Codes Structurés pour la Cryptographie : des Fondations Théoriques de Sécurité aux Applications

Maxime Bombar

INSTITUT POLYTECHNIQUE DE PARIS

NNT : 2023IPPAX109

Thèse de doctorat

Structured Codes for Cryptography: from Source of Hardness to Applications

Thèse de doctorat de l'Institut Polytechnique de Paris
préparée à École Polytechnique

École doctorale n°626 École doctorale de l'Institut Polytechnique de Paris (EDIPP)
Spécialité de doctorat : Mathématiques et Informatique

Thèse présentée et soutenue à Palaiseau, le 15/12/2023, par

MAXIME BOMBAR

Composition du Jury :

Adeline Roux-Langlois Directrice de recherche, CNRS (GREYC, Caen)	Présidente
Steven Galbraith Professor, University of Auckland	Rapporteur
Gilles Zémor Professeur, Université de Bordeaux (IMB)	Rapporteur
André Chailloux Chargé de recherche, INRIA (COSMIQ)	Examinateur
Lisa Kohl Researcher, CWI (Cryptology Group)	Examinatrice
Damien Stehlé Directeur scientifique, Cryptolab	Examinateur
Alain Couvreur Directeur de recherche, INRIA (GRACE)	Directeur de thèse
Thomas Debris-Alazard Chargé de Recherche, INRIA (GRACE)	Co-directeur de thèse
Nicolas Resch Assistant professor, University of Amsterdam	Invité

À mes parents.

Remerciements

Ces mots sont les derniers que j'écris dans ce manuscrit, pourtant ce sont probablement parmi les plus importants, et certainement ceux qui seront les plus lus. Il est pour moi difficile de mettre des mots sur l'émotion que je ressens à la fin de ce chapitre significatif de ma vie, mais je suis empli de gratitude envers toutes les personnes qui auront jaloné cette aventure.

Cette thèse a débuté sous le signe du Covid et des périodes de confinements et couvre-feux successifs qui m'ont personnellement profondément marqué; et ce travail a pu à ce moment représenter un échappatoire pour moi, me donner un objectif pour avancer. J'espère pour rien au monde devoir revivre ça, mais je reste content du résultat.

Mes premières pensées vont vers les deux formidables personnes qui ont dirigé ma thèse. Alain, merci d'avoir été là, j'ai tant appris à ton contact, aussi bien en tant que chercheur, mais aussi en tant qu'humain. Merci pour nos visios à des heures improbables. Merci pour la confiance que tu m'as accordée, pour les moments où on pensait avoir tout résolu et pour mettre le doute dans nos propres résultats. Merci pour la relecture approfondie de ce manuscrit, et de manière générale pour la relecture de tout mon travail.

Thomas, comment mettre en mots ce qu'on a vécu pendant ces plus de 3 ans ? Je ne saurais te remercier assez, aussi bien pour nos moments de travail que pour les innombrables soirées à Paris, à Budapest, à Washington, en Californie ou sur le plateau de Saclay. Parfois même les deux à la fois. Ah ces deadlines de conférence dans des bars jusqu'au bout de la nuit... Ou encore cette réflexion sur OCP à San Francisco. Merci pour ta patience devant ma lenteur en calcul, mais promis, je m'améliore. Merci pour tout ce que tu as fait, pour tes relectures attentives, pour tes remarques précises. Et merci encore de continuer même après cette soutenance au moment où je prépare mes candidatures.

Je souhaite ensuite remercier très chaleureusement les autres membres de mon jury: Adeline Roux-Langlois, Steven Galbraith, Gilles Zémor, André Chailloux, Lisa Kohl, Damien Stehlé et Nicolas Resch. *I particularly thank Steven for his thorough proofreading of this manuscript. I am sorry that it was too long, but I hope you enjoyed it anyway. And thank you so much for dedicating your Friday evening for my defense.* Merci Gilles d'avoir aussi accepté de rapporter cette thèse. Je réitère les mêmes excuses que pour Steven sur la longueur de ma thèse, mais j'espère que tu auras trouvé sa lecture intéressante tout de même.

Un grand merci à mes coauteurs qui n'ont pas encore été cités: Geoffroy et Clément. J'ai aussi beaucoup appris en travaillant avec vous et j'espère que notre collaboration n'est pas terminée.

Je voudrais aussi remercier les membres de l'équipe GRACE que je n'ai pas encore cités. Merci pour les innombrables discussions pendant les déjeuners et pauses café. Merci à Ben, Daniel, François, Françoise et Olivier. François, j'ai adoré travailler avec toi pour l'enseignement, que ce soit pour le cours de crypto ou celui de java. Même si elle s'est beaucoup vidée depuis le début, merci aussi aux non-permanents que j'ai pu connaître pendant mon passage dans l'équipe ou pour les nouvelles têtes que j'ai pu croiser en conférence, pendant le gt-c2 ou durant mes brefs retours avant ma soutenance: Adrien, Anaëlle, Anaïs, Angelo, Antonin, Azam, Bruno, Clémence,

Gustavo, Hugo, Ilaria, Isabella, Jade, Mathilde, Matthieu, Maxime, Nadja, Nihan, Rakhi, Sarah, Youssef.

Il est un événement qui a régulièrement ponctué ma thèse: le groupe de travail codes et cryptographie à Inria Paris. J'ai adoré y participer, ces moments font partie des plus importants pour moi dans la recherche. Ses membres sont trop nombreux pour pouvoir tous les citer sans erreur, mais un grand merci à Jean-Pierre pour leur organisation.

En parlant de groupes de travail, je souhaite aussi remercier profondément les membres et sympathisants de l'ANR Barracuda. Les deux retraites auxquelles j'ai participé à Métabief[i] et à Jonzac ont été exceptionnelles.

I would also like to warmly thank the members of the Cryptology group[ii] *at CWI, which I joined just before wrapping up this manuscript. Thank you to Léo, Marc, Ronald, Serge, Aron, Eamonn, Jelle, Ludo, Michael, Pedro, Shane, Simona, Yu-Hsuan, Zhe. I was away most of those first months, but I'll be more present from now on!*

Je souhaiterais également remercier mes amis. Tout d'abord les membres de la Kolok, avec qui j'ai habité pendant 5 ans, et avec qui nous avons partagé les périodes de confinement. Merci en particulier à Lev-Arcady, Lisa, Michaël. Merci à vous pour votre support. Merci aussi à Boudy, Charlie, Pagnyx, Talbot, Zadou et tous les autres amis de Cachan. Merci aussi à ceux d'avant, Loïs que j'ai recroisé par hasard lors d'une visite à Bordeaux: le monde de la recherche est si petit; Elyes, que j'ai rencontré il y a plus de 15 ans déjà.

Merci à tous ceux qui auront assisté à ma soutenance, que ce soit en présentiel comme en visio. Ça compte beaucoup pour moi.

Enfin, un énorme merci à ma famille, mes frères, et surtout mes parents à qui je dédie ce manuscrit. Vous avez toujours été là pour moi, et sans vous cette thèse ne serait pas ce qu'elle est.

[i] Savez vous comment s'appellent les habitants de Métabief ?
¡ sıs-ʇɐɥƆ sǝ⅂

[ii] that I didn't already mention before

Résumé long en Français

I start this manuscript with a detailed summary in French. The non-French speaking reader can safely jump to the Introduction.

Cryptographie post-quantique

Que ce soit pour communiquer avec notre famille ou nos amis, accéder à nos données bancaires, consulter nos informations de santé, ou encore gérer des alarmes à distance et autres objets connectés, de plus en plus de données privées circulent sur Internet. Une question essentielle se pose alors :

Comment protéger nos données privées, et s'assurer qu'elles le restent lors de nos communications en ligne ?

La cryptographie est précisément la science qui vise à répondre à cette question. Avec le développement de solutions gratuites comme *Let's Encrypt*, on estime aujourd'hui que plus de 80% de nos communications sur Internet sont chiffrées (contre à peine 50% en 2017)[iii]. On utilise pour cela, ce qu'on appelle des protocoles d'échange de clés, où deux entités (par exemple un client et un serveur) vont se mettre d'accord sur une donnée secrète (la clé), qui sera utilisée ultérieurement dans un système de chiffrement symétrique, comme AES par exemple. Aujourd'hui, la plupart de ces mécanismes d'échange de clés déployés repose sur des variantes d'un problème nommé d'après ses auteurs Diffie et Hellman ([DH76]) et qui peut se formuler comme suit : étant donné un groupe cyclique $G = \langle g \rangle$ rendu public, et deux éléments $g^a, g^b \in G$, la question est de calculer l'élément g^{ab}. Pour plusieurs choix du groupe G, ce problème est communément considéré comme extrêmement difficile, dans le sens où même le plus puissant ordinateur nécessiterait un temps supérieur à l'âge de l'univers pour les résoudre directement. Un autre problème issu de la théorie des nombres et très important en cryptographie est le problème de factorisation : étant donné un entier $N = p \times q$ produit de deux (grands) nombres premiers p et q, la question est de retrouver ces facteurs. Les cryptosystèmes RSA [RSA78] reposent entre autres sur ce problème. Cependant, d'autres domaines des mathématiques comme la théorie des réseaux euclidiens ou encore les codes correcteurs d'erreurs sont aussi sources de problèmes difficiles intéressants dans un contexte cryptographique. Néanmoins, on sait depuis les années 1990 et l'algorithme de Shor [Sho94], que le problème de factorisation, aussi bien que les problèmes Diffie-Hellman, deviennent en réalité faciles dès lors qu'un adversaire a accès à un ordinateur quantique. Si la construction d'une telle machine avec suffisamment de ressources est toujours aujourd'hui une question ouverte, elle est moins improbable qu'elle ne l'était il y a à peine une dizaine d'années. C'est pourquoi il faut commencer dès aujourd'hui à concevoir des nouvelles méthodes dites

[iii] Source : https://letsencrypt.org/stats/#percent-pageloads

« post-quantiques », c'est-à-dire résistantes à l'ordinateur quantique. Il est d'autant plus important de commencer à y réfléchir le plus tôt possible, que le déploiement d'une nouvelle technologie à grande échelle prend beaucoup de temps.[iv]

C'est dans ce contexte que l'agence américaine NIST (*National Institute for Standards and Technology*) a entamé en 2017 et 2023 deux processus de standardisation de cryptosystèmes post-quantiques dont les premiers standards ont été annoncés en Juillet 2022. Les grands gagnants de cette compétition sont les réseaux euclidiens puisque parmi les 4 algorithmes déjà sélectionnés, 3 sont à base de réseaux : un chiffrement[v] KYBER [ABDK+21], et deux signatures : DILITHIUM [LDKL+20] et FALCON [FHKL+20]. Cependant, le NIST a annoncé que le ou les prochains standards seront des constructions à base de codes correcteurs, choisis parmi BIKE [AABB+22a], HQC [AABB+22b] ou CLASSIC MCELIECE [ABCC+22]. Cet état de fait n'est en réalité pas surprenant : les cryptographies à base de réseaux euclidiens et à base de codes reposent sur des problèmes similaires dits d'algèbre linéaire bruitée (ou sous contrainte), et il n'est dès lors pas surprenant que soit les deux soient standardisées, soit aucune des deux.

Cryptographie à base de bruit

L'idée à la base de la cryptographie à base de codes, tout comme de celle à base de réseaux euclidiens, est que résoudre un système linéaire sur un corps fini $\mathbb{F}_q$ est en général très facile, alors que ce problème devient vite extrêmement difficile lorsqu'on exige que les solutions vérifient une certaine contrainte non linéaire. Plusieurs types de contraintes ont été considérées dans la litérature (pour simplifier on va supposer qu'il n'existe qu'une seule solution du système vérifiant la dite contrainte) :

- On peut exiger que les coefficients de la solution soient petits, pour une certaine mesure de grandeur. Pour un bon choix de paramètres, on peut alors montrer que ce problème est lié à des problèmes algorithmiques difficiles sur les réseaux euclidiens, d'où le nom cryptographie « à base de réseaux euclidiens ».
- On peut exiger que le vecteur solution ait beaucoup de coefficients nuls, sans borner la taille des autres coefficients. C'est-à-dire exiger que la solution ait un petit poids de Hamming. C'est l'objet de la cryptographie « à base de codes ».
- Si le système est définit sur une certaine extension $\mathbb{F}_{q^m}/\mathbb{F}_q$, alors le choix d'une base permet d'étendre chaque équation (et inconnue) sur $\mathbb{F}_q$. En particulier, on peut étendre le vecteur solution en une matrice à m lignes, à coefficients dans $\mathbb{F}_q$, et on peut exiger que la solution ait un petit rang. C'est l'objet de la cryptographie dite « en métrique rang »

Dans cette thèse, on s'intéresse principalement aux fondements de la cryptographie à base de codes, en s'inspirant de ce qui a été fait pour celle à base de réseaux, mais la première partie concerne la cryptographie en métrique rang. Plus formellement, un code correcteur d'erreurs (linéaire) de longueur n et de dimension k sur un corps fini $\mathbb{F}_q$ est un sous-espace vectoriel $\mathscr{C} \subset \mathbb{F}_q^n$ de dimension k. Il est en général spécifié par une base, écrite sous la forme d'une matrice $\mathbf{G} \in \mathbb{F}_q^{k\times n}$ appelée *matrice génératrice*, ou encore comme le noyau d'une matrice $\mathbf{H} \in \mathbb{F}_q^{(n-k)\times n}$ dite *matrice de parité*. Ils ont été initialement introduits afin de corriger les erreurs qui arrivent lors de communications numériques, d'où le nom « code correcteur d'erreur ». Plus précisément, un message est représenté par un vecteur $\mathbf{m} \in \mathbb{F}_q^k$, et est vu comme un élément de $\mathscr{C}$ via une

[iv] Par exemple, le protocole réseau IPv6, proposé dans les années 1990 comme remplacement d'IPv4, n'est toujours pas déployé partout, alors même que les addresses IPv4 sont déjà épuisées.

[v] Ou plutôt un mécanisme d'encapsulation de clé

opération d'encodage $\mathbf{c} \stackrel{\text{def}}{=} \mathbf{mG}$. Autrement dit, on rajoute $n-k$ éléments de $\mathbb{F}_q$ qui agissent comme une forme de redondance ($\mathbf{m}$ et $\mathbf{c}$ portant la même information lorsque $\mathbf{G}$ est de rang plein, ce qui est toujours le cas en pratique). À l'autre bout du canal, le recepteur voit arriver un vecteur $\mathbf{y} = \mathbf{c} + \mathbf{e}$ où certaines coordonnées de $\mathbf{c}$ ont été corrompues puisque le canal n'est pas parfait. Le nombre d'erreurs est caractérisé par le nombre de coefficients non nuls de $\mathbf{e}$, c'est-à-dire son poids de Hamming, et on espère que si ce poids de Hamming est suffisamment faible, alors on arrivera à retrouver $\mathscr{C}$. Le lien avec le problème d'algèbre linéaire sous contrainte est alors clair : si $\mathbf{H}$ est une matrice de parité du code $\mathscr{C}$, alors $\mathbf{Hy}^\top = \mathbf{He}^\top$, et retrouver $\mathbf{e}$ revient à trouver la solution de ce système vérifiant la contrainte sur le poids.

En réalité, ce problème est difficile, comme l'avait déjà observé Shannon dès les balbutiements de la théorie de l'information [Sha48 ; Sha49]. Il a même été prouvé NP-complet [BMT78] (donc difficile en pire cas). Cependant, il est aussi extrêmement difficile en moyenne, ce qui est beaucoup plus pertinent dans un contexte cryptographique. En effet, même après 70 ans de recherche, les meilleurs algorithmes de décodage ont une complexité exponentielle en le poids de l'erreur. Il est d'autant plus remarquable que même aujourd'hui, aucun algorithme quantique n'a apporté d'amélioration significative. Néanmoins, certaines familles particulières de codes bénéficient d'algorithmes de décodage efficaces, comme les codes de Reed-Solomon ([RS60]) ou de Goppa ([Gop70]), ou encore les codes LDPC ([Gal63]) ou MDPC ([MTSB13]). C'est ainsi que McEliece a proposé dès 1978[vi] le premier schéma de chiffrement à base de codes, et qui fonctionne de la façon suivante[vii] : étant donné un code $\mathscr{C}$ bénéficiant d'un algorithme de décodage efficace $\mathscr{D}$, la clé publique consiste en une base aléatoire $\mathbf{G}$ de $\mathscr{C}$, tandis que la clé secrète est $\mathcal{D}$. Dès lors, chiffrer un message $\mathbf{m}$ peut se faire simplement en l'encodant dans $\mathscr{C}$ et en rajoutant une erreur aléatoire $\mathbf{e}$: $\text{Enc}(\mathbf{m}) \stackrel{\text{def}}{=} \mathbf{mG} + \mathbf{e}$. Pour déchiffrer, il suffit alors d'utiliser l'algorithme de décodage $\mathscr{D}$ supposé secret pour éliminer $\mathbf{e}$ et retrouver le message original. La sécurité d'un tel schéma repose entre autres sur le fait qu'il doit être impossible de retrouver l'algorithme $\mathscr{D}$ à partir d'une base aléatoire du code, et donc le seul espoir d'un attaquant est de procéder à un décodage générique. En particulier, le choix du code $\mathscr{C}$ est extrêmement important, et de nombreux choix (comme les codes de Reed-Solomon ou les codes LDPC) aboutissent à un cryptosystème dont la sécurité est désastreuse. Il est toujours surprenant que le choix original de McEliece d'utiliser des codes de Goppa semble toujours résister aux attaques, et est à la base de la soumission Classic McEliece au NIST.

Dans cette thèse cependant, on s'intéresse à une autre approche, dûe à Alekhnovich [Ale03], qui est beaucoup plus proche de la cryptographie à base de réseaux euclidiens. L'idée est qu'en réalité un mot de code bruité $\mathbf{mG} + \mathbf{e}$ d'un code spécifié par une matrice aléatoire $\mathbf{G}$ est très proche d'un vecteur aléatoire, même lorsque $\mathbf{G}$ est connu. Plus précisément, on peut montrer que s'il existe un algorithme $\mathscr{A}$ capable de résoudre la variante *décisionnelle* du problème de décodage, c'est-à-dire capable de *distinguer* entre un couple de la forme $(\mathbf{G}, \mathbf{mG}+\mathbf{e})$ et un couple $(\mathbf{G}, \mathbf{y}^{\text{unif}})$ totalement aléatoire, alors on peut transformer $\mathscr{A}$ en un véritable *décodeur* pour le code de matrice génératrice $\mathbf{G}$. En d'autres termes, on dit que la distribution des couples $(\mathbf{G}, \mathbf{mG}+\mathbf{e})$ est *pseudoaléatoire*. Un tel résultat est appelé une *réduction de recherche à décision*.

Structure algébrique supplémentaire

En général, les systèmes cryptographiques dont la sécurité repose sur le problème de décodages de codes aléatoires ont des tailles de clés relativement grosses, pouvant atteindre plusieurs Mo. Afin de réduire cette taille, il a été proposé de considérer des matrices ayant une structure

[vi] Ce qui fait du cryptosystème de McEliece le plus ancien encore résistant aux attaques, même quantiques.
[vii] Description simplifiée

algébrique supplémentaire. Par exemple, on peut considérer des matrices formées de sous-matrices circulantes, c'est-à-dire telles que chaque ligne est une rotation circulaire de la première. Un code possédant une telle matrice génératrice (ou de parité) est appelé code quasi-cyclique. Il se trouve qu'aujourd'hui on ne sait toujours pas se servir de cette structure additionnelle pour améliorer les algorithmes de décodages de façon significative. Dès lors, on considère en général que le problème de *recherche* reste difficile même avec des codes quasi-cycliques. En revanche, pour le problème de décision la question est beaucoup moins claire. On part aussi du principe que cette variante décisionelle est difficile, mais aucune réduction de recherche à décision n'est connue dans le cas des codes structurés. C'est d'autant plus préoccupant que deux des trois derniers cryptosystèmes considérés par le NIST pour standardisation (BIKE et HQC) reposent sur ces versions structurés.

En revanche, la situation est très différente dans le monde de la cryptographie à base de réseaux euclidiens structurés, qui apparaît alors très en avance par rapport aux codes. En effet, depuis l'article fondateur [LPR10] qui a formellement introduit le problème Ring-LWE, et prouvé une réduction de recherche à décision, de nombreuses réductions ont été trouvées dans le cas des réseaux structurés. C'est d'ailleurs probablement ce manque de réductions qui a été un très gros frein à l'adoption de la cryptographie à base de codes. Ce fait a été en particulier mentionné par le NIST dans leurs différents rapports sur le precessus de standardisation [AAAC+20 ; AACD+22].

Calcul sécurisé

Le chiffrement simple n'est aujourd'hui plus suffisant pour garantir le respect de notre vie privée. En effet, les entreprises proposant des services web vont aussi réaliser des calculs dessus, par exemple pour suggérer des nouveaux produits en fonction de notre habitude de consommation, et bien souvent ces calculs vont être délégués à des services tiers que nous ne connaissons pas et en qui on ne peut avoir confiance. C'est pourquoi il faut aussi développer des primitives plus avancées pour nous protéger dans cet usage moderne d'Internet, et ainsi réaliser ce qu'on appelle du *calcul sécurisé*. Par ailleurs, il faut que ce calcul sécurisé puisse se faire de manière transparente pour les utilisateurs, ce qui implique une recherche de l'efficacité. Plus personne n'a envie d'attendre plusieurs minutes pour charger une page web ou une application. Comme on dit souvent en cryptographie, « si la cryptographie fonctionne, personne ne doit s'en rendre compte ».

Dans la dernière partie de cette thèse, je me suis intéressé à une forme particulière de calcul sécurisé appelé *calcul multiparti* (souvent abbrégé en MPC), qui permet à un groupe d'utilisateurs (par exemple un client, des serveurs et des services tiers) de réaliser en commun un calcul, ou une fonctionnalité, sur des données privées, en ne révélant rien d'autre que le résultat. Le calcul multiparti sécurisé a été imaginé dès les années 1980. Cependant, les protocoles proposés nécessitent en général que les participants communiquent énormément de données lors de leur exécution. Afin de les rendre pratique, il est nécessaire de limiter ce coût de communication : plus personne n'a envie de devoir attendre plusieurs minutes pour charger une application sur son téléphone mobile. Un élément de réponse est donnée par une observation due à Beaver [Bea91 ; Bea95] qui montre comment construire des protocoles extrêmement efficaces, dès lors que les m participants ont accès à une source de confiance produisant une suite de m-uplets $(r_1, \ldots, r_m)$ aléatoires et indépendants, tel que chaque m-uplet est soumis à une certaine corrélation $C(r_1, \ldots, r_m)$. Les protocoles modernes de MPC utilisant ce paradigme connu sous le nom de « MPC avec pré-calcul », sont alors divisés en deux phases : une première phase indépendante des entrées de la fonction à calculer qui sert à générer cet aléa corrélé, et une seconde phase « en ligne » où l'aléa corrélé est consommé pour limiter les échanges entre les parties. Cette seconde phase est par ailleurs sûre au sens de la théorie de l'information. En contrepartie, la quantité d'aléa corrélé est

importante, et se chiffre en millions, voire milliards, de tels m-uplets.

Depuis leur introduction récente dans [BCGI18 ; BCGI+19], les générateurs de pseudo-aléa corrélé (PCG pour *Pseudorandom Correlation Generators*) sont vus comme la solution la plus efficace pour cette phase de mise en place. Plus précisément, un PCG va permettre de distribuer une version compressée de cet aléas corrélé à tous les participants, qui sera ensuite décompressée localement sans communication supplémentaire entre les participants. On parle alors de « précalcul silencieux ». Les deux constructions mentionnées ci-dessus reposent sur la difficulté du problème de décodage, mais ne permet de ne générer de l'aléa corrélé que pour deux participants uniquement ($m = 2$) : elles ne possèdent pas la propriété supplémentaire dite de « programmabilité ». Afin de construire des PCG programmables, il a été important de se tourner vers des versions structurées du problème de décodage. En particulier, [BCGI+20b] propose d'utiliser des codes quasi-cycliques particuliers. Cependant, cette version n'est en réalité pas standard, et souffre d'une étude théorique pour supporter l'analyse de sa difficulté. Plus important encore, elle ne permet de construire de l'aléa corrélé que sur des corps suffisamment gros. En particulier, [BCGI+20b] propose d'utiliser des corps $\mathbb{F}_p$ où p est un nombre premier de 128 bits, ce qui ne permet de faire du MPC que dans de tels corps.

Contributions de cette thèse

Cette thèse est organisée autour de trois parties.

Partie I : Codes $\mathbb{F}_{q^m}$-linéaires munis de la métrique rang

Dans cette partie formée des chapitres 2 et 3, je me suis intéressé à la cryptographie en métrique rang, et plus précisément utilisant des codes linéaires sur une certaine extension $\mathbb{F}_{q^m}/\mathbb{F}_q$. Du fait de la cyclicité de cette extension (c'est-à-dire qu'elle est Galoisienne, de groupe de Galois cyclique), les codes $\mathbb{F}_{q^m}$-linéaires ont une nature assez proche des codes quasi-cycliques, mais la géométrie de l'espace ambiant est très différente. Plus précisément, on munit l'espace ambiant $\mathbb{F}_{q^m}^n$ d'une métrique définie par le rang : le choix d'une base de l'extension permet de voir chaque vecteur comme une matrice à m lignes et n colonnes à coefficients dans $\mathbb{F}_q$ (en étendant chaque coefficient dans la base), et on définit le poids d'un vecteur comme le rang de cette matrice. Cette métrique fournit des cryptosystèmes avec des très bons paramètres, mais ses fondations sont encore assez mal comprises. Dès lors, la sécurité des cryptosystèmes est surtout assurée via la cryptanalyse, c'est-à-dire en développant des attaques. En particulier, dans le chapitre 3, je donne une attaque extrêmement efficace contre deux cryptosystèmes en métrique rang (LIGA [RPW21], et RAMESSES [LLP20]), accompagnée d'une implémentation en SageMath [Ste+23], démontrant son utilisation pratique. Cette attaque repose sur une variante dite « à droite » de l'algorithme de décodage de codes de Gabidulin, qui sont les analogues en métrique rang des codes de Reed-Solomon. Cet algorithme, est présenté dans le chapitre 2.

Publications associées.

[**BC22**] Maxime Bombar et Alain Couvreur. "Right–hand side decoding of Gabidulin codes and applications". In : *WCC 2022 - Workshop on Coding Theory and Cryptography*. Rostock, Germany, mars 2022.

[**BC21**] Maxime Bombar et Alain Couvreur. "Decoding Supercodes of Gabidulin Codes and Applications to Cryptanalysis". In : *Post-Quantum Cryptography - 12th International Conference*. Sous la dir. de Jung Hee Cheon et Jean-Pierre Tillich. T. 12841. LNCS. Daejeon, South Korea : Springer, juill. 2021, p. 3-22.

Partie II : Fondements de la cryptographie à base de codes structurés

Cette partie formée des chapitres 4 et 5 représente le cœur de cette thèse, en particulier le chapitre 4. En partant de l'observation que la cryptographie à base de réseaux est assez proche de celle à base de codes, il peut sembler surprenant que l'herbe semble plus verte, du point de vue des réductions (de recherche à décision) dans le monde des réseaux structurés.

Il se trouve que dans le cas des réseaux structurés, c'est une très forte connexion avec la théorie algébrique des nombres qui a offert un cadre formidable pour faire des preuves ([SSTX09 ; LPR10 ; LS15 ; PRS17 ; RSW18 ; BJRW20a ; PS21b] pour ne citer que quelques références). Plus précisément, que ce soit dans le cas des codes ou des réseaux, la version structurée du problème de décodage admet une reformulation en terme de polynômes. En effet, en considérant un vecteur de longueur n comme les coefficients d'un polynôme de degré au plus $n-1$, et en représentant une matrice circulante (c'est-à-dire dont toutes les lignes sont des décalages cycliques de la première) par le polynôme représentant sa première ligne, il est très facile de voir que le produit d'une matrice circulante par un vecteur s'identifie naturellement au produit des polynômes correspondants, le tout pris modulo $X^n - 1$. En d'autres termes, on peut reformuler le problème de décodage en travaillant dans l'anneau $\mathbb{F}_q[X]/(X^n-1)$. Notons que dans le cas de la cryptographie à base de réseaux euclidiens, on considère plutôt des polynômes à coefficients dans le corps premier $\mathbb{F}_p \stackrel{\text{def}}{=} \mathbb{Z}/p\mathbb{Z}$, modulo (X^n+1) où n est une puissance de 2. Or, cet anneau s'identifie naturellement à la réduction modulo p (qu'on appelle aussi le « modulus ») de l'anneau $\mathcal{R} \stackrel{\text{def}}{=} \mathbb{Z}[X]/(X^n+1)$, qui est un anneau extrêmement important en théorie des nombres, qu'on appelle anneau des entiers algébriques d'un corps cyclotomique. Plus précisément, lorsque n est une puissance de 2, le corps $\mathbb{Q}(\zeta_{2n})$ obtenu en ajoutant à $\mathbb{Q}$ la racine complexe primitive $2n$-ième de l'unité $\zeta_{2n} \stackrel{\text{def}}{=} e^{\frac{i\pi}{n}}$ est appelé *corps cyclotomique*, et est égal à l'anneau de polynômes $\mathbb{Q}[X]/(X^n+1)$. L'anneau des entiers algébriques $\mathcal{R}$, est alors le plus gros sous anneau qui joue le même rôle que $\mathbb{Z}$ dans $\mathbb{Q}$; on parle de *clôture intégrale*, et il possède la propriété très importante d'être ce qu'on appelle un *anneau de Dedekind*[viii]. Le problème sous-jacent à la cryptographie à base de réseaux euclidiens structurés se ramène alors à des problèmes algorithmiques dans des idéaux de (ou plus généralement des *modules* sur) $\mathcal{R}$, dont l'algorithmique est bien comprise. Par ailleurs, cette vision permet d'exploiter des symétries qui sont héritées de la structure dite « galoisienne » des corps cyclotomiques, et c'est précisément cette symétrie qui fait fonctionner les preuves.

En d'autres termes, cette vision permet de *prendre de la hauteur*. D'ailleurs, cette prise de hauteur est aussi à prendre dans le sens littéral. En effet, mathématiquement, dans un anneau commutatif, la hauteur d'un idéal premier $\mathfrak{p}$ est définie comme le nombre h (ou ∞ si un tel nombre n'existe pas) tel qu'il existe une chaîne d'idéaux premiers différents, de la forme

$$\mathfrak{p}_0 \subsetneq \mathfrak{p}_1 \subsetneq \cdots \subsetneq \mathfrak{p}_h = \mathfrak{p}.$$

On définit alors la dimension de Krull d'un anneau commutatif comme la hauteur maximale d'un idéal premier. Or, un *anneau de Dedekind* est précisément de dimension 1 (lorsqu'il n'est pas un corps), alors que l'anneau $\mathbb{F}_p[X]/(X^n+1)$ est de dimension 0. Autrement dit, on a littéralement augmenté la hauteur.

Dans le chapitre 4, nous proposons une vision similaire, mais adapté au cas des codes correcteurs d'erreurs. Plus précisément, les corps de nombres ne sont pas les seuls corps intéressant

[viii] Un anneau commutatif intègre est dit de Dedekind lorsque tous ses idéaux peuvent se factoriser de façon unique (à l'ordre près des facteurs) en un produit d'idéaux premiers (c'est-à-dire d'idéaux par lesquels le quotient reste un anneau intègre). Ça généralise la propriété que tout nombre entier relatif peut s'écrire de façon unique comme un produit (signé) de nombres premiers croissants.

ayant les propriétés mentionnées ci-dessus. En effet, ils forment une facette de ce qu'on appelle les corps globaux. L'autre partie étant constituée des *corps de fonctions*, en caractéristique positive, c'est-à-dire des extensions algébriques du corps des fractions rationnelles $\mathbb{F}_q(T)$. Or, il existe une analogie très profonde entre ces deux mondes : le rôle joué par $\mathbb{Z}$ est alors rempli par l'anneau des polynômes $\mathbb{F}_q[T]$, qui sont tous les deux des anneaux euclidiens, dont les quotients par n'importe quel idéal sont finis, et les corps de fonctions ont des propriétés similaires aux corps de nombres. Ce langage permet alors de réinterpréter (et généraliser) le décodage de codes structurés en un nouveau problème que nous avons appelé FF-DP (*Function Field Decoding Problem*). Cependant, il reste encore à instancier ce nouveau problème avec des corps de fonctions ayant les bonnes propriétés. Je propose alors de regarder la théorie des *modules de Carlitz*, fournissant le bon analogue des corps cyclotomiques dans le monde des corps de fonctions ([Hay74]). Ceci permet alors de retrouver la symétrie qui manquait jusque là et de donner la première réduction de recherche à décision pour les codes quasi-cycliques.

Cependant, cette approche a aussi des limitations, qui étaient déjà présentes dans le cas des réseaux euclidiens. Dans le but de résoudre ce problème, nous nous tournons dans le chapitre 5 vers la technique OHCP (*Oracle with Hidden Center Problem*) introduite dans [PRS17], et qui est maintenant la manière moderne de faire des réductions en réseaux. Nous revisitons cette technique dans le monde des codes pour donner la première réduction de recherche à décision de type « pire cas - cas moyen ». Malheureusement, cette réduction échoue dans le cas des codes structurés, même si tout espoir n'est pas perdu.

Publications associées.

[**BCD23**] Maxime Bombar, Alain Couvreur et Thomas Debris-Alazard. "Pseudorandomness of Decoding, Revisited : Adapting OHCP to Code-Based Cryptography". In : *Advances in Cryptology - ASIACRYPT 2023 29th International Conference on the Theory and Application of Cryptology and Information Security*. Sous la dir. de Jian Guo et Ron Steinfeld. LNCS. Guangzhou, China : Springer, déc. 2023.

[**BCD22**] Maxime Bombar, Alain Couvreur et Thomas Debris-Alazard. "On Codes and Learning With Errors over Function Fields". In : *Advances in Cryptology - CRYPTO 2022 - 42nd International Cryptology Conference*. Sous la dir. d'Yevgeniy Dodis et Thomas Shrimpton. T. 13508. LNCS. Santa Barbara, CA, USA : Springer, août 2022.

Partie III : Applications au Calcul Multiparti Sécurisé

La dernière partie de cette thèse est dédiée aux applications récentes des codes (structurés) au calcul multiparti.

Le chapitre 6 donne une courte introduction au calcul multiparti sécurisé, et définit les notions qui seront nécessaires pour la lecture du chapitre suivant.

Enfin, le chapitre 7 renforce l'analyse du Générateur de Pseudo-aléa Corrélé (PCG) construit dans [BCGI+20b], et qui utilise des codes quasi-cycliques. De plus, je propose une généralisation en utilisant des codes plus généraux appelés *codes quasi-abéliens*. C'est d'ailleurs la première fois que de tels codes sont utilisés en cryptographie. Plus précisément, les codes quasi-cycliques sont des codes laissés stables par l'action d'un groupe cyclique, et le problème de leur décodage admet une reformulation en terme de polynômes univariés, la version quasi-abelienne revient à considérer des anneaux de polynômes multivariés de la forme $\mathbb{F}_q[X_1,\dots,X_t]/(X_i^{n_i}-1)$. Plus formellement, les codes quasi-abéliens sont stabilisés par des groupes abéliens plus généraux, et l'anneau de polynômes multivariés est en réalité une algèbre de groupe $\mathbb{F}_q[G]$. On peut alors montrer que cette généralité admet les mêmes propriétés de difficulté que le problème plus classique avec les

codes quasi-cycliques. Par ailleurs, un choix d'instanciation particulier permet d'obtenir un PCG efficace sur n'importe quel corps fini $\mathbb{F}_q$, avec $q \geqslant 3$. Cette construction est d'ailleurs en quelques sortes « complète », puisqu'on peut prouver qu'il n'existe pas d'algèbre de groupe $\mathbb{F}_2[G]$ avec les bonnes propriétés.

Enfin je termine ce chapitre par des pistes de recherche afin de dépasser cette limitation, et ce à l'aide de la théorie des modules de Carlitz.

Publication associée.

[**BCCD23**] Maxime Bombar, Geoffroy Couteau, Alain Couvreur et Clément Ducros. "Correlated Pseudorandomness from the Hardness of Quasi-Abelian Decoding". In : *Advances in Cryptology - CRYPTO 2023 - 43nd International Cryptology Conference*. Sous la dir. d'Helena Handschuh et Anna Lysyanskaya. Santa Barbara, CA, USA : Springer, août 2023.

Contents

List of Tables

List of Figures

Introduction

Remote communications are taking more and more importance in our everyday lives: from accessing our bank accounts and health data, communicating with family, friends and co-workers, to even managing alarm systems and other smart devices at home. As such, increasingly more sensitive data circulate over the Internet. In this context it is extremely important to guarantee that our communications remain private, this is the role of encryption schemes, and it is even more important to ensure that we are talking to the correct party, this is the role of digital signatures and authentication protocols.

Post-Quantum Cryptography

Modern cryptography aims, among other things, to tackle those two questions and offer strong security guarantees. The starting point is probably the work of Shannon on the theory of information and communications [Sha48; Sha49]. In particular, he proved that the concept of *one-time pad*, where a message is masked by a random string *of the same length*, is *unconditionally secure* (or *perfectly secure*) in the sense that it is completely indistinguishable from another message of the same length, hidden by another random mask. Even though this paradigm had been widely used for decades, Shannon was the first to actually *prove* the security. Conversely, he also proved that one-time pads are essentially the only possible constructions to achieve perfect secrecy. However, this approach has obvious limitations:

(i) it is not practical for long messages,

(ii) distributing the mask to all the parties engaging in the protocol remains an open problem.

Fortunately, perfect secrecy is not really needed for cryptographic purposes, and Shannon proposed instead to base the security on computationally hard problems, so that decrypting a ciphertext without the knowledge of some secret would be impossible with reasonable resources (time and/or memory).

In this context, Diffie and Hellman [DH76] imagined a solution to this second limitation almost three decades later, by introducing the first *key distribution protocol*, showing that two parties could agree on a secret by communicating over a public channel. Their solution relies on the hardness of the problem now known as the *Computational Diffie-Hellman* problem (CDH), which can be formulated as follows: given a public cyclic group $G \stackrel{\text{def}}{=} \langle g \rangle$, as well as two uniformly random elements $g^a, g^b \in G$, the goal is to compute g^{ab}. For many choices of group G, this problem is widely believed to be hard in practice,[ix] and the Diffie-Hellman protocol is nowadays a very important component of Internet telecommunications. Furthermore, they also paved the

[ix] Note that it is related to the discrete logarithm problem where given a cyclic group $\langle g \rangle$ and a random element g^s the goal is to recover s.

way to what is now known as *public-key cryptography.*[x] In this setting, the encryption key is different from the decryption key. In particular, it can be made public, while the decryption key should remain secret.[xi] The first public key cryptosystem is due to Rivest, Shamir and Adleman in 1978 [RSA78], and is based on the hardness of integer factorisation: given an integer N which is the product of two primes p and q, the goal is to recover p and q. When p and q are large and carefully chosen to avoid so-called weak primes, this problem is believed to be hard in general, and RSA cryptosystem is still widely used today.

However, a new computing paradigm shook the cryptographic world in the 1990s, following the breakthrough algorithm by Shor [Sho94]. Indeed, the aforementioned two problems on top of which most of the cryptography is built can actually be seen as instantiations of the so-called *Hidden Abelian Subgroup* problem, which can be efficiently solved using *quantum computers* [Sho94; Joz01]. If building one with a large enough computing power is still an open question, it is far less unbelievable than it was only a decade ago due to recent advances. Moreover, some applications need long term secrecy and are vulnerable to a "store-now, decrypt-later" kind of attack. It is even more important to think about future-proof solutions now, that deploying a new technology on a large scale is very complicated and takes much time.[xii] In this context, NIST, the American institute for standards and technology, launched calls in 2017 and 2023 for standardising quantum-safe cryptographic primitives. Currently, the most promising solutions for encryption schemes belong to two categories: lattice-based and code-based cryptosystems.[xiii] In particular, in July 2022 NIST announced that one lattice-based encryption (Kyber [ABDK+21]) and two signatures (Falcon [FHKL+20] and Dilithium [LDKL+20]) will be standardised.[xiv] Furthermore, the next standard that should be announced in a few months will be based on error-correcting codes, which are the main topic of this manuscript: the only candidates remaining in the fourth round of NIST competition are BIKE [AABB+22a], HQC [AABB+22b] and Classic McEliece [ABCC+22]. As we will see all along this manuscript, lattice-based and code-based cryptosystems share many properties, therefore it is not really surprising that either both or none of them would be standardised. It has to be noted that other general families of computationally hard problems have been considered in the literature, namely solving systems of multivariate polynomial equations over a finite field, and finding special maps, known as isogenies, between elliptic curves. However, even though currently no quantum algorithms are known to efficiently solve those two problems, all the cryptosystems proposed to standardisation have actually been broken *classically* [Beu22; CD23; MMPP+23; Rob23]. In particular, those cryptosystems did not benefit from *reductions* to solving those hard problems, and were actually found to be weaker.

Error-based Cryptography

The main idea behind code-based and lattice-based cryptography is that solving a linear system of equations over a finite field is easy, while it becomes completely intractable when we add some non-linear constraint. Their difference basically lies on the type of the constraint. In this manuscript we will mostly focus on cryptography based on error-correcting codes, but we will get inspired by the world of lattices.

[x] With the help of Merkle [Mer78].

[xi] For signature schemes, this is the signing key which should remain secret while the verification key is public.

[xii] We can take the example of the deployment of ipv6 protocol for internet communications, which has been proposed in the 1990s as a replacement of ipv4 and is still currently not deployed everywhere, even though ipv4 addresses have already been exhausted.

[xiii] For signatures, we may also put forth the so-called hash-based paradigm used in Sphincs+ [BDEF+22].

[xiv] the first standard drafts came out in August 2023 for Kyber and Dilithium.

