette attaque ne semble baisser significativement la sécurité que si le polynôme $\sum_{i=0}^{k-1} X^i$ admet des diviseurs de degré élevé (k étant la dimension du code). Dans certains cas l'attaque ne permet donc pas de baisser significativement la sécurité. Dans les cas où cette attaque est efficace de nouveaux paramètres pour les codes ont été trouvés qui permettent de contrer l'attaque tout en gardant une probabilité d'échec de déchiffrement arbitrairement basse. Cette protection se fait au prix d'une légère augmentation de la taille de la clé, taille qui reste de quelques milliers de bits.

L'amélioration des performances des ordinateurs, et notamment la mise au point des ordinateurs quantiques, pourrait rendre obsolètes tous les systèmes de chiffrement classiques. Il est pertinent d'étudier des cryptosystèmes ne présentant pas cette vulnérabilité. Les cryptosystèmes basés sur les codes correcteurs offrent des résultats prometteurs dans ce sens. Le cryptosystème MDPC et notre cryptosystème LRPC sont deux exemples de cryptosystème où nous parvenons à baisser significativement la taille de la clé tout en gardant une bonne sécurité.

En particulier, la cryptographie en métrique rang reste finalement assez peu étudiée. Cette métrique est peu adaptée à la théorie des codes sauf dans des cas particuliers, et ses premières applications cryptographiques ont toutes été attaquées. Le cryptosystème LRPC que nous présentons est le premier cryptosystème existant en métrique rang dont la clé est petite et qui ne soit pas vulnérable aux attaques structurelles. Peu de résultats sont connus en métrique rang alors que cette métrique est naturellement adaptée à une utilisation en cryptographie et beaucoup de résultats restent probablement à découvrir.

Bibliographie

[AJ] F. Apéry, J. P. Jouanolou - *Elimination - Le cas d'une variable*, Hermann, 2006

[AR] C. Aguilar, O. Ruatta - *Complexité : P v.s. NP, un problème qui peut vous rapporter des millions*, cours de Master 1 Cryptis, Université de Limoges, 2014

[BCGO] T.P. Berger, P.L. Cayrel, P. Gaborit, A. Otmani - *Reducing Key Length of the McEliece Cryptosystem*, AFRICACRYPT 2009, 77-97

[BGJT] R. Barbulsecu, P. Gaudry, A. Joux, E. Thomé - *A quasi-polynomial algorithm for discrete logarithme in finite fields of small characteristic*, Eurocrypt 2014

[BJMM] A. Becker, A. Joux, A. May, A. Meurer - *Decoding Random Binary Linear Codes in 2n/20 : How 1+1=0 Improves Information Set Decoding*, EUROCRYPT 2012, 520-536

[BET] E.R. Berlekamp, R.J. McEliece, H.C.A. van Tilborg - *On the inherent intractability of certain coding problems. IEEE Transactions on Information Theory*, vol.24, 1978

[Ber] E.R. Berlekamp - *Algebraic Coding Theory*, Aegean Press Park, 1984

[BL] T.P. Berger, P. Loidreau - *Designing an efficient and Secure Public-Key Cryptosystem Based on Reducible Rank Codes*, INDOCRYPT 2004, 218-229

[BL2] T. P. Berger, P. Loidreau - *How to Mask the Structure of Codes for a Cryptographic use* Designs, Codes, and Cryptography, 35, 63-79, 2005

[BW] E.R. Berlekamp, L. Welch - *Error correction of algebraic block codes* US Patent, Number 4, 633,470, 1986

[Cha] M. Chardin - *Differential resultants and subresultants*, in *Fundamentals of computation theory* Springer, vol. 529, pp. 180-189, 1991

[Co] N. Courtois - *Efficient zero-knowledge authentication based on a linear algebra problem MinRank* , *Asiacrypt 2001*, LNCS 2248, pp. 402-421

[CFS] N. Courtois, M. Finiasz, N. Sendrier - *How to achieve a McElice-base Digital Signature Scheme* Technical report, INRIA, http://eprint.iacr.org/2001/010, 2001

