MÉMOIRE D'HABILITATION À DIRIGER DES RECHERCHES

Université Pierre et Marie Curie, Paris 6

Spécialité :

Informatique

présentée par

Pierre Loidreau

Ecole Nationale Supérieure de Techniques Avancées (ENSTA)

Sujet :

Métrique rang et cryptographie

Soutenue le 25 janvier 2007, devant le jury composé de

Rapporteurs

M.	Tom	Høholdt	Technical University of Denmark (Danemark)
M.	Grigory	Kabatiansky	IPPI (Russie)
M.	Gilles	Zémor	Université de Bordeaux I

Examinateurs

M.	Thierry	Berger	Université de Limoges
M.	Daniel	Lazard	Université Paris 6
M.	François	Morain	Ecole Polytechnique
M.	Nicolas	Sendrier	INRIA Rocquencourt
M.	Gilles	Villard	INRIA Rhône-Alpes

version du 30 janvier 2007

A mon père

La partie des remerciements n'est jamais la plus simple car on craint d'oublier les personnes qui, peu ou prou, ont contribué aux orientations de la recherche ainsi qu'à l'élaboration de ce document tant par leurs conseils que par leurs encouragements.

Je souhaite remercier d'une façon toute particulière Pascale Charpin qui, depuis mes débuts, me prodigue régulièrement des conseils et avec qui les discussions ne sont jamais convenues. L'ouverture d'esprit dont elle fait preuve et la liberté qu'elle laisse sont des qualités assez rares pour que l'on veuille en faire des modèles, même si parfois la voie à suivre semble semée d'embûches.

Un grand merci à Tom Høholdt qui, outre sa présence à ce jury, m'a fait l'honneur de rapporter ce document et qui, malgré un texte écrit en français, fit preuve d'une remarquable célérité dans la rédaction du rapport.

Je remercie également Grisha Kabatiansky toujours plein de verve et d'humour, spécialement lorsqu'il s'agit de soulever des points délicats. Il avait déjà beaucoup fait en tant que rapporteur de ma thèse de doctorat. Je le remercie également pour ses talents d'organisateur de conférences.

Je n'oublierai pas Gilles Zémor, le seul francophone parmi les rapporteurs, mais qui pourrait aussi bien être rapporteur dans quelque autre langue. Je le remercie des précieux conseils qu'il m'a donnés en vue de l'amélioration de ce mémoire.

Thierry Berger est un examinateur naturel et occupé dans le domaine des codes correcteurs appliqués à la cryptographie. Je lui sais gré néanmoins d'avoir accepté de faire partie de ce jury. Ses connaissances furent très précieuses dans les articles que nous avons rédigés ensemble et j'ai pris beaucoup de plaisir à travailler avec lui et à venir à Limoges manger un bon steak tartare.

Je suis très heureux que François Morain ait accepté de faire partie de ce jury. A force de participer à des jurys de thèse ou bien d'habilitation dont les sujets relèvent des codes correcteurs et de leurs applications cryptographiques, il doit être devenu, j'en suis sûr un redoutable spécialiste.

Merci à Nicolas Sendrier également pour son humour, son soutien et tous ses bons conseils donnés tant dans le domaine de la cryptographie que celui de l'œnologie. Je suis sûr que dans ce dernier domaine aussi il acquerra une renommée internationale.

Merci à Daniel Lazard et à Gilles Villard d'avoir gentiment bien voulu faire partie de ce jury.

J'ai une pensée aussi pour Ernst Gabidulin, qui est pour beaucoup à l'origine de ce travail, et je l'en remercie.

Un grand merci à Françoise Levy-dit-Vehel qui a beaucoup contribué à ma venue à l'ENSTA. J'apprécie particulièrement son professionnalisme et l'implication dont elle fait preuve dans tous les domaines et notamment l'organisation, l'enseignement et la recherche.

Merci à Patrick Ciarlet pour son bon goût musical qui rythme nos journées pluvieuses, ainsi qu'à Michel Mauny arrivé depuis peu qui fait preuve d'un grand dynamisme et qui a su apporter une touche d'exotisme rafraîchissante au fonctionnement du laboratoire.

Je remercie toutes les personnes qui ont jalonné et continuent de jalonner mon parcours de recherche, en particulier toutes les personnes qui ont contribué par leur travail à la réalisation de ce document : Daniel Augot, Alexandre Hersans, Matthieu Finiasz, Bruno Giroire, Alexandra Petrova, Christelle Roux, Alexei Ourivski, Bassem Sakkour, Cédric Faure, Vitaly Shorin et Raphaël Overbeck.

Merci aux personnes que je côtoie régulièrement et que j'ai côtoyées à l'INRIA agrémentant le quotidien, par leurs discussions, gâteaux et autres crêpes. Ils sont si nombreux *que je crains d'en oublier : Anne Canteaut, Claude Carlet, Guy Chassé, Eric Filiol, Caroline Fontaine, Christelle Guiziou, Jean-Pierre Tillich, Ayoub Otmani, Cédric Tavernier, Fabien Galand, Marion Videau, Raghav Bhaskar Ludovic Perret, et des plus jeunes, Thomas Camara, Mathieu Cluzeau, Frédéric Didier, Cédric Lauradoux, Yann Laigle-Chapuy, Michaël Quisquater, Andrea Röck, Maria Naya Plasencia...*

De l'ENSTA je garderai entre autres le souvenir ému de ces journées de Psaumes organisées par notre cher Jérôme Pérez qui me suit de peu dans cette voie. Pour cela, je remercie Anne-Sophie Bonnet-Bendhia, Laurent Bourgeois, Pierre Carpentier, Colin Chambeyron, Jean-Luc Commeau, Maurice Diamantini, Eve-Marie Duclairoir, Christophe Hazard, Grace Hechme, Erell Jamelot, Frédéric Jean, Samir Kaddouri, Nicolas Kielbasiewicz, David Lefèvre, Guillaume Legendre, Marc Lenoir, François Loret, Eric Lunéville, Annie Marchal, Nadia Megdich, Jean-François Mercier, Francis Muguet, Jerónimo Rodriguez, Fabrice Roy, Hasnaa Zidani...

Enfin je tiens tout particulièrement à remercier Delphine, qui fait preuve quotidiennement de tendresse et qui m'a soutenu dans les moments difficiles de la rédaction de ce mémoire. Merci à elle pour tout ce qu'elle m'apporte.

Table des matières

Liste des tableaux

Overview of the document

Public-key cryptography based on error-correcting codes originates from a paper by R. J. McEliece, see [McE78]. The general security of the original system relies on the difficulty of solving the problem of bounded distance decoding for a random linear code up to its error-correcting capability in Hamming metric. This problem is related to several decisional problems that were proved to be NP-complete, see [BMvT78, Var97]. These results, as well as the complexity of the most efficient decoding algorithms give some convincing arguments that such systems are secure provided they are well designed and the parameters are sufficiently high.

More recently Gabidulin, Paramonov and Tretjakov imagined a McEliece-like cryptosystem using a new metric previously introduced by Gabidulin in [Gab85b], the so-called *rank metric*. It is suitable for correcting matricial errors of upper-bounded rank. The main advantage of systems based on rank metric is that this enables the conceiver of a cryptosystem to employ a much smaller public-key in the design than it is necessary in Hamming metric. Namely, though there is no proof that decoding in rank metric is closely related to some NP-complete decisional problem, the complexity of the best known decoding algorithms is much higher than in Hamming metric, for the same sets of parameters.

The interest of constructing new public-key cryptosystems with reduced key size motivated my interest for the properties of the metric, for the construction of families of codes with efficient decoding algorithms and the study of related problems. This formed a very important part of my research over the last few years. More than presenting the results I obtained on rank metric, codes in rank metric or cryptographic applications of rank metric, the document is an attempt to summarise and structure existing results already published in various places and by various authors related to the rank metric and more precisely to its cryptographic applications.

Organisation of the document

This document is formed of three distinct parts. The first part introduces concepts about rank metric, then some mathematical tools involved, like the family of linearised polynomials and then basic facts about the statements of decoding problems in rank metric and some existing approaches to solve them. A second part is dedicated to the study of families of codes with polynomial-time decoding algorithms initially introduced by E. M. Gabidulin. Finally a third part deals with the evolution of rank metric-based cryptography and especially public-key cryptosystems.

Chapter 1. This chapter is a general introduction to rank metric. There are some introductory elements available in the seminal paper, see [Gab85b]. From approximations on the volume of balls and spheres in the metric, I prove some properties about the non-existence of perfect codes. With the same approximations is derived an expression for an equivalent of Gilbert-Varshamov bound and an asymptotic equivalent for the minimum rank distance of codes which are *on GV*. This result is given by Proposition 4, in which is shown that for a $(n, M, d_{GV})_r$-code over $GF(q^m)$ with $m \geq n$ we have

$$\frac{d_{GV}}{m+n} \underset{n\to+\infty}{\sim} \frac{1}{2} - \frac{\sqrt{\log_q M}}{m+n}\sqrt{1+\frac{(m-n)^2}{4\log_q M}}$$

provided $mn \geq \log_q M = \lambda(n)(m+n)$, where $\lambda(n)$ tends to $+\infty$ with n. Then, we derive a bound on the packing density of optimum codes. Proposition 7 gives an approximation of the packing density of optimal codes in rank metric. This leads to the noticeable fact that, whenever the extension degree of the field is equal to the length of the code, this density is bounded by numbers depending only on the base field and on the error-correcting capability of the code. These result are not yet published but submitted for publication.

Chapter 2. In this chapter, we define different decoding problems which can arise in rank metric : the problem of Maximum-Likelihood decoding (MLD), Bounded Distance decoding and List decoding. There are no known algorithms solving MLD in rank metric than the enumeration of the code. The most efficient algorithms solving the BDD-problem up to the error-correcting capability for a linear code are described. They are derived from an algorithm finding low-rank codewords and are due to Ourivski and Johannson, see [OJ02]. They can be considered as the analogs of Information Set decoding algorithms existing for Hamming metric codes. Their complexity is exponential and shows that with equivalent parameters, it is more difficult to decode in rank metric than in Hamming metric. Namely the complexity of decoding a linear $[n, k, d]_r$-code over $GF(q^m)$ up to $t \leq (d-1)/2$ is given by

- Bases enumeration algorithm : $W_{\text{bases}} \leq (k+t)^3 q^{(t-1)(m-t)+2}$ q-ary operations.
- Coordinates enumeration algorithm : $W_{\text{coord}} \leq (k+t)^3 t^3 q^{(t-1)(k+1)}$ q-ary operations.

We conclude this chapter by a first step towards list decoding. Given a random code of size M uniformly distributed over $GF(q^m)^n$ the average number of codewords within a ball centred on a vector of the ambient space and of radius t is polynomial in m and n provided

$$t \leq \frac{m+n+1}{2} - \sqrt{\log_q (Mn^\lambda) + \frac{(m-n)^2 - 2m - 2n}{4}},$$

where λ is some given number. In the case of Gabidulin codes, this result was published in [Fau06].

Chapter 3. The ring of q-polynomials defined by Øre is presented, see [Ore33]. This ring is well-suited for the study of rank metric and codes in rank metric. Indeed, it performs the same role as classical polynomials in Hamming metric. With a noticeable exception that it is non-commutative and the role of multiplication is performed by the composition of applications.

Additionally to the definitions related to the ring, we present algorithms making simple operations in this ring such as additions and multiplications of q-polynomials, roots finding, interpolation on some coefficients and a right and left Euclidian division algorithm. We also give the complexity these algorithms. The fact that the ring is non-commutative is problematic when one tries to diminish the complexity by using the same methods as for the classical polynomial ring. In particular, this prevents from using Karatsuba's approach for the composition of q-polynomials.

And in the last section, we exhibit links between some decoding problems in rank metric and problems related to reconstruction of q-polynomials. It is the heart of proposition 14.

Chapter 4. The family of Gabidulin codes is presented in this chapter. In a first part we recall some basic properties of these codes such as : They are evaluation codes of q-polynomials, they are optimal codes in the rank metric regarding the corresponding Singleton bound, and therefore they can be seen as the equivalent in rank metric of Reed-Solomon codes. In a second part we present the principle of the polynomial-time decoding algorithms that were until recently the only known algorithms to decode Gabidulin codes up to the error-correcting capability. Despite the existence of a Berlekamp-Massey like algorithm, all these algorithms are of cubic complexity in the number of errors, see [Gab85b, Gab91, Rot91, RP04b, RP04a].

In the last section, we show that one can decode Gabidulin codes up to the error-correcting capability by solving the problem of list-decoding a Gabidulin code. Namely, in that case the size of the list is at most 1 and the list-decoding can be achieved by solving the problem of reconstructing q-polynomials with the parameters chosen accordingly. This approach enables to design a quadratic polynomial-time algorithm for decoding Gabidulin codes up to the error-correcting capability. This part of the chapter is an improvement of the algorithms presented in papers published in the *Compte Rendus de l'Académie des sciences* and in the *proceedings of WCC 2005*, see [Loi04, Loi06]. We can summarise the results presented in the chapter by :

Consider $[n, k, d]_r$-Gabidulin code over $GF(q^m)$. Errors of rank $t \leq \lfloor (d-1)/2 \rfloor$ can be corrected with complexity

- *Extended Euclidian*, [Gab85b] : $\approx n(d-1) + t^3/2 + t^2 + d^2/4$ multiplications in $GF(q^m)$.
- *Berlekamp-Massey* like, [RP04b, RP04a] : $\approx n(d-1) + 6t^2 + t^3/2$ multiplications in $GF(q^m)$.
- *Linear system solving*, [Gab91, Rot91] : $\approx n(d-1) + t^3$ multiplications in $GF(q^m)$.
- *Welch-Berlekamp* like, [Loi06] : $\approx 2n^2 - k^2 + kt$ multiplications in $GF(q^m)$.

Chapter 5. This chapter is essentially devoted to a research work achieved with E. M. Gabidulin on the structure of subcodes of Gabidulin codes, see [GL00, GL04, GL05]. Since subfield subcodes or subspace subcodes of Generalised Reed-Solomon codes have rather interesting properties like the fact that the structure of the parent code is not easily recoverable, we were interested in studying structural properties of subfield subcodes or subspace subcodes of Gabidulin codes. It so happens that the projection in a subfield of a Gabidulin code does not scramble the structure of the codes. Namely proposition 20 constructs a rank preserving isomorphism between subspace subcodes and Gabidulin codes with smaller parameters. This isomorphism makes it possible to design a systematic encoding-decoding procedure for subspace subcodes with small complexity, which would be problematic in Hamming metric.

More specifically when one considers subfield subcodes, theorem 4 establishes that when the extension degree is equal to the length of the code, a subfield subcode is equivalent to a direct sum of Gabidulin codes over the considered subfield, modulo the action defined by the general linear group over the base field.

In a final section we describe the family of reducible rank codes (denoted by RRC). These codes play an important role in cryptographic applications of rank metric. They are inspired from the structure of subfield subcodes and can be decoded by using decoders for Gabidulin codes several times, see [OGHA03].

Chapter 6. This chapter is a history of McEliece type cryptosystems based on rank metric. Theses systems originate from a paper by Gabidulin, Paramonov and Tretjakov in 1991, see [GPT91]. Since then various authors have searched on this ground on how to reduce the key-size by keeping a sufficient security against the structural attacks designed by Gibson see [Gib95, Gib96]. The different ideas involved were :

- generalise the original cryptosystem by using some left-scrambler, see [OG03],
- use the Niederreiter form of the system to control the structure of the distortion matrix, see [BL02, BL05].
- use the family of RRC-codes to scramble the structure, see [OGHA03, BL04].

In a first part I describe the different principles that oriented the design of these cryptosystems. I also show a way to improve the security of the system against some active attacks : reaction attacks and message resend attacks, results which have been published in [BL04].

Then, the principle on which is based a recent structural attack by R. Overbeck is presented. This new attack breaks most of the systems, for which the parameters are not chosen in a way to resist to this attack, see [Ove05]. It uses the fact that a Gabidulin code is *almost* stable when one applies the Frobenius automorphism of the field on the components of codewords. This weakness is then exploited into constructing a matrix from the public-key for which the right kernel is non-zero. In the case where it is of dimension exactly 1 it is possible to show that a decoder for the public code can be recovered in polynomial-time. This is what is proved in proposition 25.

In a final section we show what are sufficient conditions for the system to resist these attacks. The key-size must therefore be significantly increased rendering thus the system less interesting for use as an encryption tool.

Chapter 7. This chapter presents an alternative way of designing a public-key encryption scheme for rank metric. This system is inspired from a still-born encryption scheme published at the international conference EUROCRYPT 2003, see [AF03]. The security of the latter relies on the fact that it is impossible to list-decode a Reed-Solomon code beyond the Johnson bound. In the case of rank metric the system is based on the assumption that it is impossible to decode beyond the error-correcting capability of a Gabidulin code. Namely, no equivalent of Sudan algorithm exists in that case.

The public-key of the system is given by

$$\mathbf{K}_{\text{pub}} = \mathbf{c} + \mathbf{E},$$

where $\mathbf{c}$ is a Gabidulin codeword over $GF(q^m)$ and $\mathbf{E}$ is an error-vector of rank greater than the error-correcting capability of the code. Encrypting a message consists in encoding a vector in a code derived from the Gabidulin code and from the public-key. To this codeword, an error-vector of convenient rank is added to form the ciphertext.

After the design we show some kind of attacks that can be imagined. For instance ciphertext-only attacks imagined on the same model as the attacks designed by Coron in the case of the original cryptosystem. However, from the properties of the metric we show that we can protect the system by choosing convenient parameters. Inspired from

Overbeck's approach in the case of McEliece type cryptosystem, we design a structural attack by showing how to transform the problem of recovering the key into an instance of solving a GPT cryptosystem. However, although both constraints should be taken into account, it is still possible to design a secure against these attacks. Part of the results obtained in this chapter were published at WCC 2005, see [FL06].

Axes de recherche

L'étude de familles de codes correcteurs à des fins d'utilisation dans la conception de cryptosystèmes à clé publique constitue le moteur de ma recherche. Durant ma thèse, plus de deux bons tiers du temps consacré à la recherche le fut pour l'étude de la structure des codes de Goppa qui forment le cœur du cryptosystème de McEliece [McE78]. D'une part, avec N. Sendrier mon directeur de thèse, j'ai étudié l'application de l'algorithme de séparation des support à la cryptanalyse du système de McEliece. Cet algorithme permet de retrouver la permutation entre deux codes équivalents pourvu que leur *hull*[1] soit de petite dimension, cf. [Sen00]. Des simulations effectuées sur les codes de Goppa binaires dont le polynôme générateur était choisi aléatoirement me permirent de montrer que la distribution des dimensions de leur *hull* était identique à celle des codes binaires aléatoires, cf. [Sen97, Loi97].

D'autre part, nous avons montré comment, en utilisant ce même algorithme, construire la première attaque structurelle contre le cryptosystème de McEliece. D'abord nous avons exhibé une famille de clés faibles aisément identifiables (ce sont les codes de Goppa dont le polynôme générateur est à coefficients dans un sous-corps du corps support) puis nous avons conçu une attaque par structure réalisable sur les paramètres proposés par McEliece dans le cas où les clés faibles dont le polynôme générateur est à coefficients binaire sont utilisées, cf. [LS01]. La structure de ces clés faibles me conduisit à étudier plus particulièrement le groupe d'automorphisme des codes correspondants. J'ai montré que celui-ci contenait le sous-groupe engendré par l'automorphisme de Frobenius de l'extension. A partir de ces propriétés, j'ai construit une famille de nouveaux codes, appelés codes s-projetés issus de la projection du sous-code idempotent. Grâce à cette projection, j'ai pu en déduire des bornes sur la dimension des codes s-projetés, ainsi que sur leur distance minimale, de même qu'un algorithme de décodage en temps polynomial [Loi01].

En prenant en compte le fait que le groupe d'automorphismes des codes de Goppa contient l'automorphisme de Frobenius, j'ai élaboré une version du système de McEliece permettant d'augmenter le nombre d'erreurs que l'on pouvait décoder en les plaçant selon un schéma particulier. Ainsi, pour une taille de clé constante, la sécurité vis-à-vis des attaques par décodage pouvait être renforcée. Cependant, K. Kobara et H. Imai montrèrent comment, par des transformations convenables sur la clé publique, on pouvait se ramener à un cas proche du cas initial, cf. [Loi00, KI03].

C'est vers la fin de ma thèse que j'ai commencé à m'intéresser à une classe de cryptosystèmes fondée sur des propriétés de codes décodables en métrique rang[2] [GPT91], sujet qui allait devenir mon axe de recherche principal. L'utilisation de cette métrique permet de concevoir des cryptosystèmes fondés sur les codes avec des tailles de clés réduites de l'ordre de quelques milliers de bits, à comparer aux centaines de milliers de bits du système

[1]intersection d'un code et de son dual
[2]les codes de Gabidulin

de McEliece original. En effet, les algorithmes impliqués dans le décodage de codes en métrique rang ont, à paramètres fixés, des complexités plus importantes que leurs équivalents en métrique de Hamming (on peut notamment comparer les complexités intervenant dans [Bar98, CC98] et dans [CS96, OJ02]). La première utilisation de la métrique rang en cryptographie remonte à un article de Gabidulin, Paramonov et Tretjakov de 1991 [GPT91], dans lequel les auteurs présentent un système de type McEliece utilisant des codes de Gabidulin destructurés au moyen d'une matrice de distorsion. Récemment, avec T. Berger, nous avons élaboré une version du système de Niederreiter pour la métrique rang. Nous avons montré comment le choix de sous-codes convenables *destructurait* suffisamment les familles de codes permettant ainsi de sécuriser le système contre les attaques de Gibson, les seules jusqu'alors connues, cf. [Nie86, Gib95, Gib96, BL00].

Depuis, j'ai poursuivi l'étude des propriétés algébriques de la métrique rang en gardant à l'esprit l'objectif de développer des outils pour concevoir de nouveaux cryptosystèmes et améliorer ceux existants. Toujours avec T. Berger, nous avons étudié l'effet de l'emploi des sous-codes sur le contrôle de la résistance des cryptosystèmes contre les attaques par structure, cf. [BL02, BL05]. Cette approche dépassant le simple cadre de la métrique rang, nous avons également regardé comment l'appliquer à la famille des codes de Reed-Solomon généralisés (codes GRS) qui forment le pendant en métrique de Hamming des codes de Gabidulin. Comme ils sont fortement structurés, il s'avère indispensable de masquer cette structure, en considérant par exemple la famille des sous-codes. Cela nous a ainsi permis de concevoir des cryptosystèmes et de proposer des jeux de paramètres pour lesquels ces systèmes étaient résistants aux attaques connues et pour lesquels la clé publique était de taille significativement plus petite que les systèmes existants. Cette approche a suscité un certain intérêt car depuis, de nouvelles attaques contre ces systèmes ont été publiées, cf. [Our03, Wie06].

Cette façon de masquer la structure des codes de Gabidulin en considérant leurs sous-codes me conduisit à étudier leurs propriétés structurelles. Ce travail fut mené en commun avec E. M. Gabidulin. Dans un premier temps nous nous sommes consacrés aux *sous-codes trace*, qui correspondent au sous-ensemble du code formé par les mots à coefficients dans un sous-corps de l'alphabet. Dans un second temps, nous nous sommes intéressés de manière plus générale aux sous-codes sur les sous-espaces (l'alphabet du code est alors vu comme un espace vectoriel) cf. [GL04, GL05]. Cette démarche me paraissait intéressante essentiellement pour deux raisons :

- D'une part, dans le cas de la métrique de Hamming, la *bonne* famille de codes intervenant dans la conception des systèmes de chiffrement (les codes de Goppa binaires) est une famille de sous-codes des codes de Reed-Solomon généralisés (codes GRS). Dans ce cas précis, le choix des mots à coefficient dans sous-corps destructure le code suffisamment pour que les attaques contre les systèmes utilisant les codes GRS ne puissent plus s'appliquer[3]. En conséquence, il était intéressant de se demander également quelle était l'influence, sur la structure des code de Gabidulin, du choix des mots à coefficients dans un sous-corps ou bien dans un sous-espace de l'alphabet.
- D'autre part, du point de vue du décodage, cela permet de construire de nouvelles familles de codes disposant d'un algorithme de décodage en temps polynomial et pour lesquelles on déduit un certain nombre de bornes sur les paramètres à partir du *code parent*.

Contrairement au cas des codes RS ou bien des codes GRS pour lesquels l'obtention de la

[3]Si on écrit les équations satisfaites, on se retrouve à devoir résoudre des systèmes non-linéaires de haut degré

dimension exacte ou bien de la distance minimale exacte d'un sous-code reste un problème ouvert, nous sommes parvenus à montrer que les *sous-codes* sur des sous-espaces de codes de Gabidulin sont en fait isomorphes à des codes de Gabidulin avec des paramètres plus petits. Cela revient à dire que la projection du code sur le sous-espace n'altère pas sa structure. Cet isomorphisme est calculable en temps polynomial, conduisant ainsi à l'élaboration d'algorithmes spécifiques de décodage de ces codes. Dans le cas plus particulier des sous-codes sur des sous-corps ou *sous-codes trace* nous avons montré que, quand le degré de l'extension est égal à la longueur du code, tout *sous-code trace* est le produit direct de codes de Gabidulin de taille plus petite. De plus, celui-ci est caractérisé de manière unique par une isométrie linéaire de la métrique, cf. [GL00, GL04, GL05]. Ainsi, l'utilisation de tels sous-codes dans la conception de cryptosystèmes ne plaide pas vraiment pour le renforcement de la sécurité en comparaison des systèmes utilisant les codes de Gabidulin proprement dits.

L'utilisation de *sous-codes trace* de codes Gabidulin ne semblant pas opportune dans la conception de cryptosystèmes, A. V. Ourivski et al. conçurent une famille de codes semblant plus adaptée : les codes rang réductibles (codes RRC), cf. [OGHA03]. Dans ce même article ils décrivent un cryptosystème de type McEliece les utilisant. Avec T. Berger nous avons mis en œuvre ce système fondé sur le choix de codes RRC d'ordre 2, cf. [BL04]. Pour celui-ci, nous avons également développé une procédure permettant d'une part de renforcer la sécurité du système contre les attaques *par rejeu* et *par réaction* et, d'autre part, d'augmenter le taux de transmission du système en plaçant de l'information dans le vecteur d'erreur. Pour y parvenir, nous nous sommes servis d'une fonction de hachage *parfaite* ainsi que les boîtes S de l'AES. Dans le modèle de *l'oracle aléatoire*, nous sommes parvenus à prouver une sécurité sémantique du système contre les attaques actives précitées.

Lors de la mise en place effective des cryptosystèmes fondés sur les codes, le facteur limitant les performances est bien souvent la complexité de l'algorithme de décodage de la famille de codes. Jusqu'à une date récente, les algorithmes de décodage des codes de Gabidulin et de leurs dérivés étaient tous de complexité cubique en fonction du rang de l'erreur, cf. [Gab85b, Rot91, Gab91, RP04b]. Malgré une version récente utilisant un algorithme du type Berlekamp-Massey réduisant quelque peu la complexité, le terme cubique demeurait du fait de l'absence d'une algèbre matricielle efficace pour inverser certaines matrices structurées. Or, les codes de Gabidulin sont des des codes d'évaluation de polynômes linéaires, et leurs algorithmes de décodages sont calqués sur les algorithmes de décodage de codes RS. Cette similitude m'a conduit à utiliser les propriétés des polynômes linéaires (cf. [Ore33, Ore34]), afin de définir le problème de *reconstruction de polynôme linéaire*, puis d'établir le lien entre la résolution de ce problème et la résolution du problème de *décodage en liste* de codes de Gabidulin, cf. [Loi04]. Quand la taille de la liste est réduite à un élément, ce qui est le cas du décodage jusqu'à la capacité de correction, on peut construire un algorithme de décodage des codes en s'inspirant de l'algorithme de Welch et Berlekamp pour les codes RS, cf. [BW86]. J'ai ainsi conçu le premier algorithme de décodage en temps quadratique de codes de Gabidulin [Loi06].

Toujours dans le cadre de l'étude des performance du décodage de codes de Gabidulin en métrique rang, avec R. Overbeck, j'ai commencé un travail sur le décodage des codes de Gabidulin dont le vecteur générateur prend ses valeurs dans un sous-corps du corps formant l'alphabet du code, cf. [LO06]. En s'inspirant des algorithme de décodage de codes RS entrelacés (cf. [BKY03] et [SRB06]), nous avons montré que l'on pouvait augmenter significativement le rang de l'erreur que l'on peut corriger. Nous avons construit deux algorithmes, l'un déterministe sur un certain jeu de paramètres, et le second probabiliste mais plus général.

Le lien entre les problèmes de décodage de codes de Gabidulin et de reconstruction de polynômes linéaires, ainsi que la complexité de leur résolution dès que le rang de l'erreur est suffisamment grand nous ont amenés avec C. Faure à élaborer un système de chiffrement reposant sur le problème de la *reconstruction des polynômes linéaires*, cf. [Fau04, FL06]. Nous nous sommes inspirés d'un système similaire pour la métrique de Hamming dont la sécurité structurelle repose sur le problème dit de *reconstruction de polynômes* et dont la sécurité contre les attaques par décodage repose sur la difficulté de décoder dans un sur-code d'un code RS. Il s'agit du cryptosystème Augot-Finiasz qui fut publié en 2003 à la conférence internationale EUROCRYPT, cf. [AF03]. Peu après il fut cryptanalysé, d'abord par J.-S. Coron, cf. [Cor04] qui, en utilisant des techniques de linéarisation inspirée de l'algorithme de décodage de Welch-Berlekamp a montré que, décoder dans un *sur-code* pouvait se faire en général assez facilement. Pour ma part, j'ai montré que l'une des propositions faites dans l'article d'utiliser des sous-corps pour réduire la taille de la clé conduisait à une cryptanalyse très simple et, si le corps était suffisamment petit, souvent bien plus rapide que celle proposée par J.-S. Coron [Loi05]. Pour tenter de réparer le système, avec D. Augot et M. Finiasz, j'ai proposé de le faire évoluer en considérant des *sur-codes* de dimension plus grande que 1, cf. [AFL03]. Cependant, cette approche ouvrait la voie à un autre type de cryptanalyse qui démontrait la quasi-impossibilité de concevoir un système qui permettrait de résister aux deux attaques et ce, quels que soient les paramètres choisis, cf. [Cor04, KY04].

Dans la conception d'un système similaire utilisant les polynômes linéaires et la métrique rang, nous avons pris en compte ces remarques et construit le système en même temps que nous considérions les transpositions des attaques contre le système d'origine. L'une des attaques liées à la structure des codes pouvait facilement se transposer tandis qu'une autre ne se transposait pas. En effet, cette dernière repose sur des propriétés spécifiques de la métrique, cf. [Fau04, FL06].

Si l'essentiel de ma recherche reste consacrée à l'étude et à la conception de systèmes cryptographiques à clé publique fondés sur la théorie des codes correcteurs, cela n'en est pas l'unique objet.

Depuis 2003, avec B. Sakkour, j'ai entamé une recherche sur l'étude et l'amélioration d'algorithmes de décodage d'une famille de codes très populaire : Les codes de Reed-Muller. Bien que la distance minimale de ces codes soit exponentiellement petite en l'ordre du code, il existe un ensemble d'algorithmes de décodage probabilistes qui permettent de décoder, pour un ordre fixé, au-delà de la capacité de correction avec une probabilité d'échec tendant vers 0 quand la longueur du code tend vers $+\infty$, cf. par exemple les algorithmes de décodage récursif, cf. [DK00, DK02, Dum04]. Nous nous sommes intéressés plus particulièrement à l'algorithme de décodage des codes de Reed-Muller d'ordre 2 de V. M. Sidel'nikov et A. S. Pershakov, cf. [SP92]. Nous avons montré comment améliorer de manière pratique son efficacité en décodage sans augmenter sa complexité, cf. [LS04]. Bien qu'asymptotiquement tous les algorithmes de décodage décodent la même proportion d'erreurs, nous sommes parvenus à montrer que, pour des codes de longueurs moyennes l'amélioration que nous avions présentée permettait de corriger bien plus d'erreurs que les algorithmes par décodage récursif plus rapides en pratique puisqu'ils sont de complexité quasi-linéaire en la longueur du code. Toutefois cette méthode ne permet pas d'atteindre la borne de décodage à maximum de vraisemblance des codes de Reed-Muller d'ordre 2 qui est donnée dans [HKL03].