A (linear) error-correcting code $\mathscr{C}$ of dimension k and length n over a finite field $\mathbb{F}_q$ is a k-dimensional subspace of $\mathbb{F}_q^n$. Usually, it is specified by a basis represented as a matrix $\mathbf{G} \in \mathbb{F}_q^{k\times n}$ referred to as a generator matrix, or as the kernel of a matrix $\mathbf{H} \in \mathbb{F}_q^{(n-k)\times n}$ called a parity-check matrix. They were first introduced as a mean to actually remove errors in digital telecommunications, hence their name. More precisely, a message $\mathbf{m}$ to be transmitted is first encoded into a codeword $\mathbf{c} \stackrel{\text{def}}{=} \mathbf{mG} \in \mathscr{C}$, and the receiver gets a noisy vector $\mathbf{y} \stackrel{\text{def}}{=} \mathbf{c}+\mathbf{e}$ where $\mathbf{e}$ is hopefully *small enough* to be removed, where the size is quantified by some metric on $\mathbb{F}_q^n$ (*e.g.* Euclidean, Hamming, Lee, or rank metrics). However, it was noticed as early as in the work of Shannon [Sha48; Sha49] that in general this problem, known as *Decoding*, is very hard, both in the worst-case where it has been shown to be NP-complete [BMT78] (when the error is characterised by the Hamming metric, which will be assumed in the sequel unless otherwise specified), but also on average: after more than 70 years of research, the best decoding algorithms for generic codes (*i.e.* when the generator or parity-check matrix is picked uniformly at random) have a complexity *exponential* in the weight of the error (*i.e.* the distance between the received message and the actual codeword). It is striking to notice that, even today, no quantum algorithm is known to significantly improve on classical decoding algorithms. Nevertheless, since the main objective of error-correcting codes is to actually perform this decoding, much research in coding theory has been dedicated to building codes with an additional structure allowing efficient decoding algorithms. They basically fit into two classes: codes arising from the evaluation of algebraic functions such as Reed-Solomon [RS60] and Goppa codes [Gop70], and codes endowed with unusually sparse parity-check matrices such as LDPC [Gal63] and MDPC [MTSB13].

In this context, McEliece proposed in 1978 to use such codes to build a public-key encryption scheme. The idea is the following: starting from a code $\mathscr{C}$ endowed with an efficient decoding algorithm $\mathcal{D}$, the public-key consists of a random basis of $\mathscr{C}$ while the secret key is $\mathcal{D}$. Now, encrypting a message $\mathbf{m}$ consists in encoding it into $\mathscr{C}(\mathbf{m}) \in \mathscr{C}$ using this public random basis, before introducing a random error $\mathbf{e}$: a ciphertext is therefore of the form $\mathscr{C}(\mathbf{m})+\mathbf{e}$; and the decryption process simply consists in applying the decoding algorithm $\mathcal{D}$ to recover $\mathbf{m}$. The hope is that the public basis should be random enough so that recovering the secret decoding algorithm $\mathcal{D}$ be intractable; and the only approach to break it is to use generic decoding algorithms for random codes. In particular, the choice of the underlying code $\mathscr{C}$ is really important. In his original proposal [McE78], McEliece suggested to use the aforementioned Goppa codes, and although the parameters had to be updated to take into account the tremendous amount of research in generic decoding algorithms, it has to be noted that the general approach of McEliece with Goppa codes has still not been broken, even with the help of quantum computers, which makes it the oldest public-key encryption scheme still unbroken (either classically or quantumly). In particular, this is the approach at the core of NIST submission Classic McEliece [ABCC+22].

However, this cryptosystem suffers from a major drawback which explains why it did not receive the attention it deserved back then: the public keys are large random matrices, and therefore need a lot of storage. More precisely, for Goppa codes which allow to decode at a large distance $\theta(n/\log(n))$, and therefore inducing attacks with complexity of the form $O\left(2^{c\cdot n/\log(n)}\right)$, the size of the public key is quadratic in the security level λ.[xv] For a reasonable security, this induces keys of several hundreds of kB, or even several MB,[xvi] which did not even fit in RAM when it was proposed. For MDPC codes, the situation is even worse, since the decoders only allow to decode up to $\theta(\sqrt{n})$ errors, which induces public keys of the prohibitive size $\Omega\left(\lambda^4\right)$.

In order to cope with this limitation, it was proposed to introduce more structure on the

[xv] A cryptosystem is said to achieve λ bits of security if the best attacks need $\Omega(2^\lambda)$ operations to succeed.

[xvi] For example, NIST submission Classic McEliece has a recommended parameter set inducing a public key of roughly 1.3 MB. See Table 1.1.

underlying code. For example, using codes being able to decode even further than Goppa codes, such as Reed-Solomon codes. Unfortunately, every other choice have led to devastating attacks (*e.g.* [SS92; CGGO+13]). Another possibility is to introduce another type of structure, which does not help for decoding, but allows to somehow compress the public key: using some codes endowed with a large group of automorphisms, it is possible to publish only one row of the public basis to deduce it completely. In particular, using MDPC codes endowed with a so-called *quasi-cyclic* structure, where the generator matrix (or parity-check matrix) is formed by multiple random circulant matrices, allows to give public keys of only several kB [Gab05]. Additionally, this new algebraic structure allows to give more efficient encoding algorithms, yielding a more efficient encryption process. This led to BIKE proposal [AABB+22a] in the NIST competition.

Nevertheless, the question of assessing the actual security of those cryptosystems remains open: this is the role of *security reductions*, and *cryptanalysis*. In the McEliece situation, especially when instantiated with Goppa codes, cryptanalysis, or more precisely the lack thereof after nearly 50 years of research, gives a good confidence in the scheme. However, it has to be noted that the security of this cryptosystem not only relies on the hardness of decoding random linear codes, but also on the assumed hardness of distinguishing the public code from an actual random code, which could then be turned into an attack. In particular, most of the instantiations of McEliece have been broken using this path. If in the case of Goppa codes we are still very far away from actually achieving this goal, recent progress has been made in that direction [MT22; BMT23; CMT23], and more research needs to be done.

Still, code-based cryptography should not be limited to cryptosystems in the spirit of McEliece. Indeed, an important line of work initiated by Alekhnovich [Ale03] proposed to build cryptosystems without a decoding trapdoor. More precisely, starting from a code $\mathscr{C} \subset \mathbb{F}_2^n$ specified by a uniformly random generator matrix $\mathbf{G}$ which forms the public key, encrypting one bit $\beta \in \{0,1\}$ consists in sending either a uniformly random vector $\mathbf{u} \in \mathbb{F}_2^n$; or a noisy codeword $\mathbf{mG}+\mathbf{e}$ where $\mathbf{m}$ is a random element of $\mathbb{F}_2^k$. The decryption process now consists in *distinguishing* between those two situations given the sole knowledge of the public matrix $\mathbf{G}$, *i.e.* in other words solving a *decisional* version of the Decoding Problem. A similar approach has been used by Regev [Reg05] to define the Learning With Errors (LWE) cryptosystem, and the eponym problem, which are at the core of lattice-based cryptography. More precisely, in LWE[xvii] the adversary is given $(\mathbf{A},\mathbf{y})$ where $\mathbf{A} \in \left(\mathbb{Z}/p\mathbb{Z}\right)^{k\times n}$ is uniformly random, with p a prime number, and some vector $\mathbf{y} \in \left(\mathbb{Z}/p\mathbb{Z}\right)^{n}$; and is asked to determine whether $\mathbf{y}$ is uniformly random, or is of the form $\mathbf{sA}+\mathbf{e}$ where $\mathbf{s} \in \left(\mathbb{Z}/p\mathbb{Z}\right)^{k}$ is a secret and $\mathbf{e}$ has entries chosen independently from some discretised normal distribution. Regev proved that breaking LWE is harder than computational problems based on Euclidean lattices. This has led to a very rich and fruitful line of work in the last two decades.

Our brief description of the Alekhnovich cryptosystem is not complete. In particular, we did not specify how this secret distinguisher is actually built. This will be more precisely described in Section 1.2.4. This variant of the Decoding Problem should be reminiscent of the *decisional* version of Diffie-Hellman problem (usually referred to as DDH) used in some encryption schemes such as El Gamal, where instead of computing the element g^{ab}, the adversary is given an additional element g^c and they must tell whether it actually is g^{ab} or a uniformly random element of the group. Of course, solving the computational problem yields a solution to the decisional version, which proves that the former is harder, but there is no information in the other way around. And indeed it is known that for some groups DDH is actually *strictly easier* than CDH [JN03]. However, the situation is completely different in the code-based setting, where a

[xvii] with n samples.

search-to-decision reduction for the Decoding Problem was given by Fischer and Stern [FS96], proving that both the decisional and computational versions are equivalent in the sense that an algorithm solving one problem can be turned in polynomial time into an algorithm solving the other one. In particular, Alekhnovich-like encryption schemes are *provably secure* under the well-established assumption that decoding a random linear code is hard. However, as for McEliece-like cryptosystems, encryption schemes relying on this Decisional Decoding Problem have huge public keys. For them to be practical, it is therefore necessary to, once again, introduce structured codes. Since in this situation we do not care about decoding algorithms, it is natural to make use of random quasi-cyclic codes. This approach has led to the HQC submission [AABB+22b] to NIST competition.

Impact on the security. Nevertheless, using quasi-cyclic codes might hinder the security. If from the cryptanalytic side, no algorithm is known to exploit it in order to significantly improve on the generic decoders, those new cyptosystems now come without security proofs. More precisely, the reduction from Fischer and Stern does not extend to structured variants of the Decoding Problem.

The research described in this manuscript is motivated by this state of affairs. Indeed, LWE-based cryptosystems suffer from similar drawbacks as Alekhnovich-like cryptosystems, and in the context of lattice-based cryptography as well, many algebraically structured variants such as Polynomial-LWE [SSTX09], Ring-LWE [LPR10] and Module-LWE [LS15] have been considered in the literature. The main difference being that they also come with their search-to-decision reductions, making use of algebraic number theory, and various tools such as the *Oracle with Hidden Center Problem* (OHCP) [PRS17] which is now considered to be the modern tool for deriving reductions in the lattice-based setting. In particular, in Part II we get inspired by this tremendous literature regarding structured lattice-based cryptography to derive a search-to-decision reduction for structured codes, making use of the famous number field-function field analogy well known in algebraic number theory. This is the main topic of Chapter 4. We also adapt the OHCP-based approach to the code-based setting and single out some theoretical obstacles. This is the main topic of Chapter 5.

Conversely, when security reductions do not exist, it is tempting to try and break those cryptosystems. This is the role of cryptanalysis. In particular, in Chapter 3 we propose attacks on some cryptosystems, in the rank metric, which do not have reductions to well-established hard problems. In particular, this shows how important it is to have cryptosystems which do benefit from such reductions.

Advanced Cryptography

There is more in post-quantum cryptography than this supposed quantum security. In fact, encryption and authentication are nowadays not the only objectives of modern cryptography. An important goal which is gaining more and more popularity recently is the ability to compute a function over some data without giving plain access to it. The ultimate goal would be to delegate the computation as much as possible to a remote server, but without revealing any secret information. The applications are endless: be it for computing statistics over health data, storing a sensitive database in the cloud, but still being able to make queries over it... This goal has recently been achieved with the introduction by Gentry of the first *Fully Homomorphic Encryption* (FHE) scheme [Gen09], which allows a third-party to evaluate any function on encrypted data, to get the encryption of the result. It turns out that Gentry's construction involves structured lattice-based cryptography, but it was not really practical. Since then, the popularity of

such advanced primitives has increased every year, using more and more algebraically structured lattices to improve the efficiency.

Another important objective is the so-called Multiparty Computation (MPC) where several clients which do not trust each other want to jointly compute a function on their private inputs. For example, when only a small part of our data needs to remain private, we might want to only keep these secret data, while giving away the rest to a remote server. An alternative application could be to allow different parties to cooperate on a common goal, without revealing their initial information. In general, the main bottleneck in secure MPC protocols is the tremendous cost of the communication, and in particular using some FHE might not be the best approach. It turns out that there is a way to overcome this issue by pushing most of the communication in a preprocessing phase, *before* involving the actual private data. In this computation model, the parties can benefit from sharing long list of random elements having a useful *correlation*, such as so-called *Oblivious Transfers* or their generalisation to larger fields: *Oblivious Linear Evaluations*. However, we merely changed the issue, reducing the problem to generating these long lists of correlated random elements, whose cost might still be prohibitive.

Nevertheless, a recent paradigm put forth by Boyle *et al* [BCGI18; BCGI+19] changed this state of affairs with the introduction of a new tool called *Pseudorandom Correlation Generators* (PCG), allowing to efficiently generate those correlated random strings with only a small amount of communication, followed by local computations by each of the parties. In other words, after the initial small communication, the parties remain *silent*. It turns out that those constructions rely on error-correcting codes, and more precisely on the hardness of the decisional version of the Decoding Problem. However, this tool was initially tailored to specifically target the 2-party computation, where only two parties are involved. In order to generalise the construction to more parties, an additional property called the *programmability* is needed, and the state-of-the-art construction which can be found in [BCGI+20b] makes use of quasi-cyclic codes. Nevertheless, this construction has two limitations: first, it necessitates to work over large fields, and in particular only allows secure computation of data which can be encoded into those fields; and second the security was not fully understood.

With the work presented in Chapter 4, we are able to give a stronger theoretical foundation to this construction. Moreover, in Chapter 7 we extend the construction by introducing more general algebraically structured codes, namely *quasi-abelian codes*, to provide a construction of a programmable PCG for the so-called OLE correlation (*Oblivious Linear Evaluation*) which works over any finite field $\mathbb{F}_q$ with $q > 3$. Chapter 6, for its part, is dedicated to a more detailed presentation of secure multiparty computation, in particular in the aforementioned preprocessing model.

Contributions of this thesis

Related Publications

The material presented in this manuscript is mostly based on the following publications, ordered from the most recent to older ones.

[**BCD23**] Maxime Bombar, Alain Couvreur, and Thomas Debris-Alazard. "Pseudorandomness of Decoding, Revisited: Adapting OHCP to Code-Based Cryptography". In: *Advances in Cryptology - ASIACRYPT 2023 29th International Conference on the Theory and Application of Cryptology and Information Security*. Ed. by Jian Guo and Ron Steinfeld. LNCS. Guangzhou, China: Springer, Dec. 2023.

[BCCD23] Maxime Bombar, Geoffroy Couteau, Alain Couvreur, and Clément Ducros. "Correlated Pseudorandomness from the Hardness of Quasi-Abelian Decoding". In: *Advances in Cryptology - CRYPTO 2023 - 43nd International Cryptology Conference*. Ed. by Helena Handschuh and Anna Lysyanskaya. Santa Barbara, CA, USA: Springer, Aug. 2023.

[BCD22] Maxime Bombar, Alain Couvreur, and Thomas Debris-Alazard. "On Codes and Learning With Errors over Function Fields". In: *Advances in Cryptology - CRYPTO 2022 - 42nd International Cryptology Conference*. Ed. by Yevgeniy Dodis and Thomas Shrimpton. Vol. 13508. LNCS. Santa Barbara, CA, USA: Springer, Aug. 2022.

[BC22] Maxime Bombar and Alain Couvreur. "Right–hand side decoding of Gabidulin codes and applications". In: *WCC 2022 - Workshop on Coding Theory and Cryptography*. Rostock, Germany, Mar. 2022.

[BC21] Maxime Bombar and Alain Couvreur. "Decoding Supercodes of Gabidulin Codes and Applications to Cryptanalysis". In: *Post-Quantum Cryptography - 12th International Conference*. Ed. by Jung Hee Cheon and Jean-Pierre Tillich. Vol. 12841. LNCS. Daejeon, South Korea: Springer, July 2021, pp. 3–22.

Organisation of the manuscript

This thesis is organised into three general parts. We give a more detailed introduction to the related topics in the beginning of each chapter.

Part I: $\mathbb{F}_{q^m}$-linear codes endowed with the rank metric. [BC21; BC22] The major part of code-based cryptography, especially considering structured codes, focuses on codes endowed with the Hamming metric, and this work is no exception. Yet, other algebraically structured codes have been considered in the literature. In this part consisting in Chapters 2 and 3, we introduce some codes whose algebraic structure comes from an extension field $\mathbb{F}_{q^m}/\mathbb{F}_q$. Due to the cyclicity of this extension, they are quite similar in nature to quasi-cyclic codes considered in "traditional" code-based cryptography, but have also their specificities since the metric structure on the ambient space is different: those codes are endowed with the so-called *rank metric*. Little is known regarding the theoretical foundations of rank-metric-based cryptography, and the security of those schemes is mainly assessed through cryptanalysis regarding known attacks. In particular, in Chapter 3 we give a full practical message recovery attack against two encryption schemes based on the rank metric, which were proposed recently, namely RAMESSES [LLP20] and LIGA [RPW21]. This part is mostly independent from the rest of the manuscript, and only minor references to the rank-metric are made in subsequent chapters.

Part II: Foundations. [BCD22; BCD23] This part consisting in Chapters 4 and 5 forms the main core of this manuscript, especially Chapter 4. We start from the observation that in cryptography based on structured codes, little is known regarding the theoretical foundations, especially regarding the decisional versions of the problems on top of which rests the security of many cryptosystems, while the grass looks greener in the world of structured lattice-based cryptography. It is therefore natural to ask why it is so, and how we can cope with this state of affairs. It turns out that lattice-based cryptography has been using techniques from algebraic number theory in order to give a better perspective on the objects at stake. There is an idiom in French reading "Prendre de la hauteur" which means "getting a new perspective on something" and literally translates to "increasing the height". This perfectly captures the situation happening

in lattice-based cryptography. Indeed, starting from a finite world where all the actual objects lie, namely the vector space $\mathbb{F}_p^n = (\mathbb{Z}/p\mathbb{Z})^n$ for some large prime p ; the security is assessed by working with infinite objects, namely euclidean lattices in $\mathbb{R}^n$, *i.e.* discrete subgroups of $\mathbb{R}^n$, the most simple one being $\mathbb{Z}^n$. Structured variants of LWE are built similarly to the code-based setting by considering the action of some group. The breakthrough idea in the world of lattice-based cryptography is that those structured variants can be seen as arising from modules over the ring of integers $\mathcal{O}_K$ of some finite extension $K/\mathbb{Q}$ of degree n, which can then be embedded into $\mathbb{R}^n$. In general, K is an abelian extension, and more precisely a *cyclotomic extension* of $\mathbb{Q}$. Here the notion of "increasing the height" is two-fold in my mind: on the one hand, we increase the dimension of the extension of $\mathbb{Q}$ instead of simply looking at a direct product; but more importantly all the rings $\mathcal{O}_K$ involved have one feature in common: there exist non-trivial *prime* ideals, and they are all *maximal*. In particular, for such a non-zero prime ideal $\mathfrak{p}$, the chain $0 \subsetneq \mathfrak{p}$ is a non-trivial chain of prime ideals in $\mathcal{O}_K$. In algebraic number theory, the number of non-zero prime ideals in this chain is called its *height*. In the above example, it is of height 1. In fact, the rings $\mathcal{O}_K$ are examples of *Dedekind domains* and have so-called *Krull dimension* 1: all their non-trivial chains of prime ideals have height 1. On the other hand, the actual objects used to build the cryptosystems are quotients of those Dedekind domains, and all the chains of prime ideals have height 0. In other words, we literally "increased the height" to gain more insight and make proofs.[xviii]

In Chapter 4, we try to make the same observation in the world of (structured) code-based cryptosystems, and identify good candidates in replacement of those rings $\mathcal{O}_K$. More precisely, we make use of a famous analogy between algebraic extensions of $\mathbb{Q}$ on the one hand, and on the other hand algebraic extensions of the field $\mathbb{F}_q(T)$ of rational functions with coefficients in $\mathbb{F}_q$. The latter are known as *function fields*. Moreover, we replace *cyclotomic extensions* by their function field counterparts, the so-called *Carlitz extensions*. It turns out that this analogy is richer than it looks, and allows to derive a function field analogue of the reductions designed for lattice-based cryptography. This yields the first search-to-decision reductions for structured variants of the Decoding Problem.

However, this approach for designing reductions has some inherent limitations, which already existed in the world of lattice-based cryptography. In order to address this issue, a powerful tool called the *Oracle with Hidden Center Problem* (OHCP) has been introduced in [PRS17], and is now the modern way of designing reductions for Euclidean lattices. In Chapter 5 we revisit this technique and apply it to the code-based setting, and identify some additional difficulties. However, this technique allows to give a new method for deriving search-to-decision reductions in code-based cryptography, and is suitable for worst-case to average-case reductions.

Part III: Secure Multiparty Computation. [BCCD23] Finally, the last part of this manuscript is dedicated to a recent application of structured codes in the world of secure computation.

In Chapter 6, we give a short, but detailed, presentation of secure multiparty computation, especially in the so-called *preprocessing model*, which seems to offer the most competitive MPC protocols to date.

In Chapter 7, we strengthen the analysis of the *Pseudorandom Correlation Generator* (PCG) construction from [BCGI+20b] which used quasi-cyclic codes, and extend it using more general structured codes known as *quasi-abelian* codes. It has to be noted that this is the first time such codes are used in a cryptographic context. This allows to build a PCG for the useful correlation

[xviii] In reality, this presentation rewrites history, since things have always been constructed "from above", starting from the definition of the noise used in LWE which arises from a Gaussian distribution in $\mathbb{R}^n$, discretised so that reduction modulo p make sense. However, this presentation gives the motivation for the work presented in Chapter 4.

known as *Oblivious Linear Evaluation* with the additional *programmable* property, which works over *any* field $\mathbb{F}_q$ with $q > 2$. Finally, in the end of this chapter we discuss the $q = 2$ situation, and propose some research directions towards a complete solution, based again on number theory in function fields.

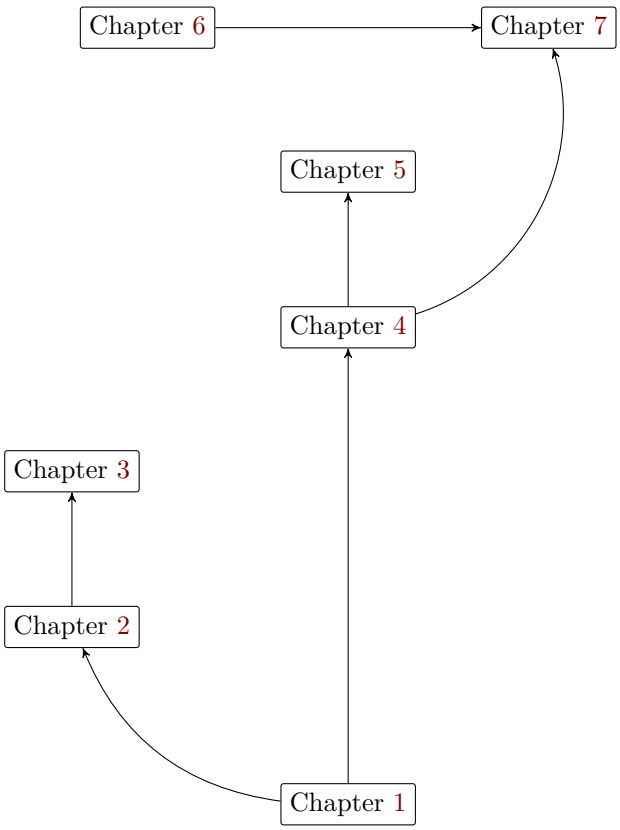


Figure 1: Outline of the thesis

Chapter 1

Structured Codes in Cryptography

In this chapter we give an overview of cryptography based on error-correcting codes, and especially those endowed with an *algebraic structure*. When talking about structured codes, we need to differentiate two notions:

(1) Codes with an additional structure which helps for decoding.

(2) Codes endowed with a large group of automorphisms, which can be represented more compactly.

In general, in this manuscript, when we consider structured codes, it will mostly be the second situation.

Outline of the current chapter

1.1 Decoding: an Intractable Problem for Cryptography

1.1.1 Introduction to Code-Based Cryptography

It is very frustrating when one is on the phone, but the quality of the call is so low that the receiver is unable to distinguish an "N" from an "M", or a "B" from a "D" or a "T". This issue is particularly present when one wants to spell their name or a word whose meaning cannot be deduced from the semantic of the sentence, *e.g.* a flight number or a ham radio call-sign. In this situation, most people would use words acrophonic to the Roman letters, like NATO phonetic alphabet, to spell that word: "A" like *Alpha*, "B" like *Bravo*, "C" like *Charlie*, and so on. This process is all the more efficient that the words do not look like each other.

More precisely, if one wants to transmit a message (say, the ham call-sign F5MMX[i]), one can *encode* it and transmit Foxtrot Five Mike Mike X-ray across the communication channel (say over the radio). This adds *redundancy* to the message: All the meaningful information is carried by the first letter of each word. On the receiver side, the message might be interpreted differently due to the noise in the communication, *e.g.* Fox-trap Hive Bike Mike X-ray. However, the receiver knows that the words Fox-trap, Hive or Bike are not valid encodings, and therefore they can *detect* that some errors happened during the transmission. A first solution might consist in asking that the message be resent, hoping that the noise alters the message differently, but this is not efficient. Instead, here it is enough to notice that the noise did not affect the message too much, and the receiver can directly *correct* the error by looking at the words in NATO alphabet that are the closest to the received ones. The message is here easily corrected and the receiver can deduce that they are indeed talking to F5MMX.

If the noise level is too high, the correction might not be possible and the information could be completely lost. Intuitively, the more redundancy is added in the original message, the more error patterns can be corrected. On the other hand, the more redundancy is added, the more communication is needed to transmit the information, which can be very costly. One of the main goals in coding theory in then to design good encodings optimising this trade-off. This is formalised through the notion of (linear) codes.

A message $\mathbf{m}$ is a sequence of k symbols from some (finite) alphabet Σ. In an information theoretic context, Σ will mostly be a finite field $\mathbb{F}_q$, and therefore $\mathbf{m}$ will be represented by an element $\mathbb{F}_q^k$. The encoding process corresponds to mapping the message into a longer vector of

[i] This is a dedication to my father who is an amateur radio operator, and this is his station call-sign. Without knowing it, he introduced me to error-correcting codes long ago

$\mathbb{F}_q^n$, adding $n-k$ redundancy symbols. This is done by first choosing a linear code $\mathscr{C} \subset \mathbb{F}_q^n$ of dimension k, represented by a generator matrix $\mathbf{G} \in \mathbb{F}_q^{k\times n}$, and then computing $\mathbf{c} \stackrel{\text{def}}{=} \mathbf{mG}$. The codeword $\mathbf{c}$ is sent through the communication channel, which adds some unknown noise $\mathbf{e}$: The receiver gets a word $\mathbf{y} \in \mathbb{F}_q^n$ of the form $\mathbf{c}+\mathbf{e}$, and the goal of the decoder is to remove the noise in order to retrieve $\mathbf{c}$, or equivalently the message $\mathbf{m}$. This concept was first formalised by Shannon in his original paper [Sha48], and is represented on Figure 1.1.

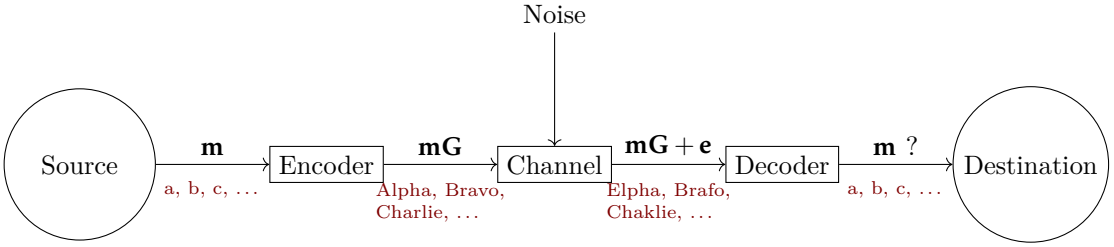


Figure 1.1: Shannon's communication channel

In general, the action of the channel on the original codeword $\mathbf{c}$ is characterised by the *Hamming weight* $|\mathbf{e}|$ of the error, *i.e.* the number of non-zero coordinates, and the goal of the decoder is usually to output a codeword $\mathbf{c} \in \mathscr{C}$ which is close (for the Hamming metric) to the received word $\mathbf{y} \stackrel{\text{def}}{=} \mathbf{c}+\mathbf{e}$, or even the closest when it is possible.

Remark 1.1. *Depending on the situation, other measures of the action of the channel might be considered. For example, the so-called* rank metric *is particularly useful for correcting errors in network communications [SK11]. In Part I we introduce it in a cryptographic context.*

Obviously, it is unthinkable to enumerate all the q^k codewords in $\mathscr{C}$ in order to find the closest one, but even after more than 70 years of research, decoding a code in general still seems out of reach,[ii] which makes it a very attractive problem for cryptographic purposes. Therefore, one of the main challenges in coding theory is to provide specific codes equipped with efficient decoding algorithms. There exist essentially two such families of codes:

(i) Codes derived from the evaluation of polynomials on a finite field such as Reed-Solomon codes [RS60],[iii] as well as their subfield subcodes, *i.e.* their restriction to a subfield, such as alternant and Goppa codes [MS86, Chapter 12].

(ii) Codes whose dual contain unusually sparse codewords, such as LDPC codes (Low Density Parity-Check) [Gal63], or MDPC codes (Moderate Density Parity Check) [MTSB13].

And what about cryptography ? In a nutshell, a code-based cryptographer[iv] will be interested in using such codes as a trapdoor. Let us do a simple thought experiment to give the intuition: consider a text and change at random some letters. As long as few characters are modified, the text can still be readable. On the other hand, it quickly becomes unintelligible when the number of mistakes is too high. In other words, when there are too many *noisy* characters, it is very hard to retrieve the original message ; the text might even look like *random*. Nevertheless, if one keeps a *secret* decoding algorithm, they can reverse the process to recover the original message.

[ii] even with the help of a quantum computer
[iii] Or more generally, algebraic-geometry codes [Gop81]
[iv] wanting to design a McEliece-like encryption scheme

1.1.2 Decoding is Hard in the Worst-Case

Consider a code $\mathscr{C}$ and a vector $\mathbf{y} \in \mathbb{F}_q^n$. As briefly recalled above, the problem of decoding $\mathscr{C}$ corresponds to finding a codeword $\mathbf{c} \in \mathscr{C}$ and an error term $\mathbf{e} \in \mathbb{F}_q^n$ such that $\mathbf{y} = \mathbf{c} + \mathbf{e}$. Of course, formulated as it is, this problem is very easy to solve with simple linear algebra. What makes it hard is to consider additional *non-linear* constraints. For instance, one may ask that $\mathbf{c}$ be *the closest* codeword from $\mathbf{y}$, or we can ask that the error satisfies some weight constraint. In the sequel we will consider the latter. More formally, the decoding problem is defined as follows:

Problem 1.2 (Decoding Problem)

Data. A code $\mathscr{C}$ given by a generator matrix $\mathbf{G} \in \mathbb{F}_q^{k \times n}$, a target distance t, and a vector $\mathbf{y} \in \mathbb{F}_q^n$ such that $\mathbf{y} = \mathbf{mG} + \mathbf{e}$ for some $\mathbf{m} \in \mathbb{F}_q^k$ and $|\mathbf{e}| = t$.

Goal. Find $\mathbf{e}$.

Note that from the knowledge of a generator matrix $\mathbf{G}$ of $\mathscr{C}$ it is easy to recover a parity-check matrix $\mathbf{H}$ (in polynomial time in n). Therefore, given a noisy codeword $\mathbf{y} = \mathbf{c} + \mathbf{e}$ with $|\mathbf{e}| \leqslant t$, one can compute its *syndrome* $\mathbf{s}^\top \stackrel{\text{def}}{=} \mathbf{Hy}^\top = \mathbf{He}^\top$. Conversely, from a syndrome $\mathbf{s}^\top = \mathbf{He}^\top$ and a parity-check matrix $\mathbf{H}$ of $\mathscr{C}$, it is easy to find a vector $\mathbf{y}$ such that $\mathbf{Hy}^\top = \mathbf{s}^\top$ by solving a linear system. Now, $\mathbf{H}(\mathbf{y} - \mathbf{e})^\top = 0$ and therefore there exists $\mathbf{c} \in \mathscr{C}$ such that $\mathbf{y} = \mathbf{c} + \mathbf{e}$, and $|\mathbf{e}| \leqslant t$ by hypothesis. In particular, Problem 1.2 can be equivalently reformulated in terms of parity-check matrices and syndromes. This alternative formulation is sometimes referred to as the Syndrome Decoding Problem, but the previous remark shows that both problems are strictly equivalent. Studying one problem or the other one is mostly a matter of taste.

Problem 1.3 ((Syndrome) Decoding Problem)

Data. A code $\mathscr{C}$ given by a parity-check matrix $\mathbf{H} \in \mathbb{F}_q^{k \times n}$, a target distance t, and a vector $\mathbf{s} \in \mathbb{F}_q^{n-k}$ such that $\mathbf{s}^\top = \mathbf{He}^\top$ with $|\mathbf{e}| = t$.

Goal. Find $\mathbf{e}$.

In 1978, Berlekamp, McEliece and van Tilborg [BMT78] proved that the Decoding Problem is NP-complete. However, one has to be careful when taking into consideration the NP-completeness for cryptographic applications. Indeed, this is nothing more than a *worst-case* result: It shows that there exists at least one instance for which we do not expect a deterministic polynomial time algorithm to succeed. However, it is possible that there exist algorithms which may fail in some occasions but still manage to solve most instances. Moreover, even if one can characterise the cases for which the known algorithms fail, forcing a cryptosystem to be built only on those hard instances is usually a bad idea if they are not really *generic*. In particular, this can be exploited to design a new attack, as we will see in Chapter 3, which is one of the contributions of this thesis.

In other words, a cryptographer is more interested in the *average-case* hardness of a problem, where the input is picked *at random* from all the valid instances. For instance, it would be wonderful if an algorithm that can efficiently solve the problem for a large fraction of the valid inputs could be turned into an algorithm solving efficiently the same problem for any instance, *i.e.* in

the *worst-case*. Such a property is called *random self-reducibility*. There is a caveat though: An NP-complete problem cannot be random self-reducible, unless the polynomial hierarchy collapses [BT06]. This is another argument in favour of having a finer grain quantification of the hardness of cryptographic problems. In fact, in general, the hard problems used in cryptography are considered in a regime where they are not even believed to be NP-hard. In this situation, reducing a worst-case problem to a problem hard on average is conceivable. This topic will be further explored in Chapter 5.

Instead, in order to assess the hardness of a putative hard problem, a cryptographer relies on actual algorithms, and on the test of time. The older and the more studied a problem is, the more confidence we can have in its hardness. This is in particular the case for the Decoding Problem, which has interested many people since the middle of the 20th century: From engineers who wanted to improve telecommunications, to cryptographers in a post-quantum era.

In the next two sections we will be interested in the hardness of the decoding problem *on average*, *i.e.* when the input is a *random code*. First, we shall begin to precisely state what we mean by *random code*.

1.1.3 Random Linear Codes

1.1.3.1 Statistical Distance

In this manuscript we will often need to estimate how close are two probability distributions (or two random variables). This can be quantified through the notion of *statistical distance*,[v] which we recall here, along with the main results. They can be found in [MG02; Sho09] for example.

Let X and Y be two discrete random variables defined over the same finite set $\mathcal{E}$. Their statistical distance $\Delta(X,Y)$ is defined as

$$\Delta(X,Y) \stackrel{\text{def}}{=} \frac{1}{2}\sum_{a\in\mathcal{E}} \left|\mathbb{P}(X=a) - \mathbb{P}(Y=a)\right|. \tag{1.1}$$

Since $\Delta(X,Y)$ only depends on the individual distributions of X and Y and not their joint distribution, we may define the statistical distance between two probability distributions $\mathscr{D}_0$ and $\mathscr{D}_1$ as the statistical distance between two independent random variables X_0 and X_1, such that $X_i \leftarrow \mathscr{D}_i$.

Equivalently, it holds that

$$\Delta(X,Y) = \max_{\mathcal{F}\subset\mathcal{E}} \left|\mathbb{P}(X\in\mathcal{F}) - \mathbb{P}(Y\in\mathcal{F})\right|.$$

Therefore, computing probabilities over X or over Y will differ by at most an additive term $\Delta(X,Y)$. A consequence of this equivalent definition is the so-called *data processing inequality*.

Proposition 1.4 (Data Processing Inequality, [MG02, Proposition 8.10])

Let X,Y be two random variables defined over a common set $\mathcal{E}$. If f is a possibly randomised function with domain $\mathcal{E}$ and whose internal randomness is independent from X and Y, then

$$\Delta(f(X),f(Y)) \leqslant \Delta(X,Y). \tag{1.2}$$

[v] The statistical distance is also known as the *total variation distance*

In particular, it implies that if $\mathcal{D}_0$ and $\mathcal{D}_1$ are two probability distributions over a common set $\mathcal{E}$, and if $\mathscr{A}$ is a probabilistic algorithm taking inputs from $\mathcal{E}$, then its "success" probability on inputs from $\mathcal{D}_0$ will differ from its "success" probability on inputs from $\mathcal{D}_1$ by at most $\Delta(X,Y)$.

Finally, when $(X_1,\dots,X_n)$ and $(Y_1,\dots,Y_n)$ are two sequences of random variables such that the X_i's (respectively the Y_i's) are pairwise independent, then

$$\Delta\big((X_1,\dots,X_r),(Y_1,\dots,Y_r)\big) \leqslant \sum_{i=1}^{r} \Delta(X_i,Y_i). \tag{1.3}$$

Remark 1.5. *This inequality is not tight. For example, if X_1, X_2 (respectively Y_1, Y_2) are two independent copies of a random variable X (respectively Y) such that $\Delta(X,Y) \leqslant \frac{1}{2}$, then Equation (1.3) yields*

$$\Delta\big((X_1,X_2),(Y_1,Y_2)\big) \leqslant 1.$$

However, the equality is reached if and only if the support[vi] *of (X_1,X_2) is disjoint from the support of (Y_1,Y_2); which would imply that the supports of X and Y are also disjoint. In particular $\Delta(X,Y) = 1$, which is a contradiction.*

In fact, we can prove the following better bound ([Kon12, Lemma 2.2])

$$\Delta((X_1,\dots,X_r),(Y_1,\dots,Y_r)) \leqslant 1 - \prod_{i=1}^{r}\Big(1-\Delta(X_i,Y_i)\Big).$$

In particular, when the X_i's (respectively the Y_i's) are independent copies of some random variable X (respectively Y), then

$$\Delta\big((X_1,X_2),(Y_1,Y_2)\big) \leqslant 1 - \Big(1-\Delta(X,Y)\Big)^r.$$

1.1.3.2 A Probabilistic Model for Random Codes

The most natural definition for a random $[n,k]_q$-code is a uniformly random element of the set of linear codes of length n and dimension k over $\mathbb{F}_q$. It is readily seen that this definition is equivalent to sampling a uniformly random matrix $\mathbf{G} \in \mathbb{F}_q^{k\times n}$ of rank k and defining $\mathscr{C}$ to be the code generated by $\mathbf{G}$. Alternatively, this coincides with the definition of sampling a uniformly random full-rank parity-check matrix $\mathbf{H} \in \mathbb{F}_q^{(n-k)\times n}$. However, this condition on the rank is often tricky to work with. Instead, we will relax this constraint and in the entirety of this manuscript a random $[n,k]_q$ code $\mathscr{C}$ will be

- either the code generated by a uniformly random matrix $\mathbf{G} \leftarrow \mathbb{F}_q^{k\times n}$:

$$\mathscr{C} \stackrel{\text{def}}{=} \{\mathbf{mG} \mid \mathbf{m} \in \mathbb{F}_q^k\}.$$

- or a code having a uniformly random parity-check matrix $\mathbf{H} \leftarrow \mathbb{F}_q^{(n-k)\times n}$:

$$\mathscr{C} = \{\mathbf{x} \in \mathbb{F}_q^n \mid \mathbf{Hx}^\top = 0\}.$$

While not being strictly equivalent since a random $k \times n$ matrix always generates a code of dimension $\leqslant k$, while a random parity-check matrix will induce a code of dimension $\geqslant k$, it

[vi] The support of a random variable X with values in a space $\mathcal{E}$ is simply $\{a \in \mathcal{E} \mid \mathbb{P}(X=a) > 0\}$

turns out that both models are exponentially close for the statistical distance, and are therefore equivalent from an algorithmic point of view. Indeed, given some event A, if we denote by $\mathbb{P}_{\mathbf{G}}(A)$ the probability of A with respect to a random generator matrix, and by $\mathbb{P}_{\mathbf{H}}(A)$ the same probability, but with respect to a random parity check matrix, one can prove ([Deb23, Lemma 2.2.2]) that

$$\left|\mathbb{P}_{\mathbf{G}}(A) - \mathbb{P}_{\mathbf{H}}(A)\right| = O\left(q^{-\min(k,n-k)}\right). \tag{1.4}$$

In particular, choosing one model or the other will be mostly a matter of taste and convenience.

1.1.3.3 Minimum Distance of Random Codes: the Gilbert-Varshamov Distance

The minimum distance of a code $\mathscr{C}$ plays an important role in coding theory. As we will see all along this document, this parameter and/or the minimum distance of the dual $\mathscr{C}^{\perp}$, also called the dual distance, give a lot of information on the hardness of code-based assumptions used in cryptography. It turns out that random codes $\mathscr{C}$ are *good codes*, *i.e.* they have good rate and minimum distance. This is formalised through the so-called *Gilbert-Varshamov* distance $d_{GV}(q,n,R)$.