[CK] D. Cantor, E. Kaltofen - *Fast Multiplication of Polynomials over Arbitrary Algebra* Acta Informatica, 28, 693-701, 1991

[CLU] L. Chaussade, P. Loidreau, F. Ulmer - *Skew codes of prescribed distance or rank* Designs, Codes and Cryptography, 50(3), 267-284 (2009)

[CS] F. Chabaud, J. Stern - *The Cryptographic Security of the Syndrome Decoding Problem for Rank Distance Codes*, ASIACRYPT 1996, 368-381

[Del] P. Delsarte - *Bilinear forms over a finite field with applications to coding theory* in *Journal of Comb. Theory A*, vol.25, 1978, pp. 226-411

[FL] C. Faure, P. Loidreau - *A New Public-Key Cryptosystem Based on the Problem of Reconstructing p-Polynomials*, WCC 2005, 304-315

[FO] E. Fujisaki, T. Okamoto - *Secure integration of asymmetric and symmetric encryption schemes* Lecture Notes in Computer Science, vol.1666, 537-554, 1999

[FOPT] J.C. Faugère, A. Otmani, L. Perret, J.P. Tillich - *Algebraic Cryptanalysis of McEliece Variants with Compact Keys*, EUROCRYPT 2010, 279-298

[Gab1] E.M. Gabidulin - *Theory of codes with maximal rank distance.* in *Problems of Information Transmission*, 1985

[Gab2] E. M. Gabidulin - *Public-key cryptosystems based on linear codes over large alphabets : efficiency and weakness*, In *Codes end Cyphers*, Formara Limited, Southend-on-sea, Essex, 1995

[Gab3] E.M. Gabidulin - *Attacks and counter-attacks on the GPT public key cryptosystem* Des. Codes Cryptography 48, 2008, 171-177

[Gal] R. G. Gallager, *Low-Density Parity Check Codes*, Mémoire de Thèse, 1963

[Gbt] P. Gaborit - *Shorter keys for code-based cryptography* International Workshop on coding and Cryptography, 2005

[GGH] O. Goldreich, S. Goldwasser, S. Halevi - *Public-Key Cryptosystems from Lattice Reduction Problems*. CRYPTO 1997, 112-131

[Gib] J. K. Gibson - *Severely Denting the Gabidulin Version of the McEliece Public-Key Cryptosystem* Designs, Codes and Cryptography, 6, 37-45, 1995

[Gie] M. Giesbrecht, *Factoring in Skew-polynomial Rings over finite fields* Journal of Symbolic Computations, 2, 187-224, 1991

[GMR] P. Gaborit, G. Murat, O. Ruatta - *Ore Polynomials, Elimination and Applications* submitted

[GMRZ] P. Gaborit, O. Ruatta, G. Murat, G. Zémor - *Low Rank Parity Check Codes and their application in cryptography*, WCC 2013, 167-179

[GNW] V. Guruswami, S. Narayanan, C. Wang - *List decoding subspace codes from insertions and deletions*, ITCS 2012, 183-189

[GO] E. M. Gabidulin, A. V. Ourivski - *Modified GPT PKC with Right Scrambler* WCC 2001, 233-242

[GPT] E.M. Gabidulin, A.V. Paramonov, O.V. Tretjakov - *Ideals over a Non-Commutative Ring and their Applications in Cryptology*. EUROCRYPT 1991, pp. 482-489

[GRS] P. Gaborit, O. Ruatta, J. Schreck - *On the complexity of the Rank Syndrome Decoding Problem* arXiv preprint, arXiv : 1301.1026, 2013

[GRSZ] P. Gaborit, O. Ruatta, J. Schreck, G. Zémor - *Ranksign : en efficient signature algorithm based on the rank metric*, eprint iacr, submitted

[GRSZ2] P. Gaborit, O. Ruatta, J. Schreck, G. Zémor - *New Results for Rank-based Cryptography* Africacrypt 2014, submitted

[GS] V. Guruswami, M. Sudan - *Improved decoding of Reed-Solomon and algebraic-geometry codes* IEEE Transactions on inf. Theory, vol.45, 1767-1767, 1999