La motivation originelle de ce travail réside dans le cryptosystème de Sidel'nikov, cryptosystème de type McEliece qui utilise comme clé publique un code de Reed-Muller permuté, et comme algorithme de décodage un algorithme conçu par V. M. Sidel'nikov et A. S. Pershakov, cf. [Sid94]. Comme les codes de Reed-Muller sont des codes faiblement auto-duaux[4] et que l'algorithme de séparation des supports de N. Sendrier est de complexité exponentielle en la dimension du *hull* du code , retrouver une permutation permettant de décoder est un problème difficile pour un attaquant, cf. [Sen00]. Cependant, si l'on considère uniquement leurs performances de décodage jusqu'à la capacité de correction, les codes de Reed-Muller constituent de piètres candidats quant à la taille de la clé nécessaire pour assurer une bonne sécurité contre les attaques par décodage. C'est pour cette raison que Sidel'nikov a utilisé un algorithme de décodage probabiliste qui permet de décoder bien au-delà de la capacité de correction avec une probabilité d'erreur tendant vers 0 quand la longueur tend vers $+\infty$, et c'est également une des raisons de l'intérêt manifesté pour les codes de Reed-Muller de longueur moyenne.

Dans un autre sujet, avec V. Shorin, venu en post-doctorat, j'ai commencé l'étude et la recherche de séquences unimodulaires parfaites, c'est-à-dire de suites finies de nombres complexes de module 1 en général, ou bien de module au moins borné, ayant les plus faibles autocorrélations possible. Les séquences sont des objets qui présentent un intérêt pour la synchronisation des communications par exemple. Pratiquement, pour en obtenir, on recherche les suites les plus longues possibles, avec l'alphabet le plus petit possible. Mathématiquement, la résolution se décrit très facilement en écrivant le système d'équations non-linéaires à coefficients entiers qui est contraint par une condition, par exemple l'unimodularité des inconnues. A paramètres fixés, simplement décider de l'existence ou bien de la non-existence de telles suites est problématique. Dans certains cas, on peut en construire à partir de séquences plus courtes en appliquant le théorème des restes chinois. En utilisant des stratégies de réécritures des systèmes non-linéaires et des algorithmes de résolution des systèmes par des méthodes utilisant des bases de Groebner d'idéaux[5], nous sommes parvenus à construire de nouvelles familles de séquences, dont une famille infinie de séquences à 6 phases qui ne peuvent pas s'obtenir à partir de la construction standard s'appuyant sur le théorème des restes chinois, cf. [SL05].

[4] l'intersection d'un code et de sont dual est égal au plus petit des deux codes
[5] algorithme F4 de J.-C. Faugère implanté dans le logiciel MAGMA [MAG]

Motivation et composition du présent document

Ce document a été construit sur la majeure partie de ma recherche de la fin de thèse jusqu'à aujourd'hui et qui concerne l'étude de la métrique rang, les propriétés des codes en métrique rang ainsi que la construction et l'étude de la sécurité des cryptosystèmes à clé publique dans cette métrique.

Le foisonnement et la dispersion dans le temps et l'espace des articles traitant de la métrique rang, notamment les problématiques concernant la sécurité effective des schémas de chiffrement utilisant cette métrique, autant que l'absence d'un cadre d'étude bien défini m'ont conduit à élaborer ce document en vue de passer une habilitation à diriger des recherches. Par ce document, j'ai cherché à donner une vision plus unifiée des propriétés de la métrique du point de vue de la théorie de l'information (bornes sur les codes, codes aléatoires, codes parfaits,. . .) ainsi que des relations et propriétés de la famille des codes de Gabidulin en gardant en point de mire les applications cryptographiques existantes. Il ne prétend pas, loin s'en faut, à l'exhaustivité en ce qui concerne les applications potentielles de cette métrique et des codes qui peuvent se trouver dans des domaines aussi divers que le codage *espace-temps*, la conception de schémas d'authentification à *divulgation nulle de connaissance*, ou bien encore la construction de fonctions de hachage pour des MAC (*Message Authentification Code*). Le lecteur désireux d'en savoir plus à ce sujet pourra se référer aux articles suivants [LK05, Che96, SNC05].

Il est constitué pour une bonne partie à partir de résultats que j'ai obtenus et déjà publiés hormis les résultats du chapitre 1, qui lui fait l'objet d'un article soumis et l'attaque structurelle sur la clé publique du cryptosystème du chapitre 7 (section 2.1). Quand il me le semblait nécessaire pour la structure du document j'ai reformulé des résultats existants en présentant une approche plus adaptée avec la structure du document. C'est le cas du chapitre 2 qui décrit les problèmes généraux du décodage en métrique rang, du chapitre 3 sur les polynômes linéaires et des attaques d'Overbeck contre le système GPT qui sont présentées au chapitre 6.

Le document se compose de trois parties. Autant que faire se peut, chaque chapitre dispose d'une introduction donnant les références bibliographiques existantes qui conduisirent à sa rédaction. De même, une section finale les clôt en donnant un ensemble de problématiques de recherche qu'il m'a semblé intéressant de considérer à partir des résultats du chapitre. Il s'agit fréquemment de problèmes pour lesquels une solution en métrique de Hamming existe effectivement et dont la méthode de résolution ne se transpose pas en métrique rang.

La première partie s'attache à établir des propriétés générales de la métrique ainsi que de définir les notions classiques de décodage de codes, et les problèmes auxquels ils sont reliés. Le premier chapitre est consacré à une mise en place de la métrique rang. Après avoir défini la métrique, on établit un équivalent de la borne d'empilement de sphères (*sphere packing bound*) et de la borne de Varshamov-Gilbert. Un résultat significaPtif réside dans la preuve de la *non-existence* de codes parfaits en métrique rang. Dans le second chapitre on définit des problèmes relatifs au décodage des codes en métrique rang. Je décris la méthode sous-tendant l'algorithme le plus efficace de recherche de mot de rang fixé dans un code. Celui-ci permet de résoudre le problème du *décodage borné* par la capacité de correction. A paramètres égaux, on constate que sa complexité est bien supérieure à celle des algorithmes résolvant le même problème en métrique de Hamming. Dans un troisième chapitre enfin on introduit l'anneau non-commutatif des polynômes linéaires définis par Øre en 1933. Cette anneau constitue le *bon* espace de polynômes permettant de caractériser les problèmes de décodage en métrique rang.

Une fois les problématiques de la métrique et des codes mises en place, la seconde partie est consacrée à la présentation de familles de codes optimaux : la famille des codes de Gabidulin, ainsi que les familles de codes qui en dérivent. Dans le premier chapitre de la partie on définit les codes de Gabidulin et on présente leurs propriétés. On décrit également le principe des algorithmes de décodage par syndrome qui sont de complexité cubiques, afin de mettre en exergue leurs différences par rapport à l'algorithme de décodage par reconstruction de polynôme linéaire de complexité quadratique. On décrit en détail cet algorithme qui est inspiré du modèle de l'algorithme de Berlekamp-Welch décodant les codes RS par résolution partielle du problème de la reconstruction des polynômes. Le second chapitre reprend un travail que nous avons effectué sur plusieurs années concernant la structure des sous-codes sur des sous-espaces de codes de Gabidulin. A la différence des codes RS, on montre que tous les paramètres fondamentaux se déduisent aisément à partir des paramètres fondamentaux des *codes parents*, et même que ces sous-codes sont isomorphes à des codes de Gabidulin ayant des paramètres plus petits. On montre que dans certains cas, on peut décoder au delà de la capacité de correction des sous-codes.

Dans une troisième et dernière partie, on présente les cryptosystèmes à clé publique fondés sur des problèmes difficiles de la métrique rang. Le premier chapitre de la partie présente une version plus générale et plus récente du cryptosystème GPT originel qui convient mieux à la description et à l'analyse. On y décrit ensuite des variantes du système : L'une utilisant la forme cryptosystème de Niederreiter utilisant des sous-codes pour masquer la structure, l'autre utilisant les codes RRC d'ordre 2. On décrit ensuite une procédure assez générale permettant de rendre le système résistant aux attaques *par réaction* et *par rejeu*. Ensuite, on décrit une version simplifiée de l'approche d'Overbeck qui, s'appuyant sur le fait que les codes de Gabidulin sont *quasi-stables* sous l'action de l'automorphisme de Frobenius, a permis de cryptanalyser les systèmes de chiffrement. Le second chapitre présente une version du cryptosystème Augot-Finiasz dont la sécurité repose sur le problème de reconstruction de polynômes linéaires. Après avoir décrit ce système, on s'attache à analyser la complexité des attaques existantes transposables à partir de la métrique de Hamming. On présente également une nouvelle attaque reliée à des propriétés structurelles de la clé publique. Enfin, on propose des paramètre pour le système permettant de les prévenir.

Notations

→ *Concernant les corps finis :*

- $GF(q^m)$, le corps fini à q^m éléments, q puissance d'un nombre premier.
- Le nombre d'espaces vectoriels de dimension t dans $GF(q)^m$ est donné par le binôme de Gauss :
$$\begin{bmatrix} m \\ t \end{bmatrix}_q = \frac{\prod_{j=0}^{t-1} q^m - q^j}{\prod_{j=0}^{t-1} q^t - q^j}.$$
- Pour $\mathbf{x} \in GF(q^m)^n$, $\mathrm{Rg}(\mathbf{x}|GF(q))$ ou bien $\mathrm{Rg}(\mathbf{x})$ s'il n'y a pas d'ambiguïté : Le rang du vecteur $\mathbf{x} = (x_1, \ldots, x_n) \in GF(q^m)$ sur le corps $GF(q)$ qui correspond au rang de la matrice obtenue étendant chaque x_i en colonne suivant ses coordonnées sur une base de $GF(q^m)/GF(q)$
- Soit $\mathbf{g} = (g_1, \ldots, g_n)$ un vecteur à coefficient dans un corps fini et soit P un opérateur (par exemple un polynôme ou bien un q-polynôme), alors on note
$$P(\mathbf{g}) = (P(g_1), P(g_2), \ldots, P(g_n))$$
- Si s divise m, alors $GF(q^s) \subset GF(q^m)$ et
$$\mathrm{Tr}_{m/s}(z) = \sum_{i=0}^{u-1} z^{q^{[si]}}$$
désigne l'*opérateur Trace* de $GF(q^m)$ dans $GF(q^s)$.

→ *Concernant les codes en général :*

- Code $[n, k, d]$: Un code linéaire de longueur n, de dimension k et de distance minimale d.
- Code (n, M, d) : Un code pas forcément linéaire de longueur n, de cardinal M, de distance minimale d.
- Code $[n, k, d]_r$: Un code linéaire de longueur n, de dimension k et de distance *rang* minimale d.
- Code $(n, k, d)_r$: Un code non nécessairement linéaire de longueur n, de dimension k et de distance *rang* minimale d.
- $Gab_k(\mathbf{g})$: Le code de GABIDULIN de dimension k de vecteur générateur $\mathbf{g}$, voir la définition 6.

Première partie

Généralités sur la métrique

Chapitre 1

Propriétés de codes en métrique rang

Sur les espaces à alphabet fini que sont par exemple les corps finis, on peut définir différentes métriques. Une métrique très commune, de par son importance pratique est la métrique de Hamming qui dénombre les coordonnées non nulles d'un vecteur, et qui est la *bonne* métrique lorsqu'on fait du décodage *dur* de mots passés au travers d'un canal binaire symétrique. Une autre métrique assez répandue est la métrique de Lee qui traite les erreurs de synchronisation, cf. [MRS98]. D'autres types de métriques utilisant des propriétés combinatoires peuvent être définies, cf. [GS98]. En correction d'erreur la *métrique rang* fit son apparition en 1985, cf. [Gab85b]. Cette métrique convient à un modèle de canal où les mots de code peuvent être vus comme des matrices à coefficients dans un corps fini et où les erreurs arrivent en bloc sur des lignes ou bien sur des colonnes. C'est le cas par exemple du stockage de données sur bande magnétique, cf. [Rot91, Rot96]. Depuis une date plus récente, la métrique rang se retrouve aussi reliée à la théorie du codage *espace-temps* dans la construction de codes dont le paramètre dénommé *diversité* est obtenu à partir de codes matriciels binaires dont la distance rang minimale est connue, cf. [LK05, Ham06].

Ce chapitre est constitué de résultats déjà anciens concernant la métrique rang que l'on peut trouver dans l'article originel de E. M. Gabidulin, cf. [Gab85b], ainsi que de résultats non-encore publiés concernant l'existence ou la non-existence de codes à paramètres déterminés en métrique rang.

Dans un premier temps, on établit des propriétés de la métrique et on définit la notion de distance rang minimale. On rappelle l'équivalent de la borne de Singleton et de la borne d'*empilement de sphères* (*sphere-packing bound*). On montre ensuite qu'il n'existe pas de codes parfaits en métrique rang. Puis nous établissons un équivalent de la borne d'existence de Varshamov–Gilbert.

Finalement, nous étudions la classe des codes optimaux pour la métrique rang (appelés codes MRD pour *Maximum rank distance codes*), dont la distribution des poids rang est connue, et se superpose avec celle de la répartition des mots d'un code aléatoire. Nous en déduisons que dans le cas où la longueur d'un code MRD est égale au degré du corps constituant l'alphabet, la densité de l'espace couvert par des boules centrées en les mots du code et de rayon égale à la capacité de correction (volume de l'espace *correctible*) dépend presque uniquement de la distance rang minimale du code.

1 Propriétés de la métrique rang

On suppose fixée une base $\mathbf{b} = (\beta_1, \ldots, \beta_m)$ de $GF(q^m)$ sur $GF(q)$.

Définition 1 ([Gab85b])

Soit $\mathbf{x} = (x_1, \ldots, x_n) \in GF(q^m)^n$. *On appelle rang de* $\mathbf{x}$ *sur* $GF(q)$, *le rang de la matrice* $\mathbf{X} = (x_{ij})$, *où* $x_j = \sum_{i=1}^{m} x_{ij}\beta_i$. *On le note* $\mathrm{Rg}(\mathbf{x}|GF(q))$, *ou plus simplement* $\mathrm{Rg}(\mathbf{x})$ *lorsqu'il n'y a pas d'ambiguïté.*

Le rang d'un vecteur est indépendant de la base choisie. Avec cette définition, il est immédiat que la métrique rang est plus grossière que la métrique de Hamming, en ce sens qu'elle discrimine moins les vecteurs. En effet, pour tout vecteur $\mathbf{x} \in GF(q^m)^n$, on a $\mathrm{Rg}(\mathbf{x}) \leq wt(\mathbf{x})$ où $wt(\mathbf{x})$ désigne le poids de Hamming du vecteur $\mathbf{x}$. D'autre part, cette métrique prend tout son sens sur des extension de corps plutôt que sur des corps premiers, puisque tout vecteur non-nul à coefficient dans un corps premier est de rang exactement égal à 1.

La taille des boules est donnée par la proposition suivante, obtenue en comptant le nombre de matrice de taille $m \times n$ de rang $\leq t$ à coefficients dans $GF(q)$. Pour la démonstration de la proposition, on peut se référer par exemple au livre [LN97]

Proposition 1

Soit $\mathbf{c}$ *un vecteur de longueur* n *à coefficients dans* $GF(q^m)$. *Soit* $\mathcal{S}(\mathbf{c}, t)$, *la sphère centrée en* $\mathbf{c}$ *de rayon* t *et soit* $\mathcal{B}(\mathbf{c}, t)$ *la boule de centre* $\mathbf{c}$ *et de rayon* t. *On a*

- $\mathcal{S}(\mathbf{c}, t) = \left(\prod_{j=0}^{t-1} (q^n - q^j)\right) \begin{bmatrix} m \\ t \end{bmatrix}_q$,
- $|\mathcal{B}(\mathbf{c}, t)| = \sum_{i=0}^{t} |\mathcal{S}(\mathbf{c}, i)|$.

On peut montrer, que le volume de la sphère de rayon t est borné par :

$$q^{(m+n-2)t-t^2} \leq |\mathcal{S}(\mathbf{0}, t)| \leq q^{(m+n+1)t-t^2}. \tag{1.1}$$

On en déduit un encadrement de la boule de rayon t :

$$|\mathcal{S}(\mathbf{0}, t)| \leq |\mathcal{B}(\mathbf{0}, t)| = \sum_{i=0}^{t} |\mathcal{S}(\mathbf{0}, i)| \leq q^{(m+n+1)t-t^2+1}. \tag{1.2}$$

Ce encadrements sont relativement grossier, mais demeurent suffisants pour la suite.

2 Codes en métrique rang

Un code $\mathcal{C}$ est un ensemble de vecteurs de longueur n à coefficients dans un alphabet qui sera soit un corps fini $GF(q^m)$, soit un sous-espace d'un corps fini (cf. le chapitre 5). La *distance rang minimale* du code $\mathcal{C}$, se définit sur le même modèle que la distance minimale habituelle d'un code.

Définition 2 ([Gab85b])

Soit $\mathcal{C}$ *un code sur* $GF(q^m)$.

- *La quantité* $d \stackrel{def}{=} \min_{\mathbf{c}_1 \neq \mathbf{c}_2 \in \mathcal{C}}(\mathrm{Rg}(\mathbf{c}_1 - \mathbf{c}_2))$ *est appelée* distance rang minimale *de* $\mathcal{C}$.
- *Si de plus* $\mathcal{C}$ *est de longueur* n *et de cardinal* M, *alors on dira que c'est un code* $(n, M, d)_r$.

Cette distance caractérise en particulier la capacité de correction du code. Dans le cas de codes additifs, on aura

$$d = \min_{\mathbf{c}\neq\mathbf{0}\in\mathcal{C}}(\mathrm{Rg}(\mathbf{c})).$$

Afin de simplifier les études suivant les différents cas, nous introduisons la notion de code transposé d'un code. Etant donnée une base $\mathcal{D} = (\gamma_1, \ldots, \gamma_n)$ de $GF(q^n)/GF(q)$, un vecteur $\mathbf{c} = (c_1, \ldots, c_n)$ à coefficients dans $GF(q^m)$ peut être considéré comme un vecteur de longueur m à coefficient dans $GF(q^n)$ par l'isomorphisme d'espace vectoriel

$$\begin{array}{ccc} GF(q^m)^n & \to & GF(q^n)^m \\ \mathbf{c} = \left(\sum_{i=1}^m c_{1i}\beta_i, \ldots, \sum_{i=1}^m c_{ni}\beta_i\right) & \mapsto & \mathbf{c}^T = \left(\sum_{j=1}^n c_{j1}\gamma_j, \ldots, \sum_{j=1}^n c_{jm}\gamma_j\right). \end{array}$$

Le vecteur $\mathbf{c}^T$ ainsi défini est appelé vecteur transposé de $\mathbf{c}$. Cet isomorphisme préserve le rang des vecteurs.

Définition 3 (Code transposé – [GP06])
Soit $\mathcal{C}$ un code $(n, M, d)_r$ sur $GF(q^m)$. On appelle code transposé de $\mathcal{C}$ noté $\mathcal{C}^T$, le code $(m, M, d)_r$ sur $GF(q^n)$ formé des transposés des mots $\mathcal{C}$, i.e.

$$\mathcal{C}^T = \{\mathbf{c}^T \mid \mathbf{c} \in \mathcal{C}\}.$$

De par la propriété de transposition de code, on obtient que le code transposé d'un code linéaire est un code additif, non nécessairement linéaire. Le fait que le code transposé ait les mêmes paramètre de taille et de distance rang minimale nous permet de ne considérer par la suite que le cas où $n \leq m$, sauf mention contraire explicite.

3 Bornes sur les codes

Bornes sur les paramètres d'un code Des propriétés de la métrique et de la distance rang minimale d'un code on déduit des bornes que doivent satisfaire les paramètres d'un code $(n, M, d)_r$.

Théorème 1 (Bornes fondamentales de la métrique rang)
Soit $\mathcal{C}$ un code $(n, M, d)_r$ sur $GF(q^m)$. Alors on a

– Borne de Singleton *:*

$$M \leq q^{\min(m(n-d+1), n(m-d+1))}.$$

– Borne d'empilement de sphères *: Si $t \overset{def}{=} \lfloor (d-1)/2 \rfloor$, alors*

$$M \times |\mathcal{B}(\mathbf{0}, t)| \leq q^{mn}, \tag{1.3}$$

Pour la démonstration de la borne de Singleton on pourra se référer aux articles [Gab85b, OGHA03]. La borne d'empilement de sphères, quant à elle, s'obtient facilement en utilisant le fait que

1. Les boules centrées en les mots du codes de rayon t sont disjointes deux à deux.
2. la réunion des volumes des boules de rayon t autour de tous les mots du code doit être inférieure au volume de l'espace vectoriel $GF(q^m)^n$.

En métrique de Hamming on définit un code *code parfait*, comme étant un code (n, M, d) dont les paramètres satisfont l'égalité $M \times |\mathcal{B}(\mathbf{0}, \lfloor (d-1)/2 \rfloor)| = q^{mn}$. Cependant on sait qu'il n'en existe pas pour tout les paramètres ni pour toutes les extensions. Il a été montré que, pour des codes à coefficients dans des corps finis, les seuls codes linéaires parfaits non triviaux sont les codes de Hamming sur n'importe quel corps d'extension et les codes de Golay binaires et ternaires, voir par exemple [MS77]. On peut donc légitimement s'interroger sur leur existence en métrique rang.

Supposons qu'il existe un tel code. La symétrie entre m et n de l'égalité impose que son code transposé est également un code parfait. On peut donc sans perte de généralité supposer que $n \leq m$.

La partie droite de l'inégalité (1.2) sur le volume des boules implique

$$Mq^{(m+n+1)t-t^2+1} \geq q^{mn}.$$

D'autre part, comme la borne de Singleton donne une borne supérieure à la taille M du code, et que $2t \leq d-1$ on a

$$q^{(m+n+1)t-t^2+1+m(n-2t)} \geq q^{mn}.$$

En prenant le logarithme en base q, et en réajustant les termes, on obtient alors que les paramètre du code doivent vérifier

$$(n-m)t \geq t^2 - t - 1.$$

Or, par construction, $n - m \leq 0$ et t est un entier supérieur ou égal à 1. Donc, pour qu'un code soit parfait, il est nécessaire que $n = m$ et $t = 1$. Mais dans ce cas, on peut calculer facilement le volume exact de la boule $\mathcal{B}(\mathbf{0}, 1)$ et l'égalité qu'il doit vérifier devient

$$\underbrace{q^{n(n-2)}}_{\text{Singleton}} \frac{q^{2n} - 2q^n + q}{q-1} \geq M \underbrace{\frac{q^{2n} - 2q^n + 1}{q-1} + 1}_{|\mathcal{B}(\mathbf{0},1)|} = q^{n^2},$$

soit nécessairement

$$1 - \frac{2}{q^n} + \frac{1}{q^{2n-1}} \geq q - 1.$$

Or, cela n'est jamais possible pour $q \geq 2$. Donc

Proposition 2 *Il n'existe pas de code parfait en métrique rang.*

La *densité d'empilement* (*packing density* en anglais) d'un code $(n, M, d)_r$ est définie comme étant le volume de l'espace ambiant couvert par les boules centrées en les mots du code et de volume égal à la capacité de correction du code, soit

$$\mathcal{D} = \frac{|\mathcal{B}(\mathbf{0}, t)| M}{q^{mn}}. \tag{1.4}$$

Nous montrerons à la section 5 que les codes optimaux en métrique rang ont une densité inférieurement bornée par une constante et, à la section 1 du chapitre 4 nous construisons une famille de codes dont la densité d'empilement tend vers 1 avec la longueur du code.

Borne d'existence de code en métrique rang La question de l'existence de codes $(n, M, d)_r$ de paramètres fixés trouve une réponse grâce à un équivalent de la borne de Varshamov–Gilbert :

Proposition 3 (Borne de Varshamov–Gilbert)
Si $M \times |\mathcal{B}(\mathbf{0}, d-1)| < q^{mn}$, alors il existe un code $(n, M+1, d)_r$ sur $GF(q^m)$.

Preuve
On fixe n et la taille M du code. La preuve se fait par récurrence sur M. Supposons avoir construit un code $(n, M, d)_r$ appelé $\mathcal{C}$. Considérons alors l'ensemble formé par la réunion des boules de rayon $d-1$ centrées sur les mots du code, soit

$$\mathcal{V} \stackrel{def}{=} \bigcup_{\mathbf{c} \in \mathcal{C}} \mathcal{B}(\mathbf{c}, d-1).$$

Alors $|\mathcal{V}| \leq M \times |\mathcal{B}(\mathbf{0}, d-1)|$. Donc si $M \times |\mathcal{B}(\mathbf{0}, d-1)| < q^{mn}$, il existe un vecteur $\mathbf{a} \in GF(q^m)^n \setminus \mathcal{V}$, qui est à distance au moins d de tous les autres mots. On constate alors que $\mathcal{C} \cup \{\mathbf{a}\}$ est un code de distance rang minimale au moins d et de cardinal $M+1$.
■

La proposition suivante nous donne un équivalent asymptotique pour un code de taille *raisonnable* de la distance de Varshamov–Gilbert :

Proposition 4
Soit un code $(n, M, d_{\mathrm{GV}})_r$ sur $GF(q^m)$ où $m \geq n$, et qui atteint la borne de Varshamov–Gilbert. Alors, on a

$$\frac{d_{\mathrm{GV}}}{m+n} \overset{n \to +\infty}{\sim} \frac{1}{2} - \frac{\sqrt{\log_q M}}{m+n} \sqrt{1 + \frac{(m-n)^2}{4 \log_q M}}$$

Pourvu que $mn \geq \log_q M = \lambda(n)(m+n)$, où $\lambda(n)$ tend vers $+\infty$ avec n.

Preuve
En utilisant les inégalités (1.1) et (1.2), on obtient comme conditions nécessaires sur d_{GV} que

$$\begin{aligned} 0 &\leq -d_{\mathrm{GV}}^2 + (m+n+3)d_{\mathrm{GV}} + \log_q M - mn - (m+n) - 2 + o(q^{-2}), \\ 0 &\geq -d_{\mathrm{GV}}^2 + (m+n+1)d_{\mathrm{GV}} + \log_q M - mn - (m+n). \end{aligned}$$

En résolvant les deux inéquations, on parvient à un encadrement de d_{GV}, qui mène au résultat souhaité dès que $\log_q M = \lambda(n)(m+n)$, où $\lambda(n)$ tend vers $+\infty$ avec n.
■

Le cas particulier où $m = n$ est intéressant puisqu'alors

$$\frac{d_{\mathrm{GV}}}{n} \overset{n \to +\infty}{\sim} 1 - \frac{\sqrt{\log_q M}}{n} = 1 - \sqrt{\frac{\lambda(n)}{n}}. \tag{1.5}$$

Donc si $\lambda(n)$ est équivalente à ϵn, où $\epsilon \leq 1$ est une constante (correspond à un taux de transmission constant), d_{GV} tend vers $1 - \sqrt{\epsilon}$. Dans le cas contraire (le taux de transmission tend vers 0), d_{GV} tend vers 1.

Remarque 1
Toutes les bornes obtenues s'inspirent de bornes existant en métrique de Hamming [MS77, PH98]. *Par rapport aux formules connues, on remplace le plus souvent le coefficient binômial dénombrant le nombre de vecteurs de poids de Hamming donné par le binôme de Gauss qui dénombre le nombre d'espaces vectoriels de dimension donnée.*

4 Codes aléatoires

Un *code aléatoire* de longueur n sur $GF(q^m)$ est un ensemble de M vecteurs de longueur n tirés uniformément et de manière indépendante dans $GF(q^m)^n$. On peut en déduire la répartition probabiliste des rangs des mots qui le composent.

Proposition 5 (Code aléatoire)
Soit $\mathcal{C}$ un code $(n, M, d)_r$ aléatoire. Alors, le nombre $\mathcal{A}_i$ de mots de rang i dans $\mathcal{C}$ vérifie en moyenne

$$E(\mathcal{A}_i) = \frac{M \times \sum_{i=0}^{t} \prod_{j=0}^{i} (q^n - q^i) \begin{bmatrix} m \\ i \end{bmatrix}_q}{q^{mn}} = \frac{M \times |\mathcal{S}(\mathbf{0}, i)|}{q^{mn}}.$$

Preuve
Pour prouver la proposition, on doit évaluer la probabilité qu'une matrice de taille $m \times n$ sur $GF(q)$ soit de rang i, que l'on multiplie par le nombre de mots du codes. Or le nombre de telles matrices est exactement $|\mathcal{S}(\mathbf{0}, i)|$.
■

5 Codes optimaux

En métrique de Hamming, on a une notion d'optimalité de codes donnée par la borne de Singleton. Les codes qui atteignent cette borne sont appelés codes MDS, [MS77]. En métrique rang, suivant le même principe, on définit la notion d'optimalité de codes.

Définition 4 (Codes MRD)
Soit un code $(n, M, d)_r$ sur $GF(q^m)$.
- *Si $n \leq m$ et si $M = q^{m(n-d+1)}$, on dira que le code est MRD, pour* Maximum Rank Distance.
- *Si $n > m$ on dira qu'il est MRD si son code transposé est MRD.*

Distribution des rangs De même que pour les codes MDS, la distribution des rang des mots d'un code MRD est connue MRD.

Proposition 6 ([Gab85b])
Soit $A_s(n, d)$, le nombre de mots de rang s d'un code MRD sur $GF(q^m)$, alors

$$A_{d+\ell}(n, d) = \begin{bmatrix} n \\ d+\ell \end{bmatrix}_q \sum_{t=0}^{\ell} (-1)^{t+\ell} \begin{bmatrix} d+\ell \\ \ell+t \end{bmatrix}_q q^{\binom{\ell-t}{2}} \left(q^{m(t+1)} - 1\right). \tag{1.6}$$

Exemple 1
Cette quantité n'est pas facile à manipuler, aussi, nous avons procédé à des simulations, qui montrent que la répartition des mots du code. Le tableau 1 donne le logarithme en bases 2 de la proportion de mots de rang d dans un code MRD longueur $n = 32$ sur un alphabet de taille $m \geq 32$.La courbe la plus à gauche correspond à $m = n = 32$, tandis que les courbes plus à droite correspondent à des degrés d'extension plus élevés. Plus le degré d'extension m est élevé, plus la proportion de mot de rang élevé est importante.

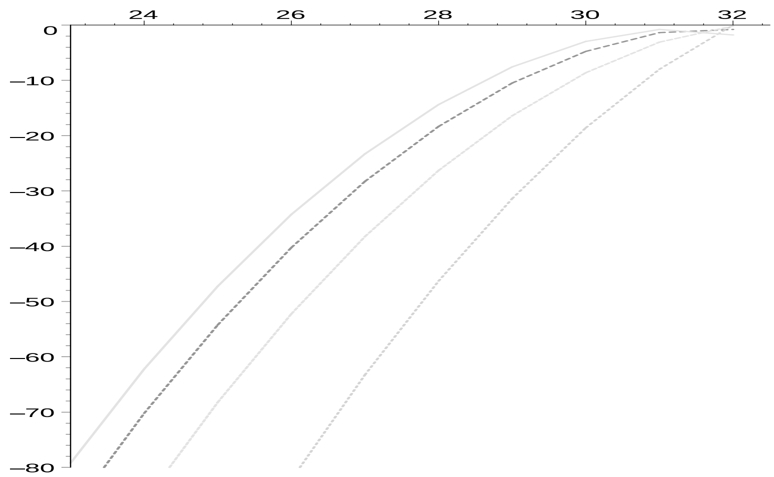

TABLEAU 1.1: Distribution des rangs de codes MRD, de longueur $n = 32$ pour des valeurs de $m \geq 32$

On constate également que la dimension des codes joue un rôle très mineur dans cette répartition pourvu que le code soit de taille suffisamment *importante.*

En effectuant des simulations, on constate que les courbes du tableau sont quasiment identiques aux courbes de distribution des poids d'un code aléatoire dont la formule est donnée par la proposition 5.

Densité d'empilement Bien que l'on ait établi dans la proposition 2 qu'il n'existe pas de codes parfaits en métrique rang, on peut se poser la question de savoir quel est le *défaut de perfectitude* d'un code MRD, c'est-à-dire quel proportion du volume de l'espace est couvert par les boules de rayon $t = \lfloor (d-1)/2 \rfloor$ centrées en les mots du code. Pour un code $(n, M, d)_r$ sur $GF(q^m)$, cette densité D est définie par

$$D \stackrel{def}{=} \frac{M|\mathcal{B}(\mathbf{0}, t)|}{q^{mn}}.$$

En utilisant les encadrements (1.2), on obtient

Proposition 7 (Densité)

Soit $\mathcal{C}$ un code MRD, $(n, q^{m(n-2t)}, 2t+1)_r$ sur $GF(q^m)$. La densité d'empilement de l'espace par le code vérifie

$$\frac{1}{q^{(m-n)t+t^2-2t}} \leq \log_q D \leq \frac{1}{q^{(m-n)t+t^2-t-1}}.$$

En particulier, si $m = n$, en fixant t et en faisant tendre n vers $+\infty$, la densité est supérieure à une constante ne dépendant que de t : q^{-t^2-2t}. Quand $t = 1$ et $m = n$, on peut utiliser l'expression exacte de la boule de rayon 1 :

Corollaire 1
Un code $(n, q^{n-2}, 3)_r$ sur $GF(q^n)$ a une densité d'empilement égale à

$$\frac{1 - 2q^{-n} + q^{-2n+1}}{q-1}. \tag{1.7}$$

Si on considère une suite $\mathcal{C}_n$ de codes MRD de longueur n, et de taille q^{n-2} sur $GF(2^n)$, alors la densité d'empilement des codes s'approche de 1 aussi près que l'on veut quand n tend vers $+\infty$.