Definition 1.6 (Gilbert-Varshamov Distance)

Let $k \stackrel{\text{def}}{=} \lfloor Rn \rfloor$ and q be integers. The Gilbert-Varshamov distance $d_{GV} \stackrel{\text{def}}{=} d_{GV}(q;n,R)$ is the largest integer such that

$$\sum_{\ell=0}^{d_{GV}-1} \binom{n}{\ell} (q-1)^{\ell} \leqslant q^{n-k}.$$

Note that the left-hand side of the inequality is exactly the volume of the Hamming ball of radius $d_{GV}-1$. Since there are exponentially many elements in this ball, its volume is well approximated by the cardinality of its sphere. In particular, they have the same asymptotic behaviour as $n \to \infty$, which can be expressed by way of the so-called q-ary entropy function h_q defined over $[0, 1-1/q]$ by

$$\begin{cases} h_q(x) = -x \log_q\left(\dfrac{x}{q-1}\right) - (1-x)\log_q(1-x) & \text{if } x > 0, \\ h_q(0) = 0 \end{cases} \tag{1.5}$$

and whose graphical representation is given on Figure 1.2.

This is a strictly increasing function, and as such h_q has an inverse over $[0, 1-1/q]$ often denoted by g_q in the literature, or simply by h_q^{-1}. The following lemma is a consequence of the Stirling identity.

Lemma 1.7

Let $\alpha, \beta > 0$ and $q \in \mathbb{N}^{\star}$. Then,

$$\binom{\alpha n}{\beta n} (q-1)^{\beta n} \sim \sqrt{\frac{\alpha}{2\pi n \beta(\alpha-\beta)}} q^{n\alpha h_q(\beta/\alpha)} \text{ as } n \to \infty.$$

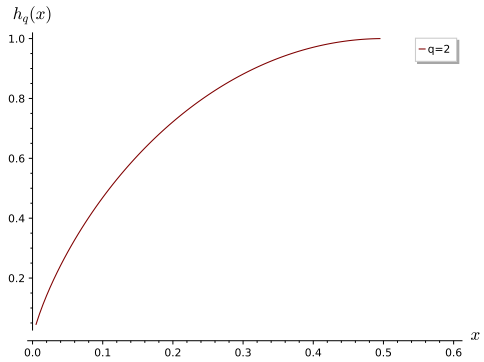


Figure 1.2: Graphical representation of h_q on $[0,1/2]$

With this lemma in hand and some computation, it can be shown that

$$\frac{1}{n} d_{GV}(q;n,R) \to h_q^{-1}(1-R), \tag{1.6}$$

where R is supposed to be constant, which is usually the case in code-based cryptography, even though other regimes also have cryptographic interest. $\delta_{GV} \stackrel{\text{def}}{=} h_q^{-1}(1-R)$ is known as the *relative Gilbert-Varshamov* bound (or distance). For example, when $q=2$ and $R=1/2$ we have $\delta_{GV} \approx 0.11$. Figure 1.3 represents δ_{GV} as a function of the code rate R for $q \in \{2,3\}$. Usually, in code-based cryptography we work with $q=2$, but we will consider the case $q=3$ in Chapter 7 for applications in secure multiparty computation.

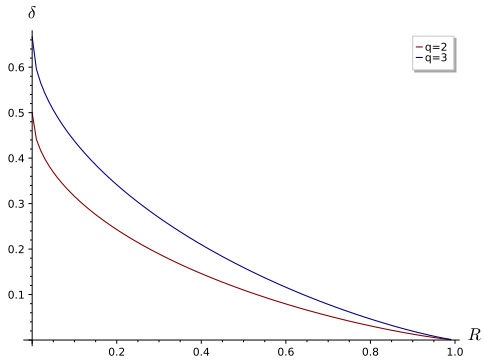


Figure 1.3: Graphical representation of $\delta_{GV}(R)$ for $q \in \{2,3\}$.

One of the most important properties of random codes is that their minimum distance reaches this bound with overwhelming probability ([Pie67; BF02; Deb23]).

Theorem 1.8 ([Deb23, Proposition 2.4.1])

Let $\varepsilon > 0$, and let $\mathscr{C}$ be a random $[n,k]_q$ code with $k = \lfloor Rn \rfloor$ for some $R \in (0,1)$. Denote by $d_{min}(\mathscr{C})$ its minimum distance. Then,

$$\mathbb{P}\left((1-\varepsilon)\delta_{GV} < \frac{d_{min}(\mathscr{C})}{n} < (1+\varepsilon)\delta_{GV} \right) \geqslant 1 - q^{-\alpha n(1+o(1))},$$

where the probability is taken over the choice of $\mathscr{C}$ and $\alpha > 0$ is an explicit constant depending on ε and R.

Remark 1.9. *Recall that in the end of the day, we want to work with more structured codes. It turns out that random structured codes will also reach this Gilbert Varshamov bound with high probability (see Section 1.3.4.3).*

1.1.4 The Average-Case Decoding Problem.

We are now ready to define the problem on which most of code-based cryptography relies: the Decoding Problem. It is parameterised by the code rate $R \stackrel{\text{def}}{=} k/n$ and the noise rate $\omega \stackrel{\text{def}}{=} w/n$.

Problem 1.10 (Decoding Problem DP$(q;n,R,\omega)$)

Let $k \stackrel{\text{def}}{=} \lfloor Rn \rfloor$ and $w \stackrel{\text{def}}{=} \lfloor \omega n \rfloor$ where R and ω are both functions of n (possibly constant).

Data. A *random* $[n,k]_q$-code $\mathscr{C}$, and a vector $\mathbf{y} \in \mathbb{F}_q^n$ such that $\mathbf{y} = \mathbf{c} + \mathbf{t}$ for $\mathbf{c} \leftarrow \mathscr{C}$ and a random $\mathbf{t} \in \mathbb{F}_q^n$ with $|\mathbf{t}| = w$.

Goal. Find $\mathbf{e} \in \mathbb{F}_q^n$ of Hamming weight w such that $\mathbf{y} - \mathbf{e} \in \mathscr{C}$.

An algorithm solving DP$(q;n,R,\omega)$ is called a *decoding algorithm*. More precisely, it is a probabilistic algorithm $\mathscr{A}$ which takes as input the description of a code $\mathscr{C}$, say via a parity-check matrix $\mathbf{H}$, as well as a syndrome $\mathbf{s}^\top \stackrel{\text{def}}{=} \mathbf{H}\mathbf{t}^\top$, and outputs some vector $\mathscr{A}(\mathbf{H},\mathbf{s})$. Its efficiency is characterised by its running time T, as well as its success probability

$$\varepsilon \stackrel{\text{def}}{=} \mathbb{P}\left(\mathscr{A}(\mathbf{H},\mathbf{s}) = \mathbf{e} \quad \text{s.t.} \quad |\mathbf{e}| = \lfloor \omega n \rfloor \text{ and } \mathbf{H}\mathbf{e}^\top = \mathbf{s}^\top \right),$$

where the probability is computed over the internal randomness of $\mathscr{A}$, and the choice of $\mathbf{H}$ and $\mathbf{s}$. With these notations, we say that $\mathscr{A}$ solves DP$(q;n,R,\omega)$ in *average time* T/ε.

Remark 1.11. *We did not specify the model of random code. Indeed, as mentioned in Section 1.1.3, whether we consider a random generator matrix or a random parity-check matrix is mostly a matter of taste. In particular, Equation (1.4) implies that an algorithm solving DP in time T with probability ε over the uniform choice of a generator matrix yields an algorithm solving DP in time $O(T+n^3)$ and with probability $\geqslant \varepsilon - O\left(q^{-\min(k,n-k)}\right)$ over the uniform choice of a parity-check matrix, where the additional $O(n^3)$ term is the cost of computing a generator matrix of $\mathscr{C}$ from a parity-check matrix.*

Remark 1.12. *In a cryptographic context, it is important to know the expected number of solutions to a cryptographic problem in order to correctly assess the security. It turns out that there is an easy criterion for the Decoding Problem. Let $\mathscr{C}$ be a code of minimum distance d. Then, it is readily seen that DP with input code $\mathscr{C}$ will have a* unique *solution as long as the number of errors is upper bounded by $\frac{d-1}{2}$. In fact, on average we can do even better. Indeed, it can be shown that as long as ω is below $\delta_{GV}(R)$, then $DP(q;n,R,\omega)$ will typically have a unique solution (see [Deb23, Proposition 2.4.2]).*

In this thesis, ω will always be sublinear*, and in particular* way below *the Gilbert-Varshamov bound, therefore we will never bother with the number of solutions.*

1.1.4.1 Average-Case Hardness of the Decoding Problem

Over the past 60 years, there has been a tremendous effort to look for the best algorithms solving Problem 1.10. For the parameters used in cryptography, the most efficient generic decoders belong to the family of Information Set Decoders (ISD) [Ste88; Dum91; BJMM12; MO15; BM17], and are all refinements of an algorithm by Prange [Pra62]. More recently, a different approach known as *Statistical Decoding* [Jab01; Ove06; DT17; CDMT22] was introduced. Even though it seems to beat ISD-based algorithms in some regimes, this approach still appears to lag behind for the parameters used in cryptography.

Since it will be central in the security analysis of the construction presented in Chapter 7, we shall begin by describing Prange algorithm (Algorithm 1.12).

The bet of linear algebra: Prange algorithm. Since an $[n,k]$-code is a k-dimensional vector space, it can be uniquely determined by a set of k-position $\mathcal{I} \subset \{1,\dots,n\}$, called an *information set*. In other words, $\mathcal{I}$ is an information set of $\mathscr{C}$ if and only if $\mathbf{G}_{\mathcal{I}}$ is invertible when $\mathbf{G}$ is any generator matrix of $\mathscr{C}$. Equivalently, $\mathcal{I}$ is an information set if and only if $\mathbf{H}_{\mathcal{I}^c}$ is invertible for a parity-check matrix $\mathbf{H}$ of $\mathscr{C}$. Now, for decoding a noisy codeword $\mathbf{y} = \mathbf{c} + \mathbf{e}$ where $|\mathbf{e}| = w$, it suffices to guess an information set $\mathcal{I}$ which does not intersect the support of $\mathbf{e}$, *i.e.* which contains no error position. In particular,

$$\forall i \in \mathcal{I},\ \mathbf{e}_i = 0, \quad \text{or equivalently} \quad \mathbf{y}_{\mathcal{I}} = \mathbf{c}_{\mathcal{I}} \tag{1.7}$$

Then, one only has to compute the unique codeword $\widehat{\mathbf{c}}$ which coincides with $\mathbf{y}$ on $\mathcal{I}$ by solving a linear system, and to check whether $|\mathbf{y} - \widehat{\mathbf{c}}| = w$. The idea of Prange's algorithm is to repeat this procedure until success.

Algorithm 1.12 : Prange algorithm

Input : $\mathbf{G}, \mathbf{y} \stackrel{\text{def}}{=} \mathbf{mG} + \mathbf{e}$
Output : $\mathbf{e}$

1 Pick a set $\mathcal{I}$ of size k until $\mathbf{G}_{\mathcal{I}}$ is full rank.
2 Compute $\widehat{\mathbf{c}} \stackrel{\text{def}}{=} (\mathbf{y}_{\mathcal{I}}\mathbf{G}_{\mathcal{I}}^{-1})\mathbf{G}$.
3 Set $\widehat{\mathbf{e}} \stackrel{\text{def}}{=} \mathbf{y} - \widehat{\mathbf{c}}$
4 **if** $|\widehat{\mathbf{e}}| = w$ **then**
5 **return** $\widehat{\mathbf{e}}$
6
7 **else**
8 Go back to Step 1

In the regime where there is a unique solution to the Decoding Problem, *e.g.* when the number w of errors is far below Gilbert-Varshamov, it can be proved that the complexity of Prange algorithm is given by

$$\frac{\binom{n}{w}}{\binom{n-k}{w}} \times T_{\text{linalg}}, \tag{1.8}$$

where T_{linalg} is the complexity of inverting a $k \times k$ matrix. The interested reader can refer to [Deb23, Chapter 3] for a detailed analysis of Prange algorithm, as well as some other ISD algorithms.

Hardness results. Prange's original approach seems to be the most simple one can think about besides exhaustive search. Yet, even after the tremendous research regarding decoding algorithms, even the best improvements still have a complexity exponential in the number of errors: the average running time of a generic decoder $\mathscr{A}$ at distance $w = \omega n$ is always of the form

$$\frac{T}{\varepsilon} = 2^{\omega n \cdot (c(q;R,\omega)+o(1))},$$

where ε is the success probability of $\mathscr{A}$ to output a solution in one run, and $c(q;R,\omega)$ is some constant depending on the chosen algorithm $\mathscr{A}$. The advanced techniques used in the different ISD algorithms aim at improving this exponent and the subexponential factors in different ranges of parameters ([Deb23, Chapter 3]). Nevertheless, there even exist some regimes of parameters where the most advanced ISD do not perform better than Prange:

- Sendrier and Canto-Torres proved in [CS16] that when the noise is sublinear in the length ($\omega = o(1)$), the best possible exponent is that of Prange.
- When $q \to \infty$, a result of Canto-Torres [Cha17] asserts that the complexity of all ISD-based algorithms converges towards that of Prange. In other words, the improvements are more mostly interesting for small values of q. This needs to be related to the fact that the Hamming metric becomes less meaningful as q increases, since it does not measure the *magnitude* of the coefficients (contrary to the Euclidean metric used in lattice-based cryptography, or the Lee metric), but only the number of non-zero elements.

Remark 1.13. *As far as we know, quantum computing does not seem to significantly improve on the best decoding algorithms, and their running time still remains exponential in the number of errors [OS09; Ber10; KT17].*

All this discussion yields to the "hardness landscape" presented in Figure 1.4. By easy, we mean in polynomial time (in the code length n). This can happen for example for a noise rate of the form $\omega = \Theta\left(\frac{\log n}{n}\right)$. Surprisingly, when the noise rate is very large, the problem becomes once again exponential in the code length for $q > 2$.

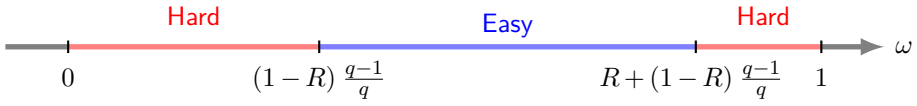


Figure 1.4: Hardness of $\mathsf{DP}(q;n,R,\omega)$ as a function of ω.

Parameters used in cryptography. In general, for code-based cryptography the rate R is considered to be constant (typically $R = 1/2$), but the noise rate varies depending on the applications:

(i) Constant noise rate for the WAVE signature scheme [DST19a]: $\omega =$ large constant C (*e.g.* $C \approx 0.95$). In this case we are in the rightmost part of Figure 1.4.

(ii) For Classic MCELIECE encryption scheme [McE78], the noise rate is $\omega = \Theta\left(\frac{1}{\log n}\right)$. However, regarding the sizes involved, this is much closer to the constant noise rate than the sublinear.

(iii) Sublinear noise rate for BIKE and HQC encryption schemes [AABB+22a; AABB+22b]: $\omega = \Theta\left(\frac{1}{\sqrt{n}}\right)$.

(iv) For building *Pseudorandom Correlations Generators*: $\omega = \Theta\left(\frac{1}{n}\right)$.

As mentioned above, we sometimes consider a very low error rate. In particular, for the applications to secure multiparty computation (MPC) presented in Part III, and more precisely in Chapter 7, the number of errors is a *constant*. In this case, Prange algorithm will run in *polynomial* time in the code length. This may be surprising at first glance in a cryptographic context. However, contrary to "traditional" code-based cryptography where the considered lengths range between 10,000 and 50,000; in the MPC situation we consider codes of *huge* lengths of several millions, even billions. In this situation, we can afford to have say 100 errors to achieve a security level of 128 bits, taking into account the cost of linear algebra with those huge matrices. This will be detailed more precisely in Chapter 7.[vii]

1.1.4.2 Relation to Learning Parity With Noise (LPN)

For cryptographic applications, there is a computational problem known as *Learning Parity with Noise* (LPN), which is closely related to the Decoding Problem. As suggested by its name, it finds its roots in computational learning theory ([Val84]). More precisely, in this domain, one of the most important problems is the following: Let $f : \mathcal{X} \mapsto \mathcal{Y}$ be any function. Given pairs $(x, f(x))$, the goal is to recover f with as few samples as possible. The main question surrounding this problem is which class of functions can be *learned* efficiently. Since the 1980s, it has been remarked that when the above problem is *noisy*, that is to say when we are not given access to perfect data, but when a small part of the samples are corrupted, even the simplest class of functions such as the class of parity functions parameterised by elements $\mathbf{s} \in \mathbb{F}_2^k$ and defined as

$$f_{\mathbf{s}} : \left\{ \begin{array}{ccc} \mathbb{F}_2^k & \to & \mathbb{F}_2 \\ \mathbf{a} & \mapsto & \langle \mathbf{a}, \mathbf{s} \rangle \end{array} \right.$$

are very hard to learn [AL87]. This motivated the introduction of this class of problems in cryptography.

In the sequel, we use the notion of *oracle* for a given probability distribution $\mathcal{D}$. Intuitively, it is a black box which we can query as we wish and which outputs independent random elements distributed according to $\mathcal{D}$. More formally, an oracle is nothing but a sequence $(X_n)_{n \in \mathbb{N}}$ of independent random variables, identically distributed according to $\mathcal{D}$.

[vii] Although we do not do that, it might be more accurate to also take into account the cost of storing and manipulating those huge matrices.

Definition 1.14 (LPN Oracle)

Let $k \in \mathbb{N}$, $\omega \in (0,1/2)$ and $\mathbf{s} \in \mathbb{F}_2^k$. The LPN oracle $\mathcal{O}_{\mathbf{s},\omega}$ is defined as follows: on each query, it outputs an independent fresh sample $(\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle + e)$ where $\mathbf{a} \leftarrow \mathbb{F}_2^k$ is uniformly distributed, and e is a Bernoulli random variable with success probability ω.

Problem 1.15 (Learning Parity with Noise (LPN(k,ω)))

Data. We are given an LPN oracle $\mathcal{O}_{\mathbf{s},\omega}$ with respect to a secret vector $\mathbf{s}$, picked uniformly at random in $\mathbb{F}_2^k$.

Goal. Recover $\mathbf{s}$.

Remark 1.16. *The above LPN problem is defined over the binary field $\mathbb{F}_q$, but it can be straightforwardly extended to larger fields.*

In other words, given an *a priori* unbounded number of corrupted samples of some random parity function $f_{\mathbf{s}}$, the goal is to *learn* the secret parameter $\mathbf{s}$. Nevertheless, this is mostly a problem of theoretical interest, and in a cryptographic context, we cannot really have unbounded samples. It turns out that when the number of samples is fixed and part of the problem, then LPN is exactly a variant of the Decoding Problem. Indeed, consider N samples (*i.e.* N queries to the oracle $\mathcal{O}_{\mathbf{s},\omega}$)

$$\big(\mathbf{a}_1, \langle \mathbf{a}_1, \mathbf{s} \rangle + e_1\big), \ldots, \big(\mathbf{a}_N, \langle \mathbf{a}_N, \mathbf{s} \rangle + e_N\big),$$

and consider the matrix $\mathbf{G} \in \mathbb{F}_2^{k \times N}$ whose i-th column is exactly $\mathbf{a}_i^\top$. Then, the above collection of N samples is nothing else than the pair $\big(\mathbf{G}, \mathbf{sG}+\mathbf{e}\big)$, where $\mathbf{e} \stackrel{\text{def}}{=} (e_1, \ldots, e_N)$ has expected Hamming weight $N\omega$. In particular, LPN with N samples corresponds to decoding a binary random code of rate k/N. This can be made more precise with the following proposition from [DR22].

Proposition 1.17 ([DR22, Proposition 3.6])

Suppose there exists an algorithm $\mathscr{A}$ able to decode random codes of length N, dimension k at distance t, with success probability ε and running in time T. Then there exists an algorithm running in time $O(T+N)$ which solves LPN$(k, t/N)$ with success probability $\Omega\left(\frac{\varepsilon}{\sqrt{t}}\right)$, and making N queries to the underlying LPN oracle.

Remark 1.18. *It turns out that LPN is a useful* intermediate *problem between the average and worst cases of the Decoding problem. We will have the occasion to discuss it further in Chapter 5.*

1.1.5 McEliece Cryptosystem

As we have seen so far, decoding a linear code is very difficult in general. In particular, DP seems to be an interesting problem to build cryptography on. The most important code-based encryption scheme is the approach of McEliece [McE78]. It is impressive to note that this article was published slightly after Diffie and Hellman's breakthrough work [DH76] and the discovery of RSA [RSA78]. It is even more impressive that the original proposal based on so-called *Goppa*

codes is still not broken, even quantumly, and is at the core of Classic McEliece [ABCC+22] which is still competing in round 4 of NIST standardisation process. This makes of McEliece the oldest cryptosystem having this property.

1.1.5.1 Wishful Thinking: Encryption from Trapdoor Error Correcting Codes

The idea of McEliece cryptosystem is simple. It makes use of the notion of *trapdoor error correcting codes*. In this section we shall give a conceptual presentation of McEliece's approach for building encryption schemes, and we shall discuss concrete instantiations in Section 1.1.5.2.

A high level description. Imagine that Alice has a very special code $\mathscr{C}$ for which she knows an efficient decoding algorithm $\mathscr{D}$ up to some distance t. She wants to keep $\mathscr{D}$ secret. This will be her trapdoor. On the other hand, she can reveal a generator matrix $\mathbf{G}$ of $\mathscr{C}$. Now, if someone, say Bob, wants to send a message $\mathbf{m}$ to Alice, he can first encode it using the public generator matrix $\mathbf{G}$. He then adds a random error $\mathbf{e}$ yielding to the ciphertext $\mathbf{y} \stackrel{\text{def}}{=} \mathbf{mG}+\mathbf{e}$. Upon receiving $\mathbf{y}$, Alice can apply here secret decoding algorithm $\mathscr{D}$ in order to recover the message $\mathbf{m}$, as long as $|\mathbf{e}| \leqslant t$.

Now, recovering the plaintext $\mathbf{m}$ from the ciphertext $\mathbf{mG}+\mathbf{e}$ is tantamount to decoding in $\mathbf{G}$. In other words, if $\mathbf{G}$ is *indistinguishable* from a random matrix, the *message security* is given by the complexity of decoding a random linear code at distance t. On the other hand, the *key security* is given by the hardness of recovering the decoding algorithm $\mathscr{D}$ from the sole knowledge of the public matrix $\mathbf{G}$. In general, this begins by distinguishing $\mathbf{G}$ from a random matrix.

Let us formally define what we mean by *trapdoor error correcting code*.

Definition 1.19 (Trapdoor Error Correcting Codes)

A family of $[n,k]$ codes $\mathcal{F} = \{\mathscr{C}(s) \mid s \in \mathcal{S}\}$ parameterised by a set of secrets $\mathcal{S}$ forms a family of *trapdoor error correcting codes* if

- The knowledge of s enables to design an efficient decoding algorithm $\mathscr{D}(s)$ for the code $\mathscr{C}(s)$ up to some decoding distance $t(s)$;
- The function
$$\mathscr{C}(\cdot) : \left\{ \begin{array}{ccc} \mathcal{S} & \longrightarrow & \mathcal{F} \\ s & \longmapsto & \mathscr{C}(s) \end{array} \right.$$
is *one-way*, *i.e.* given a description of $\mathscr{C}(s)$, it should be difficult to recover s.

This definition may seem void. However, in Section 1.1.5.2 we will discuss some codes which are believed to have this trapdoor property. For now, let us simply assume that they exist. The following abstract presentation of McEliece cryptosystem is inspired by [Cou19]. The public parameters of the system is a family $\mathcal{F}$ of $[n,k]$ trapdoor codes.

Key Generation.

- Pick a random element $s \in \mathcal{S}$.
- Compute a generator matrix $\mathbf{G}_{\text{pub}}$ of the code $\mathscr{C}(s)$, and the decoding algorithm $\mathscr{D}(s)$ up to distance $t_{\text{pub}} \stackrel{\text{def}}{=} t(s)$.

The public key is $(\mathbf{G}_{\text{pub}}, t_{\text{pub}})$, and the secret key is s.

Encryption. The message space is $\mathbb{F}_q^k$, and the ciphertext space is $\mathbb{F}_q^n$.

- Pick a random element $\mathbf{e} \in \mathbb{F}_q^n$ of weight t_{pub};
- Compute the ciphertext
$$\mathbf{y} \stackrel{\text{def}}{=} \mathbf{m}\mathbf{G}_{\text{pub}} + \mathbf{e}.$$

Decryption. Apply the secret decoding algorithm $\mathscr{D}(s)$ on the ciphertext $\mathbf{y}$ to recover the plaintext $\mathbf{m}$.

Remark 1.20. *In [Nie86], Niederreiter proposed a dual version of McEliece cryptosystem, relying on parity-check matrices instead of generator matrices. The message $\mathbf{m} \in \mathbb{F}_q^k$ to be encrypted is then first mapped to a weight t_{pub} word $\varphi(\mathbf{m}) \in \mathbb{F}_q^n$ by a public function φ and the ciphertext is now the syndrome $\varphi(\mathbf{m})\mathbf{H}^\top$. The decryption is similar. Such a function φ is described for instance in [FS96]. Since the security of both McEliece and Niederreiter cryptosystems are in reality equivalent [LDW94], we will not make the distinction in the sequel.*

1.1.5.2 Instantiating McEliece Cryptosystem

In order to instantiate an encryption scheme in the McEliece framework, the challenge is therefore to design good families $\mathcal{F} \stackrel{\text{def}}{=} \{\mathscr{C}(s) \mid s \in \mathcal{S}\}$ of trapdoor codes. The security boils down to two things:

(1) *Key security.* From the knowledge of a public code $\mathscr{C}(s)$, it should be *computationally infeasible* to recover the secret s. In particular $\mathcal{F}$ should be large enough, and $\mathscr{C}(s)$ should be as close as possible to a *random code*.

(2) *Message security.* Without the knowledge of the secret decoding algorithm $\mathscr{D}(s)$, and if $\mathscr{C}(s)$ is indistinguishable from a random code, an attacker can only try to use a generic decoding algorithm at distance $t_{\text{pub}} = t(s)$. In particular, $\mathscr{D}(s)$ should be able to decode at a distance as large as possible.

Remark 1.21. *Precise definitions for the security of a public key encryption scheme are formally defined with respect to a theoretical* attack model *which specifies how powerful can be the adversary. In general, we ask an encryption scheme to be secure under the* indistinguishability model*: an adversary $\mathscr{A}$ should not be able to distinguish the encryption of two messages*[viii] $\mathbf{m}_0$ *and* $\mathbf{m}_1$ of his choice*. This is modelled by a two players protocol between a challenger $\mathcal{C}$ and the adversary $\mathscr{A}$. In a* first phase*, $\mathscr{A}$ receives the public key and perform any computation of his choosing (and depending on the targeted security guarantee). At the end of this phase, $\mathscr{A}$ outputs two messages $\mathbf{m}_0$ and $\mathbf{m}_1$ of the same length. In a* second phase*, $\mathcal{C}$ chooses uniformly at random a bit $\beta \leftarrow \{0,1\}$ and sends to $\mathscr{A}$ the encryption of $\mathbf{m}_\beta$. Then, $\mathscr{A}$ performs any computation and outputs a bit b. The adversary is* successful *if $b = \beta$ with probability $\frac{1}{2} + \varepsilon$ with $\varepsilon > 0$.*

We can consider different indistinguishability models. The strongest is known as indistinguishability under chosen ciphertext attack *(IND-CCA). In this model, we assume that $\mathscr{A}$ has access to a* decryption oracle*, i.e. a black box algorithm $\mathscr{O}$ which takes as input the public key, as well as* any *ciphertext encrypted under this public key, and outputs the corresponding plaintext in constant time. The only restriction is that $\mathscr{A}$ is not allowed to query $\mathscr{O}$ with the challenge $\mathbf{m}_\beta$. An IND-CCA secure encryption scheme should resist most real-world attacks, and it is therefore*

[viii] of the same length,

usual in cryptography to insist that all encryption schemes should achieve this IND-CCA security. A weaker notion is known as indistinguishability under chosen plaintext attack *(IND-CPA) where $\mathscr{A}$ is not given access to such a decryption oracle. In the case of McEliece cryptosystem, under the assumption that the public code is* indistinguishable *from a random one, this means that two noisy codewords should be indistinguishable. Fortunately, it can be proven that noisy codewords (or syndromes) of a random code are in fact* indistinguishable *from random vectors of the same length. This fact will be made more precise in Section 1.2. In general, designers of encryption schemes target the IND-CPA security, since there exist generic transformations such as [HHK17] to get an encryption scheme IND-CCA secure.*

There exist essentially two classes of codes which benefit from efficient decoders:

- Codes arising from the evaluation of polynomials over a finite field such as Reed-Solomon codes [RS60]; their generalisations to algebraic curves of higher genus, known as algebraic-geometry codes [Gop81]; their restriction to a subfield such as alternant and Goppa codes [MS86, Chapter 12]; or Reed-Muller codes [Ree54; Mul54]. Such codes are endowed with a *deterministic* decoding algorithm.
- Codes whose duals have a basis of unusually sparse codewords such as LDPC [Gal63] and MDPC [MTSB13]. Such codes are endowed with a *probabilistic* decoding algorithm.

However, virtually all instantiations led to devastating key recovery attacks. The only codes which appear to resist attacks are

- Goppa codes,[ix] and more precisely *binary* Goppa codes. Interestingly enough, they were already the suggestion of McEliece in his original construction [McE78]. They are now at the core of Classic McEliece [ABCC+22] which is competing in round 4 of NIST call for post-quantum primitives.
- MDPC codes, on which relies the BIKE cryptosystem, also present in round 4 of NIST competition.

Once a good family of trapdoor codes is chosen, the main drawback of McEliece cryptosystem relies in the size of the public key. Indeed, since the public code should be indistinguishable from a random linear code over $\mathbb{F}_q$, it is necessary to give a complete generator matrix to fully describe the code, which yields a public key of $kn\log_2(q)$ bits. Some optimisations allow to reduce the key size to $k(n-k)\log_2(q)$ bits. In general, the code rate is fixed (*e.g.* to $1/2$, *i.e.* $k=n/2$), which yields a public key of size *quadratic* in n. Moreover, the decoding distance t_{pub} should be large enough to resist generic decoding algorithms, which have a complexity of the form $2^{c\cdot t_{\text{pub}}}$. In general, t_{pub} will be a $\theta(n/\log(n))$ or even $\theta(\sqrt{n})$ for MDPC codes. In the latter situation, this means that in order to achieve λ bits of security, n should scale in λ^2, and the public key scales in λ^4, which is huge. For instance, in his original article [McE78], McEliece proposed to instantiate this cryptosystem with $[n=1024, k=524]$ binary Goppa codes, which are able to correct up to 50 errors. This yields a public key of 32.750 kB, however this achieves less than 80 bits of security with current state of the art decoders. In Table 1.1, we give the key sizes for the parameters proposed in NIST submission Classic McEliece [ABCC+22], and in Table 1.2, we give the key sizes for the parameter sets originally proposed in [MTSB13] for McEliece instantiated with MDPC codes. In both cases, n and k are respectively the length and the dimension of the public code.

In [Gab05], Gaborit imagined a solution to reduce the size of the public key. More precisely, he proposed to consider families $\mathcal{F}$ of *quasi-cyclic* codes, for which it is enough to publish only

[ix] More generally alternant codes

Security level (bits)	n	k	Public key size (Bytes)
128	3,488	2,720	261,120
256	8,192	6,528	1,357,824

Table 1.1: Public key size of Classic McEliece

Security level (bits)	n	k	Public key size (Bytes)
128	19,714	9,857	12,145,056
256	65,542	32,771	134,242,305

Table 1.2: Public key size of MDPC-based McEliece as per [MTSB13]

a few rows of a generator matrix in order to deduce the whole basis by computing the action of some cyclic group G. Obviously, this automatically gives a distinguisher between such codes and random linear codes which do not benefit from such action. Nevertheless, as surprising as it may sound, it turns out that decoding random *quasi-cyclic* codes is not significantly easier than decoding random codes. In particular, using such codes does not appear to hurt too much the security. This will be precised in Section 1.3.

Nevertheless, the security of McEliece cryptosystem still relies on the hardness of distinguishing the public code from a random linear code. In particular, an encryption scheme whose security truly relies on the Decoding Problem DP (Problem 1.10) is yet to be found. This is the topic of Section 1.2.

1.2 Decisional Version of the Decoding Problem

In the last section we have recalled hardness results regarding the Decoding Problem. However, this is partly satisfying. Indeed, as we have seen so far, the security of McEliece-like cryptosystems also relies on the assumed hardness of distinguishing the public code from a random one, which is mostly assessed through the test of time (and the lack of cryptanalysis). In particular, they do not benefit from a *theoretical reduction* to the decoding problem. And indeed, virtually all the instantiations of McEliece have been broken by attacking this distinguishing problem.

However, code-based cryptography should not be limited to McEliece cryptosystems, and another line of work initiated by Alekhnovich [Ale03], managed to design such a cryptosystem truly based on the hardness of decoding. More precisely, as we will see in this section, Alekhnovich cryptosystem relies on the *decisional version* of the Decoding Problem (Problem 1.27) which asks to *distinguish* between a noisy codeword $\mathbf{c}+\mathbf{e}$ and a uniform vector $\mathbf{y}^{\text{unif}}$ given the sole knowledge of a generator (or a parity-check) matrix of the code. This could be considered as a code-based analogue of the Decisional Diffie-Hellman problem. However, contrary to the latter problem,[x] we know since the work of Fischer and Stern [FS96] that the Decisional Decoding Problem is actually *equivalent* to the computational version. This result is obtained through a *search-to-decision reduction* and will be recalled in Section 1.2.3.

[x] Decisional Diffie-Hellman is known to be strictly easier than the computational (or search) version [JN03].

1.2.1 Notion of Distinguisher

In this thesis we will often work with decisional problems, where the goal is to distinguish between two probability distributions $\mathcal{D}_0$ and $\mathcal{D}_1$ defined over the same space $\mathcal{E}$. More formally, consider a probabilistic algorithm $\mathscr{A}$ which takes as input an element of $\mathcal{E}$ and outputs a bit $b \in \{0,1\}$, and let $\beta \leftarrow \{0,1\}$ be a uniformly random bit. Consider X_β be picked according to distribution $\mathcal{D}_\beta$. Algorithm $\mathscr{A}$ is characterised by its success probability defined as $\mathbb{P}\left(\mathscr{A}(X_\beta) = \beta\right)$, where the probability is computed over the internal randomness of $\mathscr{A}$, the uniformly random bit β and X_β. Obviously, if $\mathscr{A}$ is deterministic and always returns 1, then the above probability is simply $1/2$. Therefore, the goal of a *distinguisher* is to succeed with probability *strictly greater* than $1/2$. Instead, it is more relevant to consider its *distinguishing advantage* defined as

$$\mathrm{Adv}_{\mathscr{A}}\left(\mathcal{D}_0, \mathcal{D}_1\right) \stackrel{\text{def}}{=} \frac{1}{2}\Big(\mathbb{P}\left(\mathscr{A}(X) = 1 \mid X \leftarrow \mathcal{D}_1\right) - \mathbb{P}\left(\mathscr{A}(X) = 1 \mid X \leftarrow \mathcal{D}_0\right)\Big).$$

Remark 1.22. *In the literature, the advantage is sometimes defined to be the absolute value of the above quantity. The rationale behind, is that an algorithm with binary output which is wrong most of the time can be as useful as an algorithm is right most of the time, it suffices to switch the result. In order to simplify the discussions in this manuscript, we assume that this quantity is always positive.*

It is readily seen that the success probability of Algorithm $\mathscr{A}$ satisfies

$$\mathbb{P}\left(\mathscr{A}(X_\beta) = \beta\right) = \frac{1}{2} + \mathrm{Adv}_{\mathscr{A}}\left(\mathcal{D}_0, \mathcal{D}_1\right).$$

Remark 1.23. *It is well known that the statistical distance (Equation (1.1)) $\Delta(\mathcal{D}_0, \mathcal{D}_1)$ is the advantage (after renormalisation) of an* optimal *distinguisher, in the sense that the advantage of* any *distinguisher $\mathscr{A}$ is upper bounded by $\Delta(\mathcal{D}_0, \mathcal{D}_1)$, and there exists a distinguisher whose advantage is exactly that (see [Nan06] for instance).*

Sometimes it can be handy to run the distinguisher $\mathscr{A}$ several times on independent inputs from distributions $\mathcal{D}_\beta$ in order to have a better distinguishing property. In this case, we say that $\mathscr{A}$ has *oracle access* to the distribution $\mathcal{D}_\beta$. Intuitively, an oracle is a black box that we can query arbitrarily many times and whose outputs are independent random elements distributed according to $\mathcal{D}_\beta$. This is formalised by a sequence $(X_n)_{n\in\mathbb{N}}$ of independent random variables, identically distributed according to $\mathcal{D}_\beta$. This is particularly useful, since as long as the advantage (as defined above) is strictly positive, then by querying the oracle multiple times it is possible to give a correct answer with very good probability. Indeed, let $\mathscr{A}$ be a distinguisher with advantage $\delta > 0$, and repeat the distinguishing experiment m times, where m is an integer to be determined. In other words, pick $X_0, \ldots, X_m$ independently from the unknown distribution $\mathscr{D}_\beta$, and run $\mathscr{A}$ on each of those inputs. After the m trials, perform a majority voting and output the bit which appeared the most. It turns out that it suffices to choose m as $\Omega\left(\frac{1}{\delta^2}\right)$ to be correct with good probability. More precisely, we have the following proposition:

Proposition 1.24

Let $\mu \in (0,1)$ be some parameter, and let $\mathscr{A}$ be a distinguisher with advantage δ.

Repeating the distinguishing experiment m times with $m \geqslant \ln(\frac{1}{\mu})\frac{1}{2\delta^2}$ guarantees that the majority voting outputs the correct result with probability at least $1-\mu$.

This relies on the famous Chernoff bound.

Proposition 1.25 (Chernoff bound)

Let $(X_j)_{1\leqslant j\leqslant m}$ be m independent Bernoulli random variables with success probability $\frac{1}{2}+\delta$. Let $X \stackrel{\text{def}}{=} \sum_{j=1}^{m} X_j$. Then

$$\mathbb{P}\left(X \leqslant \frac{m}{2}\right) \leqslant e^{-2m\delta^2}.$$

Proof of Proposition 1.24. Let

$$X_j \stackrel{\text{def}}{=} \begin{cases} 1 & \text{if trial } j \text{ is correct} \\ 0 & \text{otherwise.} \end{cases}$$

denote the indicator random variable that the j-th run of $\mathscr{A}$ returns the correct bit β. By definition of the distinguishing advantage, X_j is a Bernoulli random variable with success probability $\frac{1}{2}+\delta$. After m trials, the majority voting fails if and only if more than $m/2$ runs are wrong, and by Chernoff bound, this happens with probability less than $e^{-2m\delta^2}$. In other words, if $m \geqslant \ln(\frac{1}{\mu})\frac{1}{2\delta^2}$, then the majority voting strategy succeeds with probability at least $1-\mu$. □

Nevertheless, we did not say anything so far about the *efficiency* of distinguishers. In particular, for cryptographic applications it is more relevant to restrict ourselves to distinguisher which are *efficient*, say running in time polynomial, in its input, and/or some security parameter. This will be assumed implicitly in the rest of this manuscript.

Remark 1.26. *We can refine the notion of statistical distance between two distributions using the interpretation of Remark 1.23, by defining the following distance*

$$\Delta_T(\mathcal{D}_0,\mathcal{D}_1) \stackrel{def}{=} \sup_{\mathscr{A}\, running\ in\ time\ \leqslant\ T} \mathrm{Adv}_{\mathscr{A}}(\mathcal{D}_0,\mathcal{D}_1).$$

When restricted to the class of efficient *distinguisher, this refinement is sometimes referred to as the* computational distance *between $\mathcal{D}_0$ and $\mathcal{D}_1$.*

1.2.2 The Decisional Decoding Problem

In cryptography, we often work with computational problems in their *search* variants. More precisely, given a function f supposed to be hard to invert, and an image $f(x)$ for some x, the goal of the search variants is to recover x. For example, it should be computationally infeasible

to decrypt a ciphertext with the sole knowledge of the public key. However, as already hinted in Remark 1.21, sometimes the security of cryptographic primitives also relies on *decisional variants*, where we ask an adversary to *distinguish* between two distributions. Up until now, we have only considered the Decoding Problem DP (Problem 1.10) in its *search* version, namely given a random linear code and a random noisy codeword, we ask to recover the error. Let us now introduce the *decisional* variant, parameterised by an integer n, a real value $R \in (0,1)$ and a probability distribution ψ over $\mathbb{F}_q^n$.

Problem 1.27 (Decisional Decoding Problem (DDP$(q;n,R,\psi)$)

Set $k \stackrel{\text{def}}{=} \lfloor Rn \rfloor$. Let $\mathbf{m}$ be drawn uniformly at random in $\mathbb{F}_q^k$ and consider the following two distributions

- $\mathcal{D}_0 : (\mathbf{G}, \mathbf{y}^{\text{unif}})$ uniformly distributed over $\mathbb{F}_q^{k\times n} \times \mathbb{F}_q^n$,
- $\mathcal{D}_1 : (\mathbf{G}, \mathbf{mG}+\mathbf{e})$ where $\mathbf{G} \leftarrow \mathbb{F}_q^{k\times n}$, and $\mathbf{e} \leftarrow \psi$.

Given oracle access to distribution $\mathcal{D}_\beta$ where $\beta \leftarrow \{0,1\}$ is a uniform bit, the goal is to recover β.

In general, ψ will be the uniform distribution over the Hamming sphere $\mathcal{S}_t$ of given weight t.

Remark 1.28. *Similarly to the search Decoding Problem 1.10, DDP is related to the* decisional *version of the LPN problem (whose search version has been defined in 1.1.4.2), where the goal is to distinguish an LPN oracle $\mathcal{O}_{\mathbf{s},\omega}$ from an oracle $\mathcal{O}^{\text{unif}}$ which outputs samples of the form $(\mathbf{a}, \mathbf{y}^{\text{unif}})$, both uniformly distributed in $\mathbb{F}_2^n$. More precisely, an adversary against the* decisional-*LPN problem making N queries to its input oracle will in reality have to solve $DDP(q=2; N, \frac{k}{N}, \psi_N)$ where ψ_N outputs vectors of length N whose components are independent Bernoulli random variables of success probability ω.*

This problem is obviously related to Problem 1.10. More precisely, an algorithm able to solve the Decoding Problem can immediately be turned into a distinguisher between $\mathcal{D}_0$ and $\mathcal{D}_1$. In particular, this shows that DDP is *easier* than the (search) Decoding Problem DP, and it is natural to wonder if this is *strict*. As surprising as it may be, it turns out that both problems are actually *equivalent*, as we will see in the following section.