[GSH] J. von zur Gathen, V. Shoup - *Computing Frobenius map and factoring polynomials* Computational Complexity, vol.2, 187-224, 1991

[GSZ] P. Gaborit, J. Schreck, G. Zémor - *Full Cryptanalysis of the Chen Identification Protocol* PQCrypto 2001, 35-50

[GX] V. Guruswami, C. Xing - *List-decoding Reed-Solomon, algebraic-geometric, and Gabidulin subcodes up to the Singleton bound*, STOC 2013, 843-852

[GZ] P. Gaborit, G. Zémor - *On the Hardness of the Decoding and the Minimum Distance Problem for Rank Codes*, arXiv preprint, arXiv : 1404.3482, 2014

[Ham] R.W. Hamming - *Error Detecting and Error Correcting codes*
The Bell System Technical Journal, vol.29, 1950

[Hau] A. Hauteville, *Rapport de stage de Master 2*, Université de Limoges, 2014

[HPS] J. Hoffstein, J. Pipher, J.H. Silverman - *NTRU : A Ring-Based Key Cryptosystem*
ANTS 1998, 267-288

[Hua] Hua L.- *A theorem on matrices over a field and its applications.*
in *Chinese mathematical society*, vol.1, 1951, pp. 109-163

[JH] J. Justesen, T. Hoholdt, *A course in Error-Correcting Codes* (EMS textbooks in Mathematics), EMS 2004

[KI] K. Kobara, H. Imai - *Semantically secure McEliece public-key cryptosystems -conversions for pkc-*
in K. Kim, editor. Public Key Cryptography, vol. 1992 of Lecture Notes in Computer Science

[KK] R. Koetter, F. R. Kschischang - *Coding for Errors and Erasures in Random Network Coding*
IEEE Transactions on Information Theory 54(8), 3579-3591, 2008

[Li] Z. Li, *A Subresultant Theory for Linear Differential, Linera Difference and Ore Polynomials, with Applications*, mémoire de thèse, 1996

[Loi] P. Loidreau - *Métrique rang et cryptographie*, mémoire d'HDR, 2007

[Loi2] P. Loidreau - *Designing a Rank Metric Based McEliece Cryptosystem*, PQCrypto 2012, 142-152

[Loi3] P. Loidreau - *Properties of Codes in Rank Metric*, http://arxiv.org/abs/cs/0610057

[Lue] F. Luebeck - *Conway polynomials for finite fields*
http://www.math.rwth-aachen.de/~Frank.Luebeck/data/ConwayPol/index.html?LANG=en

[LP] F. Levy-dit-Vebel, L. PERRET - *Algebraic decoding of rank metric codes*, proceedings of YACC06

[LPR] V. Lyubashevsky, C. Peikert, O. Regev - *On Ideal Lattices and Learning with Errors over Rings*
J. ACM 60(6) : 43, 2013

[LYC] S. Y. R. Li, R. W. Yeung, N. Cai - *Linear network coding*
IEEE Transactions on Information Theory 49(2), 371-381, 2003

[MB] R. Misoczki, P.S.M. Barreto - *Compact McEliece Keys from Goppa Codes*
Selected Areas in Cryptography, 2009, 376-392

[McEl] R.J. McEliece - *A public-key Cryptosystem based on Algebraic Coding Theory.*
Rapport de recherche, Jet Propulsion Lab. DSN Progress Report, 1978

[Mis] R. Misoczki - *Two Approaches for Achieving Efficient Code-Based Cryptosystems.*
Mémoire de thèse, 2013

[mpfq] E. Thomé, P. Gaudry et.al. - MPFQ - *A finite field library*, http://mpfq.gforge.inria.fr/

[MR] D. Micciancio, O. Regev - chapitre *Lattice-based Cryptography*
in *Post-quantum Cryptography*, D.J. Bernstein, J. Buchmann, Springer, 2008

[MS] F.J. MacWilliams, N.J.A Sloane, *The theory of Error-Correcting Codes*
North-Holland Mathematical Library, 1977