6 Pour aller plus loin

Structure du dual en métrique rang En métrique de Hamming, la notion combinatoire de *design* permet d'obtenir une famille de transformations (les transformées de MacWilliams), reliant des propriétés d'un code et de son dual comme le lien entre les distances minimales des deux objets, cf. [MS77]. Dans le cas de la métrique rang il n'existe pas de telles transformations.

En fait, ce n'est pas tout à fait exact, et c'est plutôt le cadre que l'on a donné à la métrique qui ne s'y prête pas. Dans deux articles du milieu et de la fin des années 70, Delsarte et Goethals ont établi une notion de dualité pour une métrique similaire ainsi que des notions de transformées, cf. [DG75, Del78]. Cette métrique est caractérisée par le rang des matrices, mais une matrice de taille $n \times m$ sur $GF(q)$, est définie comme étant la matrice d'une forme quadratique de $GF(q^n) \times GF(q^m)$ dans $GF(q)$. Deux tels objets sont dits orthogonaux si la trace (au sens de trace matricielle) de la matrice produit résultante est nulle. Munis de cette orthogonalité, on parvient à construire des *designs ad hoc* pour lesquels existent des équivalents des transformées de MacWilliams. Malheureusement, cette orthogonalité n'est pas du tout équivalente à la notion d'orthogonalité de vecteurs que nous connaissons par le truchement du produit scalaire dans un espace vectoriel sur un corps fini.

Une piste de recherche intéressante consisterait à établir une théorie de la dualité adaptée au cadre considéré, et à en déduire des équivalents de transformées de MacWilliams.

Codes aléatoires Des simulations effectuées en Maple, montrent que la distribution des rangs d'un code MRD est quasiment identique à celle d'un code aléatoire. Une question intéressante est alors de trouver des paramètres qui puissent différencier les deux familles. Par exemple un travail pourrait être de comparer leurs distances rang minimales respectives. En métrique de Hamming, on sait que la distance minimale d'un code aléatoire est sur la borne GV. En métrique rang, une telle information serait par exemple précieuse pour évaluer la qualité d'un code construit.

Chapitre 2

Correction des erreurs en métrique rang

Dans le chapitre précédent, nous avons établi quelques propriétés générales de la *métrique rang*. En particulier, nous avons discuté de la possibilité de construire des codes avec de *bons* paramètres. Dans l'ordre des choses, la question suivante consiste à définir ce que représente la notion de décodage d'un code, et de tenter d'élaborer les algorithmes les plus efficaces possibles afin de *décoder*.

Décoder est une procédure qui, étant donné un vecteur de l'espace ambiant reçu, permet de retrouver un ou une liste de mots du code vérifiant des propriétés fixées. En général, on cherche à trouver les mots du code minimisant une certaine distance entre le vecteur reçu et le code. Dans le cas où l'on se place dans le modèle du *canal binaire symétrique* (BSC) on minimise la distance de Hamming et dans le cas où l'on considère le modèle canal Gaussien, il s'agit de distances dérivées de la distance euclidienne.

On peut ainsi distinguer plusieurs types de décodages dont, parmi les plus courants :

- *Décodage à maximum de vraisemblance* : Consiste à déterminer un mot du code qui minimise la distance par rapport au mot reçu pour la métrique considérée.
- *Décodage borné par t* : Consiste à déterminer, s'il existe, un mot de code à distance au plus t du mot reçu.
- *Décodage en liste jusqu'à t* : Consiste à déterminer la liste, qui peut être vide, de tous les mots du code qui sont à distance inférieure ou bien égale à t du mot reçu.

Dans le cas de la métrique de Hamming qui nous sert de fil conducteur, il a été montré qu'il était désespéré de rechercher des algorithmes en temps polynomial qui résolve le problème du *Décodage à maximum de vraisemblance*, puisque ce problème est NP-difficile pour un code linéaire binaire, cf. [BMvT78]. De même, il semble peu probable de pouvoir construire un algorithme générique résolvant le problème du *décodage borné*, puisque le simple fait de déterminer la distance minimale d'un code linéaire binaire est NP-difficile, cf. [Var97]. Ces preuves de NP-complétude du problème décisionnel sous-jacent dérivent du problème bien connu de *3-dimensional matching*, cf. [GJ79] par exemple. Les familles d'algorithmes les plus efficaces pour résoudre le décodage borné dans le cas des codes binaires, sont probabilistes de complexité exponentielle. On y trouve notamment les algorithmes de décodage par ensemble d'information raffinés (*IS decoding*), cf. [CC98, Bar98].

La résolution du problème du décodage en liste, problème si populaire pour les familles de codes d'évaluation que sont les codes de *Reed-Solomon généralisés* (codes GRS) et les codes de Reed-Muller est un problème considéré comme un problème difficile à résoudre dès que le rayon de la boule considérée dépasse la *borne de Johnson* , cf. [Sud97, Aug03, Aug04].

A l'instar de la métrique de Hamming, on décrit ainsi les problèmes qui nous intéressent en métrique rang tels qu'ils furent formalisés dans [Loi04, BL04] :

On se donne comme paramètres un code $\mathcal{C}$ sur $GF(q^m)$, un vecteur $\mathbf{y} \in GF(q^m)^n$ de l'espace ambiant, et un nombre entier t.

1. Décodage à maximum de vraisemblance :

   Décodage_MV$(\mathbf{y}, \mathcal{C})$
   Trouver, $\mathbf{c}_0 \in \mathcal{C}$ *tel que* $\mathrm{Rg}(\mathbf{y} - \mathbf{c}_0) = \min_{\mathbf{c} \in \mathcal{C}}(\mathrm{Rg}(\mathbf{y} - \mathbf{c}))$.

2. Décodage borné par un paramètre t :

   Décodage_Borné$(\mathbf{y}, \mathcal{C}, t)$
   Trouver, s'il en existe $\mathbf{c} \in \mathcal{C}$ *et* $\mathbf{e} \in GF(q^m)^n$ *avec* $\mathrm{Rg}(\mathbf{e}) \leq t$ *tels que* $\mathbf{y} = \mathbf{c} + \mathbf{e}$.

3. Décodage en liste :

   Décodage_Liste$(\mathbf{y}, \mathcal{C}, t)$
   Trouver tous les $\mathbf{c} \in \mathcal{C}$ *et* $\mathbf{e} \in GF(q^m)^n$ *avec* $\mathrm{Rg}(\mathbf{e}) \leq t$ *tels que* $\mathbf{y} = \mathbf{c} + \mathbf{e}$.

Bien que les problèmes de décodage en métrique rang ressemblent à s'y méprendre à des instances du problème décisionnel *MinRank* prouvé NP-complet dans les corps finis, cf. [BFS96], il s'avère qu'il n'existe pas de réduction connue entre les problèmes de décodage en métrique rang et ce dernier. A ce jour aucun argument ne permet de justifier que la complexité dans le pire des cas du problème du décodage borné en métrique rang, voire même de la recherche de mot de rang minimum dans un code additif est NP-difficile, sinon les complexités des meilleurs algorithmes permettant de résoudre ces problèmes, présentés dans [OJ02, CS96].

Au début du chapitre, nous décrivons le principe des algorithmes les plus performants réalisant le décodage borné par la *capacité de correction* d'un code linéaire. Ces algorithmes publiés par A. Ourivski et T. Johannson résolvent en premier lieu le problème de la recherche d'un mot de rang minimal dans un code linéaire, cf. [OJ02].

Dans un second temps, nous abordons la problématique du décodage en liste en métrique rang. Comme il n'existe pas d'approche générique connue autre que d'énumérer les mots du code puis de les tester, un point de départ consiste à évaluer la taille de la liste des mots de code susceptibles d'être candidats. Nous présentons une borne supérieure sur le rang de l'erreur que l'on peut corriger telle que le nombre moyen de mots de codes dans cette boule est polynomial. Il s'agit d'un premier pas vers la recherche d'une borne de type *borne de Johnson* en métrique rang. Ce travail a été réalisé par C. Faure, cf. [Fau06].

1 Résolution du décodage borné

De par une particularité inhérente à la métrique rang, les classes d'algorithmes de décodage si répandus pour la métrique de Hamming, qui consistent à déplacer une fenêtre d'information sur le vecteur reçu en espérant éviter les positions d'erreur ne sont pas applicables (pour un exemple d'un tel algorithme, on peut se reporter à l'article [CC98]). En effet, il est tout à fait possible que le nombre de positions corrompues soit égal à la longueur du vecteur reçu, tandis que le rang de l'erreur reste petit :

Exemple 2 *Soit $\alpha \in GF(q^m)^*$. Alors le vecteur*

$$\underbrace{(\alpha, \ldots, \alpha)}_{n \text{ fois}},$$

est de poids de Hamming n, mais de rang 1.

Dans la suite, on va décrire un algorithme de résolution de **Décodage_Borné**$(\mathbf{y}, \mathcal{C}, t)$, où $\mathcal{C}$ est un code $[n, k, d]_r$ et où $t \leq (d-1)/2$. On parle alors de *décodage borné* par la *capacité de correction* .

La résolution du problème **Décodage_Borné** jusqu'à la capacité de correction dans le cas d'un code linéaire a été en premier lieu étudiée par F. Chabaud et J. Stern en 1996 dans le but de cryptanalyser le schéma d'authentification de K. Chen qui utilise des propriétés de la métrique rang, cf. [Che96, CS96]. Plus récemment, en 2002, A. Ourivski et T. Johannson ont publié deux algorithmes, fondés sur deux méthodes de résolution du même problème et permettant d'améliorer significativement l'algorithme de Chabaud-Stern, cf. [OJ02].

Leur méthode de résolution s'appuie sur la proposition suivante qui relie le problème de la recherche d'un mot de rang minimum d'un code linéaire et le problème du décodage jusqu'à la capacité de correction :

Proposition 8 ([OJ02])
Soit $\mathcal{C}$ un code $[n, k, d]_r$ engendré par une matrice $\mathbf{G}$. Soit $\mathbf{y} = \mathbf{c} + \mathbf{e}$, où $\mathbf{c} \in \mathcal{C}$ et $\mathrm{Rg}(\mathbf{e}) \leq \lfloor (d-1)/2 \rfloor$. *Soit $\mathcal{C}'$ le code engendré par la matrice*

$$\mathbf{G}' = \begin{pmatrix} \mathbf{G} \\ \mathbf{y} \end{pmatrix}.$$

Alors, les vecteurs non nuls de rang minimum de $\mathcal{C}'$ sont de la forme $\alpha\mathbf{e}$, où $\alpha \in GF(q^m)^*$.

Une fois qu'on a obtenu un vecteur $\mathbf{e}' = \alpha\mathbf{e}$, on détermine l'élément α en utilisant la propriété

$$\mathbf{H}\mathbf{e}'^T = \alpha\mathbf{H}\mathbf{y}^T,$$

où $\mathbf{H}$ désigne une matrice de parité du code $\mathcal{C}$. La mise en forme de l'algorithme de recherche de mots de rang minimal dans le code $\mathcal{C}$ est la suivante :

1. Sans perte de généralité, on suppose que les $k+1$ premières coordonnées de $\mathcal{C}'$ forment un ensemble d'information de $\mathcal{C}'$. On considère alors la matrice $\mathbf{G}_{syst} = (\mathbf{I}_{k+1} \mid \mathbf{R})$, génératrice de $\mathcal{C}'$ mise sous forme systématique.
2. Tout vecteur $\mathbf{u} \in GF(q^m)^n$ de rang $\leq t$ s'écrit $\mathbf{u} = (1, \beta_2, \ldots, \beta_t)\mathbf{U}$ où

$$\mathbf{U} = (\underbrace{\mathbf{U}_1}_{\text{k+1 cols}} \mid \underbrace{\mathbf{U}_2}_{\text{n-k-1 cols}})$$

   est une matrice non nulle $t \times n$ à coefficients dans $GF(q)$ et où le éléments $1, \beta_2, \ldots, \beta_t$ forment une famille libre sur $GF(q)$.

Ainsi, déterminer un mot de code non nul de rang $t \leq (d-1)/2$ est équivalent à trouver une solution non nulle de l'équation

$$(1, \beta_2, \ldots, \beta_t)\,(\mathbf{U}_2 - \mathbf{U}_1\mathbf{R}) = \mathbf{0}, \tag{2.1}$$

avec comme contrainte que les éléments $1, \beta_2, \ldots, \beta_t$ soient linéairement indépendants sur $GF(q)$. A. Ourivski et T. Johannson ont envisagé deux stratégies pour résoudre le système.

1. *Enumération des bases* : On énumère les familles $\beta_2, \ldots, \beta_t$ libres sur $GF(q)$. En utilisant des symétries spécifiques du système, on se ramène à énumérer un ensemble de familles dans un espace de taille au plus $q^{(m-t)(t-1)}$. Pour chacun des candidats, on tente ensuite de résoudre dans $GF(q)$ le système (2.1) de $mt(n-k-1)$ équations (équations projetées sur $GF(q)$) à nt inconnues (matrice $\mathbf{U}$).
2. *Enumération des coordonnées* : L'autre méthode consiste à voir que, pour que (2.1) ait une solution non nulle, il faut et il suffit que la matrice $\mathbf{V} = \mathbf{U}_2 - \mathbf{U}_1\mathbf{R}$ de taille $t \times (n-k-1)$ soit de rang $\leq t-1$. Ceci implique que la matrice carrée $t \times t$ formée des t premières colonnes de $\mathbf{V}$ est non inversible. Pourvu que la matrice carrée formée des t premières colonnes de $\mathbf{U}_1$ soit inversible, on peut considérer que $\mathbf{U}_1$ s'écrit
$$\mathbf{U}_1 = (\mathbf{I}_t|\mathbf{A}).$$
Il suffit alors d'énumérer un espace de taille $q^{t(k+1)}$ et de tester le rang de chaque candidat. Une fois une matrice $\mathbf{V}$ *ad hoc* de rang non maximal déterminée, on résout alors le système linéaire.

Avec quelques raffinements supplémentaires, mais plus techniques, on parvient à construire des algorithmes de décodage dont les complexités moyennes sont données par la proposition suivante, la partie exponentielle provenant de l'énumération d'espace vectoriels, la partie polynomiale provenant de la résolution de systèmes linéaires :

Proposition 9 ([OJ02])

Soit $\mathcal{C}$, un code linéaire $[n,k,d]_r$. Alors, il existe des algorithmes permettant de résoudre **Décodage_Borné**$(\mathbf{y}, \mathcal{C}, t)$*, avec $t \leq (d-1)/2$ dont les complexités moyennes vérifient*

- *Enumération des bases :* $W_{bases} \leq (k+t)^3 q^{(t-1)(m-t)+2}$.
- *Enumération des coordonnées :* $W_{coord} \leq (k+t)^3 t^3 q^{(t-1)(k+1)}$.

Remarque 2

L'algorithme de F. Chabaud et J. Stern est un algorithme par énumération des bases. *A. Ourivski et T. Johannson en ont amélioré le facteur polynomial.*

Remarque 3

Comme on pourrait le voir dans la preuve de la proposition 8*, le fait de savoir trouver un mot de rang faible n'implique que l'on sait résoudre le problème du décodage borné que dans le cas où la borne t est inférieure ou égale à la capacité de correction du code. Si veut résoudre ce problème pour des valeurs de t plus grandes, il faut modifier son approche.*

2 Décodage en liste

Etant donné un vecteur $\mathbf{y} \in GF(q^m)^n$, choisi aléatoirement, et un code linéaire $\mathcal{C}$ de taille M donné par une matrice aléatoire, on ne peut espérer construire un algorithme polynomial de décodage en liste que si l'intersection de la boule considérée et du code est de taille polynomiale dans le pire des cas. Ce qui implique d'avoir une estimation dans le cas le pire.

Ce problème pouvant être redoutablement difficile et dépendant des propriétés du code, une première approche consiste à évaluer la nombre moyen de mots de code dans la boule. Cela revient à déterminer la taille moyenne $\mathcal{N}$ de $|\mathcal{B}(\mathbf{0}, t) \cap \mathcal{C}|$. Ce travail ainsi que les bornes ont été obtenues par C. Faure [Fau06]. En utilisant l'encadrement du volume d'une boule en métrique rang donné par l'inéquation (1.2) on obtient

$$q^{(m+n-2)t-t^2+\log_q M-mn} \leq \mathcal{N} = \frac{|\mathcal{B}(\mathbf{0},t)||\mathcal{C}|}{q^{mn}} \leq q^{(m+n+1)t-t^2+\log_q M-mn+1}.$$

Le nombre moyen de mots du code dans la boule est donc exponentiel en fonction de n, sauf dans le cas où l'exposant $(m+n+1)t - t^2 + mn - \log_q M$ est inférieur ou égal à $\lambda \log_q n$, λ fixé. Cela implique alors que

$$t \leq \frac{m+n+1}{2} - \sqrt{\log_q (Mn^\lambda) + \frac{(m-n)^2 - 2m - 2n}{4}}. \tag{2.2}$$

Un cas particulier intéressant que nous avons déjà mentionné au chapitre 1 consiste à étudier le cas de codes MRD lorsque $m = n$, puisque ce sont les codes à plus forte densité. Dans ce cas l'équation (2.2) devient

$$t \leq n - \sqrt{(n+1)k + \lambda \log_q n}, \tag{2.3}$$

les termes $\lambda \log_q n$ et n pouvant être asymptotiquement négligé.

Ainsi, une conclusion partielle de cette borne simple est que le problème du décodage en liste jusqu'à la distance t de codes aléatoires quelconques a de bonne chances de n'être pas résoluble en temps polynomial dès que le rayon de recherche de la boule dépasse la quantité $\frac{m+n}{2} - \sqrt{\log_q Mn^\lambda + \frac{(m-n)^2-2n-2m}{4}}$, la taille moyenne de la liste des candidats devenant alors exponentielle.

3 Pistes de recherches

- Autant, la complexité du décodage en métrique de Hamming a fait l'objet d'études permettant de considérer que ces problèmes sont difficiles, cf. [BMvT78, Var97], autant, en métrique rang le travail reste à faire. Il ne semble pas trivial de réduire ces problèmes à des problèmes réputés difficiles, mais qui leur sont proches dans la description. Le problème *3-dimensional-matching* utilisé dans les preuves de complexité est combinatoire et ne se transpose pas au cas de la métrique rang. De même, les problèmes prouvés NP-difficile d'algèbre linéaire comme *MinRank* et *MaxRank* ne se transposent pas aisément non plus. Le facteur bloquant réside dans le fait qu'il est nécessaire à un certain moment de pouvoir projeter sur un sous-corps. Pour la réduction de sécurité, il faut alors pouvoir remonter la pente en temps polynomial. Cependant, le nombre d'antécédents possible d'une projection est exponentiel !
- Résoudre le problème du décodage borné par la capacité de correction en métrique rang est équivalent à résoudre le système (2.1). En le *dépliant*, c'est-à-dire en projetant les équations obtenues dans le corps de base $GF(q)$, on obtient un système quadratique de $m(n-k-1)$ équations à $(m+n)t - m$ inconnues. Une approche intéressante en matière de recherche pourrait être d'appliquer justement les algorithmes de résolution de systèmes non linéaires en calculant les bases de Groebner. Bien qu'on mesure difficilement la complexité de tels algorithmes, les progrès récents en la matière méritent d'être approfondis, cf. [Fau99, Fau02]. C'est ce qu'ont fait F. Levy-dit-Vehel et L. Perret. Je présente leurs résultats dans le tableau 2.1. Les

terminologies OuJo-1 et OuJo-2 désignent les algorithmes de A. Ourivski et T. Johannson respectivement par *énumération des coordonnées* et *énumération des bases.* L'algorithme LePe est un algorithme utilisant l'algorithme F4 implanté dans le logiciel de calcul algébrique MAGMA, cf. [Lev06, Fau99, MAG]. Dans le cas où $t = 2$, on constate que l'algorithme LePe est en général le plus efficace.

m	n	k	t	OuJo−1	OuJo−2	LePe
25	30	15	2	2^{32}	2^{39}	31s.
30	30	16	2	2^{37}	2^{46}	1min. 4s.
30	50	20	2	2^{41}	2^{45}	5min. 30s.
50	50	26	2	2^{49}	2^{67}	1h. 5min.
20	20	10	3	2^{42}	2^{52}	8h.
15	15	7	3	2^{35}	2^{37}	30min. 20s.
15	15	8	3	2^{36}	2^{40}	13h. 30min.

Tableau 2.1: Comparaison entre les algorithmes résolvant le problème de décodage borné par la distance minimale en métrique rang

– Outre les algorithmes dits *classiques* de décodage par ensemble d'information qui permettent de résoudre le problème du *décodage borné* d'un code linéaire jusqu'à sa capacité de correction, il existe en métrique de Hamming une grande variété d'algorithmes de décodage permettant de résoudre des problèmes proches comme les algorithmes de décodage par *découpage du syndrome*[1] , ou bien les méthodes plus combinatoires utilisant des algorithmes du type de *descente de gradient*[2], cf. [Bar98]. On peut alors se poser la question de savoir s'il est possible de transposer ces approches dans le cadre de la métrique rang.

[1]split syndrome decoding
[2]gradient like decoding

Chapitre 3

Polynômes linéaires et métrique rang

Le tableau de la métrique rang serait incomplet si nous n'introduisions une classe de polynômes qui lui sont intrinsèquement liés à savoir les q-polynômes, ou polynômes linéaires définis par Øre [Ore33, Ore34], ou même encore appelés *polynômes de Øre*. Si les articles sus-cités sont volumineux et touffus du point de vue théorique, du point de vue algorithmique, assez peu de travaux leur ont été spécifiquement dédiés depuis. Une exception notable est la construction d'un algorithme de détermination de l'ensemble des racines de q-polynômes, cf. [Ber84, LN97].

Ce chapitre constitue un résumé de propriétés des q–polynômes utiles dans la suite du document, en mettant en exergue l'aspect algorithmique des algorithmes de calcul et de recherche dérivés. On commence par définir l'anneau des q–polynômes, ainsi que quelques-unes de ses propriétés. Nous effectuons également un tour d'horizon des complexités des algorithmes simples dans lesquels ces polynômes interviennent, en exhibant les similitudes et les différences entre ces algorithmes et leurs *alter-ego* pour les polynômes classiques. Pour ces derniers on prend le livre [GG03] comme référence. Enfin, on montre que des problèmes de reconstruction de q-polynômes sont plus spécifiquement reliés aux problèmes du *décodage borné* et du *décodage en liste* décrits au chapitre précédent. Ce chapitre constitue ainsi une synthèse et une remise en forme de résultats que l'on peut retrouver plus ou moins explicitement dans les articles sus-cités.

1 Anneau des q-polynômes

Voici quelques définitions ainsi que des propriétés des polynômes linéaires. En plus des résultats dûs à Øre, on explicitera les complexités des algorithmes.

Définition 5 ([Ore33])

On appelle q-polynôme de q-degré t à coefficient dans $GF(q^m)$, un polynôme de la forme

$$P(z) = \sum_{i=0}^{t} p_i z^{q^i},$$

où $p_t \neq 0$, et l'on note $deg_q(P) \stackrel{def}{=} t$, le q-degré du polynôme.

Comme les termes correspondent à des puissance de l'automorphisme de Frobenius $z \mapsto z^q$ de $GF(q^m)/GF(q)$, un q-polynôme est une application linéaire sur $GF(q^m)$, considéré comme espace vectoriel de dimension m sur $GF(q)$, *i.e.*

$$\forall z, y \in GF(q^m),\ \forall \lambda, \mu \in GF(q), \quad P(\lambda z + \mu y) = \lambda P(z) + \mu P(y). \tag{3.1}$$

En particulier, l'ensemble des racines d'un q-polynôme forme un espace vectoriel sur $GF(q)$ de dimension inférieure ou égale au q-degré du polynôme considéré.

Lorsqu'il n'y aura pas ambiguïté, on notera $[i] \stackrel{def}{=} q^i$. L'ensemble des q-polynômes à coefficients dans $GF(q^m)$ a une structure d'anneau non-commutatif lorsqu'on le munit des lois suivantes :

- *Addition* : $(P+Q)(z) \stackrel{def}{=} P(z) + Q(z)$.
- *Composition* : $(P \circ Q)(z) \stackrel{def}{=} P[Q(z)]$.

2 Opérations sur les q-polynômes

Dans cette section, on passe en revue les algorithmes permettant d'effectuer les opérations les plus communes ainsi que leurs complexités respectives. On ne retiendra dans le calcul de complexité que le coût de la multiplication dans $GF(q^m)$ négligeant par là les complexités des additions ainsi que des élévations à une puissance de q.

2.1 Addition et multiplication

Additionner deux q-polynômes consiste à additionner les coefficients dans $GF(q^m)$, tandis que pour les multiplier, la seule méthode connue de multiplication, consistant à effectuer la composition de deux q-polynômes permet une complexité de :

Proposition 10 (Multiplication)
Soient P et Q, deux q-polynômes de q-degré respectifs t et s, alors $P \circ Q$ se calcule en

$$\sum_{i=0}^{s+t} i + 1 = \frac{(s+t+1)(s+t+2)}{2}, \text{ multiplications dans } GF(q^m).$$

L'anneau des q-polynôme étant non-commutatif, on ne peut pas appliquer un algorithme de type Karatsuba pour la multiplication.

2.2 Algorithme d'Euclide

Pour la structure d'anneau non-commutatif des q-polynômes, Øre a établi l'existence d'un algorithme d'Euclide à droite et d'un algorithme d'Euclide à gauche [1].

Proposition 11 ([Ore33]) *Soit*

$$\begin{cases} P(z) = p_k z^{[k]} + \sum_{i=0}^{k-1} p_i z^{[i]}, & p_k \neq 0, \\ G(z) = g_s z^{[s]} + \sum_{j=0}^{s-1} g_j z^{[j]}, & g_s \neq 0, \end{cases}$$

alors il existe deux couples de q-polynômes (q_1, r_1) et (q_2, r_2) tels que

$$\begin{aligned} P &= q_1 \circ G + r_1, \quad deg_q(r_1) < deg_q(G), \\ P &= G \circ q_2 + r_2, \quad deg_q(r_2) < deg_q(G). \end{aligned}$$

Les couples (q_1, r_1) et (q_2, r_2) peuvent êtres déterminés en $s(k-s)$ multiplications dans $GF(q^m)$.

[1] pour les corps finis il existe toujours un algorithme de division euclidienne à gauche mais ce n'est pas toujours le cas.

2.3 Recherche de racines

L'ensemble des racines d'un q-polynôme est un espace vectoriel sur $GF(q)$. Rechercher les racines revient donc à déterminer une base de cet espace. Soit P un q-polynôme de q-degré t, alors la procédure décrite dans [Ber84] est la suivante :

- On se fixe une base $\beta_1, \ldots, \beta_m$ de $GF(q^m)/GF(q)$. Et on évalue $P(\beta_1), \ldots, P(\beta_n)$ sur ces éléments. Comme la complexité de l'évaluation d'un q-polynôme sur un élément de $GF(q^m)$ est de t multiplications, la complexité de l'étape d'évaluation est de mt multiplications dans $GF(q^m)$.
- On résout ensuite le système linéaire

$$\lambda_1 P(\beta_1) + \cdots + \lambda_m P(\beta_m) = 0$$

  où, pour tout $i = 1, \ldots, m$, $\lambda_i \in GF(q)$. Cela revient à résoudre un système linéaire de m équations à m inconnues dans $GF(q)$ de rang $t \leq m$. Ce qui se fait en t^3 opérations dans $GF(q)$.

Comme la résolution du système de la seconde étape s'effectue dans le corps de base, sa complexité peut être négligée par rapport à la complexité de la première étape. On obtient ainsi :

Proposition 12 ([Ber84])

Etant donné un q-polynôme à coefficients dans $GF(q^m)$ de q-degré t, il existe un algorithme qui retrouve une base de l'espace des racines en mt multiplications dans $GF(q^m)$.

2.4 Interpolation de polynôme

L'ensemble des racines d'un q-polynôme est un espace vectoriel. Réciproquement, Øre a montré que tout sous-espace vectoriel de $GF(q^m)$ vu comme espace vectoriel de dimension m sur $GF(q)$ est l'ensemble des racines d'un unique q-polynôme unitaire.

Soit $e_1, \ldots, e_t \in GF(q^m)$ linéairement indépendants sur $GF(q)$, et soit $V =< e_1, \ldots, e_t >$ l'espace vectoriel de dimension t engendré. On considère la famille P_i de q-polynômes de degré croissant construite par la récurrence suivante :

1. $P_1(z) = z^q - e_1 z$,
2. $\forall i = 2, \ldots, t,\ P_i(z) = P_{i-1}(z)^q - P_{i-1}^{q-1}(e_i) P_{i-1}(z)$.

Le q-polynôme P_t ainsi obtenu est bien le polynôme d'interpolation car

$$\begin{cases} P_t \text{ est unitaire,} \\ deg_q(P_t) = t, \\ \forall i = 1, \ldots, t,\ P_t(e_i) = 0. \end{cases}$$

L'évaluation de la complexité de cette procédure nous amène à la proposition suivante

Proposition 13 ([Ore33])

Tout sous-espace vectoriel de dimension t de $GF(q^m)/GF(q)$ est l'ensemble des racines d'un unique q-polynôme unitaire de degré t. Ce polynôme peut-être déterminé en $t(t-1)/2$ multiplications dans $GF(q^m)$.

3 Lien entre q-polynômes et métrique rang

On définit l'évaluation d'un q-polynôme V sur un vecteur $\mathbf{x} = (x_1, \ldots, x_n)$ de longueur n par

$$V(\mathbf{x}) \stackrel{def}{=} (V(x_1), \ldots, V(x_n)).$$

La structure même des q-polynômes permet de les relier aux problèmes de décodage en métrique rang décrits dans l'introduction du chapitre 2 en utilisant le corollaire suivant qui découle de la proposition 13.

Corollaire 2
Soit $\mathbf{x} = (x_1, \ldots, x_n)$ un vecteur de $GF(q^m)$. Le vecteur $\mathbf{x}$ est de rang t sur $GF(q)$ si et seulement si *le q-polynôme V unitaire de plus petit q-degré tel que $V(\mathbf{x}) = \mathbf{0}$ est de q-degré t.*

Etant donné un vecteur $\mathbf{y}$ de l'espace ambiant $GF(q^m)^n$, un code $\mathcal{C} \subset GF(q^m)^n$, et un entier $t \geq 0$, on définit les deux problèmes de recherche de q-polynômes avec certaines propriétés sur le degré :

Degré_Borné$(\mathbf{y}, \mathcal{C}, t)$
Trouver, s'il en existe $\mathbf{c} \in \mathcal{C}$ et un q-polynôme V de q-degré $\leq t$ tels que $V(\mathbf{y} - \mathbf{c}) = \mathbf{0}$.

Liste_Polynômes$(\mathbf{y}, \mathcal{C}, t)$
Trouver tous les $\mathbf{c} \in \mathcal{C}$ et V q-polynômes de q-degré $\leq t$ tels que $V(\mathbf{y} - \mathbf{c}) = \mathbf{0}$.

La proposition suivante donne le lien entre ces problèmes et les problèmes de décodage définis dans l'introduction du chapitre 2.

Proposition 14 (Equivalence de problèmes)

1. **Décodage_Borné**$(\mathbf{y}, \mathcal{C}, t)$ *et* **Degré_Borné**$(\mathbf{y}, \mathcal{C}, t)$ *sont polynomialement équivalents.*
2. **Décodage_Liste**$(\mathbf{y}, \mathcal{C}, t)$ *et* **Liste_Polynômes**$(\mathbf{y}, \mathcal{C}, t)$ *sont polynomialement équivalents.*

Preuve
On ne prouvera que la première assertion, l'autre pouvant s'en déduire facilement.
- Soit $\mathbf{c} \in \mathcal{C}$ et $\mathbf{e}$ une solution de **Décodage_Borné**$(\mathbf{y}, \mathcal{C}$,t). Soit V l'unique polynôme unitaire q-degré $\mathrm{Rg}(\mathbf{e}) \leq t$ qui interpole le vecteur $\mathbf{e}$. D'après la proposition 13, le q-polynôme V est calculable en temps polynomial et comme $\mathbf{e} = \mathbf{y} - \mathbf{c}$, on a $V(\mathbf{y} - \mathbf{c}) = \mathbf{0}$.
- Réciproquement, soit $(V, \mathbf{c}) \in \mathcal{C}$ une solution de **Degré_Borné**$(\mathbf{y}, \mathcal{C}, t)$. Alors le vecteur $\mathbf{e} = \mathbf{y} - \mathbf{c}$ dont les composantes sont racines de V est de rang t au plus.