1.2.3 Pseudorandomness of Decoding: a Search-to-Decision Reduction

The relationships between the Decisional Decoding Problem DDP (1.27) and the Search Decoding Problem DP (1.10) have been investigated by many authors, but the first to actually *give a proof* that both problems are actually equivalent are Fischer and Stern [FS96]. They achieve this result via a *search-to-decision* reduction. The aim of this section is to recall their reduction.

More precisely, we prove the following theorem:

Theorem 1.29 (Search-to-Decision Reduction)

Let $n > 0$ be an integer. Let $\mathscr{A}$ be a probabilistic algorithm running in time $T(n)$ solving $\mathrm{DDP}(n, R, \tau)$ with distinguishing advantage ε. Then there exists an algorithm $\mathscr{A}'$ which solves $\mathrm{DP}(n, R, \tau)$ in time $O\left(T(n) n^2 \left(\log\left(\frac{1}{\varepsilon}\right)\right)^3\right)$ with probability $\Omega(\varepsilon^2)$.

The proof of this reduction rests on the following technical result due to Goldreich and Levin [GL89; Lev93]. A proof can also be found in [Zém16].

Theorem 1.30 (Goldreich-Levin)

Let n, k be positive integers. Let $f : \mathbb{F}_2^k \to \mathbb{F}_2^n$ be some function, $\mathbf{x}, \mathbf{r} \leftarrow \mathbb{F}_2^k$ be two independent uniform vectors and let $\mathscr{B}$ be a probabilistic algorithm taking input from $\mathbb{F}_2^n \times \mathbb{F}_2^k$ and outputting an element of $\mathbb{F}_2$, such that

$$\mathbb{P}\left(\mathscr{B}\left(f(\mathbf{x}), \mathbf{r}\right) = \langle \mathbf{x}, \mathbf{r} \rangle\right) = \frac{1}{2} + \varepsilon,$$

where the probability is computed over $\mathbf{x}, \mathbf{r}$ and the internal randomness of $\mathscr{B}$. Denote by T the running time of $\mathscr{B}$.

Then, there exists a probabilistic algorithm $\mathscr{B}'$ with input from $\mathbb{F}_2^n$ which outputs an element of $\mathbb{F}_2^k$, such that $\mathscr{B}'$ runs in time $O\left(T k^2 \left(\log\left(\frac{1}{\varepsilon}\right)\right)^3\right)$ and

$$\mathbb{P}\left(\mathscr{B}'\left(f(\mathbf{x})\right) = \mathbf{x}\right) = \Omega\left(\varepsilon^2\right),$$

where the probability is computed over $\mathbf{x}$ and the internal randomness of $\mathscr{B}'$.

Remark 1.31. *A generalisation of this result to larger fields $\mathbb{F}_q$ is given in [GRS00, Section 2].*

Proof of Theorem 1.29, adapted from [FS96; Zém16; Deb23].

Let $k, t \leqslant n$ be positive integers, and let $\mathbf{G} \leftarrow \mathbb{F}_2^{k \times n}$ be a uniformly random matrix, let $\mathbf{x} \leftarrow \mathbb{F}_2^k$ be a uniformly random vector, and let $\mathbf{e} \leftarrow \mathcal{S}_t$ be a uniformly random vector of Hamming weight t. Let $\mathscr{A}$ be an algorithm solving $\mathrm{DDP}(n, R, t/n)$ with advantage ε. The roadmap is the following: we build an algorithm $\mathscr{B}$ satisfying the assumptions of Theorem 1.30 with the function

$$f_{\mathbf{G},\mathbf{e}} : \left\{ \begin{array}{ccc} \mathbb{F}_2^k & \longrightarrow & \mathbb{F}_2^n \\ \mathbf{x} & \longmapsto & \mathbf{x}\mathbf{G} + \mathbf{e} \end{array} \right. .$$

We then simply apply it. Algorithm $\mathscr{B}$ is described in Algorithm 1.31.

Algorithm 1.31 : Algorithm $\mathscr{B}$

Input : $\mathbf{G}, \mathbf{y} \stackrel{\text{def}}{=} \mathbf{xG} + \mathbf{e}$ and a uniformly random vector $\mathbf{r} \in \mathbb{F}_2^k$
Output : $\langle \mathbf{x}, \mathbf{r} \rangle$

1 Pick $\mathbf{s} \leftarrow \mathbb{F}_2^n$ uniformly at random.
2 Set $\mathbf{G}' \stackrel{\text{def}}{=} \mathbf{G} - \mathbf{r}^\top \mathbf{s}$. ▷ $\mathbf{G}'$ is uniformly distributed.
3 **return** $\beta \stackrel{def}{=} 1 - \mathscr{A}(\mathbf{G}', \mathbf{y})$.

Let us check that this procedure is correct. First, notice that since the matrix $\mathbf{G}$ is uniformly distributed, so is $\mathbf{G}'$. Now,

$$\mathbf{y} = \mathbf{xG} + \mathbf{e} = \mathbf{xG}' + \mathbf{x}\left(\mathbf{r}^\top \mathbf{s}\right) + \mathbf{e} = \mathbf{xG}' + \langle \mathbf{x}, \mathbf{r} \rangle \mathbf{s} + \mathbf{e}. \tag{1.9}$$

In particular, if $\langle \mathbf{x}, \mathbf{r} \rangle = 0$, then $\mathbf{y} + \mathbf{e} = \mathbf{xG}' + \mathbf{e}$ is a noisy codeword of the code generated by $\mathbf{G}$, but if $\langle \mathbf{x}, \mathbf{r} \rangle = 1$, then $\mathbf{y} + \mathbf{e}$ is uniformly random from the point of view of $\mathbf{G}'$ (since $\mathbf{s}$ is). Finally, since $\mathscr{A}$ has distinguishing advantage ε, Algorithm $\mathscr{B}$ will succeeds will probability $\frac{1}{2} + \varepsilon$.

In order to conclude the proof, it suffices to apply Theorem 1.30. □

1.2.4 Alekhnovich Encryption Scheme

In 2003, Alekhnovich [Ale03] introduced a new approach to design an encryption scheme based on error correcting codes. Unlike McEliece cryptosystem, Alekhnovich truly relies on the hardness of decoding random codes. More precisely, its security relies on the decisional version of the Decoding Problem, but as we have seen, DDP is equivalent to DP.

Alekhnovich Cryptosystem. Let k, n be integers. Starting from a random $[n, k]$ code $\mathscr{C}$, it proceeds as follows:

- *Key Generation.* Let $\mathbf{e}_{\mathsf{sk}} \leftarrow \mathrm{Ber}(\tau)^{\otimes n}$ be of small Hamming weight $\theta(\sqrt{n})$. The public key is $(\mathscr{C}, \mathbf{c} + \mathbf{e}_{\mathsf{sk}})$ where $\mathbf{c} \in \mathscr{C}$ and the secret key is $\mathbf{e}_{\mathsf{sk}}$.

- *Encryption.* To encrypt one bit $\beta \in \{0, 1\}$ set:
  - $\mathrm{Enc}(1) \stackrel{\text{def}}{=} \mathbf{u}$ where $\mathbf{u} \in \mathbb{F}_2^n$ is a uniformly random vector.
  - $\mathrm{Enc}(0) \stackrel{\text{def}}{=} \mathbf{c}^* + \mathbf{e}$ where $\mathbf{e}$ is of small Hamming weight $\theta(\sqrt{n})$ and $\mathbf{c}^*$ lies in the dual of the code $\mathscr{C}_{\mathsf{pub}}$ spanned by $\mathscr{C}$ and $\mathbf{c} + \mathbf{e}_{\mathsf{sk}}$.

- *Decryption.* The decryption of $\mathrm{Enc}(\beta)$ is $\langle \mathrm{Enc}(\beta), \mathbf{e}_{\mathsf{sk}} \rangle$, where $\langle \cdot, \cdot \rangle$ is the usual inner product on $\mathbb{F}_2^n$.

The correction of this procedure relies on the fact that

$$\langle \mathrm{Enc}(0), \mathbf{e}_{\mathsf{sk}} \rangle = \langle \mathbf{c}^* + \mathbf{e}, \mathbf{e}_{\mathsf{sk}} \rangle = \langle \mathbf{e}, \mathbf{e}_{\mathsf{sk}} \rangle,$$

where we used that $\mathbf{e}_{\mathsf{sk}} \in \mathscr{C}_{\mathsf{pub}}$ while $\mathbf{c}^*$ lies in its dual. Now, using Lemma 1.32 below, and the fact that both $\mathbf{e}_{\mathsf{sk}}$ and $\mathbf{e}$ have small Hamming weight of order $\theta(\sqrt{n})$, we can prove that this inner product is highly biased towards 0. On the other hand, $\langle \mathrm{Enc}(1), \mathbf{e}_{\mathsf{sk}} \rangle$ is a uniformly random bit.

Lemma 1.32 ([Til18, Appendix B, Lemma 6])

Let $\mathbf{e}_1$ be uniformly distributed over the vectors of $\mathbb{F}_2^n$ of Hamming weight w, and let $\mathbf{e}_2 \in \mathbb{F}_2^n$ be of Hamming weight t. Assume that both w and t are of order $\theta(\sqrt{n})$. Then

$$\mathbb{P}_{\mathbf{e}_1}\left(\langle \mathbf{e}_1, \mathbf{e}_2 \rangle = 1\right) = \frac{1}{2}\left(1 - e^{-2\frac{wt}{n}}\left(1 + O\left(\frac{1}{\sqrt{n}}\right)\right)\right).$$

In particular, the decryption will succeed with good probability. However, decryption failures are basically inherent to this scheme, and it might be necessary to repeat the procedure many times in order to lower this failure rate. Nevertheless, the strength of Alekhnovich's cryptosystem is that, contrary to McEliece cryptosystem, its security *does not* depend on hiding the description of a code:

- *Key security.* Recovering the private key from public data amounts to decoding the random code $\mathscr{C}$ at distance $\theta(\sqrt{n})$, *i.e.* solving Problem 1.10.
- *Message security.* Recovering the plaintext from the ciphertext is tantamount to *distinguishing* a noisy codeword from a uniformly random vector, *i.e.* solving the *decisional* version of the decoding problem (Problem 1.27).

In particular, since they are both equivalent, it turns out that Alekhnovich encryption scheme truly relies on the hardness of Decoding. However, as is, Alekhnovich cryptosystem is not practical. Indeed, the decryption process is *very slow*, since it might be needed to repeat the process many times *for each bit* to get a correct decryption with good enough probability. Moreover, since the number of errors in the Decoding Problems used to define the security is a $\theta(\sqrt{n})$, the complexity of breaking this scheme is about $\theta\left(2^{c\sqrt{n}}\right)$. It means that n needs to be at least *quadratic* in the targeted security parameter, and since the public key is the description of a random code of length n, it is of size *quadratic* in n, *i.e.* of the *huge* size $\Omega(\lambda^4)$ where λ is the targeted security parameter. However, the approach itself was a major breakthrough in code-based cryptography.

Remark 1.33. *In reality, since $\mathscr{C}$ is* random*, we can just specify a short* seed *to be used inside a Pseudorandom Generator, and the public key would amount to the seed and a noisy codeword $\mathbf{c} + \mathbf{e}_{sk}$ of length $n = \Omega(\lambda^2)$.*

In order to cope with those gigantic sizes, and lack of efficiency, it was proposed to consider algebraically structured codes. This is the topic of the next section.

1.3 Quasi-Cyclic Structure

1.3.1 Quasi-Cyclic Codes for Cryptography

As we have seen so far in this chapter, the main issue with code-based encryption scheme is the huge size of the public-key. Indeed, whether we consider a McEliece-like (recalled in Section 1.1.5)

cryptosystem, or the approach of Alekhnovich (recalled in Section 1.2.4), we basically need to transmit a whole random matrix, which means that the public key will be at least *quadratic* in the size of the ciphertexts. In order to mitigate this state of affairs, Gaborit proposed in [Gab05] to restrict the public keys to families of codes endowed with an additional algebraic structure, namely *quasi-cyclic codes*.

1.3.1.1 From Cyclic to Quasi-Cyclic Codes

One of the most studied families of codes is the *cyclic codes*. Indeed, they benefit from a rich algebraic structure which enables quite efficient encoding algorithms, and a compact representation. Some of them (*e.g.* BCH codes [Hoc59; BC60; GZ61]) also benefit from efficient decoders. They were introduced by Prange in a series of technical reports for the Air Force Cambridge Research Labs [Pra57; Pra58].

Cyclic Codes. A code $\mathscr{C} \subset \mathbb{F}_q^n$ is said to be *cyclic* if it is stable by the cyclic shift defined by

$$\sigma : \left\{ \begin{array}{ccc} \mathbb{F}_q^n & \longrightarrow & \mathbb{F}_q^n \\ (x_0, \ldots, x_{n-1}) & \longmapsto & (x_{n-1}, x_0, \ldots, x_{n-2}) \end{array} \right. ,$$

that is to say $\sigma(\mathscr{C}) = \mathscr{C}$. In particular, their automorphism group is unusually large compared to random linear codes, for which this situation is not expected to happen.

In order to manipulate cyclic codes, it turns out that it is more useful to consider vectors $\mathbf{c} \in \mathbb{F}_q^n$ as polynomials whose coefficients are exactly the components of $\mathbf{c}$. More formally, there is an $\mathbb{F}_q$-vector space isomorphism

$$\Phi : \left\{ \begin{array}{ccc} \mathbb{F}_q^n & \longrightarrow & \mathbb{F}_q[X]/(X^n - 1) \\ \mathbf{a} \stackrel{\text{def}}{=} (a_0, \ldots, a_{n-1}) & \longmapsto & \mathbf{a}(X) \stackrel{\text{def}}{=} \sum_{i=0}^{n-1} a_i X^i \end{array} \right. ,$$

which can be canonically made into an isometry by transporting the Hamming metric from $\mathbb{F}_q^n$ onto $\mathbb{F}_q[X]/(X^n - 1)$. With this polynomial representation, it is easy to see that the cyclic shift σ on $\mathbb{F}_q^n$ actually corresponds to multiplication by X in $\mathbb{F}_q[X]/(X^n - 1)$. In particular, cyclic codes are in one-to-one correspondence with ideals of $\mathbb{F}_q[X]/(X^n - 1)$, themselves in one-to-one correspondence with divisors of $X^n - 1$ in $\mathbb{F}_q[X]$. As rich as it is, this structure is in reality not suitable for cryptographic applications. Indeed, the last remark implies that the number of cyclic codes is way too small to be useful in a cryptographic context. Instead, it turns out that the correct notion is a slight generalisation known as *quasi-cyclic codes*.

Quasi-Cyclic Codes. Let ℓ be a positive integer, and consider codes of length ℓn, multiple of ℓ. Consider the following application known as the *ℓ-quasi-cyclic shift* $\sigma_\ell : \mathbb{F}_q^{\ell n} \to \mathbb{F}_q^{\ell n}$ the application which applies σ block-wise on blocks of size n:

$$\sigma_\ell : \left\{ \begin{array}{ccc} \mathbb{F}_q^{\ell n} & \longrightarrow & \mathbb{F}_q^{\ell n} \\ (\mathbf{x}_0 \mid \cdots \mid \mathbf{x}_{\ell-1}) & \longmapsto & (\sigma(\mathbf{x}_0) \mid \cdots \mid \sigma(\mathbf{x}_{\ell-1})). \end{array} \right.$$

Similarly to cyclic codes in the previous paragraph, a linear code $\mathscr{C} \subset \mathbb{F}_q^{\ell n}$ is said to be *ℓ-quasi-cyclic* or *quasi-cyclic of index ℓ* (or even simply *quasi-cyclic* when the context is clear) if it is

stable under the action of σ_ℓ. A visual representation of this shift is given in Figure 1.5.

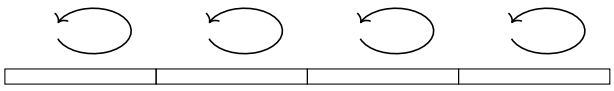


Figure 1.5: Illustration of the quasi-cyclic shift

Quasi-cyclic codes also benefit from a polynomial representation, a vector of length ℓn being represented as a collection of ℓ elements of $\mathbb{F}_q[X]/(X^n-1)$, and the action of σ_ℓ simply corresponds to the multiplication by X on each block. In other words, quasi-cyclic codes can be regarded as $\mathbb{F}_q[X]/(X^n-1)$-submodules of $\left(\mathbb{F}_q[X]/(X^n-1)\right)^\ell$.

Remark 1.34. *In Chapter 7 we consider a generalisation of quasi-cyclic codes known as* quasi-abelian codes. *More precisely, the polynomial ring* $\mathbb{F}_q[X]/(X^n-1)$ *can actually be understood as the* group algebra $\mathbb{F}_q\left[\mathbb{Z}/n\mathbb{Z}\right]$, *i.e. the free algebra generated by the cyclic group with* n *elements, endowed with the convolution. With this formalism, an* ℓ*-quasi-cyclic code is an* $\mathbb{F}_q\left[\mathbb{Z}/n\mathbb{Z}\right]$*-submodule of* $\left(\mathbb{F}_q\left[\mathbb{Z}/n\mathbb{Z}\right]\right)^\ell$. *The aforementioned quasi-abelian codes corresponds to using more general abelian groups instead of only cyclic groups.*

It turns out that quasi-cyclic codes are much more suitable for cryptographic applications than cyclic codes. One reason to explain this is that there are much more quasi-cyclic than cyclic codes. More precisely, a quasi-cyclic code will have a generator matrix formed out by multiple *circulant matrices* of the form

$$\mathbf{rot}(\mathbf{a}) \stackrel{\text{def}}{=} \begin{pmatrix} a_0 & a_1 & \dots & \dots & a_{n-1} \\ a_{n-1} & a_0 & \dots & \dots & a_{n-2} \\ \vdots & \ddots & \ddots & & \vdots \\ \vdots & & \ddots & \ddots & \vdots \\ a_1 & a_2 & \dots & a_{n-1} & a_0 \end{pmatrix} \in \mathbb{F}_q^{n\times n},$$

that is to say matrices whose rows are the cyclic shifts of a given vector $\mathbf{a} \stackrel{\text{def}}{=} (a_0,\dots,a_{n-1}) \in \mathbb{F}_q^n$. In other words, the generator matrix of a quasi-cyclic code will have the following general shape

$$\begin{pmatrix} \mathbf{rot}(\mathbf{a}^{(1,1)}) & \mathbf{rot}(\mathbf{a}^{(1,2)}) & \cdots & \mathbf{rot}(\mathbf{a}^{(1,\ell)}) \\ \vdots & \ddots & \ddots & \vdots \\ \mathbf{rot}(\mathbf{a}^{(k,1)}) & \mathbf{rot}(\mathbf{a}^{(k,2)}) & \cdots & \mathbf{rot}(\mathbf{a}^{(k,\ell)}) \end{pmatrix} \in \mathbb{F}_q^{nk\times n\ell},$$

for some integer k. The dimension of the code generated by such a matrix is likely to be kn, and this will be enforced in cryptographic applications.

For BIKE (see Section 1.3.2.2) and HQC (see Section 1.3.3), we will consider the particular case of *double circulant* codes whose generator matrices are formed by a single row of two circulant matrices (*i.e.* $k=1, \ell=2$ with the previous notations). In other words, we are mostly concerned with $[2n,n]$-quasi-cyclic codes.

Example 1.35. *The following matrix is a generator matrix for a binary double-circulant code*

$$\mathbf{G} \stackrel{def}{=} \left(\begin{array}{ccc|ccc} 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 \end{array}\right).$$

Polynomial representation. The representation of elements of $\mathbb{F}_q^n$ as polynomials in the quotient ring $\mathbb{F}_q[X]/(X^n-1)$ has another interesting property. Indeed, a simple computation shows that for two vectors $\mathbf{a}, \mathbf{b} \in \mathbb{F}_q^n$, the matrix-vector product $\mathbf{b} \cdot \mathbf{rot}(\mathbf{a})$ actually corresponds to the product of polynomials in $\mathbb{F}_q[X]/(X^n-1)$:

$$\mathbf{b} \cdot \mathbf{rot}(\mathbf{a}) = \mathbf{b}(X) \cdot \mathbf{a}(X) = \mathbf{a} \cdot \mathbf{rot}(\mathbf{b}),$$

where we identified a vector $\mathbf{c}$ of length n with its polynomial representation $\Phi(\mathbf{c}) \stackrel{\text{def}}{=} \mathbf{c}(X)$.

In particular, in the generator matrix of an ℓ-quasi-cyclic code, each circulant block $\mathbf{rot}(\mathbf{a}) \in \mathbb{F}_q^{n \times n}$ can be represented by the polynomial $\mathbf{a}(X) \in \mathbb{F}_q[X]/(X^n-1)$. In other words, the generator matrix of an $[\ell n, kn]$-quasi-cyclic code can be represented as a $k \times \ell$ matrix with entries in $\mathbb{F}_q[X]/(X^n-1)$.

Example 1.36. *Let us continue with Example 1.35. The polynomial representation of the matrix* $\mathbf{G}$ *with respect to the ring* $\mathbb{F}_2[X]/(X^3-1)$ *is nothing but*

$$\mathbf{G}(X) \stackrel{def}{=} \left(\begin{array}{c|c} 1 & X+1 \end{array}\right).$$

Duality and Parity-check matrices. In code-based cryptography it is sometimes more convenient to present the Decoding Problem in terms of syndrome and parity-check matrices. It is well-known that the *dual* of a quasi-cyclic code is still quasi-cyclic. Therefore, an $[n\ell, nk]$-quasi-cyclic code will have a parity-check matrix formed by $(\ell - k) \times \ell$ circulant blocks. In particular, it also benefits from a polynomial representation. In Chapter 7 we prove the same result for general quasi-abelian codes (Proposition 7.11).

Example 1.37. *It is readily seen that the following double-circulant matrix is a parity-check matrix for the code from Example 1.35:*

$$\mathbf{H} \stackrel{def}{=} \left(\begin{array}{ccc|ccc} 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \end{array}\right).$$

and its polynomial representation is

$$\mathbf{H}(X) \stackrel{def}{=} \left(\begin{array}{c|c} X^2+1 & 1 \end{array}\right).$$

1.3.1.2 Structured Variants of the Decoding Problems.

Random quasi-cyclic codes. Similarly to the unstructured situation, in the rest of this manuscript, a random quasi-cyclic code will simply be a code generated by a matrix $\mathbf{G}$ formed by $k \times \ell$ random

$n \times n$ circulant matrices. Such a random matrix is therefore simply represented by $k \times \ell$ polynomials in $\mathbb{F}_q[X]/(X^n-1)$, picked uniformly at random. Equivalently, we can define a random quasi-cyclic codes by means of a parity-check matrix formed out by random circulant blocks.

In a cryptographic context, it will also be more convenient to ensure that the involved matrices be full-rank. This can be done canonically by restricting the definitions of our problems with matrices in *systematic forms*. With this model, a random quasi-cyclic code will be nothing else than a code admitting a parity-check matrix having a polynomial representation of the form

$$\mathbf{H} = \left(\begin{array}{c|c} \mathbf{I}_{\ell-k} & \mathbf{A} \end{array} \right)$$

where $\mathbf{A}$ is a uniformly random matrix with coefficients in $\mathbb{F}_q[X]/(X^n-1)$.

Structured variants of the Decoding Problems. Let $\mathbf{G}$ be the generator matrix of a random quasi-cyclic code of index ℓ. For simplicity, let us first assume that $\mathbf{G}$ has only one row of circulant matrices, *i.e.* it is of the form

$$\mathbf{G} = \begin{pmatrix} \mathbf{rot}(\mathbf{a}^{(1)}) & \dots & \mathbf{rot}(\mathbf{a}^{(\ell)}) \end{pmatrix} \in \mathbb{F}_q^{n \times n\ell}$$

for some uniformly random vectors $\mathbf{a}^{(i)} \in \mathbb{F}_q^n$. Let $\mathbf{y} = \mathbf{mG} + \mathbf{e}$ be a noisy codeword, for some vector $\mathbf{m} \in \mathbb{F}_q^n$ and noise term $\mathbf{e} \stackrel{\text{def}}{=} (\mathbf{e}^{(1)}, \dots, \mathbf{e}^{(\ell)}) \in \mathbb{F}_q^{n\ell}$. In other words,

$$\mathbf{y} = \begin{pmatrix} \mathbf{m} \cdot \mathbf{rot}(\mathbf{a}^{(1)}) + \mathbf{e}^{(1)}, & \dots, & \mathbf{m} \cdot \mathbf{rot}(\mathbf{a}^{(\ell)}) + \mathbf{e}^{(\ell)} \end{pmatrix} \in \mathbb{F}_q^{n\ell}$$

and its polynomial representation $\Phi(\mathbf{y})$ is the collection of ℓ corrupted polynomials

$$\left\{ \begin{array}{c} \mathbf{m}(X) \cdot \mathbf{a}^{(1)}(X) + \mathbf{e}^{(1)}(X) \\ \vdots \\ \mathbf{m}(X) \cdot \mathbf{a}^{(\ell)}(X) + \mathbf{e}^{(\ell)}(X) \end{array} \right. \in \mathbb{F}_q[X]/(X^n-1),$$

where $\mathbf{m}(X) \stackrel{\text{def}}{=} \Phi(\mathbf{m})$. When dealing with quasi-cyclic codes, we usually consider the $\mathbf{e}^{(i)}$ to be independent but identically distributed. This error model is sometimes referred to as *regular errors*. This will always be assumed in the sequel, unless otherwise specified.

With this formalism, the Decoding Problem (1.10) restricted to the class of ℓ-quasi-cyclic codes can be formulated as follows. Let ψ be a probability distribution over $\mathbb{F}_q[X]/(X^n-1)$.

Problem 1.38 (QC-DP(ℓ, ψ), search version)

Let $\mathcal{R} \stackrel{\text{def}}{=} \mathbb{F}_q[X]/(X^n-1)$, and let $\mathbf{m} \in \mathcal{R}$ be *fixed*.

Data. ℓ samples $\left\{ (\mathbf{a}^{(i)}, \mathbf{y}^{(i)}) \right\}_{1 \leqslant i \leqslant \ell}$, where $\mathbf{a}^{(i)} \leftarrow \mathcal{R}$ and $\mathbf{y}_i \stackrel{\text{def}}{=} \mathbf{a}^{(i)}\mathbf{m} + \mathbf{e}^{(i)} \in \mathcal{R}$ with $\mathbf{e}^{(i)} \leftarrow \psi$.

Goal. Recover $\mathbf{m}$.

In general, ψ will be a distribution such that $\mathbb{E}_{\mathbf{x} \leftarrow \psi}(|\mathbf{x}|) = t$ for some integer parameter

$t \in \{1,\dots,n\}$, where the Hamming weight on $\mathcal{R}$ is the number of non-zero coefficients. This encompasses for instance the uniform distribution over polynomials of fixed Hamming weight t, or when the coefficients of the error are independently distributed according to a q-ary Bernoulli random variables of success probability t/n.

A natural way to try and solve QC-DP is to apply any generic decoding algorithm on the input. Therefore, we may wonder how easy it is to solve QC-DP compared to DP. It turns out that even after more than 50 years of extensive research, no algorithm is known to take into account the quasi-cyclic structure in order to significantly speed up the decoding process. More precisely, to this day the best approach in general is known as *Decoding One Out of Many* (DOOM) [Sen11]. However, it merely allows a $\sqrt{n}$ speed-up. This will be discussed in Section 1.3.4.1. In other words, Problem 1.38 is widely believed to be roughly as hard as the generic decoding problem in practice.[xi]

This is particularly interesting in a cryptographic context, since describing a random quasi-cyclic code of block-size n only requires the first row of each circulant block (of the generator or parity-check matrix). This is equivalent to giving the coefficients of its polynomial representation. In particular, it allows to save a factor n on the size of the public key: a random quasi-cyclic code can be represented with only $k \times \ell n$ elements of $\mathbb{F}_q$ instead of $kn \times \ell n$, while on the other hand keeping the same security level.

Obviously, Problem 1.38 also admits a decisional version, which is the quasi-cyclic version of DDP (Problem 1.27).

Problem 1.39 (QC-DP(ℓ,ψ), decisional version)

Let n be an integer, and set $\mathcal{R} \stackrel{\text{def}}{=} \mathbb{F}_q[X]/(X^n-1)$. Let $\mathbf{m}$ be drawn uniformly at random in $\mathcal{R}$ and consider the following two distributions

- $\mathcal{D}_0 : (\mathbf{a}, \mathbf{y}^{\text{unif}})$ uniformly distributed over $\mathcal{R}^2$,
- $\mathcal{D}_1 : (\mathbf{a}, \mathbf{a}\cdot\mathbf{m}+\mathbf{e})$ where $\mathbf{a} \leftarrow \mathcal{R}$, and $\mathbf{e} \leftarrow \psi$.

Given ℓ samples of the distribution $\mathcal{D}_\beta$ where $\beta \leftarrow \{0,1\}$ is a uniformly random bit, the goal is to recover β.

Remark 1.40. *It may also be interesting to consider the above problem in an LPN regime, where the number of samples ℓ is* a priori *unbounded. This structured version is sometimes referred to as* ring-*LPN in the literature (with respect to the ring $\mathbb{F}_q[X]/(X^n-1)$) ([HKLP+12; DP12]).*

Obviously, both structured variants of the Decoding Problems can equivalently be stated in terms of syndromes and parity-check matrices. Let $\mathbf{H}$ be a parity-check matrix of an ℓ quasi-cyclic code. For simplicity, consider the double-circulant situation in *systematic form*, *i.e.*

$$\mathbf{H} = (1 \mid \mathbf{h})$$

where $\mathbf{h} \leftarrow \mathbb{F}_q[X]/(X^n-1)$ is a uniformly random element. A syndrome of $\mathbf{s}^\top \stackrel{\text{def}}{=} \mathbf{H}\mathbf{e}^\top$ where

[xi] Some changes need to be made when giving a concrete set of parameters for cryptosystems, but the exponent describing the security parameter will be the same.

$\mathbf{e} = (\mathbf{e}^{(1)}, \mathbf{e}^{(2)})$ is a *regular* error vector,[xii] will then have a polynomial representation of the form

$$\mathbf{s}(X) = \mathbf{e}^{(1)}(X) + \mathbf{h}(X) \cdot \mathbf{e}^{(2)}(X) \in \mathbb{F}_q[X]/(X^n - 1)$$

where $\mathbf{e}^{(i)}$ are independent and identically distributed according to some distribution ψ over $\mathbb{F}_q[X]/(X^n - 1)$. The goal of the Decoding Problems will then be to recover the errors $\mathbf{e}^{(i)}$, or to distinguish this syndrome from a uniform element of $\mathbb{F}_q[X]/(X^n - 1)$, given the knowledge of $\mathbf{h}$.

Remark 1.41. *Interestingly enough, the polynomial representation of this syndrome is very similar to one sample of the quasi-cyclic distribution (i.e. distribution $\mathcal{D}_1$ of Problem 1.39), with the only difference being the fact that the secret (which can now be identified to $\mathbf{e}^{(2)}$) has the same distribution ψ as the error $\mathbf{e}^{(1)}$, instead of being uniformly distributed. This has sometimes been referred to as a* normal *Ring-LPN distribution in the literature.*[xiii]

A note on the systematic form. In the decisional version, we are asked to distinguish between a sample of the form $\mathbf{e}^{(1)} + \mathbf{h}\mathbf{e}^{(2)}$ and a uniform element of $\mathbb{F}_q[X]/(X^n - 1)$. Now, assume that we did not restrict ourselves to parity-check matrices in systematic form. In the general case, a (double-circulant) random parity-check matrix will therefore have a polynomial representation of the form

$$\mathbf{H} \stackrel{\text{def}}{=} \left(\begin{array}{c|c} \mathbf{h}^{(1)} & \mathbf{h}^{(2)} \end{array} \right)$$

and the syndrome $\mathbf{s}^\top \stackrel{\text{def}}{=} \mathbf{H}\mathbf{e}^\top$ will have the following representation

$$\mathbf{s}(X) = \mathbf{h}^{(1)}\mathbf{e}^{(1)} + \mathbf{h}^{(2)}\mathbf{e}^{(2)} \in \mathbb{F}_q[X]/(X^n - 1).$$

In the decisional version of the Decoding Problem, we would therefore be asked to distinguish a sample of the form

$$(\mathbf{h}^{(1)}, \mathbf{h}^{(2)}, \mathbf{h}^{(1)}\mathbf{e}^{(1)} + \mathbf{h}^{(2)}\mathbf{e}^{(2)})$$

where $\mathbf{h}^{(i)}$ are uniformly distributed in $\mathbb{F}_q[X]/(X^n - 1)$; from a sample of the form

$$(\mathbf{a}^{(1)}, \mathbf{a}^{(2)}, \mathbf{y}^{\text{unif}})$$

where all the components are uniformly distributed in $\mathbb{F}_q[X]/(X^n - 1)$. *A priori*, there seems to be little difference between the systematic version and the general case. However, there is a little caveat here. Indeed, notice that the syndrome $\mathbf{s}(X)$ actually belongs to the *ideal* of $\mathbb{F}_q[X]/(X^n - 1)$ generated by $(\mathbf{h}^{(1)}, \mathbf{h}^{(2)})$, *i.e.* the ideal generated by their gcd. In general, they are likely to be coprime, and therefore this ideal will be the full ring. However, it may happen that this ideal is not large enough, which may induce a bias in the distribution.

This discussion will be all the more important in Chapter 7 when we will consider the generalisation to *quasi-abelian codes*.

A "module" version. Finally, let us remove the condition "$k = 1$" which we merely put for simplicity. That is to say, let us consider quasi-cyclic codes generated by random matrices

[xii] Recall that a vector $\mathbf{e} \in \mathbb{F}_q^{\ell n}$ is said to be *t-regular* when it can be split into ℓ vectors of length n and Hamming weight t.

[xiii] With only one sample.

formed by *more than one* row of circulant blocks:

$$\mathbf{G} = \begin{pmatrix} \mathbf{rot}(\mathbf{a}^{(1,1)}) & \cdots & \mathbf{rot}(\mathbf{a}^{(1,\ell)}) \\ \vdots & & \vdots \\ \mathbf{rot}(\mathbf{a}^{(k,1)}) & \cdots & \mathbf{rot}(\mathbf{a}^{(k,\ell)}) \end{pmatrix}.$$

In terms of coding theory, this merely means that we consider codes with a higher rate.

In this situation, a noisy codeword $\mathbf{mG}+\mathbf{e}$ will still be represented by ℓ "noisy elements" of $\mathbb{F}_q[X]/(X^n-1)$, but their shape will be a little be more complex. Indeed, the element $\mathbf{m}$ is of the form $(\mathbf{m}_1(X),\dots,\mathbf{m}_k(X)) \in \left(\mathbb{F}_q[X]/(X^n-1)\right)^k$, and each sample will now be of the form

$$\sum_{i=1}^{k} \mathbf{m}_i \cdot \mathbf{a}^{(i,j)} + \mathbf{e}_j = \langle \mathbf{m}, \mathbf{a}^{(\cdot,j)} \rangle + \mathbf{e}_j \in \mathbb{F}_q[X]/(X^n-1), \qquad \text{for } j = 1,\dots,\ell;$$

where the inner product

$$\langle \cdot,\cdot \rangle \colon \left(\mathbb{F}_q[X]/(X^n-1)\right)^k \times \left(\mathbb{F}_q[X]/(X^n-1)\right)^k \to \mathbb{F}_q[X]/(X^n-1)$$

is canonically defined, and $\mathbf{a}^{(\cdot,j)}$ denotes the j-th column of $\mathbf{G}$.

In terms of parity-check matrices and syndromes, increasing the code-rate corresponds to having more elements in each row of the parity-check matrix (in the polynomial representation):

$$\mathbf{H} \stackrel{\text{def}}{=} \left(\; 1 \;\middle|\; \mathbf{h}^{(1)} \;\middle|\; \cdots \;\middle|\; \mathbf{h}^{(\kappa)} \; \right),$$

and a syndrome of an error $\mathbf{e} \stackrel{\text{def}}{=} (\mathbf{e}^{(0)},\dots,\mathbf{e}^{(\kappa)})$ would then be of the form

$$\mathbf{s}(X) \stackrel{\text{def}}{=} \mathbf{e}^{(0)} + \sum_{i=1}^{\kappa} \mathbf{h}^{(i)}\mathbf{e}^{(i)} = \langle \mathbf{h}^{(\geqslant 1)}, \mathbf{e}^{(\geqslant 1)} \rangle + \mathbf{e}^{(0)} \in \mathbb{F}_q[X]/(X^n-1).$$

In the context of lattice-based cryptography, a similar construction known as *Module*-LWE has been introduced in [LS15]. This explains why this version of the Decoding Problem has sometimes been called *Module*-LPN in the literature (for example [BCGI+20b, Definition 3.2]).

1.3.2 Instantiating McEliece with Quasi-Cyclic Codes

For classical instantiations of McEliece, the size of the public key can be very large, reaching 1.3 MB for Classic McEliece (See Table 1.1), one of the three remaining proposals in round 4 of NIST competition. Since QC-DP is not much easier than DP, it would be very interesting to instantiate the McEliece framework (Section 1.1.5) with families of quasi-cyclic codes. It turns out that this idea was already present in the initial paper [Gab05] where Gaborit proposed to use some quasi-cyclic Goppa codes (see [Gab05, Section 3.4]). However, this idea was not pushed further at that time and he proposed instead to use quasi-cyclic subcodes of BCH codes. Nevertheless, even though the loss in message security is negligible when working with quasi-cyclic codes instead of generic linear codes, this additional structure might damage the security of the keys (*i.e.* distinguishing the considered family from random codes). In particular, the instantiation of [Gab05] was eventually broken a few years later [OTD10]. The essence of this

attack lies in the fact that the chosen family of quasi-cyclic codes providing the keys was not big enough. This suggests to find a *very large* family of quasi-cyclic codes.

In this section, we will present two instantiations which seem to resist cryptanalysis, namely with quasi-cyclic binary Goppa codes, and with quasi-cyclic MDPC codes (or QC-MDPC for short).

1.3.2.1 Big Quake: A Quasi-Cyclic Version of Classic McEliece

For instantiations of McEliece cryptosystem with alternant codes (such as Goppa codes), attacks on the keys are still much less efficient than attacks on the messages. Therefore, the loss in the key security might be affordable. However, this should be carefully analysed. In particular, in the first round of NIST competition, the submission DAGS [BBBC+17] proposed to use some structured variants of alternant codes.[xiv] However, their choice of parameters was completely broken by algebraic attacks on the keys (see [Bar18; BC18; BBCO19]).

Nevertheless, not all hope is lost, and it may still be possible to design very competitive instantiations of McEliece. In particular, another candidate called BIG QUAKE (for BInary Goppa QUAsi-cyclic Key Encapsulation) [CBBB+17] was submitted to NIST competition. As suggested by the name, they proposed to use quasi-cyclic binary Goppa code, and is therefore in some sense a quasi-cyclic version of Classic McEliece [ABCC+22] which is still competing in the fourth round. In Table 1.3 we represent their choice of parameters as well as the corresponding size of the public key. The last column represents the size of the public key for an instantiation of McEliece with the same parameters, but forgetting the quasi-cyclicity. A little warning though, the notations in this thesis are slightly different from the notations in the supporting documentation of BIG QUAKE [CBBB+17]. Here, n corresponds to the block-size while the index of quasi-cyclicity is denoted by ℓ, and k is the number of rows of independent $n \times n$ circulant blocks. In particular, the codes have length $N = \ell n$ and dimension $K = kn$.

Security level (bits)	N	K	n	ℓ	k	Public key size (Bytes)	
						BIG QUAKE	Without quasi-cyclicity
128	3,510	2,418	13	270	186	25,389	330,057
192	7,410	4,674	19	390	246	84,132	1,598,508
256	10,070	6,650	19	530	350	149,625	2,842,875

Table 1.3: Public key size of BIG QUAKE

In particular, BIG QUAKE offers public keys of size 10 times smaller than that of the corresponding security level of Classic McEliece. Nevertheless, BIG QUAKE did not pass beyond the first round, even though it was not broken (and still is not!).

However, very recent progress have been made in distinguishing Goppa codes from random linear codes [MT22; BMT23; CMT23]. If in the case of generic Goppa codes the gap between the key and messages security is still huge, it has been largely reduced. It is therefore natural to wonder if those recent approaches can lead to better attacks in the case of structured Goppa codes. All in all, the research potential is far from being exhausted.

[xiv] In reality, in DAGS they propose to use *quasi-dyadic* codes and not quasi-cyclic codes.

1.3.2.2 BIKE: an Instantiation with QC-MDPC Codes.

In order to achieve an acceptable security level, instantiating McEliece's framework with MDPC codes requires even bigger public keys, than with Goppa codes (see Table 1.2). In order for it to be practical, it is *necessary* to introduce structured variants. This led to the submission BIKE (for Bit-flIpping Key Encapsulation) [AABB+22a] to NIST competition, named after a decoder for MDPC codes. BIKE is still running in the fourth round of the competition, and is one of the strongest code-based candidates to standardisation. For efficiency concerns, BIKE is presented in the Niederreiter variant, using parity-check matrices of quasi-cyclic MDPC (QC-MDPC) codes of index 2.

Public parameters. We use the following notations.

- n is the size of the circulant blocks.
- $w = \theta(\sqrt{n})$ is the density of the code, *i.e.* the weight of each rows.
- $t = \theta(\sqrt{n})$ is the number of errors.

The presentation is given in the polynomial representation using the ring $\mathbb{F}_2[X]/(X^n - 1)$.