[MTSB] R. Misoczki, J.P. Tillich, N. Sendrier, P. S.L.M. Barreto - *MDPC-McEliece : New McEliece Variants from Moderate Density Parity-Check Codes.*
IACR Cryptology ePrint Archive (IACR) 2012:409, 2012

[MV] H. Mahdavifar, A. Vardy - *Algebraic List-Decoding of Subspace Codes* IEEE Transactions on Information Theory, 59(12), 7814, 7828, 2013

[Nie] H. Niederreiter - *Knapsack-type Cryptosystems and Algebraic Coding Theory* Problems of Control and information Theory, 15(2), 159-166, 1986

[OGHA] A. V. Ourivski, E. M. Gabidulin, B. Honary, B. Ammar - *Reducible rank codes and their applications to cryptography*, IEEE Trans. on Inf. Theoey, 49(12), 3829-3293, 2003

[OJ] A. V. Ourivski, T. Johansson, *New Technique for Decoding Codes in the Rank Metric and its Cryptographic Aplications*, Probl. Inf. Transm (38), 237-246, 2002

[Ore1] O. Öre - *Theory of non-commutative polynomials.* in *The Annals of Mathematics*, vol.34, 1933, pp. 480-508

[Ore2] O. Öre - *On a special class of polynomials.* in *Transaction of the American Mathematical Society*, vol.35, 1933, pp. 237-246

[Ore3] O. Öre - *Contribution to the theory of finite fields.* in *Transaction of the American Mathematical Society*, vol.36, 1934, pp. 243-274

[OTD] A. Otmani, J.P. Tillich, L. Dallot - *Cryptanalysis of Two McEliece Cryptosystems Based on Quasi-Cyclic Codes*, Mathematics in Computer Science 3(2), 2010, 129-140

[Ove] R. Overbeck - *Structural Attacks for Public Key Cryptosystems based on Gabidulin Codes, Journal of Cryptology*, vol.21, 2008, pp. 280-301

[QD] Qi C., Daqing W. - *A deterministic reduction for the gap minimum distance problem* STOC 2009, pp. 33-38

[RS] I.S. Reed, G. Solomon - *Polynomial Codes Over Certain Finite Fields* in *SIAM Journal of Applied Mathematics*, vol.8, 1960, pp. 300-304

[Rol] R. Rolland - *Fonctions de Möbius - Formule de Rota*, CNRS, Institut de mathématiques de Luminy, 2006

[Sc] A. Schönhage - *Schnelle Multiplikation von Polynomen über Körpen der Charakeristik 2* Acta Informatica, 7, 395-398-1977

[ScSt] A. Schönhage, V. Strassen - *Schnelle Multiplikation grosser Zahlen*, Computing, 7, 281-292, 1971

[Sha] C.E. Shannon - *A Mathematical Theory of Communications* The Bell System Technical Journal, vol.27, 379-429, 623-656, 1948

[SiSh] V.M. Sidel'nikov, S.O. Shestakov - *On cryptosystems based on generalized Reed-Solomon codes* in *Discrete Mathematics*, vol.4, 1992, pp. 57-63

[Ste] J. Stern, *A method for finding codewords of small weight*, Lecture Notes in Computer Science vol. 388, 106-113, 1989

[Sud] M. Sudan - *Decoding of Reed-Solomons Codes beyond the Error-Correcting Bound* Journal of Complexity, vol.13, 180-193, 1997

[TSC] V. Tarokh, N. Seshadri, A. R. Calderbank - *Space-Time Codes for High Data Rate Wireless Communications : Performance criterion and Code Construction.* IEEE Transactions on Information Theory 44(2), 744-765, 1998

[Var] A. Vardy - *Algorithmic Complexity in Coding Theory and the Minimum Distance Problem.* STOC 1997, pp. 92-109

[Wer] Nickel, Werner *Endliche Körper in dem gruppentheoretischen Programmsystem GAP*, Mémoire de thèse, 1988

[Zém] G. Zémor - *Arithmétique 1 : corps finis et applications*, cours de Master CSI, université de Bordeaux 1, 2006