■

4 Pistes de recherche

Dans le cas de polynômes *classiques*, il y a pléthore de résultats et d'algorithmes permettant d'effectuer efficacement les opérations simples sur les polynômes comme l'interpolation, la recherche de racines, ou la multiplication – algorithme de Karatsuba ou transformée de Fourier. Une abondance de détails et d'algorithmes se trouvent décrits dans le livre [GG03].

Pour les q-polynômes en revanche, c'est plutôt la disette et même un algorithme aussi simple et élégant que l'algorithme de Karatsuba ne peut être construit puisque la loi de composition n'est pas commutative. Une piste de recherche pourrait consister à construire des algorithmes plus efficaces permettant d'effectuer les opérations simples sur les polynômes linéaires en commençant par exemple par l'algorithme simple de multiplication de q-polynômes. Ces algorithmes trouveraient alors des application immédiates dans le décodage des codes de Gabidulin que nous allons présenter au chapitre 4.

Deuxième partie

Codes optimaux

Chapitre 4

Les codes de Gabidulin

Dans les précédents chapitres, nous avons établi des propriétés assez générales de la métrique rang, montré que le décodage d'un code linéaire en métrique rang semblait être un problème plus difficile qu'en métrique de Hamming, malgré l'inexistence de réduction de complexité à des problèmes *durs*. Nous avons ensuite montré que la métrique et les problèmes de décodage se reliaient à l'anneau des q-polynômes et à des problèmes de reconstruction.

Cependant, nous sommes encore orphelins de codes disposant d'algorithmes de décodage en temps polynomial, et c'est l'objet de ce chapitre que d'introduire la première famille de codes avec ces propriétés. Dans son article fondateur, E. M. Gabidulin présenta une famille de codes optimaux, disposant d'un algorithme de décodage en temps polynomial. Ces codes sont construits comme codes d'évaluation de q-polynômes, sur des éléments linéairement indépendants de l'extension de corps considérée.

Dans un premier temps, nous définissons les codes de Gabidulin. Nous en dérivons les propriétés principales. Hormis la proposition 17 sur l'existence de codes de Gabidulin asymptotiquement parfaits, qui découlent de la proposition 7 du chapitre 1, ces résultats proviennent des articles [Gab85b, Ber03].

Dans un second temps, nous décrivons les algorithmes de décodage par syndrome qui furent les premiers algorithmes de décodage construits [Gab85b, Gab91, Rot91]. Malgré des raffinements successifs dont les plus récents inspirés de l'algorithme de Berlekamp-Massey, leur complexité reste cubique en fonction du rang de l'erreur à corriger, cf. [RP04b, RP04a] par exemple.

Dans une troisième partie, on décrit une nouvelle classe d'algorithmes de décodage dits par *reconstruction de q-polynôme*. Cette approche s'inspire de l'algorithme de Welch–Berlekamp de décodage des codes RS et s'appuie sur la relation forte entre les problèmes de décodage et les problèmes de reconstruction de q-polynômes (cf. chapitre 3). De par une telle approche, on parvient à améliorer significativement la complexité du décodage. La version la plus performante d'un algorithme de cette classe est de complexité quadratique en la longueur du code. Ces travaux se trouvent décrits dans les articles [Loi04, Loi06].

1 Définition et propriétés

Soit $\mathbf{g} = (g_1, \ldots, g_n)$, un vecteur d'éléments d'une extension de corps $GF(q^m)$ linéairement indépendants sur $GF(q)$. Soit

$$\mathbf{G} = \begin{pmatrix} g_1 & \cdots & g_n \\ \vdots & \ddots & \vdots \\ g_1^{[k-1]} & \cdots & g_n^{[k-1]} \end{pmatrix}, \tag{4.1}$$

où $[i] \stackrel{def}{=} q^i$ désigne la $i$ème puissance de l'automorphisme de Frobenius de $GF(q^m)/GF(q)$.

Définition 6 ([Gab85b])

Le code de Gabidulin $Gab_k(\mathbf{g})$ sur $GF(q^m)$ de dimension k et de vecteur générateur $\mathbf{g}$ est le code de matrice génératrice $\mathbf{G}$.

Il possède les propriétés suivantes, [Gab85b, Gab85a] :

- C'est un code d'évaluation de polynômes linéaires de q-degré strictement inférieur à k, sur le vecteur $\mathbf{g}$ *i.e.*

$$Gab_k(\mathbf{g}) = \left\{ (P(g_1), \ldots, P(g_n)) \mid P(z) = \sum_{i=0}^{k-1} p_i z^{[i]} \right\}$$

- C'est un code *MRD*, c'est-à-dire $[n, k, d]_r$ où $d = n-k+1$ (cf. définition 4 du chapitre 1). C'est également un code MDS. En particulier tout ensemble de k positions est une fenêtre d'information.

Des deux propriétés précédentes, on déduit la matrice génératrice du code *sous forme systématique* sur les k premières positions

Proposition 15

La matrice génératrice du code engendré par $\mathbf{G}$ est de la forme

$$\mathbf{G}_{syst} = \left(\begin{array}{cccc|ccc} 1 & 0 & \cdots & 0 & P_1(g_{k+1}) & \cdots & P_1(g_n) \\ 0 & 1 & \ddots & 0 & P_2(g_{k+1}) & \cdots & P_2(g_n) \\ \vdots & \ddots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & \cdots & 1 & P_k(g_{k+1}) & \cdots & P_k(g_n) \end{array}\right),$$

où, pour $i = 1, \ldots, k$, P_i est l'unique q-polynôme de degré k vérifiant pour tout $j \leq k$, $P_i(g_j) = \delta_{i,j}$.

Une matrice de parité de $Gab_k(\mathbf{g})$ est la matrice

$$\mathbf{H} = \begin{pmatrix} h_1 & \cdots & h_n \\ \vdots & \ddots & \vdots \\ h_1^{[d-2]} & \cdots & h_n^{[d-2]} \end{pmatrix}, \tag{4.2}$$

où $\mathbf{h} = \left(\lambda_1^{[d]}, \ldots, \lambda_n^{[d]}\right)$ et les λ_i vérifient l'équation

$$\sum_{i=1}^{n} \lambda_i g_i^{[j]} = 0,$$

pour $j = 0, 1, \ldots, n-2$. Ainsi, le code dual de $Gab_k(\mathbf{g})$ est le code $Gab_{n-k}(\mathbf{h})$.

Il est clair que l'ensemble des codes de Gabidulin est stable par permutation du support et même par multiplication à droite par une matrice inversible sur le corps de base. En effet

Proposition 16

Soit $\mathbf{T} \in GL_n(GF(q))$, et soit $\mathbf{g} = (g_1, \ldots, g_n) \in GF(q^m)^n$, linéairement indépendants sur $GF(q)$. Alors

$$Gab_k(\mathbf{g})\mathbf{T} \stackrel{def}{=} \{\mathbf{cT} \mid \mathbf{c} \in Gab_k(\mathbf{g})\} = Gab_k(\mathbf{gT}).$$

Une autre résultat important concernant la structure de ces codes a été publié par T. Berger. Il a caractérisé le groupe des isométries *semi-linéaires* ainsi que le groupe des permutations laissant un code invariant.

Théorème 2 ([Ber03])

- *Le groupe des isométries semi-linéaires de $Gab_k(\mathbf{g})$ est réduit au groupe des multiplications scalaires dans $GF(q^m)^*$. En particulier, deux codes $Gab_k(\mathbf{g})$ et $Gab_k(\mathbf{g}')$ sont égaux si et seulement s'il existe $\alpha \in GF(q^m)^*$ tel que $\mathbf{g} = \alpha\mathbf{g}'$.*
- *Le groupe de permutation de $Gab_k(\mathbf{g})$ est trivial.*

Une famille de codes asymptotiquement parfaits Comme les codes de Gabidulin sont des codes MRD, on peut appliquer les résultats de la section 5 du chapitre 1, notamment le corollaire 1. On obtient alors

Proposition 17

Soit $\mathbf{g}_i$ une suite de vecteurs de longueur i à coefficients dans $GF(2^i)$ dont les composantes sont linéairement indépendantes sur $GF(2)$. Alors la suite de codes $Gab_{i-2}(\mathbf{g}_i)$ est une suite de codes 1-correcteurs asymptotiquement parfaits*, dont la densité d'empilement $\mathcal{D}_i$ vérifie*

$$\mathcal{D}_i = 1 - 2^{-i+1} + 2^{-2i+1},$$

Ainsi, quand $i \to \infty$ ces codes s'approchent aussi près que l'on veut d'un code parfait.

2 Décodage par syndrome

On s'intéresse désormais au problème du décodage des codes de Gabidulin borné par la capacité de correction, c'est-à-dire dans la terminologie du chapitre 2 à la résolution de **Décodage_Borné**$(\mathbf{y}, \mathcal{C}, t)$, où $t = \lfloor (d-1)/2 \rfloor$. On suppose reçu le vecteur

$$\mathbf{y} = \mathbf{xG} + \mathbf{e},$$

où $\mathbf{G}$ est une matrice génératrice du code $Gab_k(\mathbf{g})$ mise sous la forme (4.1), et où $\mathbf{e}$ est un vecteur d'erreur de rang $t \leq \lfloor (d-1)/2 \rfloor$. Il existe deux grands types d'algorithmes permettant recouvrer le vecteur d'information $\mathbf{x}$.

Nous considérons en premier lieu les algorithmes de décodage par syndrome. Pour ce type d'algorithmes, le receveur calcule le syndrome $\mathbf{s} = \mathbf{Hy}^T$, où $\mathbf{H}$ est une matrice de parité de $Gab_k(\mathbf{g})$. Son but consiste à déterminer l'unique vecteur $\mathbf{e}$ de rang $t \leq \lfloor (d-1)/2 \rfloor$ vérifiant

$$\mathbf{s} = \mathbf{He}^T. \tag{4.3}$$

Une fois que le vecteur $\mathbf{e}$ est retrouvé, on détermine le vecteur d'information $\mathbf{x}$ en évaluant

$$\mathbf{x} = \mathbf{G}_k^{-1}(\mathbf{y} - \mathbf{e}),$$

où $\mathbf{G}_k$ désigne la matrice carrée $k \times k$ formée de k colonnes quelconques de $\mathbf{G}$[1].

[1]le code $Gab_k(\mathbf{g})$ est MRD donc MDS, donc tout ensemble de k positions forme une fenêtre d'information, et donc $\mathbf{G}_k$ est inversible

Principe des algorithmes Pour ce faire, on recherche un vecteur $\mathbf{b} = (\beta_1, \ldots, \beta_t)$ constitué de t éléments linéairement indépendants sur $GF(q)$ et une matrice $\mathbf{Y}$ de taille $t \times n$ à coefficients dans le corps de base $GF(q)$ tels que $\mathbf{e} = \mathbf{bY}$. Le système (4.3) se réécrit alors

$$\mathbf{s} = \underbrace{\mathbf{HY}^T}_{\mathbf{X}} \cdot \mathbf{b}^T,$$

où $\mathbf{X}$ est une matrice de taille $(d-1) \times t$. Les algorithmes [Gab85b, Gab91, Rot91, RP04b, RP04a], procèdent tous de la façon suivante :

1. Calcul du syndrome $\mathbf{s}$.
2. Détermination d'un vecteur $\mathbf{b}$, souvent appelé *base des solutions*. On construit un q-polynôme de q-degré t qui a pour racine l'espace vectoriel engendré par les composantes de $\mathbf{b}$. On en détermine les racines en utilisant par exemple l'algorithme de la section 2.3 du chapitre 3.
3. Détermination de la matrice $\mathbf{X} = \mathbf{HY}^T$, en résolvant le système $\mathbf{s} = \mathbf{Xb}^T$.
4. Détermination de la matrice $\mathbf{Y}$, puis calcul de $\mathbf{e} = \mathbf{bY}$.

Complexité des algorithmes Tous les algorithmes sus-mentionnés ont la même complexité

- pour l'étape de détermination du syndrome : $n(d-1)$ multiplications dans $GF(q^m)$,
- pour l'étape de détermination des racines du q-polynôme : t^3 opérations dans $GF(q)$, ce qui est négligeable par rapport aux autres étapes,
- pour la détermination de $\mathbf{X}$: pivot de Gauss, $t^3/2$ multiplications dans $GF(q^m)$,
- et pour la détermination de la matrice $\mathbf{Y}$: résolution d'un système dans $GF(q)$, en général cette complexité est négligeable par rapport aux autres étapes.

Là où les divers algorithmes diffèrent est l'étape de détermination du polynôme linéaire de q-degré t dont une base des racines constitue le vecteur $\mathbf{b}$. On trouve deux approches différentes :

1. Résolution d'une *équation clé* du type

   $$\Lambda(x) \circ S(x) = F(x) \bmod x^{[d-1]},$$

   où S désigne un q-polynôme de q-degré $d-1$ dont les coefficients sont les coordonnées du syndromes $\mathbf{s}$, et Λ est le q-polynôme recherché de q-degré $\leq t$. On peut utiliser, ou bien un algorithme de type Euclide étendu pour les polynômes linéaires, cf. [Gab85b], ou bien un algorithme de type Berlekamp–Massey par génération de registres, cf. [RP04b, RP04a]. La complexité de l'algorithme d'Euclide étendu est en gros $\leq t^2 + d^2/4$ multiplications dans $GF(q^m)$, tandis que celle de l'algorithme de type Berlekamp–Massey est de $6t^2$ multiplications dans $GF(q^m)$.
2. On peut également résoudre un système linéaire de taille t. Avec un algorithme générique de résolution, on parvient à une complexité de l'ordre de $t^3/2$ multiplication dans $GF(q^m)$, cf. [Gab91, Rot91].

La complexité des divers algorithmes de décodage par syndrome peut se résumer par

Propriétés 1 (Complexité du décodage par syndrome)
- Euclide étendu *:* $\approx n(d-1) + t^3/2 + t^2 + d^2/4$ *multiplications dans* $GF(q^m)$*.*
- Berlekamp–Massey *:* $\approx n(d-1) + 6t^2 + t^3/2$ *multiplications dans* $GF(q^m)$*.*

– Résolution de système linéaire *:* $\approx n(d-1)+t^3$ *multiplications dans* $GF(q^m)$.

Quel que soit l'algorithme considéré, il reste un facteur cubique en le rang de l'erreur à corriger.

Remarque 4
Dans le cas du décodage des codes RS, dans l'évaluation de la complexité des algorithmes de décodage par syndrome le facteur cubique disparaît car on peut calculer en une étape, à la fois le polynôme localisateur et le polynôme évaluateur d'erreur. On utilise un algorithme d'Euclide étendu ou bien un algorithme de type Berlekamp–Massey, qui sont tous deux de complexité quadratique. L'étape de recherche des racines est également de complexité quadratique. Une description des ces algorithmes se trouve dans la thèse de N. Sendrier par exemple [Sen91].

3 Le problème de reconstruction de polynômes linéaires et le décodage des codes de Gabidulin

Les codes de Gabidulin sont des codes d'évaluation de polynômes linéaires de q-degré borné, au même titre que les codes de Reed-Solomon peuvent être considérés comme des codes d'évaluation de polynômes sur des éléments distincts d'un corps fini. C'est cette propriété que E. R. Berlekamp et L. Welch ont exploitée dans la conception d'un algorithme reconstruisant le polynôme des positions d'erreurs du mot reçu, cf. [BW86]. M. Sudan s'est également inspiré de cette propriété pour ses algorithmes décodage en liste des codes de Reed-Solomon, cf. [Sud97, GS99].

On peut transposer, au moins partiellement, cette approche dans le cas qui nous intéresse, cf. [Loi04, Loi06]. Dans un premier temps, il convient de réadapter le problème de reconstruction de polynômes à l'anneau *non-commutatif* des polynômes linéaires. Le problème prend en argument des vecteurs $\mathbf{y}=(y_1,\ldots,y_n)$ et $\mathbf{g}=(g_1,\ldots,g_n)$ de l'espace ambiant $GF(q^m)^n$ et des entiers positifs k et t. Il s'énonce sous la forme :

Reconstruction$(\mathbf{y},\mathbf{g},k,t)$
Trouver tous les couples (V,P) où V est un q-polynôme unitaire non nul de q-degré $\leq t$ et P est un q-polynôme de q-degré $< k$, vérifiant

$$V(y_i)=V\circ P(g_i),\ \forall i=1,\ldots,n.$$

Quand les coordonnées du vecteur d'entrée $\mathbf{g}$ sont linéairement indépendantes sur $GF(q)$, le théorème suivant relie sa résolution au problème du décodage borné d'un code de Gabidulin.

Théorème 3 ([Loi06])
Soient $\mathbf{y}\in GF(q^m)^n$, et $\mathbf{g}\in GF(q^m)^n$ dont les coordonnées sont linéairement indépendantes sur $GF(q)$. Si (V,P) est une solution de **Reconstruction**$(\mathbf{y},\mathbf{g},k,t)$*, alors* $(\mathbf{c}=P(\mathbf{g}),\mathbf{e}=\mathbf{y}-\mathbf{c})$ *est solution du problème* **Décodage_Borné**$(\mathbf{y},Gab_k(\mathbf{g}),t)$.

Quand le rang de l'erreur t est inférieur à la capacité de correction $\lfloor (n-k)/2\rfloor$ du code, le couple solution de **Reconstruction**$(\mathbf{y},\mathbf{g},k,t)$ est unique. On peut alors montrer que résoudre **Reconstruction**$(\mathbf{y},\mathbf{g},k,t)$ est polynomialement équivalent à la résolution de

Décodage_Borné$(\mathbf{y}, Gab_k(\mathbf{g}), t)$. Ainsi savoir *reconstruire* permet de décoder jusqu'à la capacité de correction.

Si l'on considère comme inconnues les coefficients des polynômes (V, P) solution, résoudre **Reconstruction**$(\mathbf{y}, \mathbf{g}, k, t)$ consiste à résoudre le système quadratique de n équations dont les $k + t$ inconnues sont les coefficients des q-polynômes V et P :

$$V(y_i) = V \circ P(g_i), \ \forall i = 1, \ldots, n. \tag{4.4}$$

Les techniques les plus efficaces de résolution de systèmes non-linéaires reposent sur le calcul de la base de Groebner de l'idéal engendré par le système. Cependant la complexité de tels algorithmes en temps et en mémoire dépend également du degré du système, ainsi que du nombre de solutions au système, cf. [Fau99, Fau02]. Partant, si la taille de la liste des solutions du système est importante, la complexité de résolution du problème est elle-aussi élevée.

Pour simplifier, on *linéarise* le système, c'est-à-dire on considère le système linéaire

$$V(y_i) = N(g_i), \quad \forall i = 1, \ldots, n, \tag{4.5}$$

dont les inconnues sont les q-polynômes V de q-degré t et N de q-degré $k + t - 1$. On a le résultat suivant :

Proposition 18 ([Loi04])
Si (V, P) *est une solution de* (4.4)*, alors* $(V, V \circ P)$ *est une solution de* (4.5).

Ce résultat montre que l'on peut rechercher les solution du système quadratique comme des instances particulières du système linéaire (4.5). Dans le cas où le rang de l'erreur t est inférieur à la capacité de correction du code, la proposition suivante permet de relier les deux ensembles de solution, démontrant qu'il suffit de résoudre le système linéaire pour obtenir l'unique solution du système quadratique.

Proposition 19 ([Loi06])
Si $t \leq \lfloor (n - k)/2 \rfloor$*, et* (4.4) *a une solution* (V, P) *où* $V \neq 0$*, alors l'espace vectoriel des solutions de* (4.5) *est de dimension* 1 *et toute solution non-nulle de* (4.5) *donne l'unique solution de* (4.4)*, où* V *est unitaire.*

Il suffit d'effectuer dans l'anneau des q-polynômes la division de N par V où le couple (V, N) est solution de (4.5).

Pour résoudre ce système linéaire, on a envisagé deux approches qui donnent lieu à la conception de deux algorithmes.

3.1 Un algorithme naturel

L'idée qui vient naturellement consiste à résoudre le système par des techniques standards de résolution de systèmes linéaires comme le *pivot de Gauss*. La matrice du système linéaire a la forme :

$$\left(\begin{array}{ccc|ccc} g_1 & \cdots & g_1^{[k+t-1]} & y_1 & \cdots & y_1^{[t]} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ g_n & \cdots & g_n^{[k+t-1]} & y_n & \cdots & y_n^{[t]} \end{array}\right),$$

sa partie gauche étant indépendante du vecteur $\mathbf{y}$ reçu. En effectuant des pré-calculs, on trouve une base des solutions du système en $(k+t)(k+t^2+2t)+t^3/2$ multiplications dans

$GF(q^m)$. Ajoutant à cela la complexité d'une division afin de retrouver le mot de code, l'algorithme fonctionne en approximativement $(k+t)(k+t^2+3t)+t^3/2$ multiplications dans $GF(q^m)$, cf. [Loi04, Loi06].

3.2 Un algorithme plus efficace

L'algorithme de résolution précédent comporte encore un terme cubique en le rang de l'erreur à corriger. Ce terme est dû au fait que l'on ne dispose pas d'algorithme permettant d'inverser des matrices structurées autrement qu'avec des algorithmes génériques d'inversion matricielle.

Il existe toutefois une approche alternative qui s'inspire de celle de Berlekamp et de Welch pour les polynômes. Celle-ci consiste à reconstruire les q-polynômes candidats (V et N sont des q-polynômes) en prenant deux suites de polynômes *correctement* initialisées et, à chaque étape, en incrémentant alternativement les q-degrés de l'une des suites tout en gardant les q-degrés de l'autre suite constants, cf. [BW86].

On cherche à construire des suites de q-polynômes $\left(V_0^{(i)}(y), N_0^{(i)}(x)\right)$ et $\left(V_1^{(i)}(y), N_1^{(i)}(x)\right)$, qui vérifient pour tout $i \leq n$ la propriété suivante

$$\forall k \leq i, \begin{cases} V_0^{(i)}(y_k) - N_0^{(i)}(g_k) = 0, \\ V_1^{(i)}(y_k) - N_1^{(i)}(g_k) = 0. \end{cases}$$

Si, pour un certain i on dispose de deux tels couples, on montre qu'on peut en déduire deux autres couples satisfaisant la propriété pour $i+1$ en augmentant le q-degré des q-polynômes de l'un des deux couples de 1 tandis que le q-degré des q-polynômes de l'autre couple demeure constant.

L'algorithme ainsi obtenu est décrit dans le tableau 4.1 et se décompose en deux étapes :

1. Une étape d'initialisation des polynômes au cours de laquelle on calcule $(N_0, V_0) = (Int_{g_1,\ldots,g_k}, 0)$, où $Int_{g_1,\ldots,g_k}(x)$ est l'unique q-polynôme unitaire de q-degré k tel que
$$\forall i = 1, \ldots, k, \quad Int_{g_1,\ldots,g_k}(g_i) = 0.$$
On peut rapprocher cette méthode de la technique de réécriture dans l'algorithme de Sudan telle qu'elle est décrite par Koetter et Vardy dans [KV03].
Pour l'autre couple de polynômes, on détermine N_1 comme étant l'unique q-polynôme de q-degré $k-1$, interpolant $y_1, \ldots, y_k$ sur les éléments $g_1, \ldots, g_k$. Pour ce faire, on pré-calcule une base $\mathcal{P}_i$ des polynômes d'interpolation, et on écrit
$$N_1(x) \stackrel{def}{=} \sum_{i=1}^{k} y_i \mathcal{P}_i.$$
Ainsi, comme les $\mathcal{P}_i$, et $Int_{g_1,\ldots,g_k}$ peuvent être pré-calculés, la complexité de cette étape est exactement de k^2 multiplications dans $GF(q^m)$.
2. Une étape accroissant le q-degré de l'un des couples tandis que l'autre reste constante. La complexité de cette étape vaut
$$\sum_{i=k+1}^{n} (4i-1) = 2n^2 - 2k^2 + (n-k)$$
multiplications dans $GF(q^m)$.

Entrée :
- $Gab_k(\mathbf{g})$,
- $\mathbf{y} = (y_1, \ldots, y_n)$ à distance rang $t = \lfloor (d-1)/2 \rfloor$ de $Gab_k(\mathbf{g})$.

Sortie : Un couple de q-polynômes (V_1, N_1) vérifiant (4.5)

1. *Etape d'initialisation :*
   - $V_0(y) \leftarrow 0$ et $V_1(y) \leftarrow y$,
   - $N_0(x) \leftarrow Int_{g_1,\ldots,g_k}(x)$ et $N_1(x) \leftarrow \sum_{i=1}^{k} y_i \mathcal{P}_i$.
2. *Etape d'accroissement alterné des degrés*

   Pour $i \in \{k+1, \ldots, n\}$ faire
   - $s_0 \leftarrow V_0(y_i) - N_0(g_i)$
   - $s_1 \leftarrow V_1(y_i) - N_1(g_i)$
   - $\lambda \leftarrow s_0/s_1$,
   - Echanger N_0 et N_1, V_0 et V_1, s_0 et s_1
   - Calculer
     (a) $N_1(x) \leftarrow N_1(x) - \lambda N_0(x)$,
     (b) $V_1(y) \leftarrow V_1(y) - \lambda V_0(y)$,
     (c) $N_0(x) \leftarrow N_0(x)^q - s_0 N_0(x)$,
     (d) $V_0(y) \leftarrow V_0(y)^q - s_0 V_0(y)$.

   Fin pour
3. Retourner (V_1, N_1).

TABLEAU 4.1 : Algorithme de résolution du système (4.5)

La complexité totale de l'algorithme en comptant la division euclidienne des polynômes (N_1, V_1) (pour la complexité cf. la proposition 11) est donc de $\approx 2n^2 - k^2 + kt$ multiplications dans $GF(q^m)$, complexité quadratique pour laquelle on s'affranchit du terme en t^3.

4 Pistes de recherche

Liens entre codes de Gabidulin et codes GRS Les codes de Gabidulin sont des codes MRD donc en particulier MDS. Or, la seule famille de connue de codes MDS disposant d'algorithmes de décodage en temps polynomial est la famille des codes de Reed–Solomon généralisés (codes GRS).

Etant donnés les paramètres d'un code de Gabidulin, il existe des codes GRS ayant exactement les mêmes paramètres et à coefficients dans le même espace ambiant. Une question peut alors se poser : Existe-t-il des transformations simples comme des permutations ou des isométries semi-linéaires de la métrique de Hamming qui transforment les codes de Gabidulin en codes GRS ?

Une première approche possible consiste à étudier la forme de la matrice génératrice d'un code GRS mise sous forme systématique. Comme les k premières positions forment un ensemble d'information, si on fixe ces positions, la forme systématique est unique et se représente sous la forme

$$\mathbf{G}_{\text{syst}} = (\mathbf{I} \mid \mathbf{R}),$$

où la matrice $\mathbf{R}$ est une matrice de Cauchy généralisée (cf. [RS85]) c'est-à-dire dont le terme général R_{ij} est de la forme

$$R_{ij} = \frac{c_i d_j}{x_i + y_j}.$$

La proposition 15 donne la structure du terme général de la partie redondante de la forme systématique pour un code de Gabidulin. Pour un code $Gab_k(\mathbf{g})$, savoir s'il est un code GRS revient à savoir si l'on peut trouver des solutions à l'équation

$$P_i(g_j) = \frac{c_i d_j}{x_i + y_j},$$

dont les inconnues sont P_i, c_i, d_j, x_i et y_j pour $i = 1, \ldots, k$ et $j = k+1, \ldots, n$.

Algorithmes de décodage Le problème du décodage des codes de Gabidulin offre plusieurs directions de recherche :

- On peut tenter d'éliminer le facteur cubique en l'erreur des algorithmes d'Euclide étendu et de Berlekamp-Massey en métrique rang. Pour cela il faudrait trouver un algorithme quadratique d'inversion d'une matrice du type

$$\begin{pmatrix} x_1 & \cdots & x_t \\ \vdots & \ddots & \vdots \\ x_1^{[t-1]} & \cdots & x_t^{[t-1]} \end{pmatrix},$$

  où les x_i sont linéairement indépendants sur $GF(q)$. Pour une telle matrice que l'on peut qualifier de matrice de Vandermonde en métrique rang, le problème demeure ouvert.
- Dans un second temps, on peut tenter de réduire la complexité de l'algorithme décrit dans le tableau 4.1, en améliorant les complexités des opérations sur les q-polynômes. Par exemple, les étapes d'interpolation ou bien d'évaluation sur des éléments d'un corps fini. Comme on l'a mentionné à la section 4 du chapitre 3, cela pourrait passer par la conception d'un algorithme de type Karatsuba pour les q-polynômes puis appliquer des méthodes que l'on trouve pour les polynômes classiques, cf. le livre [GG03].
- Dans un troisième temps, on peut se demander s'il est possible de construire un algorithme de décodage en liste en temps polynomial du type algorithme de Sudan ou Guruswami-Sudan pour les codes de Gabidulin. En effet, les codes de Gabidulin étant optimaux, on peut espérer que la taille moyenne de la liste des mots candidats demeure polynomiale pour une erreur de rang inférieur à $n - \sqrt{nk}$, cf. [Fau06], ainsi que l'équation (2.3), pour un code de longueur n et de dimension k sur $GF(q^n)$. Pour imiter la méthode de M. Sudan il devient alors nécessaire de définir ce que recouvre la notion de polynôme linéaire multivarié, et de relier ses propriétés à des propriétés sur les racines. Le fait que la loi de composition des q-polynômes ne soit pas commutative constitue un sérieux écueil à la réalisation d'un tel algorithme.

Chapitre 5

Codes construits à partir de codes de Gabidulin

L'existence d'algorithmes de décodage en temps polynomial résolvant le problème du décodage borné par un entier est une propriété qui se transfère d'un code à ses sous-codes, en particulier aux sous-codes sur des sous-corps (encore appelés *sous-codes trace*) ainsi qu'aux sous-code sur des sous-espaces. Parmi les familles de codes les plus populaires en métrique de Hamming, on retrouve fréquemment les *sous-codes trace* des codes de Reed-Solomon généralisés (codes GRS), tels les *codes alternants* dont font partie les codes BCH ainsi que les codes de Goppa qui se trouvent au cœur du système de chiffrement de McEliece, cf. [Gop70, McE78, MS77]. La projection de l'alphabet du code sur un sous-corps a pour conséquence de masquer la structure du code et de protéger le système contre les attaques par structure. Pour les codes de Goppa par exemple, il n'existe pas de *distingueur*[1] fonctionnant en temps polynomial, c'est-à-dire d'algorithme en temps polynomial permettant de les différencier de codes aléatoires. Bien que les paramètres fondamentaux des *codes parents* soient parfaitement connus et que l'on puisse exprimer les mots du code dual en fonction de celui-ci et de l'action de l'opérateur Trace, les paramètres fondamentaux des sous-codes ne sont évalués au mieux, que par des bornes inférieures, cf. [Del75, Sti90].

Au début des années 90, G. Solomon s'intéressa à des sous-codes particuliers de codes de Reed-Solomon (codes RS). Pour ceux-ci, il ne regardait pas la projection des coordonnées du code sur un sous-corps, mais sur un sous-espace vectoriel du corps fini constituant l'alphabet. L'intérêt principal exprimé consistait à construire de nouveau codes avec de bonnes conditions d'optimalité tout en s'affranchissant de la contrainte inhérente aux codes RS, à savoir que la longueur du code est nécessairement inférieure au cardinal du code, cf. [Sol92]. Peu après R. J. McEliece et G. Solomon étudièrent une famille plus générale de codes, *Trace shortened RS codes* qui sont encore dérivés des codes RS. Finalement, en 1998 les résultats obtenus furent généralisés à l'ensemble des sous-codes sur des sous-espaces de codes RS, cf. [MS94, HMS98]. Les résultats essentiels de ces trois articles reposent sur l'obtention de bornes sur le cardinal des sous-codes ainsi que sur leurs distances minimales respectives, en fonction des propriétés du sous-espace vectoriel considéré. Hormis ces résultats qui sont similaires à des résultats existants sur les sous-codes trace de codes RS, un des inconvénients majeurs de ces constructions réside dans le fait que les codes obtenus ne sont pas linéaires mais simplement additifs.

[1] pour une définition rigoureuse d'un distingueur on pourra se référer au document d'habilitation à diriger des recherches de N. Sendrier [Sen01]

Donc, même la conception d'un algorithme d'encodage efficace est par elle-même problématique. Une solution partielle fut donnée par M. van Dijk et L. Tolhuisen en 1998, cf. [DT99]. Elle n'est cependant pas optimale, car elle ne permet pas d'encoder toute l'information possible.