Key generation. The secret key is the parity-check matrix of a random QC-MDPC code of density w. In order to generate it, it suffices to pick at random two polynomials $\mathbf{h}_0, \mathbf{h}_1 \in \mathbb{F}_2[X]/(X^n - 1)$, both having weight $w/2$. They form the secret key.

On the other hand, the public key is given by a systematic form of this parity-check matrix. In polynomial representation, if

$$\mathbf{H} = (\mathbf{h}_0, \mathbf{h}_1),$$

then the corresponding systematic form is given by

$$(1, \mathbf{h}_1 \mathbf{h}_0^{-1}).$$

In particular, it is necessary to ensure that $\mathbf{h}_0$ be invertible. It turns out that we have a very easy criterion when the block-length n is carefully chosen. Indeed, when n is a prime, which is also a primitive root modulo 2 (*i.e.* such that 2 generates $\mathbb{F}_n^\times$), then $X^n - 1$ has only two factors $X - 1$ and $\Phi(X) \stackrel{\text{def}}{=} 1 + X + \cdots + X^{n-1}$. In particular,

$$\mathbb{F}_2[X]/(X^n - 1) \simeq \mathbb{F}_2 \times \mathbb{F}_2[X]/(\Phi(X)) \simeq \mathbb{F}_2 \times \mathbb{F}_{2^{n-1}},$$

and it is readily seen that the invertible elements are the odd-weight polynomials. Therefore, it suffices to choose w even, such that $w/2$ is odd.

Wrapping up:

- *Secret key:* $\mathbf{h}_0, \mathbf{h}_1$, two elements of $\mathbb{F}_2[X]/(X^n - 1)$ with Hamming weight $w/2$.
- *Public key:* $\mathbf{h} \stackrel{\text{def}}{=} \mathbf{h}_1 \mathbf{h}_0^{-1} \in \mathbb{F}_2[X]/(X^n - 1)$.

Encryption. A plaintext is an error $\mathbf{e} \stackrel{\text{def}}{=} (\mathbf{e}_0, \mathbf{e}_1)$ such that $\mathbf{e}_i$ is an element of $\mathbb{F}_2[X]/(X^n - 1)$ of weight t.

The encryption is simply a syndrome with respect to the public parity-check matrix:

$$\text{Enc}((\mathbf{e}_0, \mathbf{e}_1)) \stackrel{\text{def}}{=} \mathbf{e}_0 + \mathbf{e}_1\mathbf{h}.$$

Decryption. Upon receiving a ciphertext $\mathbf{y} \stackrel{\text{def}}{=} Enc((\mathbf{e}_0, \mathbf{e}_1))$, it suffices to multiply by h_0 to get

$$\mathbf{s} \stackrel{\text{def}}{=} \mathbf{y}\mathbf{h}_0 = \mathbf{e}_0\mathbf{h}_0 + \mathbf{e}_1\mathbf{h}_1,$$

which is nothing but the syndrome of $(\mathbf{e}_0, \mathbf{e}_1)$ with respect to the (secret) parity-check matrix with moderate density. Therefore, $\mathbf{s}$ can be decoded using known MDPC-decoders to recover the plaintext.

Remark 1.42. *BIKE is somewhat in the spirit of* NTRU *encryption scheme [HPS98], whose security rests on hard lattice problems [SS11; PS21a; FPS22].*

On the security. As any McEliece-like cryptosystem, BIKE relies on two hypothesis:

(i) The public matrix is indistinguishable from a random matrix.

(ii) The Decoding Problem at the targeted distance is hard.

Under the polynomial representation, the first assumption rewrites into the hardness of distinguishing $\mathbf{h} \stackrel{\text{def}}{=} \mathbf{h}_1\mathbf{h}_1^{-1}$ from a random element of $\mathbb{F}_2[X]/(X^n - 1)$. It seems to be a bit *ad-hoc*. However, a full key recovery still needs to find small polynomials $\mathbf{h}'_0, \mathbf{h}'_1$ in the code generated by $\mathbf{H}$ such that $\mathbf{h}'_1\mathbf{h}'^{-1}_0 = \mathbf{h} = \mathbf{h}_1\mathbf{h}_0^{-1}$. In particular, the best known attacks are the same as for the Decoding Problem, namely the problem underlying the message security.

Remark 1.43. *Under the assumption that the* decisional *version of QC-DP (Problem 1.39) is hard, then the ciphertexts are indistinguishable from random elements of* $\mathbb{F}_2[X]/(X^n - 1)$.

Impact of the size of the public key. Table 1.4 summarises the parameter sets for the three security levels submitted to NIST competition. They are obtained from the official submission package [AABB+22a]. In particular, the additional quasi-cyclic structure allows to have public keys way smaller than traditional McEliece-based cryptosystems. We use the same notations as in Table 1.3. Note that BIKE makes use of double-circulant matrices, therefore $\ell = 2, k = 1$. In particular, $K = n = \frac{N}{2}$.

Security level (bits)	$N = 2n$	$K = n$	ℓ	k	Public key size (Bytes)	
					BIKE	Without quasi-cyclicity
128	24,646	12,323	2	1	1,541	18,982,041
192	49,318	24,659	2	1	3,083	76,007,273
256	81,946	40,973	2	1	5,122	209,848,341

Table 1.4: Public key size of BIKE

A note on the efficiency. It has to be noted that without the quasi-cyclicity structure, BIKE would involve huge matrices, and in particular the operations would be very inefficient. On the other hand, with the polynomial representation of quasi-cyclic codes, one only has to deal with polynomials in $\mathbb{F}_2[X]/(X^n-1)$ where n is large but very manageable. In particular, the arithmetic of $\mathbb{F}_2[X]/(X^n-1)$ can be heavily optimised to yield very good performances. The BIKE submission team even provide an optimised hardware implementation: https://github.com/Chair-for-Security-Engineering/BIKE.

1.3.3 Noisy Diffie-Hellman: HQC Cryptosystem.

In Section 1.2.4 we recalled an encryption scheme whose security truly relies on the hardness of the Decoding Problem. More precisely, it relied on the hardness of the Decisional version, but thanks to the search-to-decision reduction from [FS96] and recalled in Section 1.2.3 we know that they both are equivalent. Unfortunately, Alekhnovich cryptosystem also involves huge public keys. In order to mitigate that, HQC cryptosystem was proposed in [ABDG+16] and later submitted to the NIST competition, where it is still competing in the fourth round [AABB+22b]. It is very reminiscent of encryption schemes based on structured variants of LWE, such as Kyber [ABDK+21].

The intuition behind HQC is a "noisy version" of Diffie-Hellman key exchange protocol, where two participants, say Alice and Bob, agree on a secret element, but which is corrupted by some random noise. In general, it might be useless, since the noise can *a priori* be different for the two participants. This is where error-correcting codes come to the rescue. More precisely, Alice and Bob will agree on a common *codeword* of some *public code* which benefits from an efficient decoding algorithm. At the end of the protocol, they would get two different noisy versions of the codeword, but if the noise level is below the decoding radius, it can easily be removed.

The protocol runs in two phases:

(i) Setup phase:

- Alice and Bob first agree on a random quasi-cyclic code whose parity check matrix $\mathbf{H}$ is of the form $(1 \mid \mathbf{h})$ where $\mathbf{h}$ is a uniformly random element of $\mathbb{F}_q[X]/(X^n-1)$.
- Then, Alice computes a syndrome $\mathbf{s}_A \stackrel{\text{def}}{=} \mathbf{a} + \mathbf{h}\boldsymbol{\alpha}$, keeping secret $(\mathbf{a}, \boldsymbol{\alpha})$.
- Bob does the same thing and compute $\mathbf{s}_B \stackrel{\text{def}}{=} \mathbf{b} + \mathbf{h}\boldsymbol{\beta}$, keeping $(\mathbf{b}, \boldsymbol{\beta})$ for himself.
- They exchange their syndromes.
- Finally, Alice computes
$$\sigma_A \stackrel{\text{def}}{=} \boldsymbol{\alpha} \cdot \mathbf{s}_B = \boldsymbol{\alpha}\mathbf{b} + \mathbf{h}\boldsymbol{\alpha}\boldsymbol{\beta},$$
and Bob computes
$$\sigma_B \stackrel{\text{def}}{=} \boldsymbol{\beta} \cdot \mathbf{s}_A = \boldsymbol{\beta}\mathbf{a} + \mathbf{h}\boldsymbol{\beta}\boldsymbol{\alpha}.$$

At the end of the setup phase, Alice and Bob both know a corrupted version of $\mathbf{h}(\boldsymbol{\alpha}\boldsymbol{\beta})$.

(ii) Exchanging a codeword:

- Alice picks a codeword $\mathbf{c}$ in the public code, uniformly at random.
- She sends to Bob the vector
$$\mathbf{z} \stackrel{\text{def}}{=} \mathbf{c} + \sigma_A.$$

- Finally, Bob computes

$$\mathbf{z} - \sigma_B = \mathbf{c} + (\boldsymbol{\alpha}\mathbf{b} + \mathbf{h}\boldsymbol{\alpha}\boldsymbol{\beta}) - (\mathbf{a}\boldsymbol{\beta} + \mathbf{h}\boldsymbol{\alpha}\beta) = \mathbf{c} + \underbrace{(\boldsymbol{\alpha}\mathbf{b} - \mathbf{a}\boldsymbol{\beta})}_{\stackrel{\text{def}}{=} \mathbf{e}'},$$

  which he can decode in the public code, provided that the Hamming weight of $\mathbf{e}'$ is below the decoding radius; recovering the codeword $\mathbf{c}$.

It has to be noted that at every step of the protocol, Alice and Bob only exchange random elements of $\mathbb{F}_q[X]/(X^n - 1)$, under the assumption that the decisional version of QC-DP (Problem 1.39) is hard. Moreover, the outer code is completely public and its structure does not need to be hidden. In particular, it can well be a Reed-Solomon code for example. HQC cryptosystem is a translation of the above protocol into an encryption scheme, which we quickly describe below for completeness.

Public parameters. We use the following notations. For the NIST submission, the ambient polynomial ring is $\mathbb{F}_2[X]/(X^n - 1)$.

- $\mathscr{C}$ is a public code of dimension k, represented by a generator matrix $\mathbf{G}$, with an efficient decoding algorithm up to some threshold Δ.
- n is the size of the circulant blocks.
- w is half the number of errors in the public key.
- t is a number of additional errors.

Similarly to BIKE, the random quasi-cyclic codes involved in HQC are random double-circulant codes in systematic form. However, they are not necessary MDPC.

Key generation. The public key is a pair $(\mathbf{H}, \mathbf{H}\mathbf{e}^\top)$ where $\mathbf{H}$ is a random double-circulant matrix of length $2n$ in systematic form, and $\mathbf{e}_A \stackrel{\text{def}}{=} (\mathbf{a}, \boldsymbol{\alpha})$ is a regular error of weight $2w$; while the secret key is $\mathbf{e}_A$.

More precisely, the key generation algorithm picks $\mathbf{h} \leftarrow \mathbb{F}_2[X]/(X^n - 1)$ uniformly at random, as well as $\mathbf{a}, \boldsymbol{\alpha}$ of weight w and sets

- *Secret key*: $\mathbf{a}, \boldsymbol{\alpha}$.
- *Public key*: $(\mathbf{h}, \mathbf{a} + \mathbf{h}\boldsymbol{\alpha})$.

Encryption. A plaintext is an element $\mathbf{m} \in \mathbb{F}_2^k$. The encryption proceeds as follows:

1. Encode the plaintext into the public code to get $\mathbf{m}\mathbf{G} \in \mathbb{F}_2^n$, represented as an element of $\mathbb{F}_2[X]/(X^n - 1)$.
2. Pick $(\mathbf{b}, \boldsymbol{\beta}, \mathbf{e})$ at random in $\mathbb{F}_2[X]/(X^n - 1)$ such that they all have weight t.

The ciphertext is then formed by two elements:

$$\text{Enc}(\mathbf{m}) \stackrel{\text{def}}{=} (\mathbf{b} + \mathbf{h}\boldsymbol{\beta}, \mathbf{m}\mathbf{G} + (\mathbf{a} + \mathbf{h}\boldsymbol{\alpha})\boldsymbol{\beta} + \mathbf{e}).$$

Decryption. In order to decrypt a ciphertext $(\mathbf{u}, \mathbf{v})$, use the secret key $\boldsymbol{\alpha}$ to compute

$$\begin{aligned}\mathbf{v} - \boldsymbol{\alpha}\mathbf{u} &= \mathbf{mG} + (\mathbf{a} + \mathbf{h}\boldsymbol{\alpha})\boldsymbol{\beta} + \mathbf{e} - \boldsymbol{\alpha}(\mathbf{b} + \mathbf{h}\boldsymbol{\beta}) \\ &= \mathbf{mG} + (\boldsymbol{\beta}\mathbf{a} - \boldsymbol{\alpha}\mathbf{b}) + \mathbf{e} \\ &= \mathbf{mG} + \mathbf{e}'.\end{aligned}$$

Finally, it suffices to apply the decoding algorithm for the code $\mathscr{C}$, as long as the weight of $\mathbf{e}'$ is below Δ. Since

$$\begin{aligned}|\mathbf{e}'| &= |\boldsymbol{\beta}\mathbf{a} - \boldsymbol{\alpha}\mathbf{b} + \mathbf{e}| \\ &\leqslant |\boldsymbol{\beta}| \cdot |\mathbf{a}| + |\boldsymbol{\alpha}| \cdot |\mathbf{b}| + |\mathbf{e}| \\ &\leqslant 2tw + t,\end{aligned}$$

it suffices to tune the parameters such that this upper bound is below Δ, but since the code is public and is not part of the security, this can easily be done.

A note on the security.

- Recovering a valid secret key from the public data is tantamount to decoding the public syndrome $\mathbf{a} + \mathbf{h}\boldsymbol{\alpha}$ in the random quasi-cyclic code of parity-check matrix $(1 \mid \mathbf{h})$. In other words, it corresponds to solving QC-DP at distance $2w$.
- Contrary to a direct adaptation of Alekhnovich approach, the decryption process does not involve distinguishing a noisy codeword from a uniform vector. However, under the hardness of decisional-QC-DP, then all the elements which are exchanged are pseudorandom.

Size of the public key. Table 1.5 summarises the parameter sets for the three security levels of HQC submitted to NIST competition. They are taken from the official submission documentation [AABB+22b], and we use the same notations as in Table 1.4.

Security level (bits)	$N = 2n$	$K = n$	ℓ	k	Public key size (Bytes)	
					HQC	Without quasi-cyclicity
128	35,338	17,669	2	1	2,249	39,024,195
192	71,702	35,851	2	1	4,522	160,661,775
256	115,274	57,637	2	1	7,245	415,252,971

Table 1.5: Public key size of HQC

1.3.4 Hardness of the Structured Variants of the Decoding Problem

1.3.4.1 Decoding One Out of Many (DOOM)

Despite more than half a century of research, except in certain regime of parameters, the additional structure does not seem to significantly improve known generic decoding algorithms. The best known approach, due to Sendrier [Sen11], is known as DOOM for Decoding One Out of Many.

More precisely, Sendrier was interested in the following problem: Given N instances of the Decoding Problem for the same code, and at the same distance t, the goal is to decode at least

one of them. It turns out that variants of Information Set Decoding algorithms allow to achieve a $\sqrt{N}$ speedup in the decoding.

This observation is particularly handy when working with quasi-cyclic codes of block length N. Indeed, from one noisy codeword $\mathbf{y} \stackrel{\text{def}}{=} \mathbf{c} + \mathbf{e}$ we can generate N new noisy codewords by iterating the quasi-cyclic shift, and a solution to one of them immediately yields a solution to the original problem. Therefore, the DOOM approach allows to decode an ℓ-quasi-cyclic code of length ℓn at distance t in time $O\left(\frac{2^{c \cdot t}}{\sqrt{n}}\right)$ for some constant c, instead of $O\left(2^{c \cdot t}\right)$ for a generic code of same length.

For the regime of parameters used in cryptography, this improvement affects the concrete security, and it is necessary to take it into account when deriving parameters, however it does not change the bit security of the scheme.

Remark 1.44. *We will consider again this* DOOM *approach with more details in Chapter 7 when dealing with more general* quasi-abelian codes.

1.3.4.2 Folding the Code

When dealing with codes endowed with the action of a large group G by permutation, another class of attacks need to be considered, namely *folding attacks*. More precisely, from such a code, it is possible to derive a smaller code by summing up the codewords which belong to the same G-orbit. This concept was used in [FOPP+16a; FOPP+16b; BC18] to break variants on McEliece cryptosystems. The interested reader can refer to [Bar18] for a presentation of such attacks.

However, folding the code may also be used to solve the Decoding Problem. More precisely, following [CT19][xv], let $\mathscr{C}$ be an $[\ell n, kn]$-code endowed with the free action by permutation of a group G of size n. For example, one may consider an ℓ-quasi-cyclic code, with the action of $G \stackrel{\text{def}}{=} \mathbb{Z}/n\mathbb{Z}$. In particular, the orbits of each position have same size n. Let $i_1, \ldots, i_\ell$ be a set of representatives for each orbit and define the *folding* operation to be the map

$$\pi_G : \begin{cases} \mathbb{F}_q^{\ell n} & \longrightarrow & \mathbb{F}_q^{\ell} \\ \mathbf{x} & \longmapsto & \left(\sum_{\sigma \in G} x_{\sigma(i_j)}\right)_{1 \leqslant i_j \leqslant \ell} \end{cases}.$$

Now, let $\mathbf{y} \stackrel{\text{def}}{=} \mathbf{c} + \mathbf{e}$ be a codeword of $\mathscr{C}$ corrupted by an error $\mathbf{e}$ of Hamming weight t. Then, applying the folding map π_G yields a new decoding problem

$$\pi_G(\mathbf{y}) = \pi_G(\mathbf{c}) + \pi_G(\mathbf{e}),$$

where $\pi_G(\mathbf{c})$ belongs to a smaller code $\pi_G(\mathscr{C})$ of length ℓ, and the Hamming weight of $\pi_G(\mathbf{e})$ cannot be bigger than that of $\mathbf{e}$. It can even be slightly smaller due to the presence of collisions. The key point is that since G acts on $\mathscr{C}$ by permutation, [CT19, Proposition 1] yields that in general

$$\dim \pi_G(\mathscr{C}) = \frac{\dim \mathscr{C}}{|G|} = k.$$

In particular, the folded code has the *same rate* as that of the original $\mathscr{C}$. In other words, starting from decoding a code of rate R at distance t, we get to a decoding problem of a code of rate R at distance t' possibly slightly smaller (precise bounds are given in [CT19, Proposition 2]). However, recall that the length of $\pi_G(\mathscr{C})$ is ℓ, *i.e.* the index of quasi-cyclicity, which can a

[xv] Although we keep our notations and not that of this article.

priori be very small compared to t or t'. In other words, the *absolute* error may decrease, but the *error rate* might explode. In particular, starting from a decoding problem with a unique solution might yield a decoding problem with many solutions. This might still be manageable given the size of the problem.

However, the work is far from over, and we still need to show how to use the solutions to the smaller decoding problems in order to recover the original solution. As explained in [CT19], it turns out that lifting the solutions back to $\mathbb{F}_q^{\ell n}$ allows to slightly reduce the number of equations in the linear system induced by the original problem. More precisely, using the folding can be interpreted as a new decoding problem of an $[\ell n, kn-\ell]$ code at distance t. Note that this new problem is in general unstructured, and it is not possible to continue the folding. Unfortunately, when ℓ is very small, for example in the case of BIKE and HQC we have $\ell = 2$ and $k = 1$, this does not really improve much on the original decoding problem.

Nevertheless, nothing forces us to use the whole group G, and the analysis can be refined by using some subgroup H of G. There is a caveat though: the smaller the subgroup H, the higher is the new number of errors t' in the folded decoding problem. Yet, [CT19] shows that this approach is enough to give an exponential improvement in some range of parameters ([CT19, Table 1]). In order to avoid this kind of attacks, n was chosen to be prime in BIG QUAKE, BIKE and HQC.

Remark 1.45. *We will come back to those folding attacks in Chapter 7 with more general structured codes, namely* quasi-abelian codes.

1.3.4.3 Hardness of the Decisional Version

Little is known regarding the decisional version of the decoding problem of structured codes. In particular, the landscape of theoretical reductions is pretty much empty: the search-to-decision reduction of [FS96] and recalled in Section 1.2.3 inherently works with generic random linear codes, and does not carry over to structured variants of the Decoding Problem such as QC-DP. In particular, the decisional version is not known to be equivalent to the search problem.

When no reduction exists, a cryptographer has to rely on cryptanalysis in order to assess the security. Yet, virtually all known approaches for solving the Decisional Decoding Problem (whether it be structured or unstructured), actually solve the search version. In particular, it is widely conjectured, and admitted, that the decisional version is not significantly easier than the related Decoding Problem. This state of affairs is not satisfying though, since we do not have any proof of this potential equivalence.

Remark 1.46. *In the world of euclidean lattices, which faces similar efficiency issues when considering unstructured variants of the cryptosystems, many reductions have been derived for algebraically structured variants of the problems (e.g. [LPR10] for Ring-LWE, [LS15] for Module-LWE, [RSW18] for Polynomial-LWE, [PRS17] for an attempt of giving a unifying framework). This observation is the starting point of Part II, and especially Chapter 4, where we manage to give the first search-to-decision reduction for some quasi-cyclic codes (but not for the choice of parameter made in BIKE or HQC).*

The Linear Test Framework. Faced with this, we may wonder if we are systematically obliged to go over all the literature on attacks against the decoding problem when introducing a new instantiation, where by instantiation we mean the restriction of DP on random codes restricted to a given family. For example, QC-DP is an instantiation of DP with random quasi-cyclic codes. Fortunately, it seems that a large class of attacks can actually be seen in light of a unified framework, namely the *linear test framework*, put forth in [BCGI+20a; CRR21]. More precisely,

their observation is the following: for almost all of the known attacks (including, but not limited to, all applications of Information Set Decoding algorithms [Pra62; Ste88; Dum91; BJMM12; MO15; BM17], or Statistical Decoding [Jab01; Ove06; DT17; CDMT22]), at the end of the day a distinguisher will basically try to observe a bias on a *linear* function of the sample, whose coefficients can depend arbitrarily on the input code.

This observation should be compared with the decryption process of Alekhnovich encryption scheme 1.2.4. Indeed, recall that the secret key is an error vector $\mathbf{e}_{\mathrm{sk}}$ of Hamming weight $\theta(\sqrt{n})$, and that decrypting consists in computing the inner product between the received word $\mathbf{y}$ and $\mathbf{e}_{\mathrm{sk}}$:

- If we observe a bias towards 0, it means that $\mathbf{y}$ should be a noisy codeword.
- On the other hand, $\mathbf{y}$ should be a random vector.

For another example, consider the case of Prange algorithm [Pra62], which was recalled in Section 1.1.4.1, and assume that the decoding problem is given as a parity-check matrix $\mathbf{H}$ and a vector $\mathbf{s}$ which may or may not be the syndrome of $\mathbf{H}$ with respect to a very small weight codeword $\mathbf{e}$, say sublinear in n. For example, the typical use case with quasi-cyclic codes would be $t \stackrel{\text{def}}{=} |\mathbf{e}| = O(\sqrt{n})$. For the sake of the explanation, let us assume that we work over the binary field $\mathbb{F}_2$, and that $\mathbf{s}$ *really* is a syndrome $\mathbf{s}^\top = \mathbf{H}\mathbf{e}^\top$. Now, recall that Prange algorithm consists in guessing an information set $\mathcal{I}$ which does not meet any error position, *i.e.* a subset of k positions such that $\mathbf{H}_{\mathcal{I}^c}$ is invertible, and such that $\mathcal{I} \cap \mathrm{Supp}(\mathbf{e}) = \emptyset$. Once $\mathcal{I}$ is guessed, the algorithm computes

$$\widetilde{\mathbf{s}} \stackrel{\text{def}}{=} (\mathbf{H}_{\mathcal{I}^c})^{-1} (\mathbf{s}_{I^c})^\top,$$

which should be equal to $\mathbf{e}_{\mathcal{I}^c}^\top$ if $\mathbf{s}$ is actually a syndrome. If $\mathcal{I}$ was a good guess, $\mathrm{Supp}(\mathbf{e}) \subset \mathcal{I}^c$, and in other words we should have

$$|\widetilde{\mathbf{s}}^\top| \stackrel{\text{def}}{=} |(\mathbf{H}_{\mathcal{I}^c})^{-1} (\mathbf{s}_{I^c})^\top| = t = O(\sqrt{n}).$$

In particular, using the same idea as in Alekhnovich, we could compute the inner product between $\widetilde{\mathbf{s}}$ and a vector $\widetilde{\mathbf{v}}$ of weight $O(\sqrt{n})$, which should be biased towards 0 if $\mathbf{s}$ was a syndrome, but should be a uniformly random bit if $\mathbf{s}$ was uniformly random.

All in all, this interpretation of Prange algorithm consists in observing a bias in the inner product

$$\widetilde{\mathbf{v}}\widetilde{\mathbf{s}}^\top = \left(\widetilde{\mathbf{v}}\mathbf{H}_{\mathcal{I}^c}^{-1}\right)\mathbf{s}^\top = \langle \left(\widetilde{\mathbf{v}}\mathbf{H}_{\mathcal{I}^c}^{-1}\right), \mathbf{s} \rangle,$$

which is indeed a linear function which only depends on $\mathbf{H}$ (and *not* on $\mathbf{s}$), applied on $\mathbf{s}$.

According to the analysis of [BCGI+20a; CRR21], the only known approaches which do not fit into the *linear test framework* are algebraic attacks. Nevertheless, such attacks do not seem to take a large toll on QC-DP.

This motivates the following abstract framework: an attack in the linear test framework proceeds in two phases:

(i) First, the adversary performs any computation on the input parity-check matrix (without any resource limitation), and then outputs some vector $\mathbf{v}$. The only limitation of the adversary is that it does not know the challenge $\mathbf{s}$.

(ii) Second, it estimates the bias in the inner product $\mathbf{v} \cdot \mathbf{s}^\top$.

Note that if $\mathbf{s}$ is a syndrome of $\mathbf{H}$ with respect to a small weight error $\mathbf{e}$, then $\mathbf{v}\mathbf{s}^\top = \langle \mathbf{v}\mathbf{H}, \mathbf{e} \rangle$, which is all the more biased towards 0 than $|\mathbf{v}\mathbf{H}|$ is small. On the other hand, if $\mathbf{s}$ is uniformly

distributed, then $\mathbf{vs}^\top$ is uniform no matter what ($\mathbf{x} \mapsto \mathbf{vx}^\top$ is linear and surjective when $\mathbf{v}$ is non zero). Conversely, the existence of a nonzero vector $\mathbf{v}$ such that $\mathbf{vH}$ has an unusually low weight yields a bias in the distribution $\mathbf{vs}^\top$ when $\mathbf{s}$ is the syndrome of a low weight error.

Based on this observation, an instantiation of the Decoding Problem with a class $\mathcal{F}$ of codes (*e.g.* quasi-cyclic codes) will resist *any* attack from the linear test framework, if and only if the *dual* of a code picked uniformly at random from $\mathcal{F}$ has a large minimum distance. This quantity is also known as *dual minimum distance*, or simply *dual distance*. This statement is made formal in [CRR21, Section 3].

Remark 1.47. *From Section 1.1.3.3, we know that a random linear code has a minimum distance reaching the Gilbert-Varshamov bound. In particular, the* dual *of a random code has a minimum distance which is* linear *in n with overwhelming probability, and DP is an example of instantiation which resists linear attacks. This was obviously expected given the existence of the search-to-decision reduction.*

Remark 1.48. *An important tool in lattice-based cryptography is the notion of* smoothing *introduced in [MR04]. It has recently found its way into the code-based setting in [DDRT23]: given a binary linear code $\mathscr{C}$ of length n, the goal of smoothing bounds is to give a lower bound on the amount of noise needed so that a noisy codeword $\mathbf{c}+\mathbf{e}$ (respectively a syndrome) is* statistically close *to the uniform distribution over $\mathbb{F}_2^n$ (respectively $\mathbb{F}_2^{n-k}$), where $\mathbf{c} \in \mathscr{C}$.*

When $\mathscr{C}$ is a random linear code, one can prove (see [Deb23, Proposition 2.5.1]) that it is enough to sample $\mathbf{e}$ uniformly at random in the Hamming sphere of radius t such that $t/n = \delta_{\mathrm{GV}}$ is the Gilbert-Varshamov distance (Definition 1.6). On the other hand, smoothing bounds ask the same question when the input code is fixed, *i.e.* in the worst-case situation, *which is a much more challenging task. It turns out that the bounds given in [DDRT23] are also in direct relation with the dual distance: the higher the dual distance, the better the smoothing bounds. In particular, this observation follows the same direction as the resistance to linear attacks.*

On the Minimum Distance of Random Quasi-Cyclic Codes. Let us conclude this section with a discussion on the typical behaviour of the dual distance of quasi-cyclic codes. Since the dual of an ℓ-quasi-cyclic code with block length n is still an ℓ-quasi-cyclic, with same block length (see Section 1.3.1.1), studying the dual distance is tantamount to study the actual minimum distance of quasi-cyclic codes.

It turns out that similarly to random linear codes, many classes of random quasi-cyclic codes also have a large minimum distance. In particular, [CPJ69] proved that when p is a prime such that 2 is primitive modulo p, then double circulant codes with block size p typically achieve the Gilbert-Varshamov bound. In [GZ06], Gaborit and Zémor even showed that such codes satisfied a logarithmic improvement on this bound.

Moreover, this choice of block length is also exactly the one used in BIKE [AABB+22a] and HQC [AABB+22b]. In particular, this proves that the instantiation of QC-DP with their parameters is resistant to the large class of linear attacks.

Part I

Rank-metric codes

Chapter 2

Yet another structure: $\mathbb{F}_{q^m}$-linear codes and the rank-metric

This Chapter is dedicated to a short introduction to error correcting codes endowed with this rank metric, and their usage in cryptography. In particular, in Section 2.3, we revisit a well-known decoder for a family of rank metric codes known as *Gabidulin codes*, which is one of the ingredients in the cryptanalysis of RAMESSES presented in Chapter 3. This decoder was presented at WCC 2022 [BC22], and the cryptanalysis was published in [BC21].

Outline of the current chapter

2.1 Motivations

For m, n two positive integers, denote by

$$\sigma : \left\{ \begin{array}{ccc} \mathbb{F}_q^n & \longrightarrow & \mathbb{F}_q^n \\ (x_0, \ldots, x_{n-1}) & \longmapsto & (x_{n-1}, x_0, \ldots, x_{n-2}) \end{array} \right.$$

the cyclic shift of length n and by $\sigma_m : \mathbb{F}_q^{mn} \to \mathbb{F}_q^{mn}$ the m-quasi-cyclic shift which applies σ block-wise on blocks of size n:

$$\sigma_m : \left\{ \begin{array}{ccc} \mathbb{F}_q^{mn} & \longrightarrow & \mathbb{F}_q^{mn} \\ (\mathbf{x}_0 \mid \cdots \mid \mathbf{x}_{m-1}) & \longmapsto & (\sigma(\mathbf{x}_0) \mid \cdots \mid \sigma(\mathbf{x}_{m-1})). \end{array} \right.$$

Recall from Section 1.3 that a code $\mathscr{C} \subset \mathbb{F}_q^{mn}$ stable under the action of σ_m is called an m-quasi-cyclic code of block-length n. When representing each block of length n by an element of $\mathbb{F}_q[X]/(X^n-1)$, the m-cyclic shift corresponds to the multiplication by X on each block. This representation permits to save a factor n in the representation of a quasi-cyclic code.

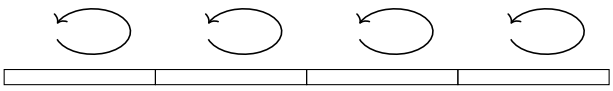


Figure 2.1: Illustration of the quasi-cyclic shift

In general, the index of quasi-cyclicity is often quite small compared to n. For example, it is only 2 for BIKE ([AABB+22a] and Section 1.3.2.2) or HQC ([AABB+22b] and Section 1.3.3). However, for large values of m, it might be interesting to consider another slicing of an element of $\mathbb{F}_q^{mn}$ into n blocks of size m, instead of m blocks of size n, and the idea is to compactly represent elements of $\mathbb{F}_q^m$. Let $\alpha \in \mathbb{F}_{q^m}$ be a primitive element of $\mathbb{F}_{q^m}$, so that $(1, \alpha, \dots, \alpha^{m-1})$ forms a basis of the extension $\mathbb{F}_{q^m}/\mathbb{F}_q$, and represent any element $(a_0, \dots, a_{m-1}) \in \mathbb{F}_q^m$ as the corresponding $a \stackrel{\text{def}}{=} \sum_{i=0}^{m-1} a_i \alpha^i \in \mathbb{F}_{q^m}$, such that a code $\mathscr{C} \subset \mathbb{F}_q^{mn}$ can be represented by a code $\widetilde{\mathscr{C}} \subset \mathbb{F}_{q^m}^n$, *a priori* non linear over $\mathbb{F}_{q^m}$.[i]

Analogously to the quasi-cyclic situation, a code $\mathscr{C} \subset \mathbb{F}_q^{mn}$ will be called α-cyclic if $\widetilde{\mathscr{C}}$ is stable under multiplication by α on each component:

$$\mathbf{c} = (c_0, \dots, c_{m-1}) \in \widetilde{\mathscr{C}} \Rightarrow \alpha \cdot \mathbf{c} \stackrel{\text{def}}{=} (\alpha c_0, \dots, \alpha c_m) \in \widetilde{\mathscr{C}}.$$

The code $\mathscr{C}$ being $\mathbb{F}_q$-linear entails that $\widetilde{\mathscr{C}}$ is also $\mathbb{F}_q$-linear. Therefore, we can immediately deduce that a code $\mathscr{C}$ is α-cyclic if and only if $\widetilde{\mathscr{C}}$ is stable under multiplication by *any* polynomial in α. Since we chose α to be a primitive element of $\mathbb{F}_{q^m}/\mathbb{F}_q$, we have that $\mathscr{C} \subset \mathbb{F}_q^{mn}$ is α-cyclic if and only if $\widetilde{\mathscr{C}}$ is $\mathbb{F}_{q^m}$-linear. In particular, $\mathscr{C}$ is α-cyclic if and only if it is β-cyclic for another primitive element β. Such a code will simply be called $\mathbb{F}_{q^m}$-linear, even if it is *a priori* only defined over $\mathbb{F}_q$.

On the other hand, contrary to the cyclic-shift, this action of $\mathbb{F}_{q^m}$ completely messes up the Hamming weights. In fact, the Hamming metric on $\widetilde{\mathscr{C}}$ has nothing to do with the Hamming metric on $\mathscr{C}$: if $\mathbf{c} \in \mathscr{C}$ and $\widetilde{\mathbf{c}}$ is the corresponding element of $\widetilde{\mathscr{C}}$, we only have the trivial bound $|\mathbf{c}| \leqslant m|\widetilde{\mathbf{c}}|$, which is far from being tight and even depends on the choice of α. Ideally, we would like to put a metric on $\mathbb{F}_{q^m}^n$ invariant under the action of $\mathbb{F}_{q^m}$. Moreover, since an α-cyclic code is in fact $\mathbb{F}_{q^m}$-linear, there is no incentive to restrict ourselves to power-bases, and this putative metric on $\mathbb{F}_{q^m}$ shall therefore be invariant under *any* change of basis.

In order to define this metric, it is more convenient to write an $\mathbb{F}_{q^m}$-linear code $\mathscr{C} \subset \mathbb{F}_q^{mn}$ as a subspace of matrices with m rows and n columns. The condition to be invariant under change of basis exactly means that this metric (on $\mathbb{F}_q^{m \times n}$) should be invariant under left multiplication by a non singular matrix (with coefficients in $\mathbb{F}_q$). In particular, this motivates to define the weight of a matrix $M \in \mathbb{F}_q^{m \times n}$ to be its *rank*, and the distance between two matrices shall be the rank of their difference. It is not hard to prove that this indeed induces a metric space structure on $\mathbb{F}_q^{m \times n}$, different from the Hamming metric, and which fulfils all of our requirements.

[i] It is $\mathbb{F}_q$-linear, though.

2.2 Generalities on the Rank Metric

Let us now formalise this motivation.

2.2.1 Codes Endowed with the Rank Metric

Let m, n be two positive integers. Let $\mathbb{F}_{q^m}/\mathbb{F}_q$ be a finite extension of $\mathbb{F}_q$, and let us fix a basis $\mathcal{B} \stackrel{\text{def}}{=} \{\beta_1, \ldots, \beta_m\}$. For any vector $\mathbf{x} \stackrel{\text{def}}{=} (x_1, \ldots, x_m) \in \mathbb{F}_{q^m}^n$, define its extension with respect to $\mathcal{B}$ to be the matrix

$$\mathrm{Ext}_{\mathcal{B}}(\mathbf{x}) \stackrel{\text{def}}{=} \begin{pmatrix} x_1^{(1)} & \cdots & x_n^{(1)} \\ \vdots & \ddots & \vdots \\ x_1^{(m)} & \cdots & x_n^{(m)} \end{pmatrix} \in \mathbb{F}_q^{m \times n} \tag{2.1}$$

where for all $i \in \{1, \ldots, m\}$

$$x_i \stackrel{\text{def}}{=} \sum_{j=1}^{m} x_i^{(j)} \beta_j$$

is the decomposition of x_i in the basis $\mathcal{B}$. With this extension map, we can associate to any $\mathbb{F}_{q^m}$-linear code $\mathscr{C} \subset \mathbb{F}_{q^m}^n$ of dimension k (over $\mathbb{F}_{q^m}$) the following vector space of $\mathbb{F}_q^{m \times n}$, with dimension km (over $\mathbb{F}_q$)

$$\mathrm{Ext}_{\mathcal{B}}(\mathscr{C}) \stackrel{\text{def}}{=} \{\mathrm{Ext}_{\mathcal{B}}(\mathbf{c}) \mid \mathbf{c} \in \mathscr{C}\} \subset \mathbb{F}_q^{m \times n},$$

which we can endow with the rank metric:

$$\forall \mathbf{X}, \mathbf{Y} \in \mathbb{F}_q^{m \times n}, \quad d(\mathbf{X}, \mathbf{Y}) \stackrel{\text{def}}{=} \mathrm{Rank}(\mathbf{X} - \mathbf{Y}).$$

Such a vector space is called a *matrix code*. Now, consider another basis $\mathcal{A} \stackrel{\text{def}}{=} \{\alpha_1, \ldots, \alpha_m\}$ of $\mathbb{F}_{q^m}/\mathbb{F}_q$. It is readily seen that

$$\forall \mathbf{x} \in \mathbb{F}_{q^m}^n, \quad \mathrm{Ext}_{\mathcal{A}}(\mathbf{x}) = \mathrm{Ext}_{\mathcal{A}}(\mathcal{B}) \cdot \mathrm{Ext}_{\mathcal{B}}(\mathbf{x}), \tag{2.2}$$

where $\mathrm{Ext}_{\mathcal{A}}(\mathcal{B}) \in \mathbb{F}_q^{m \times m}$ is the extension of the vector $(\beta_1, \ldots, \beta_m) \in \mathbb{F}_{q^m}^m$ with respect to $\mathcal{A}$. In particular, since $\mathrm{Ext}_A(\mathcal{B})$ is non-singular,

$$\forall \mathbf{x} \in \mathbb{F}_{q^m}^n, \quad \mathrm{Rank}(\mathrm{Ext}_A(\mathbf{x})) = \mathrm{Rank}(\mathrm{Ext}_B(\mathbf{x})).$$

This common quantity, which shall be denoted by $|\mathbf{x}|_R$, is called the *rank weight* of $\mathbf{x}$. This induces a well-defined *rank metric* on $\mathbb{F}_{q^m}^n$.

2.2.2 Notion of Support

One particularity with the Hamming metric compared with for example the euclidean metric used in lattice-based cryptography is that all the information on the error is contained in its support. More precisely, if we are given the syndrome $\mathbf{H}\mathbf{e}^\top$ of an error of Hamming weight t, and the t positions where $\mathbf{e}$ is non zero, then solving a linear system is in general enough to recover the actual error.

In the rank metric though, this is not enough: the rank weight of a vector depends not only on a set of coordinates, but also on the values. Nevertheless, the set of error coordinates can be

somehow bounded. Indeed, an error $\mathbf{e}$ is of rank weight t if and only if

$$\dim_{\mathbb{F}_q} \mathrm{Span}\{e_1, \dots, e_n\} = t.$$

It turns out that the knowledge of an $\mathbb{F}_q$-*basis* of this linear span is enough to recover the full error by solving a linear system (see for instance [GRS13, Section 3; AGHT18, Section 1.4]).