Ces démarches et les résultats qui en découlèrent concernant les codes RS nous conduisirent avec E. M. Gabidulin à refaire une étude similaire pour le cas des codes de Gabidulin qui présentent de grandes ressemblances avec les codes RS. Comme ce sont des codes optimaux qui, de par leur construction, ont encore plus de structure que les codes RS et qui de plus se trouvent au cœur de l'ensemble des cryptosystèmes reposant sur la métrique rang proposés jusqu'à ce jour (cf. les chapitres 6 et 7), il nous a paru naturel d'étudier les familles de sous-codes correspondant afin d'évaluer dans quelle mesure la projection sur un sous-corps ou sur un sous-espace affectait leur structure. Ce chapitre décrit les résultats de ce travail qui sont en outre publiés dans [GL00, GL04, GL05].

Dans une première partie on démontre que les sous-codes sur des sous-espaces sont structurellement équivalents à des codes de Gabidulin de longueur plus petite. Ensuite on montre que si l'on considère le code formé par la somme directe de tels sous-codes, il est possible de décoder au-delà de la distance rang minimale théorique. Dans un troisième temps on étudie plus spécifiquement une classe de *sous-codes trace* dérivant de codes dont la longueur est égale au degré de l'extension de corps. Dans une dernière partie enfin, on décrit la famille des *codes rangs réductibles* qui furent définis dans [OGHA03]. Leur construction s'inspire de la structure des *sous-codes trace* de codes de Gabidulin. Ils furent conçus à des fins spécifiquement cryptographiques, en ajoutant à une matrice génératrice de code produit de codes de Gabidulin des matrices destructurantes.

1 Sous-codes sur des sous-espaces

Dans la suite du chapitre et par commodité, on désigne par $\mathcal{G}$ le code de Gabidulin $Gab_k(\mathbf{g})$ de distance minimale d donné par sa matrice de parité sous la forme (4.2).

Soit V_s un sous-espace vectoriel de $GF(q^m)$ de dimension $s < d-1$ donné par une base $\mathbf{b} = (\beta_1, \ldots, \beta_s)$ d'éléments linéairement indépendants sur $GF(q)$. Soit

$$(\mathcal{G}|V_s) \stackrel{def}{=} \{\mathbf{c} = (c_1, \ldots, c_n) \in \mathcal{G} \mid c_j \in V_s,\ j = 1, \ldots, n\},$$

le sous-code du code $\mathcal{G}$, constitué des vecteurs dont les coordonnées appartiennent à V_s, comme défini dans [Jen95], par exemple. On considère de plus la matrice suivante obtenue à partir de la base $(\beta_1, \ldots, \beta_s)$

$$\mathbf{H}_{V_s} = \begin{pmatrix} \beta_1^{[m]} & \cdots & \beta_s^{[m]} \\ \vdots & \ddots & \vdots \\ \beta_1^{[m-d+2]} & \cdots & \beta_s^{[m-d+2]} \end{pmatrix}, \tag{5.1}$$

et on définit $\mathcal{LG}(V_s)$ comme étant le code de matrice de parité $\mathbf{H}_{V_s}$. Ce code est un code de Gabidulin de paramètres $[s, s-d+1, d]$. Le code $\mathcal{LG}(V_s)$ caractérise totalement le code $(\mathcal{G}|V_s)$ dans le sens que

Proposition 20 ([GL04])

Il existe une application bijective, linéaire sur $GF(q)$ entre $(\mathcal{G}|V_s)$ et $\mathcal{LG}(V_s)$ qui

1. *préserve le rang des vecteurs,*

2. est calculable ainsi que son inverse en temps polynomial.

L'existence de cet isomorphisme entre un sous-code sur un sous-espace et un code de Gabidulin permet, de constater que $(\mathcal{G}|V_s)$ est un code MRD, dont la distance rang minimale est exactement celle du code dont il dérive.

D'autre part, comme l'application préserve le rang des vecteurs et qu'elle est *constructible* en temps polynomial, cela permet de déduire des algorithmes d'encodage et de décodage spécifiques à partir des algorithmes d'encodage et de décodage de $\mathcal{LG}(V_s)$.

2 Somme directe de sous-codes

On s'intéresse dans cette section aux sous-codes qui sont somme directe de sous-codes sur des sous-espaces. Soit $V_{m_1}, \ldots, V_{m_u}$, un ensemble de sous-espaces vectoriels de $GF(q^m)$, qui forment une somme directe. Cela implique en particulier $\sum_{i=1}^{u} m_i \leq m$. Soit

$$\mathcal{M} \stackrel{def}{=} (\mathcal{G}|V_{m_1}) \oplus \cdots \oplus (\mathcal{G}|V_{m_u}) \subset \mathcal{G},$$

le sous-code de $\mathcal{G}$ formé de la somme directe des sous-codes $(\mathcal{G}|V_{m_i})$. Alors, en étendant la bijection définie dans la section précédente à cette somme directe, on obtient

Proposition 21 ([GL04])

Il existe une application $f : V_{m_1}^n \oplus \cdots \oplus V_{m_u}^n \to GF(q^m)^{m_1} \times \cdots \times GF(q^m)^{m_u}$ bijective, linéaire sur $GF(q)$, qui préserve le rang et telle que

$$f(\mathcal{M}) = \mathcal{LG}(V_{m_1}) \times \cdots \times \mathcal{LG}(V_{m_u})$$

De cette proposition, on déduit les paramètres de $\mathcal{M}$.

Corollaire 3

$\mathcal{M}$ est un code $(n, M, d)_r$, où
- $M = q^{m \sum_{i=1}^{u} (m_i - (d-1))}$.
- $D = d$.

Les procédures d'encodage et de décodage dérivent donc des procédures utilisées pour les sous-codes $(\mathcal{G}|V_{m_i})$. Il suffit pour cela de considérer les projections de l'alphabet sur chacun des espaces vectoriels V_{m_i}. Pour le décodage il peut arriver que, bien que le rang de l'erreur considérée soit supérieure à la capacité de correction théorique $C = \lfloor (d-1)/2 \rfloor$ du code, le rang de la projection de l'erreur sur chacun des espaces V_{m_i} soit inférieure à C pour chaque i. Dans ces cas, on peut décoder des erreurs de rang $t > C$. La proposition suivante évalue la probabilité de se trouver dans de tels cas.

Proposition 22 ([GL04])

Soit $\mathcal{M} = (\mathcal{G}|V_{m_1}) \oplus \cdots \oplus (\mathcal{G}|V_{m_u})$, le sous-code de $\mathcal{G}$ de capacité de correction égale à C. Soit $\mathbf{y} = \mathbf{c} + \mathbf{e}$ où $\mathbf{c} \in \mathcal{M}$ et $\mathbf{e}$ est aléatoire de rang $t \geq C + 1$. Alors la probabilité que $\mathbf{y}$ puisse être décodé est

$$P_{\textit{décodage}} = q^{-(N-C)(t-C)+uq^{-1}+O(q^{-2})} \quad \textit{où } N = \sum_{i=1}^{u} m_i.$$

Grâce à la bijection de la proposition 20, cette probabilité est égale à la probabilité de décoder une erreur de rang t dans le code produit de u codes de Gabidulin de capacité de correction C, dont la somme totale des longueurs est inférieure au degré de l'extension m du corps.

3 Sous-codes trace

Dans toute cette section le degré m de l'extension est égal à la longueur n du code. Concernant les *sous-codes trace* des codes de Gabidulin sur le corps $GF(q^s) \subset GF(q^n)$ nous avons le théorème suivant, qui établit une bijection entre l'ensemble des codes $(\mathcal{G}|GF(q^s))$, $\mathcal{G}$ étant un code de Gabidulin de longueur n sur $GF(q^n)$ et le groupe général linéaire $\mathrm{GL}_n(GF(q))$ des matrices carrées $n \times n$ inversible à coefficients dans le corps de base $GF(q)$:

Théorème 4 ([GL00, GL04])
Considérons la matrice

$$\mathbf{A} = \begin{pmatrix} a_1 & \cdots & a_s \\ \vdots & \ddots & \vdots \\ a_1^{[d-2]} & \cdots & a_s^{[d-2]} \end{pmatrix}$$

où $a_i \in GF(q^s) \subset GF(q^n)$ pour tout $i = 1, \ldots, s$ et où les a_i sont linéairement indépendants sur $GF(q)$. Alors, il existe une unique *matrice $\mathbf{T} \in GL_n(GF(q))$ telle que*

$$\mathbf{H}_{q^s} = \begin{pmatrix} \mathbf{A} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{A} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{A} \end{pmatrix} \mathbf{T},$$

est une matrice de parité de $(\mathcal{G}|GF(q^s))$.

Ce théorème implique que $(\mathcal{G}|GF(q^s))$ est, modulo l'action du groupe général linéaire, le produit direct de codes de Gabidulin de longueur s et de distance rang minimale d. Donc, comme dans le cas du produit direct de codes sur des sous-espaces, on peut parfois décoder au-delà de la capacité de correction du code. Le corollaire suivant découle de la proposition 22 :

Corollaire 4 (Décodage des sous-codes trace)
Soit C la capacité de correction de $\mathcal{G}$, alors la probabilité qu'une erreur de rang $t > C$ soit décodable dans le code $(\mathcal{G}|GF(q^s))$ est égale à

$$P_{décodage} = q^{-(n-C)(t-C)+\frac{n}{s}q^{-1}+O(q^{-2})}.$$

4 Les codes rang réductibles

Les codes rang réductibles (codes RRC) ont été définis par Ourivski, Gabidulin, Ammar et Honary en 2003 dans [OGHA03]. Leur construction a été motivée par le fait que projeter un code de Gabidulin sur un sous-espace ne le destructurait pas suffisamment si on les destine à des usages cryptographiques.

Définition 7 ([OGHA03])

Soient $\mathbf{G}_1, \ldots, \mathbf{G}_r$*, des matrices génératrices de codes de Gabidulin de paramètres respectifs* $[n_i, k_i, d_i]_r$*, à coefficients dans* $GF(q^m)$ *et soient* $(\mathbf{A}_{ij})_{i=1,j=1}^{r-1,r-1}$ *des matrices à coefficients dans* $GF(q^m)$ *de tailles respectives* $k_i \times n_j$*. Alors le code de matrice génératrice*

$$\mathbf{G} = \begin{pmatrix} \mathbf{G}_1 & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{A}_{11} & \mathbf{G}_2 & \mathbf{0} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ \mathbf{A}_{(r-1)1} & \mathbf{A}_{(r-1)2} & \cdots & \cdots & \mathbf{G}_r \end{pmatrix} \tag{5.2}$$

est appelé code rang réductible *(code RRC) d'ordre* r*.*

De leur construction il est facile de déduire les paramètres fondamentaux des codes RRC, ainsi qu'une procédure de décodage à distance bornée par la capacité de correction qui utilise des algorithmes de décodage de codes de Gabidulin.

Proposition 23 ([OGHA03])
Le code de matrice génératrice (5.2) *est un code* $[N, K, D]_r$ *sur* $GF(q^m)$ *où*
- $N = \sum_{i=1}^{r} n_i$,
- $K = \sum_{i=1}^{r} k_i$,
- $D = \min_{i \in \{1,\ldots,r\}}(d_i)$.

Décodage La structure triangulaire par bloc de la matrice (5.2) donne une procédure de décodage particulière récursive : Supposons reçu un vecteur $\mathbf{y} = \mathbf{c} + \mathbf{e}$ où $\mathbf{e}$ est de rang t sur $GF(q)$. Alors, on découpe le vecteur $\mathbf{y}$ en r parties $\mathbf{y} = (\mathbf{y}_1, \ldots, \mathbf{y}_r)$ où l'on a

$$\begin{cases} \mathbf{y}_r & = & \mathbf{m}_r\mathbf{G}_r & & & + & \mathbf{e}_r, \\ \mathbf{y}_{r-1} & = & \mathbf{m}_{r-1}\mathbf{G}_{r-1} & + & \mathbf{m}_r\mathbf{A}_{(r-1)(r-1)} & + & \mathbf{e}_{r-1}, \\ \vdots & \vdots & & & \vdots & & \vdots \\ \mathbf{y}_1 & = & \mathbf{m}_1\mathbf{G}_1 & + & \sum_{i=2}^{r-1}\mathbf{m}_i\mathbf{A}_{i1} & + & \mathbf{e}_1. \end{cases}$$

Le fait de connaître $\mathbf{m}_r, \mathbf{m}_{r-1}, \ldots, \mathbf{m}_i$ suffit pour retrouver $\mathbf{m}_{i-1}$ par un décodage dans le code de matrice génératrice $\mathbf{G}_{i-1}$ pourvu que $\mathbf{e}_{i-1}$ soit de rang inférieur à la capacité de correction $C_{i-1} = \lfloor (d_{i-1} - 1)/2 \rfloor$.

Remarque 5
Lorsque les codes de matrice génératrice $\mathbf{G}_i$ *ont tous les mêmes paramètres* $[n, k, d]_r$*. Le code RRC résultant est, d'une part un code MRD, suivant la définition* 4 *et, d'autre part on peut décoder au-delà de la capacité de correction dans le cas où, comme pour le cas décrit à la proposition* 22 *concernant le décodage de codes produits de codes de Gabidulin de même capacité de correction.*

De fait ces codes ont une certaine structure puisque le dual d'un code RRC est un code RRC.

Codes duals des codes RRC Le dual d'un code de Gabidulin est un code de Gabidulin, mais qu'en est-il d'un code *rang réductible*?

Proposition 24 ([OGHA03])

Le dual d'un code RRC *est un* code RRC*. Plus précisément, si la matrice génératrice est donnée par la formule* (5.2)*, alors la matrice de parité du code est*

$$\mathbf{H} = \begin{pmatrix} \mathbf{H}_1 & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{B}_{11} & \mathbf{H}_2 & \mathbf{0} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ \mathbf{B}_{(r-1)1} & \mathbf{B}_{(r-1)2} & \cdots & \cdots & \mathbf{H}_r \end{pmatrix}, \tag{5.3}$$

où les matrices $\mathbf{H}_i$ *sont des matrices* $(n_i - k_i) \times n_i$ *de parité du code engendré par* $\mathbf{G}_i$*,* i.e. $\mathbf{G}_i \mathbf{H}_i^T = \mathbf{0}$.

5 Pistes de recherche

Contrairement au cas de la métrique de Hamming, le fait de projeter l'alphabet des coordonnées sur des sous-espaces de l'alphabet ne semble pas outre-mesure destructurer les codes. Il demeure cependant un point à éclaircir. Le théorème 4 ne s'applique que dans le cas où la longueur du code est égale au degré de l'extension. Afin de généraliser ce résultat aux autres cas, une piste pourrait être d'utiliser les résultats dûs à Delsarte sur la structure du dual d'un *sous-code trace*, cf. [Del75].

Troisième partie

Cryptographie en métrique rang

Chapitre 6

Cryptosystèmes de type McEliece

L'origine de ces cryptosystèmes remonte à l'aube de la cryptographie à clé publique, en 1978 [McE78]. Ils utilisent comme famille de clés des familles de codes décodables pour une certaine métrique préalablement définie. Dans le cas de la métrique de Hamming ils constituent des alternatives intéressantes aux systèmes à clé publique couramment employés car il résistent depuis tout aussi longtemps aux attaques et leur sécurité n'est pas reliée à des problèmes *difficiles* de théorie des nombres.

Ces systèmes sont caractérisés par de très bonnes performances en vitesse de chiffrement et déchiffrement. Cependant, la taille de clé les rendait jusqu'à récemment prohibitifs d'emploi sur des systèmes à ressources limitées. La conception d'un schéma de signature très courte, ainsi que l'intérêt croissant porté aux cryptosystèmes fondés sur des résolutions de systèmes multivariés qui n'ont pas de meilleures performances ont suscité un regain d'intérêt envers ces cryptosystèmes, cf. [CFS01, Sen01].

Dans leur conception, on a besoin d'une métrique et d'une famille $\mathcal{F}$ de codes linéaires pour lesquels on dispose d'algorithmes de décodage en temps polynomial jusqu'à une distance t.

La *clé privée* est alors constituée de :

1. Un code $\mathcal{C}$ choisi aléatoire dans la famille $\mathcal{F}$. Ce code peut aussi bien être donné par une matrice génératrice $\mathbf{G}$ (cas du système de McEliece classique) que par une matrice de parité $\mathbf{H}$ (on le nomme dans ce cas *système de Niederreiter*, cf. [Nie86]).
2. Une matrice $\mathbf{T}$ de taille $n \times n$ qui est une isométrie linéaire de la métrique considérée.

La forme de la *clé publique* dépend de la version du système choisie :

– Système de McEliece :

$$\mathbf{G}_{\text{pub}} = \mathbf{GT}. \tag{6.1}$$

L'algorithme de chiffrement consiste à encoder le message que l'on veut envoyer avec la matrice génératrice de la clé publique. Ensuite, pour briser la linéarité on ajoute un vecteur d'erreur de norme t.

– Système de Niederreiter :

$$\mathbf{H}_{\text{pub}} = \mathbf{HT}. \tag{6.2}$$

Pour chiffrer, on utilise une procédure transformant l'information en un vecteur dont la norme est majorée par t. On calcule ensuite le syndrome correspondant grâce à $\mathbf{H}_{\text{pub}}$. Le syndrome obtenu correspond au chiffré.

Le déchiffrement consiste à décoder le chiffré reçu ou bien en tant que mot de code bruité (cas de la version McEliece), ou bien en tant que syndrome (cas de la version Niederreiter). Les deux problèmes sont polynomialement équivalents en métrique de Hamming. La sécurité des systèmes repose sur deux assertions :

1. Le problème du **Décodage_Borné** dans la métrique que l'on considère est un problème *difficile*. Pour la métrique de Hamming les résultats de complexité donnent de bons arguments pour affirmer que la résolution de ce problème de décodage est difficile, cf. [BMvT78, Var97].

   Pour la métrique rang, bien qu'il n'existe pas de résultats théoriques sur la complexité de résolution des problèmes de décodage, on fait reposer la sécurité des systèmes sur l'état de l'art des algorithmes de décodage existants (cf. les complexités données par la proposition 9 du chapitre 2). Ces complexités montrent qu'en métrique rang, on peut espérer atteindre une taille de clé publique bien plus petite qu'en métrique de Hamming à sécurité constante, permettant du coup à ces systèmes de s'affranchir de leur inconvénient majeur.

2. Le code engendré par la clé publique est un code indistinguable (en temps polynomial) d'un code aléatoire, cf. [Sen01]. Pour choisir la famille de codes à employer, le concepteur doit prendre en compte le fait que l'attaquant peut exploiter la structure de la clé publique qui, si le code n'est pas *suffisamment* aléatoire, permet de reconstruire un algorithme de décodage en temps polynomial du code engendré par la clé publique.

Il n'est souvent pas facile de déterminer une famille de codes satisfaisant la seconde assertion car, pour que l'on puisse décoder les codes doivent par essence avoir une structure qui peut être difficile à masquer par des transformations aussi simples que des permutations des positions des codes. Ceci constitue une faiblesse. Pour la métrique de Hamming, on peut envisager d'utiliser les codes suivants :

- *Codes de Reed-Solomon* : Grâce à leurs propriétés d'optimalité, ils forment de bons candidats. Cependant, leur grande structure les condamne à n'être pas utilisés. En effet, la partie redondante de la forme systématique de leur matrice génératrice est une matrice de Cauchy généralisée. L'attaque de Sidel'nikov–Shestakov exploite cette propriété pour retrouver un décodeur par de simples interpolations de polynômes [RS85, Gab95, SS92]. Pour en détruire la structure, on proposa des alternatives comme l'utilisation de matrices de distorsion, ou bien de sous-codes convenablement choisis, cf. [GK94, BL05]. Des développements récents modèrent cependant l'intérêt d'une telle approche. En effet, C. Wieschebrink a montré que si le sous-code était de dimension trop grande, alors une généralisation de l'attaque de Sidel'nikov–Shestakov, bien que de complexité exponentielle demeurait réalisable, cf. [Wie06].
- *Codes de Reed-Muller* : Quoiqu'encore quelque peu anecdotique, cette possibilité fut proposée par V. M. Sidel'nikov en 1994. L'auteur prend comme clé privée une matrice génératrice d'un code de Reed-Muller faiblement auto-dual. Il publie ensuite la matrice permutée, cf. [Sid94]. Comme le *hull*[1] est de grande dimension, il n'existe pas d'algorithme permettant de retrouver la permutation. Cette famille de code, à première vue intéressante souffre du fait que la distance minimale, donc la capacité de correction du code, décroît exponentiellement avec l'ordre du code.
- *Codes de Goppa binaires* : C'est cette famille de codes que proposa McEliece originellement, cf. [McE78]. C'est une famille particulière de *sous-codes traces* de codes GRS. Le fait de projeter sur un sous-corps détruit la structure inhérente des codes GRS

[1]Intersection du code avec son dual

parents. En outre, ces codes ont une bonne distance minimale, cf. [MS77] chapitre 12).

Pour une description plus complète des problématiques et des enjeux des systèmes en métrique de Hamming on pourra se reporter au mémoire d'habilitation de N. Sendrier, cf. [Sen01].

Quand le premier cryptosystème utilisant la métrique rang fut publié en 1991, seule la famille des codes de Gabidulin disposait d'algorithmes de décodage en temps polynomial pour cette métrique, cf. [GPT91]. Toutefois, ces codes étant très structurés, il s'avère indispensable de briser leur structure en ajoutant une matrice de distorsion à la clé publique d'un système de type McEliece.

Depuis, d'autres méthodes permettant de masquer la structure des codes ont été utilisées, dont certaines que j'ai publiées avec T. Berger. On peut par exemple utiliser des sous-codes bien choisis de codes de Gabidulin, ou bien des *codes RRC*, cf. [BL05, GO01, OG03, OGHA03, BL04]. Malgré cela, il reste difficile de masquer efficacement leur structure et plusieurs cryptanalyses de ces systèmes ont été réalisées. D'abord deux attaques par K. Gibson contre le système originel, puis A. V. Ourivski attaqua la version Niederreiter du système. Enfin, tout récemment R. Overbeck publia des cryptanalyses des dernières versions des systèmes [Gib95, Gib96, Our03, Ove06, Ove05]. Toutes ces attaques exploitent le fait que les codes de Gabidulin sont *quasi-stables* sous l'action de l'automorphisme de Frobenius du corps.

Ce chapitre a pour objet de faire un point sur les systèmes existants et sur les attaques auxquelles ils sont soumis. Dans une première section, on présente les différentes variantes de systèmes qui ont été publiées jusqu'à présent. La seconde partie est consacrée à la remise en forme de l'approche utilisée par R. Overbeck pour cryptanalyser les versions en attaquant la structure de la clé publique. Cela permet de proposer des pistes de recherche pour construire des cryptosystèmes qui puissent résister à ces attaques.

1 Les cryptosystèmes de type McEliece en métrique rang

Dans cette section, on décrit le cryptosystème GPT originel, qui fut publié en 1991 par E. M. Gabidulin, A. V. Paramonov et O. V. Tretjakov, cf. [GPT91]. A. Ourivski et E. M. Gabidulin en proposèrent une version modifiée utilisant une matrice de distorsion à droite qui est une isométrie linéaire de la métrique rang, cf. [OG03]. Comme cette version englobe la version d'origine comme cas particulier (cf. [Ove06] et la remarque 6), on ne présentera que cette dernière.

Dans une seconde partie de la section, on présente deux variantes que nous avons publiées avec T. Berger, l'une utilisant des sous-codes de codes de Gabidulin, l'autre reposant sur l'utilisation de codes RRC. On présente également une procédure qui, pour les cryptosystèmes de ce type permet d'augmenter significativement le taux de transmission du système, ainsi que de procurer une sécurité sémantique, cf. [BL04, BL05].

1.1 Le système d'origine

Génération des clés On se place sur le corps $GF(q^m)$. La *clé privée* est constituée de

- $\mathbf{S}$, une matrice $k \times k$ à coefficients dans $GF(q^m)$ et inversible.
- $\mathbf{G}$, une matrice génératrice de taille $k \times n$ d'un code de Gabidulin de vecteur générateur $\mathbf{g} = (g_1, \ldots, g_n)$ sous forme canonique (4.1). La capacité de correction du code est alors $t = \lfloor (n-k)/2 \rfloor$.

- $\mathbf{Z}$, une matrice de taille $k \times t_1$, à coefficients dans $GF(q^m)$.
- $\mathbf{T}$, une matrice inversible de taille $(n+t_1) \times (n+t_1)$ à coefficients dans $GF(q)$. La matrice $\mathbf{T}$ est une *isométrie linéaire* de la métrique rang [Ber03].

La clé publique est une matrice de taille $k \times (n+t_1)$:

$$\mathbf{G}_{\text{pub}} = \mathbf{S}(\mathbf{G} \mid \underbrace{\mathbf{Z}}_{t_1\ cols})\mathbf{T}. \tag{6.3}$$

Si le principe du chiffrement est identique à celui présenté dans l'introduction au chapitre, le déchiffrement en revanche est différent. Supposons reçu le chiffré

$$\mathbf{y} = \mathbf{x}\mathbf{G}_{\text{pub}} + \mathbf{e},$$

où $\text{Rg}(\mathbf{e}) \leq t$. Alors le déchiffreur calcule

$$\mathbf{y}\mathbf{T}^{-1} = \mathbf{x}(\mathbf{G} \mid \mathbf{Z}) + \mathbf{e}\mathbf{T}^{-1},$$

il supprime les t_1 dernières coordonnées de $\mathbf{y}\mathbf{T}^{-1}$ et décode dans le code engendré par $\mathbf{G}$.

Remarque 6

- *Par rapport à la présentation du système de McEliece, on a rajouté une matrice de distorsion. E. M. Gabidulin a montré que cette matrice était indispensable. Dans le cas contraire, un attaquant peut recouvrer un décodeur en temps polynomial* [Gab95], *du fait notamment de la structure de la matrice génératrice mise sous forme systématique, voir la proposition* 15.
- *Le système a été présenté différemment dans l'article* [GPT91] *. La présentation ici faite en est une généralisation. En effet, pour retrouver la présentation originelle, il suffit de prendre pour* $\mathbf{G}$ *une matrice de taille* $k \times (n+t_1)$ *sur un corps à* $GF(q^m)$ *éléments où* $n + t_1 \leq m$.

1.2 Variantes

Pour répondre aux attaques de Gibson, outre la redéfinition du système que nous venons de mentionner par le choix judicieux d'une matrice de distorsion, d'autre variantes furent proposées, cf. [Gib95, Gib96]. L'idée générale sous-jacente consiste à *casser* en la contrôlant la structure des codes employés. La première méthode qui paraît naturelle quand on considère les familles de codes mises en œuvre en métrique de Hamming consiste à briser la structure par l'utilisation de sous-codes de codes de Gabidulin, cf. [BL05]. Une autre possibilité consiste à utiliser d'autres codes décodables en métrique rang tels que les *codes RRC* définis au chapitre 5, cf. [OGHA03, BL04].

Utilisation d'un sous-code Dans l'article [BL05] nous avons utilisé la forme Niederreiter du système de chiffrement. Pour casser la structure on ajoute des lignes à la matrice de parité d'un code de Gabidulin. La matrice ainsi obtenue est une matrice de parité d'un sous-code de code de Gabidulin, et en choisissant bien les lignes que l'on ajoute, on peut contrôler la complexité des attaques de Gibson.

La clé publique du système s'écrit sous la forme

$$\mathbf{H}_{\text{pub}} = \mathbf{S}\begin{pmatrix} \mathbf{H} \\ \mathbf{A} \end{pmatrix}, \tag{6.4}$$

où $\mathbf{H}$ est une matrice de parité de $Gab_k(\mathbf{g})$, $\mathbf{A}$ une matrice quelconque et $\mathbf{S}$ une matrice inversible de taille $(n-k) \times (n-k)$ à coefficients dans $GF(q^m)$. Pour utiliser le système, nous avons introduit une procédure d'encodage très rapide qui transforme le vecteur d'information en un vecteur de rang fixé, avec peu de perte.

Si ℓ désigne le nombre de lignes de la matrice $\mathbf{A}$, alors Une matrice génératrice du code public s'écrit

$$\mathbf{G}_{\text{pub}} = \mathbf{S}'\mathbf{G}, \tag{6.5}$$

où $\mathbf{G}$ est une matrice génératrice du code privé de taille $k \times n$ et $\mathbf{S}'$ est une matrice de taille $(k-\ell) \times n$.

Utilisation des codes rangs réductibles Une autre alternative fut l'utilisation des codes RRC décrits section 4 du chapitre 5. Cette alternative est présentée dans [OGHA03]. Les auteurs utilisent des *codes rang réductibles* d'ordre $r \geq 2$ en rajoutant en plus une matrice de distorsion. Je ne m'attacherai à décrire que la version que nous avons présentée avec T. Berger [BL04]. Dans notre cas, le code privé est donné par la matrice génératrice

$$\begin{pmatrix} \mathbf{G}_1 & \mathbf{0} \\ \mathbf{A} & \mathbf{G}_2 \end{pmatrix},$$

où pour $i = 1, 2$, la matrice $\mathbf{G}_i$ engendre le code de Gabidulin $Gab_k(\mathbf{g}_i)$ de longueur n_i, de distance rang minimale d_i. Le code a une distance minimale égale à $d = \min(d_1, d_2)$ et dispose d'un algorithme de décodage jusqu'à la capacité de correction (cf. la section 4 du chapitre 5). Ainsi la clé publique du cryptosystème est donnée par

$$\mathbf{G}_{\text{pub}} = \mathbf{S} \begin{pmatrix} \mathbf{G}_1 & \mathbf{0} \\ \mathbf{A} & \mathbf{G}_2 \end{pmatrix} \mathbf{T}. \tag{6.6}$$

1.3 Amélioration des systèmes

Les systèmes de type McEliece ne résistent pas aux attaques actives, attaques pour lesquelles l'attaquant n'est plus passif, mais a accès à un *oracle de déchiffrement* qui peut lui renvoyer des informations du type *ce n'est pas un chiffré valide* ou bien *c'est un chiffré valide*. Muni de ces informations supplémentaires, l'attaquant est en mesure d'*adapter* son attaque.

En particulier, ils sont vulnérables aux attaques

- *par rejeu :* un attaquant peut distinguer quand un même message a été envoyé plusieurs fois, et en extraire de l'information.
- *par réaction :* l'attaquant envoie le chiffré à l'oracle de déchiffrement, observe sa réaction, modifie sa requête en conséquence et recommence la même procédure. En utilisant cette approche, avec accès à un oracle de décodage, un attaquant peut récupérer de l'information sur le vecteur d'erreur initial. Pour la métrique de Hamming il peut récupérer des positions d'erreurs, tandis que pour la métrique rang, il récupère de l'information sur l'espace vectoriel de l'erreur.

Avec T. Berger, toujours dans [BL04], nous avons montré, sur la version McEliece du système sur des corps de caractéristique 2, comment mettre en place une procédure générale qui permet

- d'obtenir un système de sécurité équivalente à celle du système original.

- d'assurer au système une sécurité contre les attaques actives du type attaque *par rejeu* et attaque *par réaction.*
- d'augmenter le taux de transmission du système.

Si l'information à transmettre est découpée sous la forme de deux vecteurs $\mathbf{x}$ et $\widehat{\mathbf{x}}$ alors le message chiffré transmis est

$$\mathbf{y} = (\mathbf{x} + h(\mathbf{e}))\mathbf{G}_{\text{pub}} + \mathbf{e},$$

où h est une fonction de hachage convenable et $\mathbf{e} = \mathcal{P}(r, \widehat{\mathbf{x}})$ où la fonction $\mathcal{P}$ est une fonction qui prend deux paramètres en entrée et donne un vecteur de rang fixé. Les deux fonctions doivent avoir les propriétés suivantes

- La fonction de hachage que l'on suppose parfaite (si tant est que cette hypothèse puisse être légitime) sert à renforcer l'indistinguabilité du système contre les attaques par *rejeu*, et *par réaction.*
- On utilise la fonction $\mathcal{P}$ pour placer de l'information $\widehat{\mathbf{x}}$ de manière sûre (bonne diffusion) sur le vecteur d'erreur en conservant r bits d'aléa. Dans la pratique on se sert des boîtes S de l'AES.

Si N désigne la longueur du code engendré par la matrice $\mathbf{G}_{\text{pub}}$, m le degré de l'extension, et t le rang maximum de l'erreur que l'on peut tolérer, alors le gain par rapport au taux de transmission des systèmes décrits précédemment se monte à

$$\frac{(N + m - t)t - r}{mN} \text{ bits.}$$

2 Sécurité du système contre les attaques par structure

Ces attaques s'appuient sur des propriétés spécifiques des famille de codes utilisés dans la conception du système. Les premières attaques exploitant cette structure furent publiées par Gibson [Gib95, Gib96]. Plus récemment, R. Overbeck les a utilisées comme boîtes noires dans des attaques contre des versions plus récentes du cryptosystème, cf. [Ove06]. Cependant, on peut s'en prémunir en choisissant convenablement les paramètres.

Dans un second article, il a montré cette fois comment cryptanalyser les cryptosystèmes dans *presque* tous les cas [Ove05], utilisant un algorithme probabiliste dont la probabilité de succès est expérimentalement très grande. L'efficacité de son approche provient du fait que l'intersection d'un code de Gabidulin et du code obtenu en faisant agir l'automorphisme de Frobenius sur ce code est de grande dimension.

Dans cette section, je reprends l'idée d'Overbeck pour construire une attaque contre le système GPT. Bien que de complexité légèrement supérieure, elle a le mérite de pouvoir être décrite simplement et de permettre de prouver que dans certains cas raisonnables l'attaque fonctionne. Pour le cas des systèmes utilisant des codes rang réductibles la même idée conduit à l'élaboration d'un même type d'attaque. Il convient donc d'être prudent dans le choix des paramètres des cryptosystèmes.

2.1 Attaque contre le système GPT

La clé publique est donnée par la matrice $\mathbf{G}_{\text{pub}}$ de l'équation (6.3). On rappelle que la notation $\mathbf{G}^{[i]}$ pour une matrice $\mathbf{G}$ désigne la matrice formée par les coefficients de $\mathbf{G}$ élevés à la puissance $[i] = q^i$.

Si l'on élève tous les coefficients de $\mathbf{G}_{\text{pub}}$ à différentes puissances de l'automorphisme de Frobenius, par linéarité on obtient :

$$\underbrace{\begin{pmatrix} \mathbf{G}_{\mathrm{pub}} \\ \mathbf{G}_{\mathrm{pub}}^{[1]} \\ \vdots \\ \mathbf{G}_{\mathrm{pub}}^{[n-k-1]} \end{pmatrix}}_{\mathcal{G}_{\mathrm{pub}}} = \underbrace{\begin{pmatrix} \mathbf{S} & 0 & \cdots & 0 \\ 0 & \mathbf{S}^{[1]} & \ddots & 0 \\ \vdots & 0 & \ddots & \vdots \\ 0 & \cdots & & \mathbf{S}^{[n-k-1]} \end{pmatrix}}_{\mathcal{S}} \underbrace{\left(\begin{array}{c|c} \mathbf{G} & \mathbf{Z} \\ \mathbf{G}^{[1]} & \mathbf{Z}^{[1]} \\ \vdots & \vdots \\ \mathbf{G}^{[n-k-1]} & \mathbf{Z}^{[n-k-1]} \end{array}\right)}_{(\mathcal{G} \mid \mathcal{Z})} \mathbf{T}, \tag{6.7}$$

avec

- La matrice $\mathcal{G}_{\mathrm{pub}}$ de taille $k(n-k) \times n$ est une matrice de rang exactement $n-1$, de par les propriétés des code de Gabidulin.
- Comme $\mathbf{S}$ est inversible, la matrice $\mathcal{S}$ est également inversible.
- La matrice $\mathbf{T}$ étant à coefficient dans le corps de base $GF(q)$, pour tout entier i, $\mathbf{T}^{[i]} = \mathbf{T}$, et $\mathbf{T}$ est de rang $n + t_1$.
- La matrice $\mathcal{Z}$ de taille $k(n-k) \times t_1$ est de rang $s \leq \min(k(n-k), t_1)$.

Par la suite on se place dans le cas raisonnable où t_1 est nettement inférieur à $k(n-k)$. Dans ce cas si la matrice $\mathbf{Z}$ est choisie aléatoirement, la matrice $\mathcal{Z}$ a de *bonnes chances* d'être de rang exactement t_1. Ainsi, et des simulations faites en MAGMA le confirment la matrice $\mathcal{G}_{\mathrm{pub}}$ est *très probablement* de rang $n + t_1 - 1$ et donc son noyau *à droite* est de rang égal à 1.

Dans un tel cas il est singulier de constater que l'on peut retrouver en temps polynomial un décodeur du code public cassant dès lors le système complètement.

Proposition 25

Si le noyau à droite $\ker(\mathcal{G}_{\mathrm{pub}})$ *de* $\mathcal{G}_{\mathrm{pub}}$ *est de dimension* 1, *alors*

- *Il existe un vecteur* $\mathbf{h}$ *de rang* n *sur* $GF(q)$ *tel que*

$$\ker(\mathcal{G}_{\mathrm{pub}}) = \{\ \mathbf{T}^{-1}(\alpha\mathbf{h} \mid \mathbf{0})^T \mid \alpha \in GF(q^m)\}.$$

- *Si* $\mathbf{y} \in \ker(\mathcal{G}_{\mathrm{pub}})$, *alors toute matrice* $\mathbf{Q}$ *de taille* $(n+t_1) \times (n+t_1)$ *à coefficients dans* $GF(q)$ *vérifiant* $\mathbf{Q}\mathbf{y} = (\mathbf{x} \mid \mathbf{0})^T$, *est inversible et vérifie*

$$\mathbf{T}\mathbf{Q}^{-1} = \begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{0} & \mathbf{D} \end{pmatrix},$$

où $\mathbf{A}$ *de taille* $n \times n$ *inversible, et* $\mathbf{D}$ *est de taille* $t_1 \times t_1$ *inversible. On peut trouver une telle matrice* $\mathbf{Q}$ *en temps polynomial.*

Preuve

- Comme le noyau de $\mathcal{G}_{\mathrm{pub}}$ est de dimension 1, cela implique que le noyau de $(\mathcal{G} \mid \mathcal{Z})$ est de la forme $(\alpha\mathbf{h} \mid \mathbf{0})$, où $\mathbf{h}$ engendre le noyau de $\mathcal{G}$. Or $\mathcal{G}$ engendre un code de Gabidulin de dimension $n-1$ dont le *dual* est un code de Gabidulin de dimension 1 de vecteur générateur $\mathbf{h}$. Cela implique en particulier que $\mathbf{h}$ est de rang n sur $GF(q)$.
- Soit $\mathbf{y} \in \ker(\mathcal{G}_{\mathrm{pub}})$. De par la structure du noyau précédemment décrite, on a $\mathbf{y} = \mathbf{T}^{-1}(\alpha\mathbf{h} \mid \mathbf{0})^T$. Supposons que l'on a déterminé une matrice q-aire inversible $\mathbf{Q}$ telle que

$$\mathbf{Q}\mathbf{y} = (\mathbf{x} \mid \mathbf{0})^T = \mathbf{Q}\mathbf{T}^{-1}(\alpha\mathbf{h} \mid \mathbf{0})^T.$$

Si on découpe la matrice $\mathbf{Q}\mathbf{T}^{-1}$ en quatre blocs tels que

$$\mathbf{Q}\mathbf{T}^{-1} = \begin{pmatrix} \mathbf{A}' & \mathbf{B}' \\ \mathbf{C}' & \mathbf{D}' \end{pmatrix},$$

alors on a $\mathbf{C}'\mathbf{h}^T = \mathbf{0}$, soit pour tout $i = 1, \dots t_1$, $\mathbf{c}_i\mathbf{h}^T = 0$ où $\mathbf{c}_i$ désigne la $i$ème ligne de la matrice $\mathbf{C}'$. Comme les coefficients de $\mathbf{C}'$ sont dans $GF(q)$ et que $\mathbf{h}$ est de rang n sur $GF(q)$, donc $\alpha\mathbf{h}$ est également de rang n sur $GF(q)$ et, pour tout $i = 1, \dots, t_1$ on a $\mathbf{c}_i = \mathbf{0}$. En outre, comme $\mathbf{Q}$ est inversible, l'inverse $(\mathbf{Q}\mathbf{T}^{-1})^{-1} = \mathbf{T}\mathbf{Q}^{-1}$ est également triangulaire supérieure par bloc.
Etant donné $\mathbf{y} \in \ker(\mathcal{G}_{\text{pub}})$, on détermine une matrice $\mathbf{Q}$ inversible candidate en utilisant la procédure suivante :

1. Résoudre l'équation $\mathbf{Q}_2\mathbf{y}^T = \mathbf{0}$ où $\mathbf{Q}_2$ est une matrice de taille $t_1 \times (n + t_1)$ et de rang t_1.
2. Déterminer une matrice $\mathbf{Q}_1$ telle que

$$\mathbf{Q} \stackrel{def}{=} \begin{pmatrix} \mathbf{Q}_1 \\ \mathbf{Q}_2 \end{pmatrix},$$

   soit inversible.

Concernant le second point de l'algorithme, le vecteur $\mathbf{y}$ est de rang n sur $GF(q)$ donc la matrice $\mathbf{Y}$ que l'on obtient en étendant les coefficients de $\mathbf{y}$ sur une base de $GF(q^m)/GF(q)$ est de taille $m \times (n + t_1)$, et de rang n. Partant, la dimension du noyau à droite de $\mathbf{Y}$ est égale à t_1. Déterminer $\mathbf{Q}_2$ revient donc à déterminer une base du noyau à droite de $\mathbf{Y}$, puisque on résout alors $\mathbf{Y}\mathbf{Q}_2^T = 0$. Ceci se fait en temps polynomial.

■

Lorsque $\ker(\mathcal{G}_{\text{pub}})$ est de rang 1, on applique la proposition précédente pour trouver une matrice $\mathbf{Q}$ qui vérifie alors

$$\mathbf{G}_{\text{pub}}\mathbf{Q}^{-1} = \mathbf{S}(\mathbf{G}\mathbf{A} \mid \mathbf{Z}').$$

Comme $\mathbf{A}$ est une matrice inversible à coefficient dans le corps de base $GF(q)$, la matrice $\mathbf{G}' = \mathbf{G}\mathbf{A}$ est une matrice génératrice du code $Gab_k(\mathbf{g}\mathbf{A})$. En désignant par $\mathbf{G}_1$ les n premières colonnes de la matrice $\mathbf{G}_{\text{pub}}\mathbf{Q}^{-1}$, on doit résoudre l'équation

$$\mathbf{G}_1 = \mathbf{S}\mathbf{G}',$$

c'est-à-dire que $\mathbf{G}_1$ est une matrice génératrice aléatoire de $Gab_k(\mathbf{g}\mathbf{A})$ et l'on souhaite retrouver une matrice sous forme systématique. La manière de procéder a été décrite par Gabidulin dans [Gab95], la matrice $\mathbf{S}$ ainsi déterminée est unique.

On vient de prouver la proposition suivante

Proposition 26

Si le noyau à droite de la matrice $\mathcal{G}_{\text{pub}}$ donnée par l'équation (6.7) *est de dimension* 1, *alors on peut retrouver en temps polynomial des matrices $\mathbf{Q}, \mathbf{S}$ et $\mathbf{Z}$ telles que*

$$\mathbf{G}_{\text{pub}}\mathbf{Q}^{-1} = \mathbf{S}(\mathbf{G}' \mid \mathbf{Z}'),$$

où

- $\mathbf{Q}$ *est de taille $(n + t_1) \times (n + t_1)$ à coefficient dans $GF(q)$,*
- $\mathbf{S}$ *est de taille $k \times k$ inversible,*
- $\mathbf{G}'$ *engendre un code de Gabidulin de longueur n, de dimension k,*
- $\mathbf{Z}'$ *est une matrice de taille $k \times t_1$.*

Choix de paramètres Comme la dimension du noyau à droite de $\mathcal{G}_{\text{pub}}$ dépend du rang de $\mathcal{Z}$ qui lui-même dépend du rang de la matrice $\mathbf{Z}$, Le corollaire suivant nous donne une condition suffisante pour que la dimension du noyau $\ker(\mathcal{G}_{\text{pub}})$ soit supérieure à un entier ℓ.

Corollaire 5

Soit $\mathbf{G}_{\text{pub}} = \mathbf{S}(\mathbf{G} \mid \mathbf{Z})\mathbf{T}$ *de taille* $k \times (n + t_1)$*. Si il existe un entier* ℓ *tel que*

$$1 \leq \text{Rg}(\mathbf{Z}) \leq \frac{t_1 - \ell}{n - k},$$

alors la dimension de $\ker(\mathcal{G}_{\text{pub}})$ *est supérieure ou égale à* $1 + \ell$*.*

Preuve

Si $s = \text{Rg}(\mathbf{Z})$, alors $\text{Rg}(\mathcal{Z}) \leq s(n-k)$ donc $\text{Rg}(\mathcal{G}_{\text{pub}}) \leq s(n-k)+n-1$. Donc si $s(n-k) \leq t_1 - \ell$, alors le noyau à droite de $\mathcal{G}_{\text{pub}}$ est de dimension supérieure ou égal à $1+\ell$.
■

L'attaque décrite précédemment ne s'appliquant plus dès que $\ell \geq 1$, il *suffit* de choisir la matrice de distorsion $\mathbf{Z}$ telle que

$$\text{Rg}(\mathbf{Z}) \leq \frac{t_1 - \ell}{n - k}.$$

Cela implique de choisir $t_1 > (n - k)$. Pour atteindre une sécurité suffisante, on peut voir facilement que la taille de la clé publique devient ainsi conséquente.

Exemple 3

Pour qu'un cryptosystème puisse être considéré comme sûr*, la complexité de la meilleure attaque doit être au moins de* 2^{80} *opérations binaires. Outre la complexité des attaques par décodages de Ourivski et Johannson décrites dans la section* 1 *du chapitre* 2*, il faut prendre en compte les conditions données par le corollaire* 5*. Ces conditions donnent des taux de transmission relativement faibles. Afin d'y remédier partiellement, on peut en s'appuyant sur le modèle décrit dans la section* 1.3*, augmenter le taux de transmission d'une valeur*

$$\frac{(m+n-t)t-r}{m(n+t_1)}.$$

La quantité r *désigne un nombre de bits aléatoires assurant une sorte de* sécurité sémantique *au système. Cependant, ce remède n'est que partiel, puisque la définition du taux de transmission (* $= k/(n+t_1)$*), ainsi que la contrainte sur* t_1 *donnée dans le corollaire* 5*, font que la borne supérieure est égale à* $k/((s+1)n - sk + \ell)$*. Un jeu de paramètres envisageables sont donnés par le tableau* 6.1*.*

$m = n$	k	s	t_1	Taille clé	Décodage	k/n	Transfo. BeLoi2004
24	12	3	40	18 432bits	$> 2^{83}$	19%	30%
24	12	4	52	21 888bits	$> 2^{83}$	$15,8\%$	$24,3\%$

Tableau 6.1 : Paramètres possibles pour le système GPT

2.2 Attaque contre les variantes

Dans le cas des codes RRC on peut transposer l'approche de la section précédente. On se limitera à l'étude du cas pour un code RRC d'ordre 2, car c'est le type de code employé dans la conception des cryptosystèmes, cf. [OGHA03, BL04], et que l'approche se transpose de la même manière aux codes RRC d'ordre supérieur.

Clé publique Sous sa forme la plus générale, la clé publique d'un cryptosystème utilisant un *code RRC* d'ordre 2 est donnée par l'équation (6.6), soit

$$\mathbf{G}_{\text{pub}} = \mathbf{S}\begin{pmatrix} \mathbf{G}_1 & \mathbf{0} \\ \mathbf{A} & \mathbf{G}_2 \end{pmatrix}\mathbf{T},$$

où

- pour $i = 1, 2$, $\mathbf{G}_i$ est matrice génératrice d'un code $Gab_{k_i}(\mathbf{g}_i)$ de longueur n_i.
- $\mathbf{A}$ est une matrice quelconque de taille $k_2 \times n_1$ dont le rang est s.
- $\mathbf{S}$ est inversible à coefficients dans $GF(q^m)$.
- $\mathbf{T}$ est inversible à coefficients dans $GF(q)$.

Principe de l'attaque L'équation suivante s'obtient à partir de (6.6), en *empilant* les matrices publiques dont les coefficients sont élevés aux puissances successives de l'automorphisme de Frobenius.

$$\underbrace{\begin{pmatrix} \mathbf{G}_{\text{pub}} \\ \vdots \\ \mathbf{G}_{\text{pub}}^{[n_2-k_2-1]} \end{pmatrix}}_{\mathcal{G}_{\text{pub}}} = \underbrace{\begin{pmatrix} \mathbf{S} & \cdots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots & \mathbf{S}^{[n_2-k_2-1]} \end{pmatrix}}_{\mathcal{S}} \underbrace{\left(\begin{array}{l|l} \mathbf{G}_1 & \mathbf{0} \\ \mathbf{A} & \mathbf{G}_2 \\ \hline \vdots & \vdots \\ \hline \mathbf{G}_1^{[n_2-k_2-1]} & \mathbf{0} \\ \mathbf{A}^{[n_2-k_2-1]} & \mathbf{G}_2^{[n_2-k_2-1]} \end{array}\right)}_{(\mathcal{Z} \mid \mathcal{G})} \mathbf{T}. \tag{6.8}$$

La matrice $\mathcal{G}$ de l'équation engendre le code $Gab_{n_2-1}(\mathbf{g}_2)$, de co-dimension 1. Ainsi, pourvu que le noyau à droite de $\mathcal{G}_{\text{pub}}$ soit de dimension 1, la méthode décrite dans la preuve de la proposition 25 s'applique et on obtient :

Proposition 27
Si le noyau à droite de la matrice $\mathcal{G}_{\text{pub}}$ donnée par l'équation (6.8) *est de dimension* 1, *alors on peut retrouver en temps polynomial une matrice triangulaire inférieure par bloc inversible, à coefficients dans $GF(q)$*

$$\mathbf{Q} = \begin{pmatrix} \mathbf{Q}_1 & \mathbf{0} \\ \mathbf{Q}_3 & \mathbf{Q}_2 \end{pmatrix},$$

telle que

$$\mathbf{G}_{\text{pub}}\mathbf{Q} = \mathbf{S}\begin{pmatrix} \mathbf{G}_1\mathbf{Q}_1 & \mathbf{0} \\ \mathbf{A}' & \mathbf{G}_2\mathbf{Q}_2 \end{pmatrix}.$$

Si cette proposition s'applique, un attaquant retrouve en temps polynomial une instance d'un cryptosystème fondé sur les codes RRC sans matrice de distorsion. En effet, pour $i = 1, 2$, la matrice $\mathbf{Q}_i$ est inversible et la matrice $\mathbf{G}_i\mathbf{Q}_i$ engendre le code $Gab_{k_i}(\mathbf{g}_i\mathbf{Q}_i)$.

Pour conclure l'attaque, R. Overbeck a montré que, en considérant une matrice de taille $k_2 \times (n_1 + n_2)$ formée de k_2 lignes aléatoires de $\mathbf{G}_{\mathrm{pub}}\mathbf{Q}$ qui soit de rang maximum, on retrouvait une instance du cryptosystème GPT. Ensuit, il suffit ré-appliquer la procédure de la proposition 26 cryptanalysant ainsi le système [Ove06].

Remarque 7

On pourrait imaginer l'ajout d'une matrice de distorsion supplémentaire à la clé publique, comme ce fut proposé originellement. On peut dans ce cas ramener la clé publique à une forme utilisable dans une variante des attaques décrites précédemment.

Quels paramètres pour un système sûr ? On parvient à une condition suffisante sur les paramètres de la clé publique pour que le noyau à droite de $\mathcal{G}_{\mathrm{pub}}$ soit de dimension égale à $1 + \ell$, où $\ell \geq 1$.

Corollaire 6

Soit $\mathbf{A}$ *la matrice définie à l'équation (6.6). Pourvu que*

$$1 \leq \mathrm{Rg}(\mathbf{A}) \leq \frac{n_1 - k_1}{n_2 - k_2} - \frac{\ell - 1}{n_2 - k_2} - 1, \tag{6.9}$$

alors la dimension de $\ker(\mathcal{G}_{\mathrm{pub}})$ *est supérieure ou égale à* $1 + \ell$.

Preuve

Le rang de $\mathcal{Z}$ est inférieur à n_1. Donc, $\mathrm{Rg}(\mathcal{Z}) \leq (n_2 - k_2)\,\mathrm{Rg}(\mathbf{A}) + n_2 - k_2 + k_1$. Pour que $\ker(\mathcal{G}_{\mathrm{pub}})$ soit de dimension $\geq 1 + \ell$, il suffit que $(n_2 - k_2)\,\mathrm{Rg}(\mathbf{A}) + n_2 - k_2 + k_1 - 1 \leq n_1 - \ell$, ce qui nous donne alors la condition du corollaire.
■

D'après la proposition 24 du chapitre 5, le dual d'un code RRC est encore un code RRC. Connaissant $\mathbf{G}_{\mathrm{pub}}$, un attaquant peut déterminer une matrice de parité du code public qui est de la forme

$$\mathbf{H}_{\mathrm{pub}} = \mathbf{S}' \begin{pmatrix} \mathbf{H}_1 & \mathbf{0} \\ \mathbf{B} & \mathbf{H}_2 \end{pmatrix} (\mathbf{T}^{-1})^T.$$

S'il se retrouve dans de bonnes conditions, il peut lui appliquer les résultats de la proposition 27. Ainsi une condition suffisante sur les paramètres consiste à remplacer $n_2 - k_2$ par k_1 et $n_1 - k_1$ par k_2. On obtient alors

$$1 \leq \mathrm{Rg}(\mathbf{B}) \leq \frac{k_2}{k_1} - \frac{\ell}{k_1} - 1.$$

Il faudrait donc s'assurer que, si le rang de $\mathbf{A}$ est petit, alors le rang de $\mathbf{B}$ est petit lui-aussi. Ces deux matrices sont reliées par l'équation

$$\mathbf{A}\mathbf{H}_1 + \mathbf{G}_2\mathbf{B} = \mathbf{0}.$$

Nous ne disposons pas de résultats à ce sujet, mais des simulations montrent, que en général les rangs de $\mathbf{A}$ et de $\mathbf{B}$ sont proches.

Exemple 4

Pour obtenir des conditions optimales de sécurité, il convient de choisir les paramètres tels que

$$\frac{k_2}{k_1} = \frac{n_1 - k_1}{n_2 - k_2}. \tag{6.10}$$

De plus ils doivent être suffisamment grand pour que les complexités des attaques par décodage deviennent prohibitives.

Le tableau suivant donne un exemple de paramètres résistants aux attaques connues. On prend comme taille de clé la partie redondante de la forme systématique de la clé publique. *Pour un degré d'extension m, on prend*

$$m = \max(n_1, n_2).$$

La complexité de l'attaque est celle donnée par l'algorithme d'Ourivski et Johannson le plus adapté.

n_1	k_1	n_2	k_2	t	Taille clé	Décodage	Taux
39	7	22	14	4	97 539 bits	$> 2^{90}$	30%

TABLEAU 6.2: Paramètres admissibles pour le cryptosystème GPT utilisant des codes RRC

3 Conclusion et perspectives

Vu les percées récentes concernant les attaques par structures contre les cryptosystèmes de type McEliece utilisant des codes de Gabidulin ou bien leurs dérivés, il convient de rester très prudent dans la conception et l'utilisation de tels systèmes.

Les attaques présentées par R. Overbeck sont très récentes et ont mis à mal les systèmes de chiffrement de type McEliece utilisant des codes de Gabidulin ou leurs dérivés. Tous les paramètres proposés jusqu'alors tombent dans leur rayon d'efficacité. Pour y pallier, il est nécessaire de prendre des matrices de tailles beaucoup plus grandes. Ainsi, l'intérêt de considérer la métrique rang dans la conception de systèmes de chiffrement en serait amoindri. Il faut plusieurs dizaines de milliers de bits pour le moins. Cela ne les rend donc pas plus attractives que les cryptosystèmes en métrique de Hamming, qui résistent depuis bien plus longtemps.

Il serait donc intéressant

- De parvenir à construire un algorithme de signature du type CFS, cf. [CFS01], à condition que cela soit possible.
- De déterminer des familles de codes autre ou bien de mieux masquer la structure des codes de Gabidulin dans la conception des cryptosystèmes.

Chapitre 7

Cryptosystèmes fondés sur la reconstruction de polynômes linéaires

En 2003, à la conférence internationale EUROCRYPT, D. Augot et M. Finiasz présentèrent un nouveau type de cryptosystème à clé publique. La sécurité de celui-ci est fondée sur la difficulté de reconstruire des polynômes sur un corps fini, c'est-à-dire : Etant donné deux ensembles ordonnés de points d'un corps fini, déterminer la liste des polynômes de degré majoré interpolant le premier ensemble sur le second, cf. [AF03]. Ce problème est profondément relié au problème du décodage en liste des codes RS. Bien qu'il existe un algorithme en temps polynomial de décodage en liste quand le nombre d'erreurs est majoré par la *borne de Johnson*[1], ce problème est conjecturé comme étant difficile dès que l'on dépasse cette borne, la taille de la liste de candidats pouvant augmenter de manière exponentielle, cf. [KY02].

La clé publique du cryptosystème est formée par un vecteur

$$\mathbf{K}_{\text{pub}} = \mathbf{c} + \mathbf{E} \subset GF(q^m)^n$$

où $\mathbf{c}$ est un vecteur pris dans un code RS de longueur n sur $GF(q^m)$ et $\mathbf{E}$ est un vecteur de poids de Hamming supérieur à la borne de Johnson. L'opération de chiffrement du message $\mathbf{x}$ consiste plus ou moins à encoder $(\mathbf{x}, \alpha)$, où α est un élément aléatoire du code considéré *sur-code* du code RS contenant $\mathbf{c}$. On ajoute ensuite une erreur de poids de Hamming convenable afin de briser la linéarité de l'encodage. La matrice génératrice du *sur-code* est formée par la matrice génératrice du code RS, à laquelle on ajoute le vecteur $\mathbf{K}_{\text{pub}}$. Le concepteur connaissant les positions d'erreur de $\mathbf{E}$, il utilise cette information pour poinçonner le vecteur reçu et récupérer le message. Ainsi, avec des tailles de clé de l'ordre de 80 kbits on obtient des sécurités supérieures à un facteur de travail de 2^{80}. Dans le même article, les auteurs ont proposé de réduire la taille des clés en considérant la clé publique sur un sous-corps de $GF(q^m)$.

Le système subit de sévères cryptanalyses, dont la première par J.-S. Coron, cf. [Cor04]. Ce dernier montra comment, dans la majorité des cas, décrypter le chiffré intercepté en

1. Ecrivant le système quadratique vérifié par le texte clair en fonction des paramètres connus du système.
2. Linéarisant ce système d'équation et en montrant que, puisqu'il y avait nécessairement une solution, un des paramètre inconnus du système devait être racine du

[1] algorithmes de Sudan et Guruswami-Sudan, cf. [Sud97, GS99]

déterminant d'un sous-système. Un simple calcul des racines de ce polynôme univarié de bas degré permettait de retrouver ce paramètre et par là de recouvrer le chiffré en temps polynomial.

Peu après j'ai montré que, réduire la taille de la clé publique en la considérant dans un sous-corps de $GF(q^m)$, permettait de concevoir une attaque efficace permettant de récupérer le texte clair à partir d'un chiffré intercepté. Bien qu'elle soit de complexité exponentielle, elle est plus rapide que les précédentes sur les paramètres donnés par les auteurs du système, cf. [Loi05]. Puis, A. Kiayias et M. Yung établirent que l'attaque de J.-S. Coron avait une probabilité d'échec exponentiellement petite. Dans le même article, ils montrèrent que même sous un choix optimal de paramètre, on pouvait étendre cette attaque [KY04].

Avec D. Augot et M. Finiasz, j'ai proposé un nouveau cryptosystème résistant à ces attaques. Il utilise les propriétés de la Trace sur les corps finis qui permet de considérablement augmenter la complexité des attaques par linéarisation. En effet, le déterminant du système linéaire devient, suivant l'approche considérée ou bien un polynôme de très grand degré ou bien un polynôme multivarié. Trouver les racines de ces polynômes est un exercice difficile, cf. [AFL03]. Pour un tel choix, attaquer la clé publique s'apparente à décoder dans un code RS entrelacé, cf. [BKY03]. C'est donc cette borne sur le décodage des codes RS entrelacés que l'on doit prendre en compte dans la construction de la clé publique. Malheureusement cette contrainte est incompatible avec la résistance du système à une variante des attaques par linéarisation publiée par J.-S. Coron et le système s'avère, encore une fois, peu sûr.

Il m'a semblé nécessaire de donner ici une place importante à l'historique des versions du cryptosystème Augot-Finiasz original ainsi que des attaques, afin d'éclairer la démarche qui nous a conduit, avec C. Faure, à l'élaboration d'un système similaire en métrique rang ainsi qu'à l'étude de la transposition des diverses attaques à ce nouveau système. Ce chapitre résume en partie des travaux que nous avons publiés dans [Fau04, FL06], sur la conception d'un cryptosystème analogue au système Augot–Finiasz en métrique rang. Les rôles des codes RS et des polynômes y sont respectivement tenus par les codes de Gabidulin et par les q-polynômes.

Dans la section 2.1 notamment, je montre que si les paramètres du système sont mal choisis, on peut ramener le problème de retrouver des candidats pour la clé privée au problème d'attaquer la clé publique d'un cryptosystème GPT, comme nous l'avons vu à la section 2 du chapitre 6. Cette attaque, non-encore publiée utilise, au même titre que les attaques d'Overbeck, des propriétés de stabilité des codes de Gabidulin sous l'action de l'automorphisme de Frobenius. Le lecteur remarquera que les paramètres que nous avons proposés dans [FL06] considérés comme sûrs envers cette attaque. C'est pourquoi, dans une section finale, je reprends les résultats obtenus sur l'*effectivité* des attaques pour dériver des jeux de paramètres *ad hoc*.

1 Construction du système

Soit $GF(q^s)$ le corps à q^s éléments. Pour un paramètre entier u, on pose $m = su$. On a alors $GF(q^s) \subset GF(q^m)$. On se donne également une base $\mathcal{B} = (\gamma_1, \ldots, \gamma_u)$ de $GF(q^m)/GF(q^s)$, ainsi qu'un entier $n \leq s$ et un vecteur $\mathbf{g} = (g_1, \ldots, g_n)$ de longueur n à coefficients dans $GF(q^s)$ formé d'éléments linéairement indépendants sur $GF(q)$.

Génération des clés La *clé privée* du système est constituée par

– Un q-polynôme
$$P(z) = \sum_{i=0}^{k-1} p_i z^{[i]}, \tag{7.1}$$
à coefficients dans $GF(q^m)$. On impose en outre comme condition que les u coefficients de plus haut degré $p_{k-1}, \ldots, p_{k-u}$ forment une base de $GF(q^m)/GF(q^s)$.

– Un vecteur $\mathbf{E} = (E_1, \ldots, E_n) \in GF(q^m)^n$ aléatoire de rang $W > (n-k)/2$, à valeurs dans $GF(q^m)$.

La *clé publique* est le vecteur à coefficients dans $GF(q^m)$:
$$\mathbf{K}_{\text{pub}} \stackrel{def}{=} P(\mathbf{g}) + \mathbf{E}. \tag{7.2}$$
La taille en bits de ce vecteur est environ égale à $nsu \log_2(q)$.

Chiffrement Le texte clair est un vecteur $\mathbf{x} = (x_0, \ldots, x_{k-u-1})$ où $x_i \in GF(q^s)$ pour tout $i = 0, \ldots, k-u-1$. note $X(z) \stackrel{def}{=} \sum_{i=0}^{k-u-1} x_i z^{[i]}$. L'émetteur choisit aléatoirement

1. un élément $\alpha \in GF(q^m)^*$;
2. un vecteur $\mathbf{e}$ de rang $\omega \leq \frac{(n-W)-k}{2}$.

Le chiffré est le vecteur $\mathbf{y}$ donné par
$$\mathbf{y} = X(\mathbf{g}) + \text{Tr}_{m/s}(\alpha \mathbf{K}_{\text{pub}}) + \mathbf{e}, \tag{7.3}$$
où $\text{Tr}_{m/s}(z) = \sum_{i=0}^{u-1} z^{[si]}$ désigne l'opérateur Trace de $GF(q^m)$ dans $GF(q^s)$.

Déchiffrement Comme $\mathbf{K}_{\text{pub}} = P(\mathbf{g}) + \mathbf{E}$, on a
$$\mathbf{y} = (X + \text{Tr}_{m/s}(\alpha P))(\mathbf{g}) + \text{Tr}(\alpha \mathbf{E}) + \mathbf{e}.$$
Le vecteur $\mathbf{E}$ étant de rang W sur $GF(q)$, il existe une matrice $\mathbf{T}$, q-aire inversible de taille $n \times n$ telle que
$$\mathbf{ET} = (\underbrace{\mathbf{0}}_{n-W} \mid \mathbf{E}').$$
De plus, comme $z \mapsto \text{Tr}_{m/s}(z)$ est une forme $GF(q^s)$-linéaire, partant $GF(q)$-linéaire, on a
$$\mathbf{yT} = (X + \text{Tr}_{m/s}(\alpha P))(\mathbf{gT}) + (\mathbf{0} \mid \text{Tr}_{m/s}(\alpha \mathbf{E}')) + \mathbf{eT}.$$
En poinçonnant $\mathbf{yT}$ sur les W dernières positions puis, en appliquant un algorithme de décodage de codes de Gabidulin au $Gab_k(\mathbf{gT})$ poinçonné sur ces mêmes positions, on détermine le q-polynôme $X + \text{Tr}_{m/s}(\alpha P)$.