For this reason, this linear span is often called the *rank support* and denoted by $\mathrm{Supp}(\mathbf{e})$. However, recall that by choosing a basis $\mathcal{B}$, an element $\mathbf{e} \in \mathbb{F}_{q^m}^n$ can be seen as the matrix $\mathrm{Ext}_{\mathcal{B}}(\mathbf{e}) \in \mathbb{F}_q^{m\times n}$. With this matrix representation, the aforementioned rank support corresponds exactly to the *column space* of $\mathrm{Ext}_{\mathcal{B}}(\mathbf{e})$ (or its image by $\mathrm{Ext}_{\mathcal{B}}^{-1}$ if one really wants to consider an $\mathbb{F}_q$-subspace of $\mathbb{F}_{q^m}^n$). Nevertheless, another related element can be considered, namely the *row space*, which we denote by $\mathrm{RowSupp}_{\mathcal{B}}(\mathbf{e})$:

$$\mathrm{RowSupp}_{\mathcal{B}}(\mathbf{e}) \stackrel{\text{def}}{=} \left\{\mathbf{y}\mathrm{Ext}_{\mathcal{B}}(\mathbf{x}) \mid \mathbf{y} \in \mathbb{F}_q^m\right\} \subset \mathbb{F}_q^n.$$

By definition of the rank weight, this is a also an $\mathbb{F}_q$ vector space of dimension t. It turns out that this row support can be sometimes more useful, especially in the hands of a cryptanalyst:

Proposition 2.1

Let $\mathcal{A}$ and $\mathcal{B}$ be two different basis of $\mathbb{F}_{q^m}/\mathbb{F}_q$. Then,

$$\forall \mathbf{x} \in \mathbb{F}_{q^m}^n, \quad \mathrm{RowSupp}_{\mathcal{B}}(\mathbf{x}) = \mathrm{RowSupp}_{\mathcal{A}}(\mathbf{x}).$$

Proof. By definition,

$$\mathrm{RowSupp}_{\mathcal{A}}(\mathbf{x}) = \left\{\mathbf{y}\mathrm{Ext}_{\mathcal{A}}(\mathbf{x}) \mid \mathbf{y} \in \mathbb{F}_q^m\right\}$$

and by Equation (2.2) we have

$$\mathrm{RowSupp}_{\mathcal{A}}(\mathbf{x}) = \left\{\mathbf{y}\mathrm{Ext}_{\mathcal{A}}(\mathcal{B}) \cdot \mathrm{Ext}_{\mathcal{B}}(\mathbf{x}) \mid \mathbf{y} \in \mathbb{F}_q^m\right\} = \left\{\mathbf{y}'\mathrm{Ext}_{\mathcal{B}}(\mathbf{x}) \mid \mathbf{y}' \in \mathbb{F}_q^m\right\} = \mathrm{RowSupp}_{\mathcal{B}}(\mathbf{x})$$

where $\mathbf{y} \stackrel{\text{def}}{=} \mathbf{y}\mathrm{Ext}_{\mathcal{A}}(\mathcal{B})$ ranges all over $\mathbb{F}_q^m$ since $\mathrm{Ext}_{\mathcal{A}}(\mathcal{B})$ is of full rank. □

In particular, the row support of an element of $\mathbb{F}_{q^m}^n$ *does not* depend on the choice of the extension basis, and we may simply denote by $\mathrm{RowSupp}(\mathbf{x})$ this common vector space contrary to the column support, where we need to *remember* the basis. In this sense, the row support seems more *canonical*. In particular, it cannot be hidden by a change of basis, and this can prove to be a weakness point in some cryptosystems.

2.2.3 The Rank Decoding Problem

Cryptography based on rank-metric codes essentially relies on the hardness of the so-called rank metric decoding problem.

Let $\mathscr{C} \subset \mathbb{F}_q^{m\times n}$ be a matrix code of dimension K, and denote by $\mathbf{M}_1, \dots, \mathbf{M}_K$ a basis of $\mathscr{C}$. Given a matrix $\mathbf{Y} \in \mathbb{F}_q^{m\times n}$ of the ambient space, decoding at rank distance r exactly means

finding elements $x_1, \ldots, x_K \in \mathbb{F}_q$ such that

$$\mathrm{Rank}\left(\mathbf{Y} - \sum_{i=1}^{K} x_i \mathbf{M}_i\right) \leqslant r.$$

In other words, decoding a matrix code is exactly an instance of the *MinRank* at the core of many multivariate cryptosystems. MinRank was introduced and proved to be NP-complete in [BFS99] for some parameters. This proof was improved in [Cou01] with a reduction from the Decoding Problem in the Hamming metric, which proves NP-completeness for all the practical parameters, and in particular hard *on the worst-case*.

When restricting to the $\mathbb{F}_{q^m}$-linear situation, this problem has a formulation much closer to the more familiar Decoding Problem in the Hamming metric (1.10).

Problem 2.2 (Rank Decoding Problem (RDP))

Let m, n, k, t be integers, such that $k, t \leqslant n$. Given an $\mathbb{F}_{q^m}$-linear code $\mathscr{C} \subset \mathbb{F}_{q^m}^n$, generated by a matrix $\mathbf{G} \in \mathbb{F}_{q^m}^{k \times n}$, and a vector $\mathbf{y}$ of the form $\mathbf{mG} + \mathbf{e}$ where $\mathbf{e} \in \mathbb{F}_{q^m}^n$ has rank weight t, the goal is to find $\mathbf{e}$.

By using the extension map (Equation 2.1), Problem 2.2 can be seen as a particular instance of MinRank. However, $\mathbb{F}_{q^m}$-linear codes are more algebraically structured, and this might hinder the security. In fact, contrary to general MinRank, the Rank Decoding Problem is *not* known to be NP-complete. Nevertheless, a *probabilistic* reduction from the Decoding Problem (in the Hamming metric) was given in [GZ16].

Remark 2.3. *This reduction requires to work with short codes, of length $n < \sqrt{m}$, while in practice one tends to use rank metric codes in a regime where $n = \theta(m)$.*

In this manuscript we will mostly be concerned with this problem in a worst case regime where the rank metric code is fixed. However, some rank metric based cryptosystems may rely on the hardness of this problem, on average, similarly to more traditional code-based cryptography.

Remark 2.4. *Similarly to the Hamming metric situation, we can define a rank metric analogue of the Gilbert-Varshamov bound (see [Loi06b] for more precise information, and close formulae). This is both the typical minimum (rank) distance of a random $\mathbb{F}_{q^m}$-linear code*[ii]*, and the limit below which we typically expect a unique solution to the rank decoding problem, and above which the number of solutions is typically exponential. In particular, most of the rank metric submissions to NIST's new call for post-quantum signatures propose to instantiate their systems with parameters close to this bound.*

There exist essentially two general approaches for decoding an $\mathbb{F}_{q^m}$-linear rank metric code.

The combinatorial approach. The general idea to solve Problem 2.2 is to recover the support of the error, or at least a vector space which contains the support. Usually, most algorithms were concerned in recovering the column support. The state-of-the-art generic algorithm for decoding an $\mathbb{F}_{q^m}$-linear rank metric code of length n and dimension k can be found in [AGHT18] and

[ii] It can also be defined for more general matrix codes.

needs to perform

$$O\left((n-k)^{\omega}m^{\omega}q^{t\left\lfloor\frac{(k+1)m}{n}\right\rfloor-m}\right)$$

operations in $\mathbb{F}_q$, where ω denotes the linear algebra exponent. In particular, for parameters of interest where t, k and m are *linear* in n, the complexity is exponential in $n \times m$ (while in the Hamming world, the complexity would only be exponential in n). Similarly to using quasi-cyclic codes, this observation allows for a finer tuning of the parameters, and yields to cryptosystems with a smaller key size compared to unstructured code-based cryptosystems. Up until very recently, this was the best known approach for solving RDP, especially for small values of q. However, recent progress in algebraic cryptanalysis proved that for some regimes algebraic approaches drastically improved on the combinatorial one.

The algebraic approach. Solving MinRank problem can be restated into a multivariate quadratic system, which may then be solved using Gröbner bases (see for instance [FLP08]). Some specific work such as [LP06b; LP06a; GRS13] tried to specifically target RDP, but were considered to perform way poorer than direct combinatorial attacks.

This state of affairs changed with the breakthrough papers [BBBG+20; BBCG+20], which showed how the $\mathbb{F}_{q^m}$-linearity could drastically improve algebraic attacks against RDP, using the so-called MaxMinors modelling. This line of work was further improved in [BBBG+22] with the introduction of a new modelling called SupportMinor, which is designed to specifically retrieve the column support of the error via an algebraic system.

For small values of t and q, these new algebraic attacks are now considered as the most efficient approaches for solving a generic instance of RDP. In particular, two rank metric encryption schemes were still present in Round 2 of the (first) NIST call for post-quantum cryptosystems, namely ROLLO [ABDG+19], which can be thought of as a rank metric analogue of BIKE [AABB+22a] cryptosystem, *i.e.* a McEliece-like cryptosystem using so-called LRPC codes; and RQC [AABB+20], which is a rank metric analogue of HQC [AABB+22b] ; however because of those algebraic attacks NIST did not select neither of them to move forward in the third round. Nevertheless, in the status report on the second round [AJAC+20], NIST still encouraged further research in rank metric based cryptography.

On the other hand, when the rank t of the error increases, algebraic approaches still seem to behave similarly, or even worse, than the combinatorial attacks.

A note on the decisional Rank Decoding Problem. Similarly to what happens with codes endowed with the Hamming metric, some cryptosystems actually rely on the *decisional* version of RDP, where the goal is to distinguish between the following two distributions

- $\mathcal{D}_0 : (\mathbf{G}, \mathbf{y}^{\text{unif}})$ uniformly distributed over $\mathbb{F}_{q^m}^{k\times n} \times \mathbb{F}_{q^m}^{n}$
- $\mathcal{D}_1 : (\mathbf{G}, \mathbf{mG}+\mathbf{e})$, where $G \leftarrow \mathbb{F}_{q^m}^{k\times n}$ is a uniformly random generator matrix of an $\mathbb{F}_{q^m}$-linear code, $\mathbf{m} \leftarrow \mathbb{F}_{q^m}^{k}$ and $\mathbf{e}$ is uniformly distributed in the set of vectors of $\mathbb{F}_{q^m}^{n}$ of rank-weight t.

Remark 2.5. *This problem could equivalently have been formulated in terms of syndromes and parity-check matrices.*

However, contrary to the Hamming metric counterpart, no reduction is known from the rank decoding problem to the rank decisional version, even for decoding matrix codes, *i.e.* solving the MinRank problem. The only known reductions are Hamming to rank (see for instance [GHPT16,

Appendix B.2]). In particular, this appears to be a very challenging task, and solving this issue would increase the confidence in cryptosystems based on the rank metric.

2.3 Gabidulin codes and their decoding algorithms

Recall from Section 1.1.5 that in order to build an encryption scheme in the McEliece framework, one is interested in having codes which are known to be efficiently decodable at a rather large distance. However, contrary to the Hamming metric, few families of rank metric codes are known to have efficient decoding algorithms, and most of them are $\mathbb{F}_{q^m}$-linear. They lie in essentially three families:

- *Gabidulin codes*[iii] [Del78; Gab85], which can be seen in many ways as rank metric analogues of Reed-Solomon codes. In particular, an $[n,k]$ Gabidulin code can be deterministically and efficiently decoded up to $\frac{n-k}{2}$ rank errors.
- *LRPC codes* [GMRZ13; AGHR+18], which are can be thought of as rank metric analogues of LDPC and MDPC codes (see Remark 2.6 below). They benefit from a probabilistic decoder inspired from that of MDPC codes, but with the specificities of the rank metric. In particular, they can induce decoding failures.
- So-called *simple codes*, introduced in [SK11], and revisited in [GHPT16].

Remark 2.6. *An $\mathbb{F}_{q^m}$-linear code $\mathscr{C}$ is LRPC if and only if there exists a small d-dimensional $\mathbb{F}_q$-vector space $F \subset \mathbb{F}_{q^m}^n$ such that $\mathscr{C}^\perp$ has a basis $(\mathbf{h}_1, \dots, \mathbf{h}_{n-k})$ with*

$$\mathrm{Supp}(\mathbf{h}_i) \subset F, \quad \forall i \in \{1, \dots, n-k\},$$

where Supp *denotes the* column support*.*

The first rank-metric based cryptosystem is due to Gabidulin, Paramonov and Tretjakov, and is referred to as the GPT cryptosystem [GPT91]. It is a McEliece-like encryption scheme based on Gabidulin codes, and was sequentially broken then repaired during its first years before Overbeck [Ove05] designed a very efficient structural attack which virtually shows that Gabidulin codes have too much structure to be used in a McEliece-like encryption scheme. LRPC codes are at the core of the design of ROLLO [ABDG+19], which made it to the second round of NIST post-quantum competition, but was not selected to advance forward.

In this chapter, and the following, we will focus on Gabidulin codes.

2.3.1 The ring of q-polynomials

We want to define a rank-metric analogue of Reed-Solomon codes, that is we will define Gabidulin codes as an evaluation code of a class of polynomials on elements of $\mathbb{F}_{q^m}$. In order to account for the rank metric, we consider polynomials which are actually *linear*, and evaluate them on *linearly independent* elements of $\mathbb{F}_{q^m}$.

[iii] actually introduced by Delsarte, but popularised by Gabidulin.

Definition 2.7 (q-Polynomial)

A q-polynomial of q-degree d with coefficients in $\mathbb{F}_{q^m}$ is a univariate polynomial of the form

$$P(X) \stackrel{\text{def}}{=} a_0 \cdot X + a_1 \cdot X^q + \cdots + a_d \cdot X^{q^d}, \quad \text{with } a_i \in \mathbb{F}_{q^m} \text{ and } a_d \neq 0$$

They were introduced and studied by Ore in [Ore33]. Since the only monomials which appear in a q-polynomials are of the form X^{q^k}, a q-polynomial P induces an $\mathbb{F}_q$-linear map $P\colon \mathbb{F}_{q^m}^n \to \mathbb{F}_{q^m}^n$, and we define the rank of P to be the rank of the corresponding linear map. In the sequel, we denote by $\mathbb{F}_{q^m}\langle X\rangle$ the set of q-polynomials, and for any integer k, we denote by $\mathbb{F}_{q^m}\langle X\rangle_{<k}$ (resp. $\mathbb{F}_{q^m}\langle X\rangle_{\leqslant k}$) the set of q-polynomials of q-degree less than k (resp. less than or equal to k).

Equipped with the regular addition of polynomials and the composition law, $\mathbb{F}_{q^m}\langle X\rangle$ is endowed with a ring structure.

Remark 2.8. *$\mathbb{F}_{q^m}\langle X\rangle$ is non commutative:*

$$X^q \cdot aX = a^q X^q, \quad \textit{but} \quad aX \cdot X^q = aX^q,$$

and those two quantities differ when $a \in \mathbb{F}_{q^m} \setminus \mathbb{F}_q$.

Although not being commutative, $\mathbb{F}_{q^m}\langle X\rangle$ has a rich arithmetic structure since it is both left and right euclidean:

Proposition 2.9 ([Ore33])

Let $A, B \in \mathbb{F}_{q^m}\langle X\rangle$ be non zero. Then there exist two pairs of q-polynomials (Q_ℓ, R_ℓ) and (Q_r, R_r) such that

- (Left division) $A = B \cdot Q_\ell + R_\ell$ and $\deg_q(R_\ell) < \deg_q(B)$
- (Right division) $A = Q_r \cdot B + R_r$ and $\deg_q(R_r) < \deg_q(B)$.

Moreover, (Q_ℓ, R_ℓ) and (Q_r, R_r) are uniquely determined.

Remark 2.10. *This proposition entails that any left (resp. right) ideal of $\mathbb{F}_{q^m}\langle X\rangle$ is principal.*

Remark 2.11. *It is not hard to adapt the usual Euclidean division algorithm for actual polynomials to the q-polynomial setting: besides the non-commutativity, everything works similarly. In particular, left and right euclidean division algorithms can be efficiently implemented.*

When working with q-polynomials, it is often very convenient to switch from the point of view of actual polynomials, to that of $\mathbb{F}_q$-endomorphisms of $\mathbb{F}_{q^m}$. The existence of this euclidean division makes those equivalent. Indeed, two q-polynomials P and Q induce the same $\mathbb{F}_{q^m}$-endomorphism if and only if their difference is (left or right) divisible by $X^{q^m} - X$. In particular, the two sided ideal $(X^{q^m} - X)$ is the kernel of the canonical map

$$\mathbb{F}_{q^m}\langle X\rangle \to \mathrm{Hom}_q(\mathbb{F}_{q^m}, \mathbb{F}_{q^m}).$$

Moreover,

$$\dim_q \left(\mathbb{F}_{q^m}\langle X\rangle \Big/ (X^{q^m} - X) \right) = m^2 = \dim_q \mathrm{Hom}_q(\mathbb{F}_{q^m}, \mathbb{F}_{q^m}),$$

which induces an isomorphism

$$\mathcal{L} \stackrel{\text{def}}{=} \mathbb{F}_{q^m}\langle X\rangle \Big/ (X^{q^m} - X) \simeq \mathrm{Hom}_q(\mathbb{F}_{q^m}, \mathbb{F}_{q^m}).$$

Remark 2.12. *Any element of $\mathcal{L}$ can be uniquely lifted to $\mathbb{F}_{q^m}\langle X\rangle$ as a q-polynomial of q-degree less than m. For this reason, we identify $\mathcal{L}$ with $\mathbb{F}_{q^m}\langle X\rangle_{<m}$. Since we only work in $\mathcal{L}$, we even drop the subscript $< m$ and all our q-polynomials will be of q-degree less than m.*

In particular, the roots of a q-polynomial form an $\mathbb{F}_q$-vector subspace of $\mathbb{F}_{q^m}$, of dimension less than or equal to its q-degree, and a basis can be determined by simply solving a linear system. The converse is also true (see for instance [Wac13]).

Definition 2.13 (Vanishing Polynomial)

For any given vector space $\mathcal{E} \subset \mathbb{F}_{q^m}$ of dimension t, there exists a unique monic q-polynomial of q-degree less than t whose roots are exactly $\mathcal{E}$. Such a q-polynomial is called the (minimum) *vanishing polynomial* of $\mathcal{E}$, and can be determined explicitly given a basis of $\mathcal{E}$ (see for instance [Wac13]).

2.3.2 Gabidulin codes and their decoding algorithms

Gabidulin codes can be seen as rank metric analogues for Reed-Solomon codes [RS60], where one replaces polynomials by q-polynomials.

Definition 2.14 (Gabidulin code)

Let $k \leqslant n \leqslant m$ be integers, and let $\mathbf{g} \stackrel{\text{def}}{=} (g_1, \ldots, g_n) \in \mathbb{F}_{q^m}^n$ be of rank weight n. The Gabidulin code of length n, dimension k and evaluation vector $\mathbf{g}$ is

$$\mathrm{Gab}_k(\mathbf{g}) \stackrel{\text{def}}{=} \{(P(g_1), \ldots, P(g_n)) \mid P \in \mathbb{F}_{q^m}\langle X\rangle_{<k}\}.$$

It is well known that Gabidulin codes bare a strong similarity with Reed-Solomon codes, their Hamming metric counterpart. In particular, Gab_k reaches the Singleton bound, *i.e.* has minimum rank distance $n-k+1$.[iv] Moreover, many decoders for Reed-Solomon codes have their rank metric counterpart to efficiently decode Gabidulin codes up to the unique decoding radius $\frac{n-k}{2}$ (see for example [Loi06a]). However, there is a strong difference between the Hamming and the rank metric worlds. Indeed, since the breakthrough works of Guruswami and Sudan [GS98], Reed-Solomon codes are well-known to be decodable slightly further (up to the Johnson bound) at the cost of allowing to return a (polynomially bounded) list of solutions. On the other hand, no such algorithm is known for Gabidulin codes. Even more, it was proved in [RW15] that there even exists infinite families of Gabidulin codes for which the number of solutions to the decoding problem was exponential even for decoding at distance $\frac{n-k}{2} + 1$. Therefore, there is no hope to break this barrier in the worst-case, and this is considered to be a very hard problem. In particular, this seems to be interesting for cryptographic applications and both LIGA [RPW21]

[iv] Such a code is known as Maximum Rank Distance (MRD). Note that an MRD code is also MDS for the Hamming metric.

and RAMESSES [LLP20] cryptosystems, which will be introduced in Section 3.3, were trying to make their security rely on this problem.

Note that the map

$$\begin{cases} \mathbb{F}_{q^m}\langle X\rangle_{<k} & \longrightarrow & \mathrm{Gab}_k(\mathbf{g}) \\ P & \longmapsto & (P(g_1),\dots,P(g_n)) \end{cases}$$

is a rank preserving isomorphism. In particular, we can, and will, identify a Gabidulin code with a space of q-polynomials of bounded degree.

Remark 2.15. *This is another difference between Gabidulin and Reed-Solomon codes. Indeed, for the latter the metric cannot be defined directly from the space of polynomials, while for the rank metric both points of view are interchangeable. It is even more surprising that the knowledge of the evaluation vector* $\mathbf{g}$ *is in reality not needed for designing decoding algorithms (see [CZ23, Section 2.1]).*

Loidreau's approach for decoding. In [Loi06a], Loidreau adapted the famous Berlekamp-Welch algorithm for Reed-Solomon codes, to decode Gabidulin codes up to $\frac{n-k}{2}$ rank errors. We will revisit this algorithm in Section 3.2.3 with cryptanalytic applications in mind.

The idea of the algorithm is to *locate* the error. Namely, consider a corrupted codeword $\mathbf{y} = \mathbf{c} + \mathbf{e}$ where $\mathbf{y}$ is known to the receiver, $\mathbf{c} \overset{\text{def}}{=} P(\mathbf{g}) \in \mathrm{Gab}_k(\mathbf{g})$ and $\mathbf{e}$ is of rank t. In particular, the column support of $\mathbf{e}$ is a t-dimensional subspace of $\mathbb{F}_{q^m}$, and there exists a (unique) monic q-polynomial Λ of q-degree less than t which exactly vanishes on $\mathrm{Supp}(\mathbf{e})$. This yields to a (non linear) system of n equations

$$\Lambda(y_i) = \Lambda \circ P(g_i), \qquad \text{for } i \in \{1,\dots,n\} \tag{2.3}$$

whose unknowns are the coefficients of Λ (t unknowns since it is monic and $\deg_q(\Lambda) = t$) and P (k unknowns since it has q-degree at most $k-1$). Inspired by the Reed-Solomon situation, one can linearise the system and set $N \overset{\text{def}}{=} \Lambda \circ P$ to be an (unknown) q-polynomial of q-degree at most $k+t-1$, in order to get

$$\Lambda(y_i) = N(g_i), \qquad \text{for } i \in \{1,\dots,n\} \tag{2.4}$$

System 2.4 now has n equations and $t + (k+t) = k + 2t$ unknowns. In other words, it has more equations than unknowns, as long as $t \leqslant \frac{n-k}{2}$, that is the radius for which we know that the solution is unique. In [Loi06a], Loidreau proved[v] that any non-zero solution (Λ, N) of System 2.4 satisfies $N = \Lambda \circ P$, which allows to recover P by (left) Euclidean division.

2.3.3 The Overbeck Distinguisher

If Gabidulin codes and Reed-Solomon codes have many features in common from a coding theory point of view, they also share their structural flaws. Indeed, similarly to the Hamming metric situation, Gabidulin codes and many of their variants, can in reality easily be distinguished from random $\mathbb{F}_{q^m}$-linear codes, and therefore cannot be used inside a McEliece-like cryptosystem. This breaks in particular the original instantiation [GPT91]. This distinguisher, due to Overbeck [Ove05], is based on the Frobenius operator, and has recently been revisited and extended in [CZ23]. It has to be seen in light of the star product distinguisher for algebraic codes in the Hamming metric.

[v] Actually, there is a mistake in the proof of [Loi06a, Proposition 2], but it can be fixed.

Remark 2.16. *The goal of this section is to introduce the Overbeck distinguisher, since it is important to understand the construction of LIGA cryptosystem which is the central point of Chapter 3. A detailed presentation of complete attacks on GPT-like cryptosystems is out of scope of this section, and the interested reader can refer to [CZ23] for a recent presentation.*

Given a vector $\mathbf{x} \in \mathbb{F}_{q^m}^n$, and a non negative integer s, denote by $\mathbf{x}^{[s]}$ the vector

$$\mathbf{x}^{[s]} \stackrel{\text{def}}{=} (x_1^{q^s}, \dots, x_n^{q^s}).$$

Similarly, given an $\mathbb{F}_{q^m}$-linear code $\mathscr{C}$, the code $\mathscr{C}^{[s]}$ is defined as

$$\mathscr{C}^{[s]} \stackrel{\text{def}}{=} \{\mathbf{c}^{[s]} \mid \mathbf{c} \in \mathscr{C}\}.$$

In general, $\mathscr{C}$ and $\mathscr{C}^{[s]}$ have nothing in common, and even for $s = 1$, $\mathscr{C} \cap \mathscr{C}^{[1]}$ will be zero with high probability if $\dim \mathscr{C} < \frac{n}{2}$. For a non negative integer s, denote by

$$\mathbf{\Lambda}_s(\mathscr{C}) \stackrel{\text{def}}{=} \mathscr{C} + \mathscr{C}^{[1]} + \cdots + \mathscr{C}^{[s]}. \tag{2.5}$$

In particular, for a random $\mathbb{F}_{q^m}$-linear code of dimension $< \frac{n}{2}$, we expect that $\dim_{\mathbb{F}_{q^m}} \mathbf{\Lambda}_1(\mathscr{C}) = 2k$. In fact, for a random $\mathbb{F}_{q^m}$-linear code $\mathscr{C}$ of length n and dimension k, and $s < k$, the following holds for any $\varepsilon > 0$

$$\mathbb{P}\left(\dim_{\mathbb{F}_{q^m}} \mathbf{\Lambda}_s(\mathscr{C}) \leqslant \min(n, (s+1)k) - \varepsilon\right) = O(q^{-m\varepsilon}).$$

(See [CC20] for a proof of this result). On the other hand, the situation is completely different for Gabidulin codes. Indeed, if $\mathrm{Gab}_k(\mathbf{g})$ is a k-dimensional Gabidulin code, then it is readily seen that

$$\mathbf{\Lambda}_s(\mathrm{Gab}_k(\mathbf{g})) = \mathrm{Gab}_{k+s}(\mathbf{g}).$$

In particular, it has dimension $k+s$ (when $k+s < n$), which is in general much smaller than $(s+1)k$.

This distinguisher does not *per se* yield a complete attack, but can be used to recover much information on the hidden Gabidulin code. It also already breaks the security of McEliece encryption schemes based on (variants of) Gabidulin codes. Moreover, contrary to their Hamming metric counterpart, subfield subcodes of Gabidulin codes are not suitable either, since they can be split into sum of smaller Gabidulin codes [GL05; GL08]. In particular, it is not possible to design an analogue of Classic McEliece in the rank metric.

However, all hope of using the rank metric in cryptosystems is not lost. Indeed, replacing Gabidulin codes with LRPC codes yields the ROLLO[vi] proposal, which made it to the second round of NIST first call for post-quantum primitives. It can be thought of as the rank metric analogue of BIKE (in the Hamming metric), or NTRU(in the Euclidean metric). One could also try different approaches. For example, RQC, which also made it to the second round, is the rank metric analogue of HQC, currently competing in the fourth round, and can also be thought of as a rank metric analogue of Ring-LWE-based encryption scheme.

[vi] Actually, ROLLO is a merge between different close schemes.

Chapter 3

Cryptanalysis in the Rank Metric

Contributions of this thesis. In this chapter, based on two publications: [BC21] and [BC22], we revisit a decoder for Gabidulin codes by working "on the right hand side". This put the emphasis on the notion of *row support* in the rank metric, with cryptanalytic purposes in mind. Finally, we provide message recovery attacks against two code-based encryption schemes with short keys, based on the rank metric. Namely LIGA [RPW21] and RAMESSES [LLP20].

Outline of the current chapter

3.1 Wishful Thinking: Code-Based Encryption Schemes with Short Keys

Since Gabidulin codes, as well as Reed-Solomon, are easily distinguishable from random codes, they cannot be used inside a McEliece-like encryption scheme. With this observation in hand, one can still wonder if it is possible to build a cryptosystem with such codes, without masking the structure. A first answer was given by Augot and Finiasz in [AF03], who proposed to build an

encryption scheme based on the hardness of decoding a public code beyond a certain threshold (*e.g.* the decoding radius of Guruswami-Sudan algorithm, namely $n - \sqrt{kn}$ for an $[n, k]$ Reed-Solomon codes). The idea is the following: Let $\mathbf{G} \in \mathbb{F}^{k \times n}$ be the generator matrix of a public code $\mathscr{C}$ (*e.g.* Reed-Solomon, Gabidulin). In the spirit of McEliece's construction, in order to encrypt a message $\mathbf{m} \in \mathbb{F}^k$, we would like to encode it in $\mathscr{C}$ and corrupt it by an error $\mathbf{e}$ of large weight W so that

$$\mathrm{Enc}(\mathbf{m}) \stackrel{\text{def}}{=} \mathbf{mG} + \mathbf{e}.$$

The secret key should be an efficient decoding algorithm $\mathcal{D}$ at distance W. However, this is incompatible with the wanted security, unless $\mathcal{D}$ could remove part of the error. For example, if $\mathbf{e}$ split into two parts

$$\mathbf{e} = \mathbf{e}_{\text{sec}} + \mathbf{e}_{\text{rand}}$$

where $\mathbf{e}_{\text{sec}}$ was part of the secret key and $\mathbf{e}_{\text{rand}}$ was a random error, then $\mathcal{D}$ could first remove $\mathbf{e}_{\text{sec}}$ and then decode normally $\mathrm{Enc}(\mathbf{m}) - \mathbf{e}_{\text{sec}}$, provided that the weight of $\mathbf{e}_{\text{rand}}$ is small enough.

In order to do that, Augot and Finiasz proposed that the public key $\mathbf{k}_{\text{pub}}$ itself should be a secret codeword of $\mathscr{C}$, corrupted by the large secret error $\mathbf{e}_{sec}$:

$$\mathbf{k}_{\text{pub}} \stackrel{\text{def}}{=} \mathbf{m}_{\text{sec}}\mathbf{G} + \mathbf{e}_{\text{sec}},$$

where $\mathbf{m}_{\text{sec}} \in \mathbb{F}^k, \mathbf{e}_{\text{sec}} \in \mathbb{F}^n$ form the secret key. In order to give an intuition, and since the Faure-Loidreau cryptosystem precisely described below is the rank-metric analog of Augot-Finiasz proposal, we will consider an oversimplified (and trivially unsecure!) version, where the encryption of a message $\mathbf{m} \in \mathbb{F}_q^k$ simply is

$$\begin{aligned}\mathrm{Enc}(\mathbf{m}) &\stackrel{\text{def}}{=} (\mathbf{mG} + \mathbf{e}_{\text{rand}}) + \mathbf{k}_{\text{pub}} \\ &= (\mathbf{m} + \mathbf{m}_{\text{sec}})\mathbf{G} + (\mathbf{e}_{\text{rand}} + \mathbf{e}_{\text{sec}}),\end{aligned}$$

where $\mathbf{e}_{\text{rand}}$ is a random error of *small weight*. The idea for decrypting is also natural: since the owner of the secret key knows $\mathbf{e}_{\text{sec}}$, she can remove it from the ciphertext, and since $\mathbf{e}_{\text{rand}}$ has a small weight, then she can deduce $\mathbf{m} + \mathbf{m}_{\text{sec}}$ and finally $\mathbf{m}$ by applying an efficient decoding algorithm for $\mathscr{C}$, and removing $\mathbf{m}_{\text{sec}}$ (which is also part of the secret key).

The originality of this proposal, and what makes it very attractive at first glance, is that the public key is a *single* word of length n, and therefore its size only scales *linearly* in the security parameter. Unfortunately, it was subject to an efficient attack by Coron [Cor04]. At a high level, it consists in crafting a low degree polynomial P vanishing at some secret data,[i] which can be recovered since $\deg P$ is small. Augot, Finiasz and Loidreau [AFL03] subsequently found a way to mitigate this attack by considering the public key into an extension $\mathbb{L}/\mathbb{F}$, and using the trace operator. More precisely, their fix consisted in setting

$$\mathbf{k}_{\text{pub}} \stackrel{\text{def}}{=} \widetilde{\mathbf{m}_{\text{sec}}}\mathbf{G} + \widetilde{\mathbf{e}_{\text{sec}}} \in \mathbb{L}^n,$$

where the code is still defined over $\mathbb{F}$, and $\widetilde{\mathbf{m}_{\text{sec}}} \in \mathbb{L}^k, \widetilde{\mathbf{e}_{\text{sec}}} \in \mathbb{L}^n$. It turns out that Coron's approach builds a polynomial whose degree is actually *exponential* in $[\mathbb{L} : \mathbb{F}] - 1$.

If this proposal resists the first attack by Coron, it is still insecure. Indeed, the public key can be considered as $[\mathbb{L} : \mathbb{F}]$ noisy codewords over the small field $\mathbb{F}$, with an error of large weight. However, Coron noticed that those errors over the small field are *not* independent, and in fact errors occur at the very same positions, *i.e.* the error vectors all have the *same support*. In other

[i] This secret element is not present in our "simple" presentation of Augot-Finiasz.

words, the public key can be considered as a codeword of an *interleaved* version of $\mathscr{C}$, and when $\mathscr{C}$ is a Reed-Solomon code, this allows to decode slightly beyond the unique decoding radius with high probability.

Nevertheless, using extension fields motivates to use the rank metric instead of the Hamming metric, especially since no list decoding algorithm is known for Gabidulin codes beyond the unique decoding radius. This is the starting point of the proposal by Faure and Loidreau, which is the main topic of this chapter.

Unfortunately, Faure-Loidreau encryption scheme was subject to an efficient key recovery attack by Gaborit, Otmani and Talé-Kalachi [GOT18], using an approach *à la Overbeck*. Two independent repairs were proposed to thwart this attack: namely RAMESSES [LLP20], and LIGA [RPW21]. They will be presented in Section 3.3, and Section 3.4 will show that those proposals are still insecure.

Remark 3.1. *In his PhD thesis [Fau09, in French], Faure mentions another motivation: after applying the trace operator, the rank supports of each error vector over the small field are not correlated anymore, which seems to thwart the aforementioned approach by Coron. In reality, as we will see in Section 3.2.2, this lack of correlation is only true for the* column support*, while we can observe a correlation on the* row support*, making the scheme vulnerable to this kind of attack. This provides an alternative attack to [GOT18].*

3.2 Another Attack on Faure-Loidreau Cryptosystem

3.2.1 Faure-Loidreau cryptosystem

After the intuitive presentation made in Section 3.1, we shall now precisely describe the original Faure-Loidreau encryption scheme.

3.2.1.1 Description of the original Faure-Loidreau cryptosystem

Public Parameters. Let k, m, n, q, u, w be positive integers such that q is a prime power, $u < k < n$ and $n - k > w > \lfloor \frac{n-k}{2} \rfloor$ and consider the three finite fields

$$\mathbb{F}_q \subset \mathbb{F}_{q^m} \subset \mathbb{F}_{q^{mu}}.$$

Let $t_{\text{pub}} \stackrel{\text{def}}{=} \left\lfloor \frac{n-k-w}{2} \right\rfloor$ and let $\mathbf{G}$ be a generator matrix of a public Gabidulin code $\mathrm{Gab}_k(\mathbf{g})$ of length n and dimension k, defined over the intermediate field $\mathbb{F}_{q^m}$. All the rank weights are considered over $\mathbb{F}_q$.

Key Generation. Alice, the owner of the secret key, picks a random vector $\mathbf{x} \leftarrow \mathbb{F}_{q^{mu}}^k$ whose last u entries form an $\mathbb{F}_{q^m}$ basis of the extension $\mathbb{F}_{q^{mu}}/\mathbb{F}_{q^m}$, and then constructs a vector $\mathbf{z} \in \mathbb{F}_{q^{mu}}^n$ of rank weight w by choosing a random w-dimension column support. More precisely, she picks a full rank vector of $\mathbb{F}_{q^{mu}}^w$ and a non-singular matrix $\mathbf{P} \in \mathrm{GL}_n(\mathbb{F}_q)$, and sets

$$\mathbf{z} \stackrel{\text{def}}{=} (\mathbf{s} \mid \mathbf{0}_{n-w}) \cdot \mathbf{P}^{-1} \in \mathbb{F}_{q^{mu}}^n.$$

The public key is then

$$\mathbf{k}_{\text{pub}} \stackrel{\text{def}}{=} \mathbf{x} \cdot \mathbf{G} + \mathbf{z} \in \mathbb{F}_{q^{mu}}^n, \tag{3.1}$$

while the secret key is $(\mathbf{x}, \mathbf{z}, \mathbf{P})$.

Encryption. A plaintext is a vector $\boldsymbol{\mu} \stackrel{\text{def}}{=} (m_1, \ldots, m_{k-u}) \in \mathbb{F}_{q^m}^{k-u}$, padded with 0's at the end, to form a vector of length k: let $\mathbf{m} \stackrel{\text{def}}{=} (\boldsymbol{\mu} \mid \mathbf{0}_u) \in \mathbb{F}_{q^m}^{k}$. The encryption works as follows:

1. Pick $\alpha \leftarrow \mathbb{F}_{q^{mu}}^{\times}$;
2. Pick $\mathbf{e} \in \mathbb{F}_{q^m}^{n}$ with $|\mathbf{e}|_R = t_{\text{pub}}$ (rank weight).

$$\text{Enc}(\boldsymbol{\mu}) \stackrel{\text{def}}{=} \mathbf{m} \cdot \mathbf{G} + \text{Tr}_{\mathbb{F}_{q^{mu}}/\mathbb{F}_{q^m}}(\alpha \mathbf{k}_{\text{pub}}) + \mathbf{e}.$$

Remark 3.2. *By $\mathbb{F}_{q^m}$-linearity of the trace operator, we have*

$$\text{Tr}_{\mathbb{F}_{q^{mu}}/\mathbb{F}_{q^m}}(\alpha \mathbf{k}_{\text{pub}}) = \text{Tr}_{\mathbb{F}_{q^{mu}}/\mathbb{F}_{q^m}}(\alpha \mathbf{x}) \cdot \mathbf{G} + \text{Tr}_{\mathbb{F}_{q^{mu}}/\mathbb{F}_{q^m}}(\alpha \mathbf{z})$$

i.e.

$$\text{Enc}(\boldsymbol{\mu}) = \left(\mathbf{m} + \text{Tr}_{\mathbb{F}_{q^{mu}}/\mathbb{F}_{q^m}}(\alpha \mathbf{x})\right) \cdot \mathbf{G} + \left(\text{Tr}_{\mathbb{F}_{q^{mu}}/\mathbb{F}_{q^m}}(\alpha \mathbf{z}) + \mathbf{e}\right) \tag{3.2}$$

In other words, the public key $\mathbf{k}_{\text{pub}}$ acts two-fold in the encryption: first, it acts as a one-time pad on the plaintext $\mathbf{m}$, and adds a random error of large weight (generally $w+t_{\text{pub}}$, see [RPW21, Appendix C] for a detailed discussion).

Decryption. Upon receiving a ciphertext $\mathbf{c}$, one first computes

$$\mathbf{c} \cdot \mathbf{P} = \left(\mathbf{m} + \text{Tr}_{\mathbb{F}_{q^{mu}}/\mathbb{F}_{q^m}}(\alpha \mathbf{x})\right) \cdot \mathbf{G} \cdot \mathbf{P} + \left(\left(\text{Tr}_{\mathbb{F}_{q^{mu}}/\mathbb{F}_{q^m}}(\alpha \mathbf{s}) \mid \mathbf{0}_{n-w}\right) + \mathbf{e} \cdot \mathbf{P}\right) \in \mathbb{F}_{q^m}^{n},$$

whose last $n-w$ components are of the form

$$\mathbf{c}' \stackrel{\text{def}}{=} \left(\mathbf{m} + \text{Tr}_{\mathbb{F}_{q^{mu}}/\mathbb{F}_{q^m}}(\alpha \mathbf{x})\right) \cdot \mathbf{G}' + \mathbf{e}' \in \mathbb{F}_{q^m}^{n-w},$$

where $\mathbf{G}' \in \mathbb{F}_{q^m}^{k \times (n-w)}$ and $\mathbf{e}' \in \mathbb{F}_{q^m}^{n-w}$. In other words, one removes the $\mathbf{z}$-part from the ciphertext (Equation (3.2)) by projecting it on a subspace which does not contain the support of $\mathbf{z}$. Note that, since $\mathbf{P} \in \text{GL}_n(\mathbb{F}_q)$ we have $|\mathbf{e} \cdot \mathbf{P}|_R = |\mathbf{e}|_R = t_{\text{pub}}$, and therefore the projection on the last $n-w$ entries yields an error $\mathbf{e}'$ of rank weight at most t_{pub}. Moreover, $\mathbf{G}$ being a generator matrix of $\text{Gab}_k(\mathbf{g})$, it is readily seen that $\mathbf{G} \cdot \mathbf{P}$ is a generator matrix of $\text{Gab}_k(\mathbf{g} \cdot \mathbf{P})$ which is still an $[n,k]$-Gabidulin code, and keeping only the last $n-w$ entries yields a $[n-w,k]$-Gabidulin code. Since

$$t_{\text{pub}} = \left\lfloor \frac{n-w-k}{2} \right\rfloor,$$

$\mathbf{c}'$ can be decoded and one recovers

$$\mathbf{m}' \stackrel{\text{def}}{=} \mathbf{m} + \text{Tr}_{\mathbb{F}_{q^{mu}}/\mathbb{F}_{q^m}}(\alpha \mathbf{x}) = \left(\boldsymbol{\mu} + \text{Tr}_{\mathbb{F}_{q^{mu}}/\mathbb{F}_{q^m}}(\alpha \mathbf{x}') \mid \text{Tr}_{\mathbb{F}_{q^{mu}}/\mathbb{F}_{q^m}}(\alpha x_{k-u+1}), \ldots, \text{Tr}_{\mathbb{F}_{q^{mu}}/\mathbb{F}_{q^m}}(\alpha x_k)\right).$$

In order to recover $\boldsymbol{\mu}$, recall that $(x_{k-u+1}, \ldots, x_k) \in \mathbb{F}_{q^{mu}}^{u}$ form a basis of $\mathbb{F}_{q^{mu}}/\mathbb{F}_{q^m}$ by hypothesis. Denote by $(x_{k-u+1}^*, \ldots, x_k^*) \in \mathbb{F}_{q^{mu}}^{u}$ its *dual basis* for the bilinear form $(a,b) \mapsto \text{Tr}_{\mathbb{F}_{q^{mu}}/\mathbb{F}_{q^m}}(ab)$, which can easily be computed, that is the basis of $\mathbb{F}_{q^{mu}}/\mathbb{F}_{q^m}$ such that

$$\text{Tr}_{\mathbb{F}_{q^{mu}}/\mathbb{F}_{q^m}}(x_i x_j^*) = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}, \quad \text{for } i,j \in \{k-u+1, \ldots, k\}.$$

One can now compute

$$\sum_{i=k-u+1} m'_i x_i^* = \sum_{i=k-u+1}^{k} \mathrm{Tr}_{\mathbb{F}_{q^{mu}}/\mathbb{F}_{q^m}}(\alpha x_i) x_i^* = \alpha.$$

Knowing both α and $\mathbf{x}$, the receiver can finally recover the plaintext $\boldsymbol{\mu}$.