Or, par construction, X est de q-degré $\leq k-u-1$. Donc les u coefficients de plus haut degré de $X + \text{Tr}_{m/u}(\alpha P)$ sont également les u coefficients de plus haut degré du q-polynôme $\text{Tr}_{m/u}(\alpha P)$, soit $\text{Tr}_{m/u}(\alpha p_{k-1}), \ldots, \text{Tr}_{m/u}(\alpha p_{k-u})$, qui sont donc les coordonnées de l'élément α dans la base duale de $(p_{k-1}, \ldots, p_{k-u})$. Par changement de base, on obtient les coordonnées de α dans la base $\mathcal{B}$.

Une fois l'élément α obtenu, on trouve le q-polynôme X dont les coordonnées donnent le texte clair $\mathbf{x}$.

Complexité de chiffrement–déchiffrement Le chiffrement est de complexité quadratique et se résume pour l'essentiel à l'encodage d'un mot de $Gab_{k-u}(\mathbf{g})$, soit

$$W_{enc} \sim (k-u)n \text{ multiplications dans } GF(q^s).$$

Le déchiffrement quant à lui se résume en complexité au décodage d'un mot bruité de $Gab_k(\mathbf{g})$ poinçonné sur les $n-W$ dernières coordonnées. Avec les algorithmes présentés au chapitre 4, on peut décoder une erreur de rang jusqu'à $\frac{(n-W)-k}{2}$. Si on considère l'algorithme de Welch-Berlekamp, la complexité du déchiffrement est

$$W_{dec} \sim 2(n-W)^2 - k^2 + k\frac{(n-W)-k}{2} \text{ multiplications dans } GF(q^s).$$

Problèmes difficiles sous-jacents La sécurité du cryptosystème s'envisage de deux manières :

1. *Sécurité structurelle de la clé publique* : La clé publique est constituée d'un mot d'un code de Gabidulin public (le vecteur $\mathbf{g}$ est nécessairement public) auquel un vecteur d'erreur $\mathbf{E}$ de rang W est ajouté. Donc W doit être supérieur à la capacité de correction du code considéré. La sécurité de notre approche repose sur le fait que le problème **Décodage_Borné**$(\mathbf{K}_{\text{pub}}, \mathcal{C}, W)$ est difficile. Comme, il n'existe pas *encore* d'algorithme permettant de décoder les codes de Gabidulin au-delà de leur capacité de correction, nous avons fait l'hypothèse que le problème était *difficile* dès que W est supérieur strictement à la capacité de correction du code $Gab_k(\mathbf{g})$, à savoir $W > (n-k)/2$.
2. *Attaque par décodage* : Recouvrer le texte clair consiste à décoder dans un code $\mathcal{C}$ de dimension $k+u$, dont $\mathrm{Gab}_k(\mathbf{g})$ est un sous-code. Ce code a pour matrice génératrice

$$\mathbf{G} = \begin{pmatrix} \mathbf{g} \\ \vdots \\ \mathbf{g}^{[k-1]} \\ \mathrm{Tr}_{m/s}(\gamma_1 \mathbf{K}_{\text{pub}}) \\ \vdots \\ \mathrm{Tr}_{m/s}(\gamma_u \mathbf{K}_{\text{pub}}) \end{pmatrix}.$$

Si le code $\mathcal{C}$ est aléatoire, résoudre **Décodage_Borné**$(\mathbf{y}, \mathcal{C}, \omega)$ est un problème réputé difficile. Les complexités des meilleurs algorithmes le résolvant sont données par la proposition 9 du chapitre 2.

Remarque 8

Dans [Fau04, FL06], on présente une version simplifiée du système où le chiffrement prend, à la place de la forme (7.3)*, la forme*

$$\mathbf{y} = X(\mathbf{g}) + \alpha \mathbf{K}_{\text{pub}} + \mathbf{e},$$

le reste demeurant identique. On montre dans [FL06] que $\mathbf{y}$ *peut presque toujours être décrypté en temps polynomial en utilisant des techniques de linéarisation. Il n'y a pas de choix de paramètres possibles pour le rendre plus sûr.*

2 Approches cryptanalytiques

On peut envisager deux types d'attaques, chacune s'appliquant aux problèmes sous-tendant la sécurité du système. L'attaque par structure de la section 2.1 n'étant pas encore publiée on la détaillera plus particulièrement. Elle s'inspire de résultats récents de R. Overbeck présentés au chapitre 6. Elle exploite le fait que les codes de Gabidulin sont relativement stables sous l'action de l'automorphisme de Frobenius.

Afin de simplifier les notations, on pose

$$\forall i = 1, \ldots, u, \quad \mathrm{Tr}_{m/s}(\gamma_i \mathbf{K}_{\mathrm{pub}}) \stackrel{def}{=} \mathbf{K}_i, \tag{7.4}$$

qui sont les projections de la clé publique $\mathbf{K}_{\mathrm{pub}}$ dans le sous-corps $GF(q^s)$, coordonnée par coordonnée.

2.1 Attaque sur la clé publique

Le vecteur générateur $\mathbf{g}$ du code étant à coefficients dans $GF(q^s)$, on a, en utilisant les notations (7.4)

$$\forall i = 1, \ldots, u, \quad \mathbf{K}_i = P_i(\mathbf{g}) + \mathbf{E}_i, \tag{7.5}$$

où, si $P(z) = \sum_{i=0}^{k-1} p_i z^{[i]}$, alors $P_i(z) = \sum_{j=0}^{k-1} \mathrm{Tr}_{m/s}(\gamma_i p_j) z^{[j]}$, et $\mathbf{E}_i = \mathrm{Tr}_{m/s}(\gamma_i \mathbf{E})$.

Soit ℓ un paramètre entier. Si on élève les coefficients de la matrice

$$\begin{pmatrix} \mathbf{K}_1 \\ \vdots \\ \mathbf{K}_u \end{pmatrix}$$

aux puissances successives de l'automorphisme de Frobenius de 0 à ℓ, on obtient l'équation donnée dans le tableau 7.1, soit

$$\mathcal{K} = \mathcal{P}\mathcal{G} + (\underbrace{\mathbf{0}}_{n-W} \mid \mathcal{E})\mathbf{T}.$$

Si le rang de la matrice $\mathcal{P}$ est égal à $k + \ell$ alors, en prenant $k + \ell$ lignes linéairement indépendantes de $\mathcal{P}$, on se ramène au cas où $\mathcal{K}$ est clé publique d'un cryptosystème GPT (voir la section 2 du chapitre 6). Une condition pour que l'attaque décrite ne puisse être mise en œuvre consiste à choisir le paramètre ℓ de telle sorte que

$$\ell \geq \frac{k-u}{u-1}. \tag{7.6}$$

Donc une condition suffisante pour que l'attaque décrite ne puisse fonctionner est de choisir les paramètres tels que $k + \ell \geq n - W$. Avec la condition (7.6), cela implique de choisir les paramètres du système de telle sorte que

$$W \geq (n-k) - \frac{k-u}{u-1}.$$

Si on écrit $W = (n-k)/2 + \delta$, δ étant entier ou demi-entier, alors l'inégalité se réécrit

$$n - k \leq \frac{2(k-u)}{u-1} + 2\delta.$$

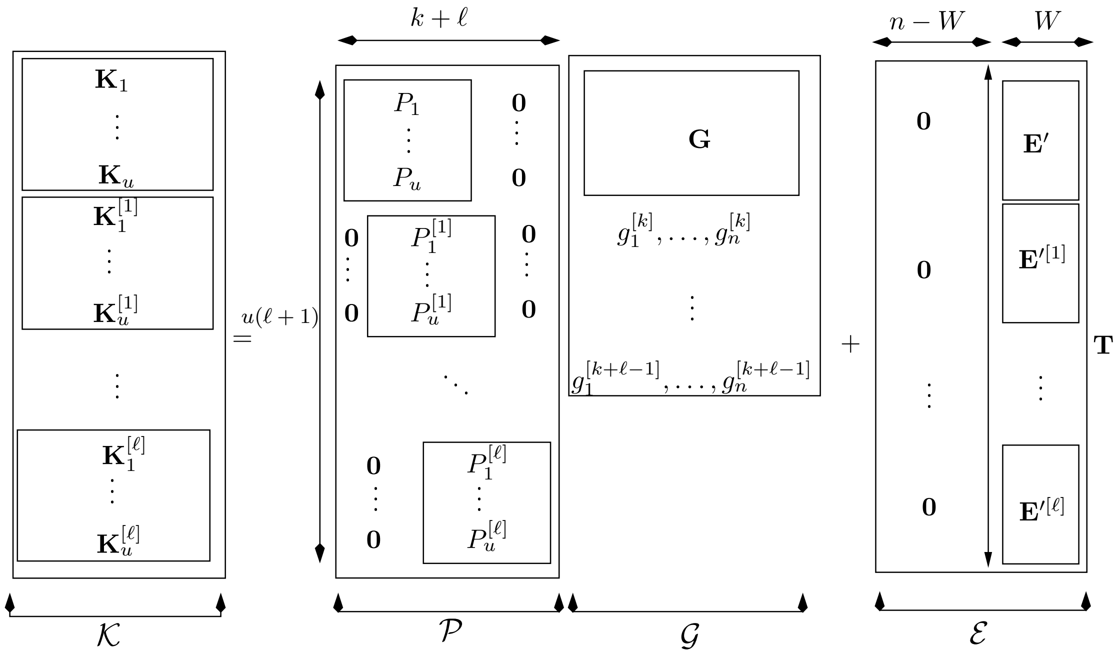

TABLEAU 7.1 : Attaque sur la structure

Pour être en mesure de décoder il faut prendre en considération le fait que $\omega \leq \frac{n-k-W}{2} = \frac{n-k}{4} - \frac{\delta}{2}$ qui implique l'inégalité

$$4\omega + 2\delta \leq n - k.$$

On obtient donc la proposition suivante

Proposition 28

Dans le cas où les paramètres du cryptosystème vérifient

$$4\omega + 2\delta \leq n - k \leq \frac{2(k-u)}{u-1} + 2\delta. \tag{7.7}$$

l'attaque décrite à la section 2 du chapitre 6 ne s'applique pas.

Exemple 5

Comme exemple, on peut choisie par exemple les paramètres suivants :

- u = 3 : $\omega = 6, \delta = 2, k = 26, n = 54$.
- u = 4 : $\omega = 6, \delta = 2, k = 40, n = 68$.

2.2 Attaque sur les chiffrés

Sur la base $\mathcal{B}$, on a $\alpha = \sum_{i=1}^{u} \alpha_i \gamma_i$, où $\alpha_i \in GF(q^s)$ pour tout $i = 1, \dots, u$. En tenant compte des notations (7.4), l'équation vérifiée par le chiffré se réécrit alors

$$\mathbf{y} = X(\mathbf{g}) + \sum_{i=1}^{u} \alpha_i \mathbf{K}_i + \mathbf{e}.$$

Comme $\mathbf{e}$ est de rang ω, de par la proposition 13, il existe un q-polynôme V de q-degré ω tel que

$$V(\mathbf{y}) = V \circ X(\mathbf{g}) + \sum_{i=1}^{u} V(\alpha_i \mathbf{K}_i). \tag{7.8}$$

Ce système est un système quadratique dont les inconnues sont les coefficients de V, de X, et les éléments α_i pour $i = 1, \ldots, u$. Deux méthodes peuvent s'appliquer pour le résoudre.

Résolution du système quadratique Nous avons programmé la résolution du système (7.8) en utilisant l'algorithme F4 de Faugère de résolution par bases de Groebner, implanté dans le langage MAGMA [MAG], et qui tourne sur un processeur de type OPTERON cadencé à 2,4Ghz. Les simulations du tableau 7.2, montrent, que pour $u = 3$ avec un degré de système égal à 31 on ne parvient pas le résoudre.

Nombre de variables	Rang de l'erreur	Degré	Magma 2.11-2/F4
$u = 2$	$\omega = 2$	7	0s
	$\omega = 3$	15	0s
	$\omega = 4$	31	3s
	$\omega = 5$	63	100s
	$\omega = 6$	127	Mémoire$>$ 2Gbs
$u = 3$	$\omega = 2$	7	0.120
	$\omega = 3$	15	34s
	$\omega = 4$	31	Trop de mémoire nécessaire

Tableau 7.2 : Simulation effectuées sur les paramètres $m = 36, k = 10, W = 14$

Attaque par linéarisation Un autre moyen de résoudre un tel système consiste à le *linéariser*, résoudre le système linéaire et, lorsque le nombre de solution du système linéaire n'est pas trop élevé, les tester pour voir celles correspondant à des solutions du système quadratique (7.8).

La méthode de linéarisation a déjà été détaillée et appliquée au chapitre 2 pour la conception de l'algorithme de décodage de Welch-Berlekamp. C'est également une approche de ce type qui permit à J.-S. Coron puis à A. Kiayias et M. Yung de montrer les faiblesses du système Augot–Finiasz. Soit le système linéaire

$$V(\mathbf{y}) = N(\mathbf{g}) + \sum_{i=1}^{u} R_i(\mathbf{K}_i), \tag{7.9}$$

où

- V est un q-polynôme de q-degré ω,
- N est un q-polynôme de q-degré $\omega + k - u - 1$,
- pour $i = 1, \ldots, u$, R_i est un q-polynôme de q-degré ω.

Ce système est un système linéaire de n équations à $k + (u+2)\omega + 1$ inconnues pour lequel on peut montrer facilement la proposition suivante

Proposition 29

Si $(V, X, \alpha_1, \ldots, \alpha_u)$ *est solution du système quadratique* (7.8), *alors*

$$(V, N = V \circ X, R_1(z) = V(\alpha_1 z), \ldots, R_u(z) = V(\alpha_u z))$$

est solution du système linéaire (7.9).

Dans le cas où la dimension de l'espace de solutions de (7.9) est égale à 1, on peut déterminer la solution de (7.8) et par là même retrouver tous les éléments nécessaires pour décrypter le chiffré $\mathbf{y}$. Dans le cas où la dimension est strictement supérieure à 1, il faut énumérer l'espace des solutions possibles et, pour chacune d'entre elle de tester si elle provient d'une solution du système quadratique ou non.

Afin de contrôler la dimension de l'espace des solutions de (7.9) on s'appuie sur la proposition suivante :

Proposition 30

Une condition suffisante pour que l'espace des solutions du système (7.9) *soit de dimension* $\geq \lambda$*, où* $\lambda > 1$ *est*

$$n - k = \omega(u + 2) - \lambda \tag{7.10}$$

On va même imposer comme condition que $\omega(u + 2) = n - k + \lambda$, où $\lambda \geq 1$ est un paramètre tel que l'espace des solutions de (7.9) ne peut être dénombré. Le cardinal de cet espace est de dimension $\geq q^{s\lambda}$.

3 Choix des paramètres

On va choisir les paramètres du système de telle sorte que

1. l'attaque contre la clé privée d'un système GPT ne s'appliquent pas. Cela correspond au choix des paramètres de la proposition 28.
2. la méthode de résolution du système par linéarisation soit impraticable, en utilisant le résultat de la proposition 30.

Les paramètres du système doivent donc vérifier

$$\left\{\begin{array}{l} n-k+\lambda = \omega(u+2), \quad \lambda \geq 1, \\ 4\omega + 2\delta \leq n-k \leq \frac{2(k-u)}{u-1} + 2\delta. \end{array}\right. \Rightarrow \left\{\begin{array}{l} n-k+\lambda = \omega(u+2), \quad \lambda \geq 1, \\ 4\omega + 2\delta \leq \omega(u+2) - \lambda \leq \frac{2(k-u)}{u-1} + 2\delta. \end{array}\right.$$

De ce système, pour des paramètres u, λ, et δ fixés, on a des conditions concurrentes sur le nombre d'erreurs ω que l'on peut corriger. D'un côté, pour résister aux attaques par décodage, il est nécessaire que ω soit grand, et d'un autre côté, il est nécessaire qu'il ne le soit pas trop pour résister aux attaques sur la clé publique. La partie gauche de l'inégalité implique en particulier que l'on doit avoir $u \geq 3$.

Voici deux exemples de jeux de paramètres possibles donnant des tailles de clés de quelques milliers de bits.

Conditions $u = 3, \delta = 2, \lambda = 2$

Les conditions à satisfaire deviennent

$$\left\{\begin{array}{l} n-k+\lambda = 5\omega, \quad \lambda \geq 1, \\ k + 4\omega + 2\delta \leq n \leq 2k + 2\delta - 3 \end{array}\right.$$

Les paramètres suivants sont admissibles dans la conception de cryptosystème.

- $n = s = 56$,
- $k = 28$,
- $W = 16$,
- $\omega = 6$.

Le système a une taille de clé égale à $56 \times 56 \times 3 = 9408$ bits de clé pour un taux de transmission égal à $25/56 \approx 44\%$.

Conditions $u = 4, \delta = 2, \lambda = 2$

Les conditions à satisfaire sont

$$\begin{cases} n - k + \lambda = 6\omega, \quad \lambda \geq 1, \\ k + 4\omega + 2\delta \leq n \leq \frac{5}{3}k + 2\delta - \frac{8}{3}. \end{cases}$$

Les paramètres suivants sont admissibles dans la conception de cryptosystème.
- $n = s = 54$,
- $k = 32$,
- $W = 13$,
- $\omega = 4$.

Le système a une taille de clé égale à $54 \times 54 \times 4 = 11664$ bits de clé pour un taux de transmission égal à $28/54 \approx 44\%$.

4 Pistes de recherche

Ce cryptosystème étant très récent, il est difficile d'établir qu'il est sûr et de certifier qu'il peut être employé en pratique pour chiffrer des documents, d'autant plus que le cryptosystème Augot-Finiasz est cassé, sans réparation possible semble-t-il. Il est donc essentiel d'explorer diverses facettes de sa sécurité et notamment
- de voir si l'on peut construire un algorithme en temps polynomial décodant un code de Gabidulin au delà de la capacité de correction, et si oui jusqu'à quelle distance. L'estimation moyenne de la borne de Johnson donnée par l'inéquation (2.3) indique que la taille de la liste de candidats pourrait être polynomiale jusqu'à un rang égal à $T = n - \sqrt{nk + \lambda \log_q n}$. Il serait donc nécessaire de prendre $W > T$. Cela pourrait remettre en question les équations obtenues.
- de considérer si le fait de prendre le vecteur générateur du code dans un sous-corps ne constitue pas une faiblesse insurmontable. Dans la section 2.1, nous avons considéré ce cas qui n'est pas rédhibitoire en métrique rang. Contrairement au cas de la métrique de Hamming et du système originel où l'on se ramène au décodage d'un code RS entrelacé dont la borne de décodage est incompatible avec la sécurité du système contre les attaques par linéarisation.

Si l'on parvient à des résultats satisfaisants, alors on pourra commencer tenter d'améliorer le système, notamment en augmentant son taux de transmission par une procédure s'inspirant de celle présentée à la section 1.3 du chapitre 6.

Etat des lieux et Perspectives de recherche

En guise de conclusion au document, je souhaiterais faire un point sur les perspectives de recherches en métrique rang, hormis celles que j'ai effectuées. Dans une intention plus prospective, je voudrais également parler un peu de l'état de mes travaux en cours.

La métrique rang aujourd'hui

Aujourd'hui on peut dire que l'intérêt de la métrique rang suscite des vocations de recherches tant dans le domaine de la cryptographie, abondamment décrit dans le document que du point de vue de la théorie des codes, et notamment pour de possibles applications dans la construction de codes *espace-temps*.

En cryptographie

Mis à part les cryptosystèmes présentés dans les chapitres 6 et 7, concernant les applications de la métrique rang en cryptographie, on pourrait rajouter l'existence d'un schéma d'identification à divulgation nulle de connaissance dû à K. Chen datant de 1996, cf. [Che96]. Un algorithme de décodage pour la métrique rang de F. Chabaud et de J. Stern montra que les paramètres publiés n'étaient pas sûrs, cf. [Che96, CS96]. Cela ne remet pas en cause le schéma mais il est nécessaire d'augmenter les tailles de paramètres. En les accroissant suffisamment, on peut rendre le schéma sûr même contre les attaques reposant sur des algorithmes plus performants comme ceux de Ourivski et Johannson. Dans un autre cadre, la métrique trouve des applications dans la conception de fonctions de hachage satisfaisantes pour être utilisées dans des MACs (*Message Authentification Codes*) [Joh96, SNC05].

Cependant, afin de convaincre que de tels protocoles ou fonctions reposent sur de solides bases de sécurité, il est nécessaire d'être en mesure de préciser la complexité des problèmes de décodage sur lesquels ils reposent. A première vue, ce problème semble plutôt complexe. En effet, il n'existe pas de problème simple connu, réputé difficile, auquel on puisse réduire les problèmes de décodage en métrique rang. Cela constitue une piste de recherche importante. Un autre point d'investigation qui me semble également d'un grand intérêt concerne la conception de nouvelles familles de codes *décodables* en métrique rang et qui ne se déduisent pas de la famille des codes de Gabidulin par des transformations linéaires simples. C'est ce type de familles que l'on recherche pour jouer le rôle de l'espace des clés publiques dans la conception de cryptosystèmes de type McEliece.

En théorie de l'information

L'avènement de nouveau domaines de recherches tels que les codes *espaces-temps* a donné de nouvelles perspectives d'applications intéressantes aux codes en métrique rang, cf. par exemple [LK05, Ham06, GY06]. Un code espace-temps de diversité d est défini comme étant un ensemble de matrices de même tailles à coefficients complexes dont la différence deux à deux est de rang supérieur à d. Dans les articles cités, il est montré comment construire des codes espace-temps additifs[2], de diversité d à partir de codes binaires de distance rang minimale d. D'autre part, bien que le stockage sur bande magnétique soit tombé en désuétude, la métrique rang reste intéressante pour corriger les erreurs arrivant sur des lignes ou bien des colonnes de matrice, erreurs qui peuvent arriver par paquets.

Mes perspectives de recherche

Pour la métrique rang les perspectives ont été données à la fin de chacun des chapitres du document. En particulier, avec C. Faure nous avons commencé à travailler sur la construction d'un algorithme de décodage en liste des codes de Gabidulin sur le modèle de l'algorithme de proposé par M. Sudan. Pour ce faire il est nécessaire de définir un cadre dans lequel on puisse relier les racines de polynômes bivariés, concept à définir pour les q-polynômes, à des problèmes de factorisation de ces mêmes q-polynômes.

Le document ainsi présenté ne traite que des travaux que j'ai réalisés concernant la métrique rang et qui ont constitué la très grande majorité de mon travail de recherche d'après la thèse. Cependant ils n'en forment pas le tout, ni leurs perspectives ne forment l'essentiel de ma recherche future. De plus en plus, je m'intéresse aux applications cryptologiques des codes de *Reed-Muller*.

1. Depuis 2003, avec B. Sakkour, nous sommes parvenus à améliorer de manière significative l'algorithme de décodage de Sidel'nikov-Pershakov pour les codes de Reed-Muller d'ordre 2 sans en augmenter la complexité. Sur des simulations effectuées pour des codes de longueur moyenne (de quelques milliers de bits) les simulations montrent qu'on peut en moyenne décoder beaucoup plus loin que les algorithmes les plus rapides existants, cf. [LS04]. Asymptotiquement, pourtant, tous ces algorithmes ont de bonnes chances d'être équivalents, mais toujours assez loin de la borne du décodage à maximum de vraisemblance. L'intérêt d'une telle étude serait d'améliorer l'efficacité du cryptosystème de Sidel'nikov, cf. [Sid94]. C'est un cryptosystème de type McEliece dont la clé publique est constituée par un code de Reed-Muller permuté. Des perspectives intéressantes dans ce cadre pourraient consister à étendre les résultats existants à des codes de Reed-Muller d'ordre supérieur.
2. Plus récemment, j'ai commencé à réaliser une implémentation efficace en langage C de l'algorithme de décodage en liste des codes de Reed-Muller d'ordre 1, algorithme décrit dans la thèse de C. Tavernier, cf. [Tav04]. Comme il est écrit dans le document de thèse de ce dernier, cette approche pourrait conduire à l'obtention, sur des sorties bien spécifiées de systèmes de chiffrement par blocs tels le DES, une liste d'équations linéaires vérifiées par les bits d'entrée et les bits de clé. Dans ce cadre, un nombre suffisant d'équations pourrait contribuer à une amélioration significative de la cryptanalyse linéaire de Matsui.

[2] stables par addition

Bibliographie

[AF03] Augot (D.) et Finiasz (M.). – A public key encryption scheme based on the polynomial reconstruction problem. *In : EUROCRYPT 2003*, LNCS, n° 2656, pp. 222–233. – 2003.

[AFL03] Augot (D.), Finiasz (M.) et Loidreau (P.). – Using the trace operator to repair the polynomial reconstruction based cryptosystem presented at Eurocrypt 2003. – Cryptology ePrint Archive, Report 2003/209, 2003. http ://eprint.iacr.org/.

[Aug03] Augot (D.). – Les travaux de M. Sudan sur les codes correcteurs d'erreurs. *Gazette des Mathématiciens*, n°98, octobre 2003.

[Aug04] Augot (D.). – Madhu Sudan's work on error-correcting codes. *European Mathematical Society Newsletter*, n°51, mars 2004, pp. 8–10. – http ://www.emis.de/.

[Bar98] Barg (A.). – *Handbook of Coding Theory, Vol. 1*, chap. 7, pp. 649–754. – North–Holland, 1998.

[Ber84] Berlekamp (E. R.). – *Algebraic Coding Theory.* – Aegean Press Park, 1984.

[Ber03] Berger (T. P.). – Isometries for rank distance and permutation group of Gabidulin codes. *IEEE Transactions on Information Theory*, vol. 49, n°11, novembre 2003, pp. 3016–3019.

[BFS96] Buss (J. F.), Frandsen (G. S.) et Shallit (J. O.). – *The computational complexity of some problems of linear algebra.* – Rapport de recherche RS-96-33, BRICS, Université de Aarhus, 1996. http ://www.brics.dk/RS/96/33/index.html.

[BKY03] Bleichenbacher (D.), Kiayias (A.) et Yung (M.). – Decoding of interleaved Reed–Solomon codes. *In : Proceedings of ICALP 2003*, pp. 97–108. – 2003.

[BL00] Berger (T. P.) et Loidreau (P.). – A Niederreiter version of the GPT public-key cryptosystem. *In : Seventh International Workshop on Algebraic and Combinatorial Coding Theory*, pp. 72–78. – Bansko, Bulgarie, juin 2000.

[BL02] Berger (T. P.) et Loidreau (P.). – Security of the Niederreiter form of the GPT public-key cryptosystem. *In : 2002 IEEE International Symposium on Information Theory, ISIT'02.* – 2002.

[BL04] Berger (T. P.) et Loidreau (P.). – Designing an efficient and secure public-key cryptosystem based on reducible rank codes. *In : Proceedings of INDOCRYPT 2004.* – 2004.

[BL05] Berger (T. P.) et Loidreau (P.). – How to mask the structure of codes for a cryptographic use. *Designs, Codes and Cryptography*, vol. 35, 2005, pp. 63–79.

[BMvT78] Berlekamp (E. R.), McEliece (R. J.) et van Tilborg (H. C.). – On the inherent intractability of certain coding problems. *IEEE Transactions on Information Theory*, vol. 24, n°3, mai 1978.

[BW86] Berlekamp (E. R.) et Welch (L.). – Error correction of algebraic block codes. – US Patent, Number 4,633,470, 1986.

[CC98] Canteaut (A.) et Chabaud (F.). – A new algorithm for finding minimum-weight words in a linear code : Application to McEliece's cryptosystem and to narrow-sense BCH codes of length 511. *IEEE Transactions on Information Theory*, vol. 44, n°1, janvier 1998, pp. 367–378.

[CFS01] Courtois (N.), Finiasz (M.) et Sendrier (N.). – How to achieve a McEliece-based signature scheme. *In : Advances in Cryptology - ASIACRYPT'2001*, éd. par Boyd (C.), LNCS, n° 2248. pp. 151–174. – Springer, 2001.

[Che96] Chen (K.). – A new identification algorithm. *In : Cryptographic policy and algorithms*, éd. par Dawson (E.) et Golic (J.). pp. 244–249. – Springer, 1996.

[Cor04] Coron (J.-S.). – Cryptanalysis of a public-key encryption scheme based on the polynomial reconstruction problem. *In : PKC 2004, 7th International Workshop on Theory and Practice in Public Key Cryptography*, éd. par Bao (F.), Deng (R.) et Zhou (J.). pp. 14–28. – Springer, 2004.

[CS96] Chabaud (F.) et Stern (J.). – The cryptographic security of the syndrome decoding problem for rank distance codes. *In : Advances in Cryptology - ASIACRYPT '96*, éd. par Kim (K.) et Matsumoto (T.), LNCS. – Springer, novembre 1996.

[Del75] Delsarte (P.). – On subfield subcodes of modified Reed–Solomon codes. *IEEE Transactions on Information Theory*, vol. 20, 1975, pp. 575–576.

[Del78] Delsarte (P.). – Bilinear forms over a finite field, with applications to coding theory. *Journal of Combinatorial Theory, Series A*, vol. 25, 1978, pp. 226–241.

[DG75] Delsarte (P.) et Goethals (J.-M.). – Alternating bilinear forms over $GF(q)$. *Journal of Combinatorial Theory, Series A*, vol. 19, 1975, pp. 26–50.

[DK00] Dumer (I.) et Krichevskiy (R. E.). – Soft-decision majority decoding of Reed-Muller codes. *IEEE Transactions on Information Theory*, vol. 46, n°1, 2000, pp. 258–264.

[DK02] Dumer (I.) et K.Shabunov. – Recursive list decoding of reed-muller codes. *In : Information, Coding and Mathematics*, éd. par Blaum (M.), Farrell (P.) et van Tilborg (H. C. A.). – 2002.

[DT99] Dijk (M. van) et Tolhuizen (L.). – Efficient encoding for a class of subspace subcodes. *IEEE Transactions on Information Theory*, vol. 45, n°6, 1999, pp. 2142–2146.

[Dum04] Dumer (I.). – Recursive decoding and its performance for low rate Reed-Muller codes. *IEEE Transactions on Information Theory*, vol. 50, n°5, mai 2004, pp. 811–822.

[Fau99] FAUGÈRE (J.-C.). – A new efficient algorithm for computing Gröbner basis : F_4. *Journal of Pure and Applied Algebra*, vol. 139, 1999, pp. 61–68.

[Fau02] FAUGÈRE (J.-C.). – A new efficient algorithm for computing Gröbner basis without reduction to zero : F_5. *In : Proceedings of ISSAC*. pp. 75–82. – ACM press, juillet 2002.

[Fau04] FAURE (C.). – *Etude d'un système de chiffrement à clé publique fondé sur le problème de reconstruction de polynômes linéaires.* – Rapport de DEA, Université Paris 7, 2004.

[Fau06] FAURE (C.). – Average number of Gabidulin codewords within a sphere. *In : Tenth International Workshop on Algebraic and Combinatorial Coding Theory, ACCT'2006*, pp. 86–89. – septembre 2006.

[FL06] FAURE (C.) et LOIDREAU (P.). – A new public-key cryptosystem based on the problem of reconstructing p-polynomials. *In : Coding and Cryptography - WCC 2005, 4th International workshop on Coding and Cryptography*, éd. par YTREHUS (Ø.), LNCS, n° 3969. pp. 304–315. – Springer, 2006.

[Gab85a] GABIDULIN (E. M.). – Optimal array error-correcting codes. *Problems of Information Transmission*, vol. 21, n°2, 1985, pp. 102–106.

[Gab85b] GABIDULIN (E. M.). – Theory of codes with maximal rank distance. *Problems of Information Transmission*, vol. 21, juillet 1985, pp. 1–12.

[Gab91] GABIDULIN (E. M.). – A fast matrix decoding algorithm for rank-error correcting codes. *In : Algebraic coding*, éd. par COHEN (G.), LITSYN (S.), LOBSTEIN (A.) et ZÉMOR (G.), LNCS. pp. 126–133. – Springer, 1991.

[Gab95] GABIDULIN (E. M.). – Public-key cryptosystems based on linear codes over large alphabets : efficiency and weakness. *In : Codes and Cyphers*, éd. par FARRELL (P. G.). pp. 17–31. – Formara Limited, Southend-on-sea, Essex, 1995.

[GG03] GATHEN (J. VON ZUR) et GERHARD (J.). – *Modern Computer Algebra.* – Cambridge University Press, 2003, 2ème édition.

[Gib95] GIBSON (J. K.). – Severely denting the Gabidulin version of the McEliece public-key cryptosystem. *Designs, Codes and Cryptography*, vol. 6, 1995, pp. 37–45.

[Gib96] GIBSON (J. K.). – The security of the Gabidulin public-key cryptosystem. *In : EUROCRYPT'96*, éd. par MAURER (U.), pp. 212–223. – 1996.

[GJ79] GAREY (M. R.) et JOHNSON (D. S.). – *Computers and Intractability - A Guide to the Theory of NP-Completeness.* – Freeman, 1979.

[GK94] GABIDULIN (E. M.) et KJELSEN (O.). – How to avoid the Sidel'nikov-Shestakov attack. *In : Error control, cryptology, and speech compression*, éd. par CHMORA (O.) et WICKER (S. B.), LNCS. pp. 33–40. – Springer, 1994.

[GL00] GABIDULIN (E. M.) et LOIDREAU (P.). – Subfield subcodes of maximum-rank distance codes. *In : Seventh International Workshop on Algebraic and Combinatorial Coding Theory, ACCT'2000*. pp. 151–156. – Bansko, Bulgarie, juin 2000.

[GL04] GABIDULIN (E. M.) et LOIDREAU (P.). – On subspace subcodes of rank codes. *In : Ninth International Workshop on Algebraic and Combinatorial Coding Theory, ACCT'2004*, pp. 178–184. – Kranevo, Bulgarie, juin 2004.

[GL05] Gabidulin (E. M.) et Loidreau (P.). – On subcodes of codes in rank metric. *In : 2005 IEEE International Symposium on Information Theory, ISIT'05*, pp. 121–123. – septembre 2005.

[GO01] Gabidulin (E. M.) et Ourivski (A. V.). – Modified GPT PKC with right scrambler. *In : Proceedings of the 2nd International workshop on Coding and Cryptography, WCC 2001*, éd. par Augot (D.) et Carlet (C.), pp. 233–242. – 2001. ISBN Number : 2-761-1179-3.

[Gop70] Goppa (V. D.). – A new class of linear error-correcting codes. *Problems of Information Transmission*, vol. 6, n°3, 1970, pp. 207–212.

[GP06] Gabidulin (E. M.) et Pilipchuk (N. I.). – Symmetric matrices and codes correcting beyond the $\lfloor (d-1)/2 \rfloor$ bound. *Discrete Applied Mathematics*, vol. 154, 2006, pp. 305–312.

[GPT91] Gabidulin (E .M.), Paramonov (A. V.) et Tretjakov (O. V.). – Ideals over a non-commutative ring and their application in cryptology. *In : Advances in Cryptology - EUROCRYPT'91*, éd. par Davies (D. W.), LNCS. pp. 482–489. – Springer, 1991.

[GS98] Gabidulin (E. M.) et Simonis (J.). – Metrics generated by families of subspaces. *IEEE Transactions on Information Theory*, vol. 44, n°3, mai 1998, pp. 1336–1342.

[GS99] Guruswami (V.) et Sudan (M.). – Improved decoding of Reed-Solomon and algebraic-geometry codes. *IEEE Transactions on Information Theory*, vol. 45, n°6, septembre 1999, pp. 1757–1767.

[GY06] Gadouleau (M.) et Yan (Z.). – Properties of codes with the rank metric. *In : Proceedings of Globecom 2006*. – 2006.

[Ham06] Hammons (A. R.). – Space-time code designs based on the generalized binary rank criterion with applications cooperative diversity. *In : Coding and Cryptography - WCC 2005, 4th International Workshop on Coding and Cryptography*, éd. par Ytrehus (Ø), LNCS. pp. 69–84. – Springer, 2006.

[HKL03] Helleseth (T.), Kløve (T.) et Levenshtein (V.). – Bounds on the error-correcting capability of codes beyond half the minimum distance. *In : Proceedings of the 3rd International Workshop on Coding and Cryptography, WCC 2003*, éd. par Augot (D.), Charpin (P.) et Kabatianski (G.), pp. 243–251. – 2003.

[HMS98] Hattori (M.), McEliece (R. J.) et Solomon (G.). – Subspace subcodes of Reed–Solomon codes. *IEEE Transactions on Information Theory*, vol. 44, n°5, septembre 1998.

[Jen95] Jensen (J. M.). – Subgroup subcodes. *IEEE Transactions on Information Theory*, vol. 41, n°3, mai 1995, pp. 781–785.

[Joh96] Johannson (T.). – Authentication codes for nontrusting parties obtained from rank metric codes. *Designs, Codes and Cryptography*, vol. 6, 1996, pp. 205–218.

[KI03] Kobara (K.) et Imai (H.). – On the one-wayness against chosen-plaintext attacks of the Loidreau's modified McEliece PKC. *IEEE Transactions on Information Theory*, vol. 49, n°12, décembre 2003, pp. 3160–3168.

[KV03] Koetter (R.) et Vardy (A.). – A complexity reducing transformation in algebraic list decoding of Reed–Solomon codes. *In : 2003 Information Theory Workshop.* – 2003.

[KY02] Kiayias (A.) et Yung (M.). – Cryptographic hardness based on the decoding of Reed-Solomon codes with applications. *In : Proceedings of ICALP*, pp. 232–243. – 2002.

[KY04] Kiayias (A.) et Yung (M.). – Cryptanalyzing the polynomial-reconstruction based public-key system under optimal parameter choice. *In : Advances in Cryptology - ASIACRYPT 2004*, LNCS, pp. 401–416. – 2004. Cryptology ePrint Archive, Report 2004/217.

[Lev06] Levy-dit-Vehel (F.). – Algebraic decoding of rank metric codes. – Special Semester on Gröbner Bases, Linz, Austria, May 2006.

[LK05] Lu (H. F.) et Kumar (P. V.). – A unified construction of space-time codes with optimal rate-diversity tradeoff. *IEEE Transactions on Information Theory*, vol. 51, n°5, mai 2005, pp. 1709–1730.

[LN97] Lidl (R.) et Niederreiter (H.). – *Finite Fields.* – Cambridge University Press, 1997, 2 édition.

[LO06] Loidreau (P.) et Overbeck (R.). – Decoding rank errors beyond error-correcting capability. *In : Tenth International Workshop on Algebraic and Combinatorial Coding Theory, ACCT'2006*, pp. 186–189. – septembre 2006.

[Loi97] Loidreau (P.). – *Éléments sur les codes de Goppa en relation avec le protocole de McEliece.* – Rapport de DEA, École Polytechnique, 1997.

[Loi00] Loidreau (P.). – Strengthening McEliece public-key cryptosystem. *In : Advances in Cryptology - ASIACRYPT 2000*, éd. par Okamoto (T.), LNCS. IACR. – Springer, décembre 2000.

[Loi01] Loidreau (P.). – Codes derived from binary Goppa codes. *Problems of Information Transmission*, vol. 37, n°2, avril 2001.

[Loi04] Loidreau (P.). – Sur la reconstruction des polynômes linéaires : un nouvel algorithme de décodage des codes de Gabidulin. *Comptes Rendus de l'Académie des Sciences : Série I*, vol. 339, n°10, 2004, pp. 745–750.

[Loi05] Loidreau (P.). – *An Algebraic attack against Augot-Finiasz cryptosystem.* – Rapport de recherche RR-5662, http ://www.inria.fr/rrrt/rr-5662.html, INRIA, 2005.

[Loi06] Loidreau (P.). – A Welch-Berlekamp like algorithm for decoding Gabidulin codes. *In : Coding and Cryptography - WCC 2005, 4th International workshop on Coding and Cryptography*, éd. par Ytrehus (Ø.), LNCS, n° 3969. pp. 36–45. – Springer, 2006.

[LS01] Loidreau (P.) et Sendrier (N.). – Weak keys in McEliece public-key cryptosystem. *IEEE Transactions on Information Theory*, vol. 47, n°3, mars 2001.

[LS04] Loidreau (P.) et Sakkour (B.). – Modified version of Sidel'nikov-Pershakov decoding algorithm for binary second order Reed-Muller codes. *In : Ninth International Workshop on Algebraic and Combinatorial Coding Theory, ACCT'2004*, pp. 266–271. – Kranevo, Bulgarie, juin 2004.

[MAG] The magma computational algebra system for algebra, number theory and geometry. http ://magma.maths.usyd.edu.au/.

[McE78] McEliece (R. J.). – *A public-key cryptosystem based on algebraic coding theory.* – Rapport de recherche, Jet Propulsion Lab. DSN Progress Report, 1978.

[MRS98] Marcus (B. H.), Roth (R. M.) et Siegel (P. H.). – *Handbook of Coding Theory, Vol. 2*, chap. 20. – North–Holland, 1998.

[MS77] MacWilliams (F. J.) et Sloane (N. J. A.). – *The Theory of Error–Correcting Codes.* – North Holland, 1977.

[MS94] McEliece (R. J.) et Solomon (G.). – *Trace-shortened Reed–Solomon codes.* – Rapport de recherche 42-117, TDA Progress Report, mai 1994.

[Nie86] Niederreiter (H.). – Knapsack-type cryptosystems and algebraic coding theory. *Problems of Control and Information Theory*, vol. 15, n°2, 1986, pp. 159 – 166.

[OG03] Ourivski (A. V.) et Gabidulin (E. M.). – Column scrambler for the GPT cryptosystem. *Discrete Applied Mathematics*, vol. 128, n°1, mai 2003, pp. 207–221. – Special issue of the second International Workshop on Coding and Cryptography (WCC2001).

[OGHA03] Ourivski (A. V.), Gabidulin (E. M.), Honary (B.) et Ammar (B.). – Reducible rank codes and their applications to cryptography. *IEEE Transactions on Information Theory*, vol. 49, n°12, décembre 2003, pp. 3289–3293.

[OJ02] Ourivski (A. V.) et Johannson (T.). – New technique for decoding codes in the rank metric and its cryptography applications. *Problems of Information Transmission*, vol. 38, n°3, septembre 2002, pp. 237–246.

[Ore33] Ore (Ö.). – On a special class of polynomials. *Transactions of the American Mathematical Society*, vol. 35, 1933, pp. 559–584.

[Ore34] Ore (Ö.). – Contribution to the theory of finite fields. *Transactions of the American Mathematical Society*, vol. 36, 1934, pp. 243–274.

[Our03] Ourivski (A. V.). – Recovering a parent code for subcodes of maximal rank distance codes. *In : Proceedings of the 3rd International workshop on Coding and Cryptography, WCC 2003*, éd. par Augot (D.), Charpin (P.) et Kabatianski (G.), pp. 357–363. – 2003. ISBN Number : 2-7261-1205-6.

[Ove05] Overbeck (R.). – A new structural attack for GPT and variants. *In : Proceedings of MyCrypt 2005*, éd. par Dawson (E.) et Vaudenay (S.), LNCS. pp. 50–63. – Springer, 2005.

[Ove06] Overbeck (R.). – Extending Gibson's attacks on the GPT cryptosystem. *In : Coding and Cryptography - WCC 2005, 4th International workshop on Coding and Cryptography*, éd. par Ytrehus (Ø), n°3969. pp. 178–188. – Springer, 2006.

[PH98] Pless (V. S.) et Huffman (W. C.) (édité par). – *Handbook of Coding Theory.* – Elsevier Science B.V., 1998.

[Rot91] Roth (R. M.). – Maximum-Rank array codes and their application to criss-cross error correction. *IEEE Transactions on Information Theory*, vol. 37, n°2, mars 1991, pp. 328–336.

[Rot96] Roth (R. M.). – Tensor codes for the rank metric. *IEEE Transactions on Information Theory*, vol. 42, n°6, novembre 1996, pp. 2146–2157.

[RP04a] Richter (G.) et Plass (S.). – Error and erasure decoding of rank-codes with a modified Berlekamp-Massey algorithm. *In : 5th Int. ITG Conference on Source and Channel Coding (SCC 04)*. – 2004.

[RP04b] Richter (G.) et Plass (S.). – Fast decoding of rank-codes with rank errors and column erasures. *In : 2004 IEEE International Symposium on Information Theory, ISIT'04*. – 2004.

[RS85] Roth (R. M.) et Seroussi (G.). – On generator matrices of MDS codes. *IEEE Transactions on Information Theory*, vol. 31, n°6, 1985, pp. 826–830.

[Sen91] Sendrier (N.). – *Codes correcteurs d'erreurs à haut pouvoir de correction*. – Thèse de doctorat, Université Paris 6, décembre 1991.

[Sen97] Sendrier (N.). – On the dimension of the hull. *SIAM Journal on Discrete Mathematics*, vol. 10, n°2, mai 1997, pp. 282–293.

[Sen00] Sendrier (N.). – Finding the permutation between equivalent codes : the support splitting algorithm. *IEEE Transactions on Information Theory*, vol. 46, n°4, juillet 2000, pp. 1193–1203.

[Sen01] Sendrier (N.). – *Cryptosystèmes à clé publique basés sur les codes correcteurs d'erreurs*. – Habilitation à diriger des recherches, Université Paris 6, 2001.

[Sid94] Sidel'nikov (V. M.). – A public-key cryptosystem based on binary Reed-Muller codes. *Discrete Mathematics and Applications*, vol. 4, n°3, 1994, pp. 191–207.

[SL05] Shorin (V. V.) et Loidreau (P.). – Application of Groebner bases techniques for searching new sequences with good periodic correlation properties. *In : 2005 IEEE International Symposium on Information Theory, ISIT'05*. – septembre 2005.

[SNC05] Savafi-Naini (R.) et Charnes (C.). – MRD hashing. *Designs, Codes and Cryptography*, vol. 37, 2005, pp. 227–242.

[Sol92] Solomon (G.). – *Nonlinear, nonbinary cyclic group codes*. – Rapport de recherche 42-108, TDA Progress Report, février 1992.

[SP92] Sidel'nikov (V. M.) et Pershakov (A. S.). – Decoding of Reed-Muller codes with a large number of errors. *Problems of Information Transmission*, vol. 28, n°3, 1992, pp. 80–94.

[SRB06] Schmidt (G.), R.Sidorenko (V.) et Bossert (M.). – Error and erasure correction of interleaved Reed-Solomon codes. *In : Coding and Cryptography - WCC 2005, 4th International workshop on Coding and Cryptography*, éd. par Ytrehus (Ø.), LNCS, n° 3969. pp. 22–35. – Springer, 2006.

[SS92] Sidel'nikov (V. M.) et Shestakov (S. O.). – On cryptosystems based on generalized Reed-Solomon codes. *Discrete Mathematics*, vol. 4, n°3, 1992, pp. 57–63.

[Sti90] Stichtenoth (H.). – On the dimension of subfield subcodes. *IEEE Transactions on Information Theory*, vol. 36, 1990.

[Sud97] Sudan (M.). – Decoding Reed-Solomon codes beyond the error-correction diameter. *In : Proceedings of the 35th Annual Allerton Conference on Communication, Control and Computing*, pp. 215–224. – 1997.

[Tav04] Tavernier (C.). – *Testeurs, problèmes de reconstruction univariés et multivariés, et application à la cryptanalyse du DES.* – Thèse, Ecole Polytechnique, janvier 2004.

[Var97] Vardy (A.). – The intractability of computing the minimum distance of a code. *IEEE Transactions on Information Theory*, vol. 43, n°6, novembre 1997, pp. 1757–1766.

[Wie06] Wieschebrink (C.). – An attack on a modified Niederreiter encryption scheme. *In : Proceeding of PKC 2006*, éd. par Yung (M.), Dodis (Y.), Kiayias (A.) et Malkin (T.), LNCS, n° 3958, pp. 14–26. – avril 2006.

Annexe A

Curriculum Vitæ

Fonctions

Depuis mai 2001 :
- Enseignant–chercheur à l'unité de mathématiques appliquées de l'ENSTA.
- Responsable pédagogique du Mastère Spécialisé en *Architecture des Systèmes d'Information* de l' ENSTA.

Diplômes

- 4 mai 2001 : Docteur en sciences de l'Ecole Polytechnique, spécialité *Algorithmique*, mention Très Honorable. Sujet de la thèse : *Etude et Optimisation de cryptosystèmes à clé publique fondés sur la théorie des codes correcteurs*, effectuée au Projet CODES de l'INRIA sous la direction de Pascale Charpin
- juil. 1997 : DEA d'Algorithmique, mention Bien - Ecole Polytechnique et Paris 6
- sept. 1996 : Diplôme d'*Ingénieur de l'Ecole Polytechnique*
- sept. 1995 : Maîtrise de mathématiques - mention "Mathématiques pures", Université de Franche-Comté.
- sept. 1994 : Licence de mathématiques, Université de Franche-Comté.

Publications

Articles acceptés dans des revues internationales (avec comité de lecture)

- T. P. Berger et P. Loidreau. How to mask the structure of codes for a cryptographic use. *Designs, Codes and Cryptography*, 35 :63-79, 2005.
- P. Loidreau. Codes derived from binary Goppa codes. *Problems of Information Transmission*, 37(2) :91-9, 2001.
- P. Loidreau et N. Sendrier. Weak keys in McEliece public-key cryptosystem. *IEEE Transactions on Information Theory*, 47(3) :1207-1211, Mars 2001.

Articles soumis pour publication à des revues internationales (avec comité de lecture)

- P. Loidreau. Properties of codes in rank metric soumis.

- E. M. Gabidulin et P. Loidreau. Properties of subspace subcodes of optimum codes in rank metric, soumis.

Notes aux Compte Rendus de l'Académie des Sciences

- P. Loidreau. Sur la reconstruction des polynômes linéaires : un nouvel algorithme de décodage des codes de Gabidulin. *Comptes Rendus de l'Académie des Sciences : Série I*, 339(10) :745–750, 2004.

Publication dans des actes de conférences internationales avec comité de lecture

- P. Loidreau. A Welch-Berlekamp like algorithm for decoding Gabidulin codes. In *Proceedings of the 4th International Workshop on Coding and Cryptography, WCC 2005*, Mars 2005.
- C. Faure et P. Loidreau. A new public-key cryptosystem based on the problem of reconstructing p-polynomials. In *Proceedings of the 4th International Workshop on Coding and Cryptography, WCC 2005*, Mars 2005.
- T. P. Berger et P. Loidreau. Designing an efficient and secure public-key cryptosystem based on reducible rank codes. In *Proceedings of INDOCRYPT 2004*, Décembre 2004.
- P. Loidreau. Strengthening McEliece public-key cryptosystem. In T. Okamoto, editor, *Advances in Cryptology - ASIACRYPT 2000*, LNCS. IACR, Springer-Verlag, Décembre 2000.

Conférences invitées

- P. Loidreau How to reduce public-key size in McEliece public-key cryptosystems *CLC2006 - Workshop on Codes and Lattices in Cryptography*, Septembre 2006, Darmstadt, Allemagne.
- P. Loidreau Decoding of Gabidulin code and problem of linearized polynomial reconstruction Russian-French mini-conference *Mathematics of Communication*, Novembre 2003, Moscou

Conférences internationales (avec comité de sélection)

- P. Loidreau et R. Overbeck. Decoding rank errors beyond error-correcting capability In *Tenth International Workshop on Algebraic and Combinatorial Coding Theory, ACCT'2006*, Septembre 2006, Zvenigorod, Russie.
- V. V. Shorin et P. Loidreau. Application of Groebner bases techniques for searching new sequences with good periodic correlation properties. In *2005 IEEE International Symposium on Information Theory, ISIT'05*, Septembre 2005.
- E. M. Gabidulin et P. Loidreau. On subcodes of codes in rank metric. In *2005 IEEE International Symposium on Information Theory, ISIT'05*, pages 121–123, Septembre 2005.
- E. M. Gabidulin et P. Loidreau. On subspace subcodes of rank codes. In *Ninth International Workshop on Algebraic and Combinatorial Coding Theory, ACCT'2004*, pages 178–184, Kranevo, Bulgarie, Juin 2004.

- P. Loidreau et B. Sakkour. Modified version of Sidel'nikov-Pershakov decoding algorithm for binary second order Reed-Muller codes. In *Ninth International Workshop on Algebraic and Combinatorial Coding Theory, ACCT'2004*, pages 266–271, Kranevo, Bulgarie, Juin 2004.
- T. P. Berger et P. Loidreau. Security of the Niederreiter form of the GPT public-key cryptosystem. In *2002 IEEE International Symposium on Information Theory, ISIT'02*, Juillet 2002.
- P. Loidreau. Large weight patterns decoding in Goppa codes and application to cryptography. In *Proceedings of 2000 IEEE International Symposium on Information Theory*, page 186, Juin 2000.
- T. P. Berger et P. Loidreau. A Niederreiter version of the GPT public-key cryptosystem. In *Seventh International Workshop on Algebraic and Combinatorial Coding Theory*, pages 72–78, Bansko, Bulgarie, Juin 2000.
- E. M. Gabidulin et P. Loidreau. Subfield subcodes of maximum-rank distance codes. In *Seventh International Workshop on Algebraic and Combinatorial Coding Theory, ACCT'2000*, pages 151–156, Bansko, Bulgarie, Juin 2000. European Community Press.
- P. Loidreau. On the factorization of trinomials over GF(3) In *Fifth International Conference on Finite Fields and Applications*, Augsburg, Allemagne, juillet 1999.
- P. Loidreau. Codes derived from Goppa codes. In *Sixth International Workshop on Algebraic and Combinatorial Coding Theory, ACCT'98*, pages 179–182, Pskov, Russie, Septembre 1998.
- P. Loidreau et N. Sendrier. Some weak keys in McEliece public-key cryptosystem. In *IEEE International Symposium on Information Theory, ISIT'98*, page 382, Cambridge, MA, USA, 1998.

Rapports de recherche

- P. Loidreau. An algebraic attack against Augot-Finiasz cryptosystem. Rapport de recherche RR-5662, INRIA, http ://www.inria.fr/rrrt/rr-5662.html, 2005.
- D. Augot, M. Finiasz, et P. Loidreau. Using the trace operator to repair the polynomial reconstruction based cryptosystem presented at Eurocrypt 2003. Cryptology ePrint Archive, Report 2003/209, 2003. http ://eprint.iacr.org/.
- P. Loidreau. Etude et Optimisation de cryptosystèmes à clé publique fondés sur la théorie des codes correcteurs. Rapport de Thèse de doctorat, Ecole Polytechnique, mai 2001.
- P. Loidreau. On the factorization of trinomials over GF(3). Rapport de recherche RR-3918, INRIA, http ://www.inria.fr/rrrt/rr-3918.html, April 2000.
- P. Loidreau. Eléments sur les code de Goppa en relation avec le protocole de McEliece Rapport de stage de DEA, Ecole Polytechnique, juillet 1997.

Vulgarisation scientifique

- P. Loidreau. Pour quelques bits d'information. *MISC - Le magazine de la sécurité informatique*, (20), Juillet-Août 2005.
- P. Loidreau. Génération d'aléa en cryptographie. *MISC - Le magazine de la sécurité informatique*, (6), 2003.
- P. Loidreau. Le partage de secret. *MISC - Le magazine de la sécurité informatique*, (3), 2002.

- P. Loidreau. Le transfert inconscient. *MISC - Le magazine de la sécurité informatique*, (2), 2002.
- P. Loidreau. L'identification à divulgation nulle de connaissance. *MISC - Le magazine de la sécurité informatique*, (1), 2002.
- P. Loidreau, Introduction à la cryptographie. *Linux magazine - Hors-série Sécurité*, (8), 2001.

Présentations et séminaires donnés au niveau national

- *Métrique rang et applications en cryptographie.* Seminaire de Cryptographie de l'Université de Rennes 1.
- *Métrique rang, reconstruction et cryptographie.* Séminaire Cryptographie de l'Université de Caen, mai 2004
- *Un nouvel algorithme de décodage pour la métrique rang.* Séminaire AZUR'CRYPT, Luminy, février 2004
- *Cryptosystèmes fondés sur la métrique rang.* Journées "Codes et Cryptographie", Luminy, Novembre 2002

Activités d'encadrement de travaux de recherche

Encadrement de stage Post-Doctoral

- Vitaly Shorin : Septembre 2004 - Septembre 2005.

  Résumé : Le sujet du stage a été la recherche de séquences unimodulaires parfaites. Pour déterminer de telles séquences, il faut résoudre des systèmes non-linéaires à coefficients entiers. Les techniques précédentes s'appuyaient sur l'utilisation des résultantes. Le stagiaire s'est approprié les techniques de résolutions de système non-linéaire par bases de Groebner. Grâce à l'algorithme F4 implanté dans les dernières versions du logiciel MAGMA, nous avons obtenu notamment une famille infinie de séquences unimodulaires parfaites à 6 phases.

Encadrement de thèse

- Cédric Faure, en co-tutelle de thèse depuis septembre 2004. Soutenance prévue septembre 2007

  Intitulé de la thèse : *Études de systèmes cryptographiques construits à l'aide de codes correcteurs*

  co-directeur de Thèse : Nicolas Sendrier.

  Résumé : Le doctorant s'intéresse aux propriétés algébriques des codes, en particulier en métrique rang en vue de leur utilisation dans la conception de systèmes de chiffrement à clé publique. Il s'agit en particulier d'étudier la sécurité d'un système de chiffrement fondé sur la reconstruction des polynômes linéaires. C. Faure a montré comment protéger le système contre les attaques connues en utilisant les propriétés de projections sur un sous-corps. Actuellement celui-ci travaille sur l'évaluation du nombre de mots d'un code de Gabidulin dans une sphère de l'espace ambiant. Il a obtenu un résultat en ce qui concerne le nombre moyen et s'intéresse désormais à l'établissement de l'équivalent d'une borne de Johnson pour les codes de Gabidulin.

Il s'agit d'un premier pas vers la conception d'un algorithme de décodage en liste pour les codes de Gabidulin.

– Bassem Sakkour, en co-tutelle de thèse depuis septembre 2003. Soutenance prévue 1er trimestre 2007.
  Intitulé de la thèse : *Etude du décodage des codes de Reed-Muller et application à la cryptographie à clé secrète.*
  co-directrice de Thèse : Pascale Charpin.

  Résumé : L'objet de cette thèse consiste en l'étude et l'amélioration d'algorithmes de décodage de codes de Reed-Muller. Parti de l'algorithme de Sidel'nikov-Pershakov pour les codes de Reed-Muller d'ordre 2 étudié au cours de son DEA, il est parvenu à construire un algorithme augmentant très significativement la performance de décodage pour les codes de longueur moyenne, tout en préservant la complexité. Il a implémenté plusieurs algorithmes de décodage en langage C++, dont des algorithmes de décodage récursif. Il a ensuite comparé leurs performances, ce qui lui permis de montrer que, sur des codes de longueur moyenne, l'algorithme qu'il avait conçu corrigeait plus d'erreurs que les algorithmes existants.

Encadrement de stages de DEA

– Cédric Faure,
  Etude d'un système de chiffrement à clé publique fondé sur le problème de reconstruction de polynômes linéaires, stage de DEA, (Mars à Septembre 2004).

  Résumé : Dans ce stage l'étudiant a repris le principe du cryptosystème Augot-Finiasz fondé sur le problème de reconstruction de polynôme publié à la conférence eurocrypt 2003 et en a transposé le principe en métrique rang. Cela a abouti à la conception d'un nouveau cryptosystème dont la sécurité repose sur le problème de reconstruction de polynômes linéaires.

– Bassem Sakkour,
  Décodage des codes de Reed-Muller au delà de la distance minimale, stage de DEA, (Mars à juin 2003).

  Résumé : Le but du stage était dans un premier temps de comprendre l'algorithme de décodage de Sidel'nikov-Pershakov pour les codes de Reed-Muller d'ordre 2. Dans un second temps, l'étudiant en a effectué une implémentation efficace en langage C et a comparé ses performances en décodage, par rapport aux bornes proposées par les concepteurs.

– Alexandre Hersans,
  Etude et implémentation des attaques de Gibson contre le cryptosystème GPT, (Avril-Août 2000).

  Résumé : Ce travail a consisté d'une part à la lecture et la compréhension des attaques que K. Gibson a publiées contre le cryptosystème GPT fondé sur la métrique rang, et d'autre part en leur implémentation dans le langage de calcul algébrique MAGMA. Ces résultats ont permis de valider en pratique l'efficacité des attaques ainsi que leur complexité. Cela permis de déterminer des jeux de paramètres pour lesquels le système de chiffrement pouvait être considéré comme résistant à ce type d'attaques.

Encadrement d'autres stages

- Christelle Roux,
  Implantation en langage C du cryptosystème Augot–Finiasz fondé sur la reconstruction de polynômes, projet personnel en laboratoire de 2ème année de l'ENSTA, (Mai-Juin 2003).

  Résumé : L'étudiante a programmé le cryptosystème Augot-Finiasz original en langage C à l'aide de la bibliothèque de calcul ZEN. Elle a implémenté deux algorithmes de décodage pour les codes de Reed-Solomon et a étudié les performances du cryptosystème en chiffrement et déchiffrement en fonction de l'algorithme de décodage utilisé.
- Alexandra Petrova,
  Etude et Implantation logicielle d'instances du cryptosystème Gabidulin-Paramonov-Tretjakov, stage de fin d'étude de l'Ecole Polytechnique (Avril à Juin 2003).

  Résumé : Le stage a consisté en l'implantation logicielle en langage C du système de chiffrement GPT, publié en 1991, et reposant sur les propriétés de la métrique rang
- Bruno Giroire,
  Implantation d'un algorithme de décodage de codes de Gabidulin, projet personnel en laboratoire de 2ème année de l'ENSTA, (Juillet-Août 2002).

  Résumé : Il s'agit de la réalisation en langage C d'une implémentation efficace l'algorithme de décodage par syndrome des codes de Gabidulin. L'étudiant a utilisé la bibliothèque de calcul sur les corps finis ZEN.

Activités liées à la recherche

Comités de programmes

- Membre du comité de programme du congrès *5th International Workshop on Coding and Cryptography, WCC 2007*, avril 2007, INRIA Rocquencourt.

Responsabilité de programme :

- Responsable de l'action *Mathématiques discrètes appliquées au codage et à la protection de l'information* du programme ECO-NET du Ministère des Affaires Etrangères, consistant à développer les liens de recherches avec des pays de l'ancien bloc de l'Est. Durée : 2 ans, terminée fin décembre 2005.

Participation à des Actions :

- Membre de l'ACI OCAM (Opérateurs Cryptographiques et Arithmétique Matérielle) dontle but est d'étudier l'implantation matérielle de primitives cryptographiques utilisant la théorie algébrique des codes, septembre 2003 à août 2006.
- Membre de l'ACI ACCESS (Outils algébriques et combinatoires pour la construction et l'étude de systèmes à clé publique), 2001 à 2004.

Organisation d'événements et de séminaires :

- Membre du comité d'organisation de la conférence *Fast Software Encryption, FSE 2005*, 21–23 février 2005, ENSTA.
- Président du comité d'organisation du congrès *3rd International Workshop on Coding*

and Cryptography, WCC 2003, 24–28 mars 2003, INRIA Rocquencourt.
- Membre du comité d'organisation du congrès *2nd International Workshop on Coding and Cryptography, WCC 2001*, 08–12 janvier 2001, Paris, Cercle Militaire.
- Organisateur du séminaire mensuel, *Codage, Cryptographie et Algorithmique* à l'ENSTA. http ://www.ensta.fr/~loidreau/CCA/

Travail de rapporteur pour des revues :
- IEEE Transaction on Information Theory.
- Designs, Codes and Cryptography.
- Comptes Rendus de l'Académie des Sciences, série I.
- Journal of Cryptology.

Autres activités

- Membre du conseil d'administration de la *Société Mathématique de France* (SMF) depuis juin 2006.
- Membre de la commission d'enseignement de la *Société Mathématique de France* (SMF) depuis septembre 2005.
- Responsable de la formation Mastère Spécialisé en *Architecture des Systèmes d'Information* de l'ENSTA. Cette formation est désormais depuis peu conventionnée avec le CNAM (Conservatoire National des Arts et Métiers).

  Il s'agit d'un cycle d'une année de formation continue labellisée par la *Conférence des Grandes Ecoles* (CGE). Dans ce cadre, je m'occupe de la coordination des enseignements, recherche et contacts avec les enseignants. Ce cycle regroupe une dizaine d'étudiants chaque année.
- Enseignements :
  - TDs du cours *Fonction de la variable complexe* de 2ème année de l'ENSTA.
  - Cours et TDs d'*Eléments de mathématiques discrètes*, de 2ème année de l'ENSTA.
  - TPs du cours *Primitives cryptographiques* de 3ème année de l'ENSTA, 2001–
  - TDs et TPs du cours *Algorithmiques en langage C* de l'ESIEA, 1999

Index