Remark 3.3. *Note that this cryptosystem does not benefit from strong security guarantees such as a reduction to a well studied hard problem, and indeed it was subject to several attacks. This observation advocates for the importance of designing such reductions.*

3.2.1.2 A First Key Recovery Attack

In [GOT18], Gaborit, Otmani and Talé-Kalachi provided a very efficient key recovering attack by showing that the iterated Frobenius operator $\mathbf{\Lambda}$ (from (2.5)) could be used to recover a valid secret key in polynomial time, from the sole knowledge of the public key.

More precisely, they choose a basis $(\gamma_1, \ldots, \gamma_u)$ of the extension $\mathbb{F}_{q^{mu}}/\mathbb{F}_{q^m}$, and split $\mathbf{k}_{\mathrm{pub}}$ into u noisy codewords of the *same* public Gabidulin code by setting

$$\mathbf{k}_{\mathrm{pub}}^{(i)} \stackrel{\mathrm{def}}{=} \mathrm{Tr}_{\mathbb{F}_{q^{mu}}/\mathbb{F}_{q^m}}(\gamma_i \mathbf{k}_{\mathrm{pub}}) = \mathrm{Tr}_{\mathbb{F}_{q^{mu}}/\mathbb{F}_{q^m}}(\gamma_i \mathbf{x}) \cdot \mathbf{G} + \mathrm{Tr}_{\mathbb{F}_{q^{mu}}/\mathbb{F}_{q^m}}(\gamma_i \mathbf{z}) \in \mathbb{F}_{q^m}^n, \qquad \text{for } i \in \{1, \ldots, u\}. \tag{3.3}$$

Remark 3.4. *Note that* $(\mathbf{k}_{\mathrm{pub}}^{(i)})$ *is nothing else than the decomposition of* $\mathbf{k}_{\mathrm{pub}}$ *in the dual basis* $(\gamma_1^*, \ldots, \gamma_u^*)$.

Now, let

$$\mathbf{z}_i \stackrel{\mathrm{def}}{=} \mathrm{Tr}_{\mathbb{F}_{q^{mu}}/\mathbb{F}_{q^m}}(\gamma_i \mathbf{z}) \tag{3.4}$$

and let

$$\mathscr{Z} \stackrel{\mathrm{def}}{=} \mathrm{Span}_{\mathbb{F}_{q^m}}\{\mathbf{z}_1, \ldots, \mathbf{z}_u\}$$

be the (secret) code generated by the components of the error term in $\mathbf{k}_{\mathrm{pub}}$, and consider the (public) code

$$\mathscr{C}_{\mathrm{pub}} \stackrel{\mathrm{def}}{=} \mathrm{Gab}_k(\mathbf{g}) + \sum_{i=1}^{u} \mathbf{k}_{\mathrm{pub}}^{(i)}$$

which can be generated by the following matrix

$$\mathbf{M} \stackrel{\mathrm{def}}{=} \begin{pmatrix} \mathbf{G} \\ \mathbf{k}_{\mathrm{pub}}^{(1)} \\ \vdots \\ \mathbf{k}_{\mathrm{pub}}^{(u)} \end{pmatrix} \tag{3.5}$$

In the sequel, this is what we call a *supercode* of the Gabidulin code. Let $\mathbf{\Lambda}_i$ denote the i-th iterated Frobenius operator (with respect to q). Using the peculiar behaviour of $\mathrm{Gab}_k(\mathbf{g})$ with respect to $\mathbf{\Lambda}$, Gaborit, Otmani and Talé-Kalachi proved that when

$$\dim_{\mathbb{F}_{q^m}} \mathbf{\Lambda}_{n-k-w-1}(\mathscr{Z}) = w, \tag{3.6}$$

where w is the $\mathbb{F}_q$-rank of the error $\mathbf{z}$, then $\mathbf{\Lambda}_{n-w-k-1}(\mathscr{C}_{\text{pub}})^{\perp}$ is a one dimensional code, any basis of which reveals the secret key by solving a single linear system in $\mathbb{F}_{q^{mu}}$.

For Condition (3.6) to be verified, it is necessary that $w \leqslant \frac{u}{u+1}(n-k)$. On the other hand, taking w larger imposes to have a small t_{pub}, which makes the scheme vulnerable to generic decoding attacks. Moreover, they noticed that this condition was always verified in their experiments. In the rest of this section we propose an alternative attack based on decoding Gabidulin codes *on the right hand side*.

3.2.2 Interleaving Interpretation: a New Attack Against Faure-Loidreau

This attack, which we presented in [BC22], is inspired by the (second) attack from Coron on Augot-Finiasz cryptosystem in the Hamming metric. Starting from a basis $(\gamma_1,\dots,\gamma_u)$ of the extension $\mathbb{F}_{q^{mu}}/\mathbb{F}_{q^m}$, consider the decomposition of the public key into u noisy codewords of a Gabidulin code from Equation (3.3), and let $\mathbf{z}_i$ be defined as in Equation (3.4). It is readily seen that

$$\text{Rank}_q(\mathbf{z}_i) \leqslant \text{Rank}_q(\mathbf{z}) = w, \qquad \text{for } i \in \{1,\dots,u\},$$

and we could imagine applying a usual decoder such as Welch-Berlekamp to recover some $\mathbf{z}_i$ if their rank weight is too small. However, it turns out that in fact they have rank weight w with good probability. Furthermore, recovering $\mathbf{z}$ is equivalent to decoding *all of them*. In the Hamming metric though, the main part of Coron's attack was that the errors $\mathbf{z}_i$ (defined as in Equation (3.4)) were in fact *correlated* in the sense that more than having the same weight, they actually had the *same support* as the original $\mathbf{z}$. Such an error model in the Hamming metric is known as *interleaving*, and it is well-known that interleaved Reed-Solomon codes can be decoded slightly beyond than half the minimum distance with overwhelming probability (see for instance [Ber03; Zap20]). Indeed, the fact that the errors have the same support allows to use the same polynomial to *locate* all the errors, which decreases the overall number of unknowns in the linear system.

In the rank-metric situation though, this is not clear anymore. Loidreau and Overbeck adapted this approach in [LO06] when the errors have the same rank support, but unfortunately this is not the case here .... for the *column support*.

However, let $\mathcal{B} \stackrel{\text{def}}{=} \{\beta_1,\dots,\beta_m\}$ be a basis of $\mathbb{F}_{q^m}/\mathbb{F}_q$. Then, $\gamma\otimes\mathcal{B} \stackrel{\text{def}}{=} \{\beta_1\gamma_1,\dots,\beta_m\gamma_1,\beta_1\gamma_2,\dots,\beta_m\gamma_u\}$ is a basis of $\mathbb{F}_{q^{mu}}/\mathbb{F}_q$, and with the previous ordering

$$\text{Ext}_{\gamma\otimes\mathcal{B}}(\mathbf{z}) = \begin{pmatrix} \text{Ext}_{\mathcal{B}}(\mathbf{z}_1) \\ \vdots \\ \text{Ext}_{\mathcal{B}}(\mathbf{z}_u) \end{pmatrix} \in \mathbb{F}_q^{um\times n}$$

In other words, since the row support is an intrinsic notion which does not depend on the considered basis for the extension, we have $\text{RowSupp}(\mathbf{z}_i) \subset \text{RowSupp}(\mathbf{z})$ for $i \in \{1,\dots,u\}$. Recall that the basic idea of Welch-Berlekamp decoding algorithm is to find a q-polynomial $\mathbf{\Lambda}$ which vanishes on the (column) support of the error, namely such that $\mathbf{\Lambda}(\mathbf{z}_i) = 0$. In terms of matrices this corresponds to a *left-hand side* annihilator of $\text{Ext}(\mathbf{z}_i)$. Having information on the row supports motivates to working on the *right-hand side* instead.

3.2.3 Decoding on the right-hand side

Let $\mathbf{g} \stackrel{\text{def}}{=} (g_1,\dots,g_n) \in \mathbb{F}_{q^m}^n$ be of rank-weight n, and consider the Gabidulin code $\text{Gab}_k(\mathbf{g})$. Suppose we are given a noisy codeword $\mathbf{y} \stackrel{\text{def}}{=} \mathbf{c}+\mathbf{e}$, where $\mathbf{c} \stackrel{\text{def}}{=} C(\mathbf{g}) \in \text{Gab}_k(\mathbf{g})$ and $\mathbf{e}$ are unknown.

The goal is to recover $\mathbf{c}$. Let $t \stackrel{\text{def}}{=} |\mathbf{e}|_R$ be the rank weight of the error.

When working on the right-hand side, it is more convenient to consider the decoding problem at the q-polynomial level. Since the g_i's are linearly independent, we can find a q-polynomial Y of q-degree less than $n-1$ such that $Y(g_i) = y_i$. Indeed, let L_i be the vanishing polynomial of $\mathrm{Span}\{g_1, \ldots, g_{i-1}, g_{i+1}, \ldots, g_n\}$. In particular it is a q-polynomial of q-degree at most $n-1$ and $L_i(g_i) \neq 0$. Then

$$P \stackrel{\text{def}}{=} \sum_{i=1}^{n} \frac{y_i}{L_i(g_i)} L_i$$

interpolates $y_1, \ldots, y_n$ at $g_1, \ldots, g_n$. In other words, after linear interpolation (which can be performed at the receiver side), the decoding problem is equivalent to finding two q-polynomials C and E such that $\deg_q(C) < k$ and $\mathrm{Rank}(E) \leqslant t$ satisfying

$$Y = C + E \mod (X^{q^m} - X). \tag{3.7}$$

More precisely, let Y be the interpolator of $\mathbf{y}$ at $\mathbf{g}$, of minimal degree (*i.e.* $\deg_q(Y) < n$), and define $E \stackrel{\text{def}}{=} Y - C$. Since $\deg_q(C) < k \leqslant n$, this entails that $\deg_q(E) < n$.

The algorithm (which is summed up in Algorithm 3.10) will make use of the *adjoint* of a (class of) q-polynomial in $\mathbb{F}_{q^m}\langle X\rangle \big/ (X^{q^m} - X)$. Let $\mathbf{P} \in \mathbb{F}_{q^m}\langle X\rangle \big/ (X^{q^m} - X)$. Regarding $\mathbf{P}$ as an $\mathbb{F}_q$-linear endomorphism, we may consider its adjoint[ii] $P^\vee$ with respect to the non degenerate bilinear form given by the trace:

$$\begin{cases} \mathbb{F}_{q^m} \times \mathbb{F}_{q^m} & \longrightarrow & \mathbb{F}_q \\ (a, b) & \longmapsto & \mathrm{Tr}_{\mathbb{F}_{q^m}/\mathbb{F}_q}(ab) \end{cases}.$$

It can be explicitly computed with the following proposition.

Proposition 3.5

Let $P \stackrel{\text{def}}{=} \sum_{i=0}^{m-1} a_i X^{q^i} \in \mathbb{F}_{q^m}\langle X\rangle \big/ (X^{q^m} - X)$. Then

$$P^\vee \stackrel{\text{def}}{=} \sum_{i=0}^{m-1} a_i^{q^{m-i}} X^{q^{m-i}}.$$

Proof. First, note that the adjoint is anticommutative: for two $\mathbb{F}_q$-endomorphisms f, g of $\mathbb{F}_{q^m}$, then $(f \circ g)^\vee = g^\vee \circ f^\vee$.

With respect to the trace bilinear form, it is clear that for $\alpha \in \mathbb{F}_{q^m}^\times$, the scalar multiplication map $x \mapsto \alpha x$ is self-adjoint, *i.e.* $(\alpha X)^\vee = \alpha X$. Moreover, the Frobenius X^q is orthogonal. Indeed, for $x, y \in \mathbb{F}_{q^m}$

$$\mathrm{Tr}_{\mathbb{F}_{q^m}/\mathbb{F}_q}(xy^q) = \mathrm{Tr}_{\mathbb{F}_{q^m}/\mathbb{F}_q}((xy^q)^{q^{m-1}}) = \mathrm{Tr}_{\mathbb{F}_{q^m}/\mathbb{F}_q}(x^{q^{m-1}} y)$$

[ii] Let k be any field, and let V be a finite dimensional k vector-space endowed with a symmetric, non-degenerate bilinear form $B\colon V \times V \to k$. For an endomorphism $\varphi \in \mathrm{End}_k(V)$ its adjoint relative to B is defined as the endomorphism $\varphi^\vee$ characterised by $B(u, \varphi v) = B(\varphi^\vee u, v)$ for any $u, v \in V$.

In other words, $(X^q)^\vee = X^{q^{m-1}}$.

Therefore, by linearity we have

$$\begin{aligned} P^\vee &= \left(\sum_{i=0}^{m-1} a_i X^{q^i}\right)^\vee = \sum_{i=0}^{m-1} \left((a_i X) \circ (X^{q^i})\right)^\vee \\ &= \sum_{i=0}^{m-1} \left((X^{q^i})^\vee \circ (a_i X)^\vee\right) = \sum_{i=0}^{m-1} \left((X^{q^{m-i}}) \circ (a_i X)\right) \\ &= \sum_{i=0}^{m-1} a_i^{q^{m-i}} X^{q^{m-i}} \end{aligned}$$

□

We are now ready to describe the right-hand side decoding algorithm. We will consider two cases.

Case $n = m$. In this situation, $(g_1, \ldots, g_n)$ form a basis of the extension field $\mathbb{F}_{q^m}/\mathbb{F}_q$, and therefore $\mathrm{Rank}_q(E) = \mathrm{Rank}_q(\mathbf{e}) = t$.

In the usual Welch-Berlekamp algorithm, we look for a q-polynomial Λ of q-degree t such that $\Lambda \circ E = 0$. The following proposition is basically a dual version, working on the right-hand side.

Proposition 3.6

Let $\mathscr{V}$ be an $\mathbb{F}_q$-vector subspace of $\mathbb{F}_{q^m}$ of dimension d. Then, there exists a q-polynomial Λ of q-degree at most $m-d$ whose image is exactly $\mathscr{V}$.

Proof. Let $\mathscr{V}^\perp$ denote the orthogonal of $\mathscr{V}$ for the inner product $(a,b) \mapsto \mathrm{Tr}_{\mathbb{F}_{q^m}/\mathbb{F}_q}(xy)$. It is of degree $m-d$. Let P be the vanishing polynomial of $\mathscr{V}^\perp$ (as per Definition 2.13). It is of the form

$$P \stackrel{\text{def}}{=} \sum_{i=0}^{m-d} a_i X^{q^i}.$$

Let $\Lambda \stackrel{\text{def}}{=} (X^{q^d} \cdot P)^\vee$. We have

$$\Im(\Lambda) = \ker(X^{q^d} \cdot P)^\perp = \ker(P)^\perp = \left(\mathscr{V}^\perp\right)^\perp = \mathscr{V}.$$

Moreover,

$$\Lambda = P^\vee \cdot \left(X^{q^d}\right)^\vee = \left(\sum_{i=0}^{m-d} a_i^{q^{m-i}} X^{q^{m-i}}\right) \cdot X^{q^{-d}} = \sum_{i=0}^{m-d} a_i^{q^{m-i}} X^{q^{m-d-i}}.$$

In other words, the q-polynomial Λ, of q-degree $\leqslant m-d$, has image $\mathscr{V}$. □

Setting $\mathscr{V} \stackrel{\text{def}}{=} \ker(E)$ in Proposition 3.6, we immediately deduce the existence of a right annihilator.

Corollary 3.7

Let E be a q-polynomial of rank t. Then there exists a q-polynomial $\mathbf{\Lambda}$ such that $\deg_q(\mathbf{\Lambda}) \leqslant t$ and $E \circ \mathbf{\Lambda} = 0 \mod (X^{q^m} - X)$.

Remark 3.8. *Such a $\mathbf{\Lambda}$ of minimal degree is unique when we force it to be* monic.

Let $\mathbf{\Lambda}$ be the (unknown) monic q-polynomial given by Proposition 3.7. Using Equation (3.7), $\mathbf{\Lambda}$ satisfies

$$Y \circ \mathbf{\Lambda} = C \circ \mathbf{\Lambda} + E \circ \mathbf{\Lambda} = C \circ \mathbf{\Lambda} \mod (X^{q^m} - X),$$

which yields to a non linear system of n equations by evaluation on $\mathbf{g}$:

$$\begin{cases} (Y \circ \mathbf{\Lambda})(g_i) = (C \circ \mathbf{\Lambda})(g_i) \\ \deg_q(\mathbf{\Lambda}) \leqslant t \\ \deg_q(C) \leqslant k-1 \end{cases} \tag{3.8}$$

whose unknowns are the $t+1$ coefficients of $\mathbf{\Lambda}$, and the k coefficients of C, *i.e.* $t+k+1$ unknowns. This system can be linearised by setting $N \stackrel{\text{def}}{=} C \circ \mathbf{\Lambda}$, a q-polynomial of q-degree at most $k+t-1$. System (3.8) now becomes

$$\begin{cases} (Y \circ \mathbf{\Lambda})(g_i) = N(g_i) \\ \deg_q(\mathbf{\Lambda}) \leqslant t \\ \deg_q(N) \leqslant k-1+t \end{cases} \tag{3.9}$$

which has n equations and $k+2t+1$ unknowns. Obviously, any solution to (3.8) yields a solution to (3.9), however the latter is *a priori* more general. The following proposition addresses that gap when t is smaller than the unique decoding radius.

Proposition 3.9

Assume that E is of rank $t \leqslant \lfloor \frac{m-k}{2} \rfloor$ and $Y = C + E$ with $\deg_q(C) = k$. If $(\mathbf{\Lambda}, N)$ is *any* non-zero solution of (3.9), then $N = C \circ \mathbf{\Lambda}$.

Proof. Let $(\mathbf{\Lambda}, N) \neq (0,0)$ be a solution to (3.9), and let C be the q-polynomial of q-degree strictly less than k which interpolates the codeword. In particular,

$$Y \circ \mathbf{\Lambda} = N \mod (X^{q^m} - X).$$

Let $R \stackrel{\text{def}}{=} N - C \circ \mathbf{\Lambda}$. It is a q-polynomial of degree at most $k-1+t$. Assume that it is nonzero (in $\mathbb{F}_{q^m}\langle X \rangle / (X^{q^m} - X)$). Then,

$$(Y - C) \circ \mathbf{\Lambda} = Y \circ \mathbf{\Lambda} - C \circ \mathbf{\Lambda} = R \mod (X^{q^m} - X),$$

i.e.

$$E \circ \Lambda = R \mod (X^{q^m} - X).$$

In particular, $\mathrm{Rank}(R) \leqslant \mathrm{Rank}(E) = t$. On the other hand, since $R \neq 0$, we have $\dim \ker R \leqslant \deg_q R \leqslant k - 1 + t$. Therefore, by the rank-nullity theorem

$$m = \dim \ker R + \mathrm{Rank}(R) \leqslant k + 2t - 1 \leqslant k + 2\frac{m-k}{2} - 1 \leqslant m - 1 < m,$$

which is a contradiction. Hence, R must be zero, *i.e.* $N = C \circ V$. □

In other words, when $t \leqslant \lfloor \frac{n-k}{2} \rfloor$ (recall that $m = n$ here), solving (3.9) allows to recover C by (right) euclidean division, *i.e.* to decode. However, despite the transformation, the system is only linear over $\mathbb{F}_q$, while the unknowns are in $\mathbb{F}_{q^m}$. Fortunately, we can address this issue by using the adjoint once more. For $1 \leqslant i \leqslant n$, let $y_i^\vee \stackrel{\text{def}}{=} Y^\vee(g_i)$ (which can be computed at the receiver side). Equation (3.9) is equivalent to

$$\begin{cases} \Lambda^\vee(y_i^\vee) = N^\vee(g_i) \\ \deg_q(\Lambda) \leqslant t \\ \deg_q(N) \leqslant k - 1 + t \end{cases} \tag{3.10}$$

which is now *linear* over $\mathbb{F}_{q^m}$, and can efficiently be solved. Since there is a one-to-one correspondence between the coefficients of $\Lambda^\vee$ and $N^\vee$, and that of Λ and N, solving (3.10) allows to recover Λ and N.

This decoding algorithm is summed up in Algorithm 3.10.

Algorithm 3.10 : Right–hand side variant of Welch–Berlekamp

Input : q a prime power, k, n, m integers, $\mathbf{g} = (g_1, \ldots, g_n)$ a basis of $\mathbb{F}_{q^m}/\mathbb{F}_q$, a Gabidulin code $\mathscr{C}$ of dimension k and evaluation vector $\mathbf{g}$, an integer $t \leqslant \lfloor \frac{n-k}{2} \rfloor$, and $\mathbf{y} \in \mathbb{F}_{q^m}^n$.

Output : $\mathbf{c} \in \mathscr{C}$ such that $\mathbf{y} = \mathbf{c} + \mathbf{e}$ for some $\mathbf{e} \in \mathbb{F}_{q^m}^n$ with $\mathrm{Rank}(\mathbf{e}) \leqslant t$.

1 Find Y the q–polynomial of q–degree strictly less than n such that $Y(g_i) = y_i$
2 Compute $Y^\vee$ and evaluate on $\mathbf{g}$ to get $\mathbf{y}^\vee \stackrel{\text{def}}{=} Y^\vee(\mathbf{g}) \in \mathbb{F}_{q^m}^n$
3 Find a non zero solution (Λ_0, N_0) of the linear system (3.10)
4 Compute $\Lambda \stackrel{\text{def}}{=} \Lambda_0^\vee$ and $N \stackrel{\text{def}}{=} N_0^\vee$
5 Recover C by computing the right–hand side Euclidean division of N by Λ
6 **return** $\mathbf{c} \stackrel{def}{=} C(\mathbf{g})$

Case $n < m$. In this situation, it is still possible to interpolate the received word as

$$Y = C + E,$$

where both Y and E have q-degree less than n, and $\deg_q(C) < k$. However, it is no longer true that E has rank t as an endomorphism of $\mathbb{F}_{q^m}$: we can only affirm that *the restriction* of E to the linear span of $\mathbf{g}$ has rank t, but the whole E might have a larger rank. In order to remedy that, we will use Proposition 3.6 again: let G be the q-polynomial of q-degree at most $m - n$

whose image is exactly $\mathrm{Supp}(\mathbf{g})$. Then,

$$Y \cdot G = C \cdot G + E \cdot G$$

and $\deg_q(C \cdot G) < k + m - n$ corresponds to a codeword of $Gab_{k+m-n}(\mathbf{g})$. Note that Y and G can be computed at the receiver side, without the knowledge of the error. Moreover, since $\Im(G) = \mathrm{Supp}(\mathbf{g})$, we have that $\mathrm{Rank}(E \cdot G) = t$. Using the previous right hand side decoding algorithm, it is possible to recover $C \cdot G$, and then C by euclidean division on the right, as long as

$$t \leqslant \frac{m - (k + m - n)}{2} = \frac{n - k}{2}.$$

3.2.4 An alternative key recovery attack against Faure-Loidreau

Using the above right-hand side decoder, we can provide an alternative key recovery attack, in the spirit of that of Coron in the Hamming metric.

Recall from Equation (3.1) that the public key is of the form

$$\mathbf{k}_{\mathrm{pub}} = \mathbf{x} \cdot \mathbf{G} + \mathbf{z} \in \mathbb{F}_{q^{mu}},$$

where $\mathbf{G}$ is a generator matrix of a (public) Gabidulin code, and from Equation (3.3) that it can be seen as u codewords by using the trace operator

$$\mathbf{k}_{\mathrm{pub}}^{(j)} \stackrel{\mathrm{def}}{=} \mathrm{Tr}_{\mathbb{F}_{q^{mu}}/\mathbb{F}_{q^m}}(\gamma_j \mathbf{k}_{\mathrm{pub}}) = \mathbf{x}_j \cdot \mathbf{G} + \mathbf{z}_j, \quad \text{for } 1 \leqslant i \leqslant u.$$

Recovering the secret key is therefore equivalent to decoding the u components $\mathbf{k}_{\mathrm{pub}}^{(j)}$, in parallel. *A priori*, the $\mathbf{z}_j$'s have a too large rank weight w to be decodable. Nevertheless, they are strongly correlated since they have the same row support, namely $\mathrm{RowSupp}(\mathbf{z})$, of dimension at most w. In particular, they have the same right-hand side annihilator $\mathbf{\Lambda}$ of degree at most w.

Denote by $K_j \stackrel{\mathrm{def}}{=} C_j + Z_j$ the q-polynomials of q-degree less than n which interpolate $\mathbf{k}_{\mathrm{pub}}^{(j)}$ at $\mathbf{g}$. Even if it means multiplying on the right by a q-polynomial of q-degree $\leqslant m - n$, we may assume that $m = n$ here. The previous remark yields the following system

$$\begin{cases} (K_j \circ \mathbf{\Lambda})(g_i) = (C_j \circ \mathbf{\Lambda})(g_i) \\ \deg_q(\mathbf{\Lambda}) \leqslant w \\ \deg_q(C_j) \leqslant k - 1 \end{cases}, \quad \text{for } j \in \{1, \ldots, u\}. \tag{3.11}$$

which can be linearised by setting $N_i \stackrel{\mathrm{def}}{=} C_i \circ \mathbf{\Lambda}$:

$$\begin{cases} (K_j \circ \mathbf{\Lambda})(g_i) = N_j(g_i) \\ \deg_q(\mathbf{\Lambda}) \leqslant w \\ \deg_q(C_j) \leqslant k - 1, \end{cases}, \quad \text{for } j \in \{1, \ldots, u\}. \tag{3.12}$$

This system has $n \times u$ equations and $w + 1 + u(k + w)$ unknowns, and therefore one can expect to retrieve $(\mathbf{\Lambda}, N_1, \ldots, N_u)$, and then $C_1, \ldots, C_u$ by euclidean division on the right, whenever

$$w \leqslant \frac{u}{u+1}(n - k),$$

which is slightly better than $\frac{n-k}{2}$, and always verified by the parameters of Faure-Loidreau. This is exactly the same condition as in [GOT18].

Remark 3.11. *To be rigorous, there needs an additional step using the adjoint operator, similarly as when decoding a single word.*

3.3 Two Independent Repairs: LIGA and RAMESSES

The alternative attack presented in Section 3.2.4 is actually an example of a decoding algorithm for an *interleaved Gabidulin* code, and the approach by decoding on the right hand side is not the only way of decoding such code. For example, [SJB11] provides an efficient syndrome-based algorithm, and analyse their decoding failures. See also [Wac13] for a survey on this topic.

The relationship between the attack from [GOT18] and decoding interleaved Gabidulin codes was already emphasised by Renner, Puchinger and Wachter-Zeh in [RPW21][iii]. In this paper, they introduce a variant of Faure-Loidreau called LIGA[iv] where they propose to introduce an additional structure on the cryptosystem in order to force every known interleaved decoder to fail: they explicit a decoding failure condition, and change the key generation algorithm. In particular, their proposal was not vulnerable to the attack from [GOT18].

An independent repair of Faure-Loidreau cryptosystem called RAMESSES was proposed in [LLP20]. It can be somehow considered as a dual version of Faure-Loidreau, where the plaintext is encoded as the row-space of some error. This avoids the necessity to use an additional extension $\mathbb{F}_{q^{mu}}$ and the trace map.

In reality, as we will see, in order to thwart attacks on the key, those cryptosystems introduce too much structure, which makes them vulnerable to a direct message recovery attack, as we have shown in [BC21].

3.3.1 LIGA Encryption scheme

Recall that the public key of Faure-Loidreau is of the form

$$\mathbf{k}_{\text{pub}} \stackrel{\text{def}}{=} \mathbf{x} \cdot \mathbf{G} + \mathbf{z} \in \mathbb{F}_{q^{mu}}^{n},$$

where $\mathbf{x}, \mathbf{z} \in \mathbb{F}_{q^{mu}}$ such that $\text{Rank}_{\mathbb{F}_q}(\mathbf{z}) = w$ and $\mathbf{G}$ is the generator matrix of a public Gabidulin code of dimension k.

From Section 3.2.1.2, the attack from [GOT18] fails when Condition 3.6 is not satisfied, *i.e.*

$$\dim_{\mathbb{F}_{q^m}} \Lambda_{n-k-w-1}(\mathscr{Z}) < w$$

where $\mathscr{Z}$ is the vector space generated (over $\mathbb{F}_{q^m}$) by the u components of $\mathbf{z}$. Let $\zeta \stackrel{\text{def}}{=} \text{Rank}_{\mathbb{F}_{q^m}}(\mathbf{z})$. Based on the behaviour of known decoders for interleaved Gabidulin codes, [RPW21, Theorem 4] shows that for Condition 3.6 to fail, it is enough to enforce

$$\zeta < \frac{w}{n-k-w}. \tag{3.13}$$

This leads to the following modification of Faure-Loidreau cryptosystem, where the Encryption and Decryption procedure remain identical to the original proposal, while the Key Generation is modified in order to enforce Condition 3.13.

[iii] A short version of this paper already appeared in [WPR18]

[iv] because it is based on the hardness of List and Interleaved decoding of Gabidulin codes

Key Generation LIGA. The owner of the secret key picks two random bases $\boldsymbol{\gamma} = (\gamma_1, \ldots, \gamma_u)$ and $\boldsymbol{\beta} \stackrel{\text{def}}{=} (\beta_1, \ldots, \beta_u)$ of $\mathbb{F}_{q^{mu}}/\mathbb{F}_{q^m}$, a non singular matrix $\mathbf{P} \leftarrow \mathrm{GL}_n(\mathbb{F}_q)$, and a random vector $\boldsymbol{\xi} \leftarrow \mathbb{F}_{q^{mu}}^{k-u}$. They then set $\mathbf{x} \stackrel{\text{def}}{=} (\boldsymbol{\xi} \mid \boldsymbol{\beta}) \in \mathbb{F}_{q^{mu}}^{k}$.

Furthermore, they generate randomly a matrix

$$\mathbf{S} = \begin{pmatrix} \mathbf{s}_1 \\ \vdots \\ \mathbf{s}_u \end{pmatrix} \in \mathbb{F}_{q^m}^{u \times n}$$

of $\mathbb{F}_{q^m}$-rank ζ and such that the $\mathbb{F}_q$-rank of all the rows is w; and set

$$\mathbf{s} \stackrel{\text{def}}{=} \mathrm{Ext}_{\gamma}^{-1}(\mathbf{S}) = \sum_{i=1}^{u} \mathbf{s}_i \gamma_i^*.$$

Finally, they set

$$\mathbf{z} \stackrel{\text{def}}{=} (\mathbf{s} \mid \mathbf{0}_{n-w}) \cdot \mathbf{P}^{-1} \in \mathbb{F}_{q^{mu}}^{n}.$$

As for Faure-Loidreau cryptosystem, the public key is

$$\mathbf{k}_{\text{pub}} \stackrel{\text{def}}{=} \mathbf{x}\mathbf{G} + \mathbf{z} \in \mathbb{F}_{q^{mu}}^{n},$$

while the secret key is formed by $(\mathbf{x}, \mathbf{z}, \mathbf{P})$.

The generation of $\mathbf{S}$ (and hence $\mathbf{z}$) is done by rejecting poorly formed matrices. In practice, a matrix of $\mathbb{F}_{q^m}$-rank ζ will satisfy this condition with very high probability.

Interpretation of ζ in view of the right-hand side decoding. Using the right-hand side decoder presented in Section 3.2.4, this quantity ζ has in reality a very simple explanation. Indeed, this is *exactly* the number of errors which are linearly independent. In terms of decoding, this corresponds to the number of independent decoding problems (a linear dependency in the errors yields a linear dependency in the different decoding problems, even if it means replacing an unknown codeword by another unknown codeword of the same code). In other words, Systems 3.11 and 3.12, have exactly $\zeta \times n$ independent equations, and $w + 1 + \zeta(k + w)$ unknowns. The very same attack is therefore expected to succeed, as long as

$$w \leqslant \frac{\zeta}{\zeta + 1}(n - k),$$

which is exactly equivalent to (3.13).

3.3.2 RAMESSES

In this section we present RAMESSES encryption scheme in terms of q-polynomials, using linear interpolation. In particular, we will fix an evaluation vector $\mathbf{g} \in \mathbb{F}_{q^m}^n$ of rank weight n, as well as a basis $\mathcal{B}$ of $\mathbb{F}_{q^m}/\mathbb{F}_q$, and the space $\mathbb{F}_{q^m}\langle X\rangle_{<k}$ will be regarded as a Gabidulin code of dimension k. Basis $\mathcal{B}$ allows to represent any element of $\mathbb{F}_{q^m}\langle X\rangle \big/ (X^{q^m} - X)$ as an $m \times m$ matrix, and reciprocally any $m \times m$ matrix defines a unique q-polynomial of q-degree less than m in the basis $\mathcal{B}$. We consider the situation of full length Gabidulin codes.

Remark 3.12. *Note that this is not exactly the point of view adopted in the original presentation [LLP20], however they are both completely equivalent, and it makes the presentation of the attack easier. We discuss this equivalence in [BC21, Appendix A].*

Public Parameters. Let k, ℓ, m, t, w be positive integers such that $1 \leqslant k, \ell, t, w \leqslant m$ and

$$t \leqslant \frac{m-k-\ell-w}{2},$$

and

$$w > \frac{m-k}{2}.$$

Key Generation. The owner of the secret key picks a uniformly random q-polynomial K_{sec} of rank w to be the secret key, while the public key is the affine space

$$\mathscr{C}_{\text{pub}} \stackrel{\text{def}}{=} \mathbb{F}_{q^m}\langle X\rangle_{<k} + K_{\text{sec}}.$$

In practice, the secret key is a uniformly random vector $\mathbf{k}_{\text{sec}} \in \mathbb{F}_{q^m}^m$ of rank w, and the public key is its syndrome

$$\mathbf{k}_{\text{pub}} \stackrel{\text{def}}{=} \mathbf{H}\mathbf{k}_{\text{sec}}^\top$$

with respect to a fixed parity-check matrix of the Gabidulin code.

Encryption. The plaintext $\mathbf{m}$ is a t-dimensional $\mathbb{F}_q$–subspace of $\mathbb{F}_{q^m}$. It is encrypted as follows:

1. Pick a uniformly random $T \in \mathbb{F}_{q^m}\langle X\rangle$ of q–degree ℓ
2. Pick a uniformly random $E \in \mathbb{F}_{q^m}\langle X\rangle_{<m}$ whose matrix representation admits $\mathbf{m}$ as its row space, equivalently E is such that $\mathbf{m}$ is the image of $E^\vee$. In particular, $\mathrm{Rank}(E) = t$.
3. Pick a uniformly random $C \in \mathbb{F}_{q^m}\langle X\rangle_{<k}$
4. Pick a uniformly random $C_0 \in \mathbb{F}_{q^m}\langle X\rangle_{<k}$, yielding a uniformly random
$$C' = C_0 + K_{\text{sec}} \in \mathscr{C}_{\text{pub}}.$$

The ciphertext is

$$Y \stackrel{\text{def}}{=} C + C' \circ T + E.$$

Note that, this ciphertext satisfies

$$Y = C_1 + K_{\text{sec}} \circ T + E, \tag{3.14}$$

where

$$C_1 = C + C_0 \circ T \text{ is of } q\text{-degree } < k + \ell.$$

More precisely, C_1 lies in the code $\mathbb{F}_{q^m}\langle X\rangle_{<k} + \mathbb{F}_{q^m}\langle X\rangle_{<k} \circ T$ which is slightly bigger than the public Gabidulin code. This C_1 is *a priori* unknown by anyone.

Decryption. The knowledge of the secret key K_{sec} means knowledge of the minimal vanishing q-polynomial of the rank support of $\mathbf{k}_{\text{sec}}$, *i.e.* a q-polynomial V of q-degree at most w such that $V \circ K_{\text{sec}} = 0 \mod (X^{q^m} - X)$.

Upon receiving a ciphertext Y, they can therefore compute

$$V \circ Y = V \circ C_1 + V \circ E \mod (X^{q^m} - X).$$

Now, $V \circ C_1 \in \mathbb{F}_{q^m}\langle X \rangle_{k+\ell+w}$ lies in a Gabidulin code of dimension $k+\ell+w$, while

$$\text{Rank}(V \circ E) \leqslant \text{Rank}(E) = t \leqslant \frac{m-k-\ell-w}{2}.$$

In particular, $V \circ E$ can be recovered, and when it has rank exactly t, its row space is that of E. In that case, the plaintext can be recovered. Otherwise, there is a decryption failure, but this happens with low probability.

3.4 A Message Recovery Attack Against LIGA and RAMESSES

In this section, we prove that the structure introduced in the public key of LIGA in order to thwart the key recovery attack by decoding an interleaved Gabidulin code, actually makes it vulnerable to a direct message recovery attack.

Indeed, the public key $\mathbf{k}_{\text{pub}} = \mathbf{x} \cdot \mathbf{G} + \mathbf{z}$ is such that the $\mathbb{F}_{q^m}$-rank of $\mathbf{z}$ is small (while keeping its $\mathbb{F}_q$-rank large). Denote by ζ this rank. In other words, there exist $\mathbf{z}_1, \ldots, \mathbf{z}_\zeta \in \mathbb{F}_{q^m}^n$ and $\mu_1, \ldots, \mu_\zeta \in \mathbb{F}_{q^{mu}}$, both linearly independent over $\mathbb{F}_{q^m}$, such that

$$\mathbf{z} = \sum_{i=1}^{\zeta} \mu_i \mathbf{z}_i.$$

Therefore, a ciphertext will be of the form (see (3.2))

$$\mathbf{c} \stackrel{\text{def}}{=} \mathbf{m}' \cdot \mathbf{G} + \underbrace{\sum_{i=1}^{\zeta} \text{Tr}_{\mathbb{F}_{q^{mu}}/\mathbb{F}_{q^m}}(\alpha\mu_i)\mathbf{z}_i + \mathbf{e}}_{\stackrel{\text{def}}{=} \mathbf{e}'}. \tag{3.15}$$

The main point of Faure-Loidreau, and then LIGA, is that $\mathbf{c}$ is a codeword of a Gabidulin code, corrupted by the error $\mathbf{e}'$ of very large weight. But what if we added part of the error inside the code ? Indeed, the major part of the error $\mathbf{e}'$ is carried by the $\mathbf{z}_i$, and in reality the error $\mathbf{e}$ is very small. In general, adding a random subspace to a given code $\mathscr{C}$ drastically deteriorates the decoding capabilities since it increases the dimension, and it may also destroy the minimum distance. However, Gabidulin codes are so good, and the error $\mathbf{e}$ is so small (by design), that it may be possible to remove it somehow. This is the idea of the message recovery attack.

Indeed, let

$$\mathscr{C} \stackrel{\text{def}}{=} \text{Gab}_k(\mathbf{g}) + \text{Span}_{\mathbb{F}_{q^m}}\{\mathbf{z}_1, \ldots, \mathbf{z}_\zeta\} \subset \mathbb{F}_{q^m}^n$$

so that a ciphertext $\mathbf{c}$ can be considered as a codeword of $\mathscr{C}$ corrupted by the small error $\mathbf{e}$. The idea is that, by design, in order to thwart the attack from [GOT18], $\mathscr{C}$ was forced to be very close to a Gabidulin code. We call such a code a *supercode* of a Gabidulin code (in opposition to subcodes). First, we show how we can modify Welch-Berlekamp algorithm in order to decode

such supercodes, at the price of decreasing the decoding capability, before we apply it to LIGA (and RAMESSES).

3.4.1 Decoding Supercodes of Gabidulin Codes

It will be more convenient for this section to give the algorithm directly at the q-polynomial level. In particular, we will omit linear interpolations, and all the codes are considered as subspaces of $\mathbb{F}_{q^m}\langle X\rangle \big/ (X^{q^m} - X)$.

Consider a code $\mathscr{C}$ of the form

$$\mathscr{C} \stackrel{\text{def}}{=} \mathcal{G} \oplus \mathcal{T},$$

where $\mathcal{G} \stackrel{\text{def}}{=} \mathbb{F}_{q^m}\langle X\rangle_{<k}$ is a Gabidulin code of dimension k, and $\mathcal{T} \subset \mathbb{F}_{q^m}\langle X\rangle \big/ (X^{q^m} - X)$, and let

$$Y \stackrel{\text{def}}{=} (C_0 + T) + E$$

be a noisy codeword, $C_0 \in \mathcal{G}, T \in \mathcal{T}$ and E is a q-polynomial of rank t. Let Λ with $\deg_q(\Lambda) \leqslant t$ be a left annihilator for E. Solving the decoding problem amounts therefore to solve

$$\Lambda \circ Y = \Lambda \circ C = \Lambda \circ C_0 + \Lambda \circ T \mod (X^{q^m} - X),$$

where the unknowns are Λ and C. As usual, this system can be linearised into

$$\Lambda \circ Y = N \mod (X^{q^m} - X), \tag{3.16}$$

where $N \in (\mathbb{F}_{q^m}\langle X\rangle_{\leqslant t} \circ \mathbb{F}_{q^m}\langle X\rangle_{<k}) + (\mathbb{F}_{q^m}\langle X\rangle_{\leqslant t} \circ \mathscr{T}) = \mathbb{F}_{q^m}\langle X\rangle_{<k+t} + \mathbb{F}_{q^m}\langle X\rangle_{\leqslant t} \circ \mathscr{T}$.

Lemma 3.13

Under the assumption that $(\mathbb{F}_{q^m}\langle X\rangle_{<k+t} \circ \mathbb{F}_{q^m}\langle X\rangle_{\leqslant t} \circ \mathscr{T}) \cap (\mathbb{F}_{q^m}\langle X\rangle_{\leqslant t} \circ E) = \{0\}$, then any nonzero solution (Λ, N) of (3.16) satisfies $\Lambda \circ E = 0$.

Proof. Let (Λ, N) be such a nonzero solution. Then

$$\Lambda \circ Y - \Lambda \circ C = \Lambda \circ E \mod (X^{q^m} - X).$$

On the other hand, by definition $\Lambda \circ Y = N \mod (X^{q^m} - X)$, and therefore the left hand side is contained in $(\mathbb{F}_{q^m}\langle X\rangle_{<k+t} \circ \mathbb{F}_{q^m}\langle X\rangle_{\leqslant t} \circ \mathscr{T})$, while the right hand side is contained in $\mathbb{F}_{q^m}\langle X\rangle_{\leqslant t} \circ \mathscr{T}$. Therefore, by assumption, both sides are zero. □

Under the hypothesis of Lemma 3.13, the decoding can be performed as follows

1. Solve (3.16).
2. Take any nonzero solution (Λ, N) and compute the right kernel of Λ. This kernel contains the image of E, and hence the (column) support of the error.
3. Knowing the support of E, one can recover E completely by solving a linear system.

Remark 3.14. *Note that step* 3 *is important, since as for the decryption of RAMESSES, left euclidean division of N by Λ is not sufficient. Indeed, Lemma 3.13 only permits to assert that $N = (\Lambda \circ C) \mod (X^{q^m} - X)$, but not all representative are equivalent. In the Gabidulin situation, the fact that $\deg_q(C) < k$ yields $\deg_q(\Lambda \circ C) < m$ and therefore the equality $N = \Lambda \circ C$ also holds directly in the full ring $\mathbb{F}_{q^m}\langle X\rangle$ of q-polynomials. In our setting, this is no longer true.*

For the decoding to succeed, the condition $(\mathbb{F}_{q^m}\langle X\rangle_{<k+t} \circ \mathbb{F}_{q^m}\langle X\rangle_{\leqslant t} \circ \mathscr{T}) \cap (\mathbb{F}_{q^m}\langle X\rangle_{\leqslant t} \circ E) = \{0\}$ needs to be satisfied. In the Gabidulin situation, *i.e.* when $\mathscr{T} = \{0\}$, this condition is guaranteed by a minimum distance argument entailing that $\mathbb{F}_{q^m}\langle X\rangle_{<k+t} \cap \mathbb{F}_{q^m}\langle X\rangle_{\leqslant t} \circ E = \{0\}$ as long as $t \leqslant \frac{n-k}{2}$. However, when $\mathscr{T}$ is larger, it is a difficult task to estimate the minimum distance. Nevertheless, in general this will hold as long as the sum of the dimension is less than that of the ambient space. Therefore, heuristically we can expect to correct almost any error of rank t as long as

$$k + 2t + \dim(\mathbb{F}_{q^m}\langle X\rangle_{\leqslant t} \circ \mathscr{T}) \leqslant n. \tag{3.17}$$

Remark 3.15. *For $\mathscr{T} = \{0\}$, this recovers exactly the decoding radius of Gabidulin codes.*

For the cryptanalysis of RAMESSES we actually need to be able to decode on the right-hand side. Fortunately, this extension is straightforward, and decoding is also possible when

$$k + 2t + \dim(\mathscr{T} \circ \mathbb{F}_{q^m}\langle X\rangle_{\leqslant t}) \leqslant n. \tag{3.18}$$

3.4.2 Applications to RAMESSES

We shall begin with describing the attack against RAMESSES, since the attack against LIGA will be more challenging.

Recall from (3.14) that a ciphertext of RAMESSES is of the form

$$Y = C_1 + K_{\text{sec}} \circ T + E,$$

where $\deg_q(C_1) < k + \ell, \deg_q(T) = \ell$ and E is a q-polynomial of rank t, whose row-space (*i.e.* the image of $E^\vee$) represents the plaintext. In other words, it suffices to recover E to get the plaintext.

Let

$$\mathscr{T} \stackrel{\text{def}}{=} K_{\text{sec}} \circ \mathbb{F}_{q^m}\langle X\rangle_{\leqslant \ell}$$

so that Y can be seen as a codeword of the supercode $\mathscr{C} \stackrel{\text{def}}{=} \mathbb{F}_{q^m}\langle X\rangle_{<k+\ell} + \mathscr{T}$ (of dimension $k + \ell$) corrupted by the error E of rank t. According (3.18), the right-hand side version of the above decoder will likely return pairs of the form (Λ, N) with $E \circ \Lambda = 0$ as soon as

$$n \geqslant (k + \ell) + 2t + \dim\left(K_{\text{sec}} \circ \mathbb{F}_{q^m}\langle X\rangle_{\leqslant \ell} \circ \mathbb{F}_{q^m}\langle X\rangle_{\leqslant t}\right) = k + 2\ell + 3t + 1 \tag{3.19}$$

Table 3.1 shows that this condition is always satisfied with the parameters proposed in [LLP20], which are therefore broken in polynomial time. The column representing the security (in bits) was computed in the paper using known attacks such as generic decoding algorithms for the rank metric. The last row further targets a decryption failure rate of less than 2^{-128}.

$m\ (=n)$	k	w	ℓ	t	Claimed Security (bits)	$k+3t+2\ell+1$
64	32	19	3	5	141	54
80	40	23	3	7	202	68
96	48	27	3	9	265	82
164	116	27	3	9	$\geqslant 256$	150

Table 3.1: This table compares the values of the formula (3.19) with the parameters proposed for RAMESSES. The first three rows are parameters for RAMESSES as a KEM and the last one are parameters for RAMESSES as a PKE. Note that for any proposed parameter set, we have $m = n$.

3.4.3 A message recovery attack against LIGA

Finally, let us consider LIGA encryption scheme. Starting from (3.15), a ciphertext of LIGA is of the form

$$\mathbf{c} \stackrel{\text{def}}{=} \mathbf{m}'\mathbf{G} + \sum_{i=1}^{\zeta} \mathrm{Tr}_{\mathbb{F}_{q^{mu}}/\mathbb{F}_{q^m}}(\alpha\mu_i)\mathbf{z}_i + \mathbf{e}$$

and can be considered as a codeword of the code

$$\mathscr{C} \stackrel{\text{def}}{=} \mathrm{Gab}_k(\mathbf{g}) + \mathrm{Span}\{\mathbf{z}_1, \ldots, \mathbf{z}_\zeta\} \subset \mathbb{F}_{q^m}^n$$

corrupted by an error of rank weight $t \leqslant \frac{n-k-w}{2}$. Note that, $\mathscr{C}$ is a supercode of $\mathrm{Gab}_k(\mathbf{g})$ with $\mathscr{T} \stackrel{\text{def}}{=} \mathrm{Span}\{\mathbf{z}_1, \ldots, \mathbf{z}_\zeta\}$ of dimension ζ. However, $\mathscr{T}$ is *a priori* secret, and it is no longer possible to directly apply a decoding algorithm for this supercode. Nevertheless, it can be computed from public data. The attack will proceed in three steps.

Step 1: Get rid of the small error. Let $\gamma_1, \ldots, \gamma_\zeta \in \mathbb{F}_{q^{mu}}$ be linearly independent over $\mathbb{F}_{q^m}$ (a priori not a basis since $\zeta < u$), and define the public code

$$\mathscr{C}_{\text{pub}} \stackrel{\text{def}}{=} \mathrm{Gab}_k(\mathbf{g}) + \sum_{i=1}^{\zeta} \mathrm{Tr}_{\mathbb{F}_{q^{mu}}/\mathbb{F}_{q^m}}(\gamma_i \mathbf{k}_{\text{pub}}).$$

If γ was allowed to form a basis of the extension, then both codes $\mathscr{C}$ and $\mathscr{C}_{\text{pub}}$ would be equal. However, for the application it is important that we keep γ of minimal size ζ, and therefore it is not clear anymore that the equality holds. Nevertheless, it will be true with overwhelming probability over the choice of the γ_i's, as shown by the following theorem.

Theorem 3.16

Over the uniform choice of linearly independent $\gamma_1 \ldots, \gamma_\zeta$, we have

$$\mathbb{P}(\mathscr{C} = \mathscr{C}_{\text{pub}}) = 1 - e^{O\left(\frac{1}{q^m}\right)}.$$

The proof of Theorem 3.16 rests on the following technical counting argument.

Lemma 3.17

Let F be a linear subspace of dimension m in a linear space E of dimension n over a finite field $\mathbb{F}_q$. Then, $|\{G \mid F \oplus G = E\}| = q^{m(n-m)}$.

Proof. Let $\mathrm{Stab}(F)$ denote the stabiliser of F under the action of $\mathrm{GL}(E)$. It is isomorphic to the group of the matrices of the form $\begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{0} & \mathbf{C} \end{pmatrix}$ with $\mathbf{A} \in \mathrm{GL}_m(\mathbb{F}_q), \mathbf{C} \in \mathrm{GL}_{n-m}(\mathbb{F}_q)$ and $\mathbf{B} \in \mathbb{F}_q^{m\times(n-m)}$. In other words,

$$\mathrm{Stab}(F) \simeq \mathbb{F}_q^{m\times(n-m)} \rtimes (\mathrm{GL}_m(\mathbb{F}_q) \times \mathrm{GL}_{n-m}(\mathbb{F}_q))$$

given by the split exact sequence

$$1 \to \begin{pmatrix} \mathbf{I}_m & \mathbb{F}_q^{m\times(n-m)} \\ \mathbf{0} & \mathbf{I}_{n-m} \end{pmatrix} \to \mathrm{Stab}(F) \to \begin{pmatrix} \mathrm{GL}_m(\mathbb{F}_q) & \mathbf{0} \\ \mathbf{0} & \mathrm{GL}_{n-m}(\mathbb{F}_q) \end{pmatrix} \to 1.$$

Moreover, $\mathrm{Stab}(F)$ acts transitively on the complement spaces of F. Indeed, let G and G' be such that $F \oplus G = F \oplus G' = E$. Let $(f_1, \ldots, f_m)$ be a basis of F and $(g_1, \ldots, g_{n-m})$ (respectively $(g'_1, \ldots, g'_{n-m})$) be a basis of G (resp. G'). Then the linear map that stabilises F and maps g_i onto g'_i is an element of $\mathrm{Stab}(F)$ that maps G onto G'. The stabiliser of a complement G under this action is simply $\mathrm{GL}_m(\mathbb{F}_q) \times \mathrm{GL}_{n-m}(\mathbb{F}_q)$. Therefore, the orbit-stabiliser theorem yields

$$|\{G \mid F \oplus G = E\}| = \frac{|\mathrm{Stab}F|}{|\mathrm{Stab}G|} = \frac{|\mathrm{GL}_m(\mathbb{F}_q) \times \mathrm{GL}_{n-m}(\mathbb{F}_q)| \times q^{m\times(n-m)}}{|\mathrm{GL}_m(\mathbb{F}_q) \times \mathrm{GL}_{n-m}(\mathbb{F}_q)|} = q^{m\times(n-m)}.$$

□

We are now ready to prove Theorem 3.16.

Proof of Theorem 3.16.. We wish to estimate the probability that $\mathscr{C} = \mathscr{C}_{\mathrm{pub}}$. Note first that inclusion $\supseteq$ is always satisfied. Indeed, let $\mathbf{c} \in \mathscr{C}_{pub}$. Then, there exist $\mathbf{m} \in \mathbb{F}_{q^m}^k$ and $\lambda_1, \ldots, \lambda_\zeta \in \mathbb{F}_{q^m}$ such that

$$\begin{aligned} \mathbf{c} &= \mathbf{m}\mathbf{G} + \sum_{i=1}^{\zeta} \lambda_i \mathrm{Tr}_{\mathbb{F}_{q^{mu}}/\mathbb{F}_{q^m}}(\gamma_i \mathbf{k}_{\mathrm{pub}}) \\ &= \left(\mathbf{m} + \sum_{i=1}^{\zeta} \lambda_i \mathrm{Tr}_{\mathbb{F}_{q^{mu}}/\mathbb{F}_{q^m}}(\gamma_i \mathbf{x})\right)\mathbf{G} \quad + \sum_{i=1}^{\zeta}\sum_{j=1}^{\zeta} \lambda_j \mathrm{Tr}_{\mathbb{F}_{q^{mu}}/\mathbb{F}_{q^m}}(\gamma_j \mu_i)\mathbf{z}_i \end{aligned}$$

and $\mathbf{c} \in \mathscr{C}$. Therefore, we are reduced to study the probability that $\mathscr{C} \subseteq \mathscr{C}_{\mathrm{pub}}$.

Let $\mathbf{c} \in \mathscr{C}$. There exists $\mathbf{m} \in \mathbb{F}_{q^m}^k$ and $\lambda_1, \ldots, \lambda_\zeta \in \mathbb{F}_{q^m}$ such that

$$\mathbf{c} = \mathbf{m}\mathbf{G} + \sum_{i=1}^{\zeta} \lambda_i \mathbf{z}_i.$$

If we can find $\boldsymbol{\alpha} \stackrel{\text{def}}{=} (\alpha_1, \ldots, \alpha_\zeta) \in \mathbb{F}_{q^m}^\zeta$ such that

$$\mathbf{c} - \sum_{i=1}^{\zeta} \alpha_i \mathrm{Tr}_{\mathbb{F}_{q^{mu}}/\mathbb{F}_{q^m}}(\gamma_i \mathbf{k}_{\text{pub}}) \in \mathrm{Gab}_k(\mathbf{g}),$$

then we are done. Unfolding the computation,

$$\mathbf{c} - \sum_{i=1}^{\zeta} \alpha_i \mathrm{Tr}_{\mathbb{F}_{q^{mu}}/\mathbb{F}_{q^m}}(\gamma_i \mathbf{k}_{\text{pub}}) =$$

$$\left(\mathbf{m} - \sum_{i=1}^{\zeta} \alpha_i \mathrm{Tr}_{\mathbb{F}_{q^{mu}}/\mathbb{F}_{q^m}}(\gamma_i \mathbf{x}) \right) \mathbf{G} + \sum_{i=1}^{\zeta} \left(\lambda_i - \sum_{j=1}^{\zeta} \alpha_j \mathrm{Tr}_{\mathbb{F}_{q^{mu}}/\mathbb{F}_{q^m}}(\gamma_j \mu_i) \right) \mathbf{z}_i.$$

It suffices to choose α such that $\lambda_i - \sum_{j=1}^{\zeta} \alpha_j \mathrm{Tr}_{\mathbb{F}_{q^{mu}}/\mathbb{F}_{q^m}}(\gamma_j \mu_i) = 0$ for $i \in \{1, \ldots, \zeta\}$, *i.e.*

$$(\lambda_1, \ldots, \lambda_\zeta) = (\alpha_1, \ldots, \alpha_\zeta) \begin{pmatrix} \mathrm{Tr}_{\mathbb{F}_{q^{mu}}/\mathbb{F}_{q^m}}(\gamma_1\mu_1) & \cdots & \mathrm{Tr}_{\mathbb{F}_{q^{mu}}/\mathbb{F}_{q^m}}(\gamma_1\mu_\zeta) \\ \vdots & \ddots & \vdots \\ \mathrm{Tr}_{\mathbb{F}_{q^{mu}}/\mathbb{F}_{q^m}}(\gamma_\zeta\mu_1) & \cdots & \mathrm{Tr}_{\mathbb{F}_{q^{mu}}/\mathbb{F}_{q^m}}(\gamma_\zeta\mu_\zeta) \end{pmatrix}.$$

Let $\mathbf{M}$ denote this last matrix. The previous remark implies

$$\mathbb{P}(\mathscr{C} \subseteq \mathscr{C}_{\text{pub}}) \geqslant \mathbb{P}(\mathbf{M} \text{ is non singular }),$$

therefore it suffices to prove that $\mathbf{M}$ is non singular with overwhelming probability over the choice of $\gamma_1, \ldots, \gamma_\zeta$.

Let

$$\mathscr{G} \stackrel{\text{def}}{=} \mathrm{Span}(\gamma_1, \ldots, \gamma_\zeta) \quad \text{and} \quad \mathscr{M} \stackrel{\text{def}}{=} \mathrm{Span}(\mu_1, \ldots, \mu_\zeta).$$

Then, $\mathbf{M}$ is singular if and only if $\mathscr{G} \cap \mathscr{M}^\perp \neq \{0\}$. Since $\mathscr{G}$ and $\mathscr{M}$ have the same dimension ζ over $\mathbb{F}_{q^m}$, we have $\mathscr{G} \cap \mathscr{M}^\perp = \{0\}$ if and only if $\mathscr{G} \oplus \mathscr{M}^\perp = \mathbb{F}_{q^{mu}}$. Therefore,

$$\mathbb{P}(\mathbf{M} \text{ is non singular }) = \frac{\left| \{\mathscr{G} \mid \mathscr{M}^\perp \oplus \mathscr{G} = \mathbb{F}_{q^{mu}}\} \right|}{\left| \{\mathscr{G} \mid \dim_{\mathbb{F}_{q^m}}(\mathscr{G}) = \zeta\} \right|}.$$

Recall the Gaussian binomial coefficient $\begin{bmatrix} u \\ \zeta \end{bmatrix}_{q^m}$ denotes the number of $\mathbb{F}_{q^m}$–linear subspaces

of dimension ζ in an $\mathbb{F}_{q^m}$–vector space of dimension u. Applying Lemma 3.17, we have

$$\mathbb{P}(\mathbf{M} \text{ is non singular}) = \frac{q^{m\zeta(u-\zeta)}}{\begin{bmatrix} u \\ \zeta \end{bmatrix}_{q^m}} \geqslant \left(1 - \frac{1}{q^m}\right)^{\frac{q^m}{q^m-1}},$$

where the inequality on the right–hand side can be found for instance in [CC19, Appendix A]. This yields Theorem 3.16. □

Set

$$\mathscr{T} \stackrel{\text{def}}{=} \bigoplus_{i=1}^{\zeta} \text{Span}_{\mathbb{F}_{q^m}} \left\{ \text{Tr}_{\mathbb{F}_{q^{mu}}/\mathbb{F}_{q^m}}(\gamma_i \mathbf{k}_{\text{pub}}) \right\},$$

so that $\mathscr{C}_{\text{pub}}$ is the supercode

$$\mathscr{C}_{\text{pub}} = \text{Gab}_k(\mathbf{g}) + \mathscr{T},$$

regarded as a subspace of q-polynomials by interpolation.

The idea to get rid of $\mathbf{e}$ is now to decode in the public supercode $\mathscr{C}_{\text{pub}}$. Since

$$\dim(\mathbb{F}_{q^m}\langle X\rangle_{\leqslant t} \circ \mathscr{T}) \leqslant \zeta(t+1),$$

we expect to decode in $\mathscr{C}_{\text{pub}}$ whenever

$$k + 2t + \zeta(t+1) \leqslant n \tag{3.20}$$

If this fails, it suffices to try with another choice of γ_i's.

Table 3.2 compares (3.20) with the proposed parameters for LIGA in [RPW21, Section 7]. As observed, Inequality (3.20) is satisfied for any proposed parameter set.

Name	n	k	w	t	ζ	u	Claimed Security (bits)	$k+2t+\zeta(t+1)$
LIGA-128	92	53	27	6	2	5	128	79
LIGA-192	120	69	35	8	2	5	192	103
LIGA-256	148	85	43	10	2	5	256	127

Table 3.2: This table compares the values of the formula (3.20) with the parameters proposed for LIGA.

Remark 3.18. *Decoding in the supercode succeeds for small values of ζ, therefore one could try to increase it in order to thwart our attack (note that ζ is upper bounded by u, though). However, doing so, one exposes to the attack by decoding an interleaved code. According to section 3.2.4, this happens when $w \leqslant \frac{\zeta}{\zeta+1}(n-k)$*[v]*. Hence, one also needs to increase w. But this automatically decreases $t \stackrel{def}{=} \lfloor \frac{n-k-w}{2} \rfloor$, which needs to be greater than 1. In other words, the task seems impossible.*

Theorem 3.19 summarises Step 1.

[v] Both attacks are not incompatible. For instance, setting $\zeta = 3$ in the three proposed parameter sets exposes to both attacks

Theorem 3.19

If $\mathbf{c} = \mathbf{m} \cdot \mathbf{G} + \mathrm{Tr}_{\mathbb{F}_{q^{mu}}/\mathbb{F}_{q^m}}(\alpha \mathbf{k}_{\mathrm{pub}}) + \mathbf{e}$ is the encryption of a plaintext $\mathbf{m}$, then we can recover the support of the error $\mathbf{e}$ and the corrupted codeword $\mathbf{m} \cdot \mathbf{G} + \mathrm{Tr}_{\mathbb{F}_{q^{mu}}/\mathbb{F}_{q^m}}(\alpha \mathbf{k}_{\mathrm{pub}})$ in polynomial time using only the knowledge of the public key.

Note that this does not fully recover the plaintext, since we are still facing a codeword of a Gabidulin code, corrupted by an error of too large weight. The purpose of the last two steps is to show how to finish the attack.

Step 2: Removing the z dependency. From now on, we can do as if the ciphertext was

$$\begin{aligned} \mathbf{c}' &\stackrel{\text{def}}{=} \mathbf{m} \cdot \mathbf{G} + \mathrm{Tr}_{\mathbb{F}_{q^{mu}}/\mathbb{F}_{q^m}}(\alpha \mathbf{k}_{\mathrm{pub}}) \\ &= (\mathbf{m} + \mathrm{Tr}_{\mathbb{F}_{q^{mu}}/\mathbb{F}_{q^m}}(\alpha \mathbf{x})) \cdot \mathbf{G} + \mathrm{Tr}_{\mathbb{F}_{q^{mu}}/\mathbb{F}_{q^m}}(\alpha \mathbf{z}). \end{aligned} \tag{3.21}$$

This is a codeword of the Gabidulin code $\mathrm{Gab}_k(\mathbf{g})$, corrupted by an error of rank $w > \lfloor \frac{n-k}{2} \rfloor$. However, thanks to the knowledge of the public key, one can easily recover the affine space

$$\mathscr{A} \stackrel{\text{def}}{=} \left\{ \beta \in \mathbb{F}_{q^{mu}} \mid \mathbf{c}' - \mathrm{Tr}_{\mathbb{F}_{q^{mu}}/\mathbb{F}_{q^m}}(\beta \mathbf{k}_{\mathrm{pub}}) \in \mathrm{Gab}_k(\mathbf{g}) \right\}$$

using linear algebra. A series of easy lemmata will show that this suffices to remove anything which depends on $\mathbf{z}$.

Lemma 3.20

Let $\beta \in \mathbb{F}_{q^{mu}}$. Then $\beta \in \mathscr{A}$ if and only if $\mathrm{Tr}_{\mathbb{F}_{q^{mu}}/\mathbb{F}_{q^m}}((\alpha - \beta)\mathbf{z}) = 0$.

The proof of Lemma 3.20 rests on the following Lemma which proves that the trace map cannot increase the $\mathbb{F}_q$-rank

Lemma 3.21

Let $\mathbf{v} \in \mathbb{F}_{q^{mu}}^n$. Then

$$\mathrm{Rank}(\mathrm{Tr}_{\mathbb{F}_{q^{mu}}/\mathbb{F}_{q^m}}(\mathbf{v})) \leqslant \mathrm{Rank}(\mathbf{v}),$$

where the rank is the $\mathbb{F}_q$-rank.

Proof. Let $\mathbf{v} \stackrel{\text{def}}{=} (v_1, \ldots, v_n) \in \mathbb{F}_{q^{mu}}^n$ be a vector of $\mathbb{F}_q$-rank w and assume for the sake of contradiction that

$$\rho \stackrel{\text{def}}{=} \mathrm{Rank}\left(\mathrm{Tr}_{\mathbb{F}_{q^{mu}}/\mathbb{F}_{q^m}}(\mathbf{v})\right) > w.$$

In particular, $\left\{\mathrm{Tr}_{\mathbb{F}_{q^{mu}}/\mathbb{F}_{q^m}}(v_1), \ldots, \mathrm{Tr}_{\mathbb{F}_{q^{mu}}/\mathbb{F}_{q^m}}(v_n)\right\}$ spans a vector space of dimension ρ

(over $\mathbb{F}_q$), of which we can extract a basis

$$\left\{\mathrm{Tr}_{\mathbb{F}_{q^{mu}}/\mathbb{F}_{q^m}}(v_{i_1}),\dots,\mathrm{Tr}_{\mathbb{F}_{q^{mu}}/\mathbb{F}_{q^m}}(v_{i_\rho})\right\}.$$

Since $\mathbf{v}$ has rank strictly less than ρ, the (v_{i_j}) cannot be linearly independent (over $\mathbb{F}_q$). In other words, there exist elements $\tau_1,\dots,\tau_\rho \in \mathbb{F}_q$ such that they are not all zero and

$$\sum_{j=1}^{\rho} \tau_j v_{i_j} = 0.$$

Now, by $\mathbb{F}_q$-linearity of $\mathrm{Tr}_{\mathbb{F}_{q^{mu}}/\mathbb{F}_{q^m}}$, this yields

$$\sum_{j=1}^{\rho} \tau_j \mathrm{Tr}_{\mathbb{F}_{q^{mu}}/\mathbb{F}_{q^m}}(v_{i_j}) = 0,$$

which is a non trivial linear combination summing up to zero. This contradicts the fact that the $(v_{i_j})_{1\leqslant j\leqslant \rho}$ are linearly independent. □

We are now ready to prove Lemma 3.20.

Proof of Lemma 3.20. Let $\beta \in \mathbb{F}_{q^{mu}}$. We may assume without loss of generality that $\beta - \alpha \neq 0$. Notice that by Equation (3.21),

$$\mathbf{c}' - \mathrm{Tr}_{\mathbb{F}_{q^{mu}}/\mathbb{F}_{q^m}}(\beta \mathbf{k}_{\mathrm{pub}}) = \left(\mathbf{m} + \mathrm{Tr}_{\mathbb{F}_{q^{mu}}/\mathbb{F}_{q^m}}((\alpha-\beta)\mathbf{x})\right)\mathbf{G} + \mathrm{Tr}_{\mathbb{F}_{q^{mu}}/\mathbb{F}_{q^m}}((\alpha-\beta)\mathbf{z}).$$

Therefore,

$$\beta \in \mathscr{A} \text{ if and only if } \mathrm{Tr}_{\mathbb{F}_{q^{mu}}/\mathbb{F}_{q^m}}((\alpha-\beta)\mathbf{z}) \in \mathrm{Gab}_k(\mathbf{g}). \tag{3.22}$$

In order to conclude, it suffices to prove that its rank weight is below the minimum distance of $\mathrm{Gab}_k(\mathbf{g})$.

Lemma 3.21 implies that

$$\mathrm{Rank}(\mathrm{Tr}_{\mathbb{F}_{q^{mu}}/\mathbb{F}_{q^m}}((\alpha-\beta)\mathbf{z})) \leqslant \mathrm{Rank}((\alpha-\beta)\mathbf{z}). \tag{3.23}$$

Moreover, since $\alpha - \beta \neq 0$, it is readily seen that $\mathbf{z}$ and $(\alpha-\beta)\mathbf{z}$ have the same *row support*. In particular, they have the same rank:

$$\mathrm{Rank}((\alpha-\beta)\mathbf{z}) = \mathrm{Rank}(\mathbf{z}) = w. \tag{3.24}$$

Equations (3.23) and (3.24) immediately yield

$$\mathrm{Rank}(\mathrm{Tr}_{\mathbb{F}_{q^{mu}}/\mathbb{F}_{q^m}}((\alpha-\beta)\mathbf{z})) \leqslant \mathrm{Rank}((\alpha-\beta)\mathbf{z}) = w < n-k < d_{\min}(\mathrm{Gab}_k(\mathbf{g})).$$

Since it has rank weight less than the minimum distance of $\mathrm{Gab}_k(\mathbf{g})$, it follows that

$$\mathrm{Tr}_{\mathbb{F}_{q^{mu}}/\mathbb{F}_{q^m}}((\alpha-\beta)\mathbf{z}) = 0.$$

□

Lemma 3.22

Let

$$\mathscr{E} \stackrel{\text{def}}{=} \bigcap_{i=1}^{\zeta} \langle \mu_i \rangle^{\perp}.$$

Then $\mathscr{A}$ is the affine space $\alpha + \mathscr{E}$.

Proof. $\beta \in \mathscr{A}$ if and only if

$$\mathrm{Tr}_{\mathbb{F}_{q^{mu}}/\mathbb{F}_{q^m}}((\alpha-\beta)\mathbf{z}) = \sum_{i=1}^{\zeta} \mathrm{Tr}_{\mathbb{F}_{q^{mu}}/\mathbb{F}_{q^m}}((\alpha-\beta)\mu_i)\mathbf{z}_i = 0.$$

By the linear independence of the $\mathbf{z}_i$'s, it follows that $\mathrm{Tr}_{\mathbb{F}_{q^{mu}}/\mathbb{F}_{q^m}}((\alpha-\beta)\mu_i) = 0$ for all i, *i.e.*

$$\mathscr{A} = \alpha + \bigcap_{i=1}^{\zeta} \langle \mu_i \rangle^{\perp}.$$

□

We are now able to remove the $\mathbf{z}$ dependency in the ciphertext. Indeed, let $\beta \in \mathscr{A}$. It is of the form $\alpha + \gamma$ where $\gamma \in \mathscr{E}$, and

$$\mathbf{c}' - \mathrm{Tr}_{\mathbb{F}_{q^{mu}}/\mathbb{F}_{q^m}}(\beta \mathbf{k}_{\text{pub}}) = \left(\mathbf{m} - \mathrm{Tr}_{\mathbb{F}_{q^{mu}}/\mathbb{F}_{q^m}}(\gamma \mathbf{x})\right)\mathbf{G}.$$

In other words, the knowledge of $\mathscr{A}$ gives finally access to the affine space $\mathbf{m} + \mathscr{F}$, where

$$\mathscr{F} \stackrel{\text{def}}{=} \left\{ \mathrm{Tr}_{\mathbb{F}_{q^{mu}}/\mathbb{F}_{q^m}}(\gamma \mathbf{x}) \mid \gamma \in \mathscr{E} \right\}.$$

We now have all the tools to fully recover the plaintext.

Step 3: Recovering the plaintext. Denote by $f \stackrel{\text{def}}{=} \dim_{\mathbb{F}_{q^m}} \mathscr{F}$. Since $\mathscr{F}$ is the image of $\mathscr{E}$ by a surjective map, we have $f \leqslant \dim \mathscr{E} = u - \zeta \leqslant u - 1$. Let $\mathbf{s}$ be some random element of $\mathbf{m} + \mathscr{F}$. Notice that from a description of the affine space $\mathbf{m} + \mathscr{F}$ it is possible to recover a basis $(\mathbf{e}_1, \ldots, \mathbf{e}_f)$ of $\mathscr{F}$. Then, $\mathbf{s}$ can be decomposed as

$$\mathbf{s} \stackrel{\text{def}}{=} \mathbf{m} + \sum_{i=1}^{f} \lambda_i \mathbf{e}_i$$

for some unknown coefficients $\lambda_i \in \mathbb{F}_{q^m}$. Furthermore, recall that the last u positions of $\mathbf{m}$ are 0. In other words, $\mathbf{m}$ is a solution of the following linear system of $k + f \leqslant k + u - 1$ unknowns and $u + k$ equations:

$$\begin{cases} \mathbf{m} + \sum_{i=1}^{f} \lambda_i \mathbf{e}_i = \mathbf{s} \\ \mathbf{m}_{k-u+1} = \cdots = \mathbf{m}_k = 0 \end{cases} \tag{3.25}$$

Finally, the following lemma shows that $\mathbf{m}$ can be recovered from *any* solution of (3.25).

Lemma 3.23

Let $(\mathbf{m}', \lambda')$ be another solution of (3.25). Then $\mathbf{m}' = \mathbf{m}$.

Proof. Since $\mathbf{m} - \mathbf{m}' = \sum_{i=1}^{f} (\lambda_i' - \lambda_i)\mathbf{e}_i \in \mathscr{F}$, it is of the form $\mathrm{Tr}_{\mathbb{F}_{q^{mu}}/\mathbb{F}_{q^m}}(\gamma \mathbf{x})$ for some $\gamma \in \mathscr{E}$. Moreover, its last u positions are 0. Recall that $(\mathbf{x}_{k-u+1}, \dots, \mathbf{x}_k)$ is a basis of $\mathbb{F}_{q^{mu}}/\mathbb{F}_{q^m}$. Then, the last u positions of $\mathrm{Tr}_{\mathbb{F}_{q^{mu}}/\mathbb{F}_{q^m}}(\gamma \mathbf{x})$ are the coefficients of γ in the dual basis $\{\mathbf{x}_{k-u+1}^*, \dots, \mathbf{x}_k^*\}$. Hence, $\gamma = 0$ and $\mathbf{m} = \mathbf{m}'$. □

Wrapping up, the attack works as follows:

1. Decode in a public supercode of a Gabidulin code to get rid of the small error $\mathbf{e}$ and recover
$$\mathbf{c}' = \mathbf{m}\mathbf{G} + \mathrm{Tr}_{\mathbb{F}_{q^{mu}}/\mathbb{F}_{q^m}}(\alpha \mathbf{k}_{\mathrm{pub}}).$$
2. Using linear algebra, deduce the affine space
$$\mathscr{A} = \left\{ \beta \in \mathbb{F}_{q^{mu}} \mid \mathbf{c}' - \mathrm{Tr}_{\mathbb{F}_{q^{mu}}/\mathbb{F}_{q^m}}(\beta \mathbf{k}_{\mathrm{pub}}) \in \mathrm{Gab}_k(\mathbf{g}) \right\}.$$
3. Recover the affine space $\mathbf{m} + \mathscr{F}$ where
$$\mathscr{F} = \left\{ \mathrm{Tr}_{\mathbb{F}_{q^{mu}}/\mathbb{F}_{q^m}}(\gamma \mathbf{x}) \mid \alpha + \gamma \in \mathscr{A} \right\}.$$
4. Deduce a basis of $\mathscr{F}$.
5. Solve (3.25) to recover the plaintext $\mathbf{m}$.

Note that the attack only involves linear algebra. In particular, it runs in polynomial time, and is very efficient in practice (see Table 3.3).

A proof of concept implementation using SageMath [Ste+23] is available on Github `https://github.com/mbombar/Attack_on_LIGA`. On a personal laptop[vi], it is able to recover the plaintext on the three LIGA proposals. The average running times are listed in Table 3.3.

Name	Parameters $(\mathbf{q}, \mathbf{n}, \mathbf{m}, \mathbf{k}, \mathbf{w}, \mathbf{u}, \zeta)$	Claimed security level	Average running time
LIGA-128	$(2, 92, 92, 53, 27, 5, 2)$	128 bits	8 minutes
LIGA-192	$(2, 120, 120, 69, 35, 5, 2)$	192 bits	27 minutes
LIGA-256	$(2, 148, 148, 85, 43, 5, 2)$	256 bits	92 minutes

Table 3.3: Average running times for the attack on LIGA.

[vi] Intel® Core™ i5-10310U CPU

Part II

Foundations

Chapter 4

A Function Field Approach to Search-to-Decision Reductions

Rien n'est plus fécond, tous les mathématiciens le savent, que ces obscures analogies, ces troubles reflets d'une théorie à une autre, ces furtives caresses, ces brouilleries inexplicables ; rien aussi ne donne plus de plaisir au chercheur.

André Weil

Algebraically structured variants of LWE, such as Polynomial-LWE [SSTX09], Ring-LWE [LPR10], or Module-LWE [LS15] used the theory of algebraic number fields to derive search-to-decision reductions, instantiated with cyclotomic extensions of $\mathbb{Q}$.

Contributions of this thesis. In this chapter based on [BCD22] and published at CRYPTO 2022, we define a function field analogue, making use of the famous number field-function field analogy, and propose to instantiate with analogues of cyclotomic number fields, namely *Carlitz extensions* of $\mathbb{F}_q(T)$. This enables a new number theoretic interpretation of the structured variants of the Decoding Problem such as QC-DP (Problem 1.38); and allows us to derive the first search-to-decision reduction for problems involving structured codes.

Outline of the current chapter

4.1 Introduction

As recalled in Chapter 1, it is a long-standing open problem in code-based cryptography to understand the hardness of structured variants of the Decoding Problem such as for quasi-cyclic codes, and especially regarding the decisional version. Recall that a quasi-cyclic code defined over the finite field $\mathbb{F}_q$ is a code generated by a matrix $\mathbf{G}$ formed out by multiple circulant blocks:

$$\mathbf{G} = \begin{pmatrix} \mathbf{a}^{(1)} & \cdots & \mathbf{a}^{(\ell)} \\ \circlearrowleft & & \circlearrowleft \end{pmatrix}$$

where the symbol $\circlearrowleft$ simply denotes the fact that each block is the circulant matrix whose rows are the cyclic shifts of a given vector $\mathbf{a} \stackrel{\text{def}}{=} (a_0, \ldots, a_{n-1}) \in \mathbb{F}_q^n$, *i.e.* of the form

$$\begin{pmatrix} a_0 & a_1 & \ldots & \ldots & a_{n-1} \\ a_{n-1} & a_0 & \ldots & \ldots & a_{n-2} \\ \vdots & \ddots & \ddots & & \vdots \\ \vdots & & \ddots & \ddots & \vdots \\ a_1 & a_2 & \ldots & a_{n-1} & a_0 \end{pmatrix}.$$

From Chapter 1, any such circulant matrix can be conveniently represented by the polynomial of degree at most $n-1$ defining its first row through the isomorphism

$$\Phi : \begin{cases} \mathbb{F}_q^n & \longrightarrow & \mathbb{F}_q[X]/(X^n - 1) \\ \mathbf{a} \stackrel{\text{def}}{=} (a_0, \ldots, a_{n-1}) & \longmapsto & \mathbf{a}(X) \stackrel{\text{def}}{=} \sum_{i=0}^{n-1} a_i X^i \end{cases},$$

such that the matrix-vector product (and therefore the encoding map) is represented by products in the quotient ring $\mathbb{F}_q[X]/(X^n - 1)$. In other words, if $\mathbf{e} \stackrel{\text{def}}{=} (\mathbf{e}^{(1)}, \ldots, \mathbf{e}^{(\ell)}) \in \mathbb{F}_q^{\ell \cdot n}$ is an error

vector, the noisy codeword $\mathbf{mG}+\mathbf{e}$ is represented by the collection of ℓ corrupted polynomials

$$\begin{cases} \mathbf{m}(X)\cdot\mathbf{a}^{(1)}(X)+\mathbf{e}^{(1)}(X) \\ \qquad\vdots \\ \mathbf{m}(X)\cdot\mathbf{a}^{(\ell)}(X)+\mathbf{e}^{(\ell)}(X) \end{cases} \in \mathbb{F}_q[X]/(X^n-1),$$

where $\mathbf{m}(X) \stackrel{\text{def}}{=} \Phi(\mathbf{m})$. With this formalism, the Decoding Problem restricted to the class of ℓ-quasi-cyclic codes can be formulated as follows. Let Ψ be a probability distribution over $\mathbb{F}_q[X]/(X^n-1)$.

Problem 4.1 (QC-DP(ℓ,Ψ), search version)

Let $\mathcal{R} \stackrel{\text{def}}{=} \mathbb{F}_q[X]/(X^n-1)$, and let $\mathbf{y}^{(1)},\ldots,\mathbf{y}^{(\ell)}$ be of the form

$$\mathbf{y}_i \stackrel{\text{def}}{=} \mathbf{a}^{(i)}\mathbf{m}+\mathbf{e}^{(i)} \in \mathcal{R},$$

for some *fixed* $\mathbf{m}\in\mathcal{R}$, where $\mathbf{a}^{(i)} \leftarrow \mathcal{R}$ is uniformly distributed and $\mathbf{e}^{(i)} \leftarrow \Psi$.

Given the ℓ samples $(\mathbf{a}^{(i)},\mathbf{y}^{(i)})$, the goal is to recover $\mathbf{m}$.

In general, Ψ will be a distribution such that $\mathbb{E}_{\mathbf{x}\leftarrow\Psi}(|\mathbf{x}|) = t$ for some integer parameter $t \in \{1,\ldots,n\}$, where the Hamming weight on $\mathcal{R}$ is the number of non-zero coefficients. This encompasses for instance the uniform distribution over polynomials of weight t, also known as the *regular* noise distribution, or when the coefficients of the error are independently distributed according to a q-ary Bernoulli random variables of success probability t/n.

Problem 4.1 also admits a decisional version.

Problem 4.2 (QC-DP(Ψ), decisional version)

Let n be an integer, and set $\mathcal{R} \stackrel{\text{def}}{=} \mathbb{F}_q[X]/(X^n-1)$. Let $\mathbf{m}$ be drawn uniformly at random in $\mathcal{R}$ and consider the following two distributions

- $\mathcal{D}_0 : (\mathbf{a},\mathbf{y}^{\text{unif}})$ uniformly distributed over $\mathcal{R}^2$,
- $\mathcal{D}_1 : (\mathbf{a},\mathbf{a}\cdot\mathbf{m}+\mathbf{e})$ where $\mathbf{a}\leftarrow\mathcal{R}$, and $\mathbf{e}\leftarrow\Psi$.

Given oracle access to distribution $\mathcal{D}_b$ where $b\leftarrow\{0,1\}$, the goal is to recover b.

Such structured variants are at the core of BIKE [AABB+22a] and HQC [AABB+22b] encryption schemes, two of the three code-based submissions still competing in the round 4 of NIST competition, and are syntactically similar structured variants of LWE such as Polynomial-LWE [SSTX09] or Polynomial-LWE [LPR10]. Besides the choice of the error distribution Ψ, what basically changes is the base ring $\mathcal{R}$. Indeed, in the lattice-based setting, the ring $\mathcal{R}$ is defined as $\mathcal{O}_K/p\mathcal{O}_K$, where $p\in\mathbb{Z}$ is a prime integer usually called the *modulus* and $\mathcal{O}_K$ is the ring of integers of a number field K. Usually, K is a cyclotomic field $\mathbb{Q}[\zeta_m]$ where ζ_m is a primitive m-th root of unity, which can also be written as $\mathbb{Q}[X]/(\Phi_m)$, and $\mathcal{O}_K = \mathbb{Z}[X]/(\Phi_m)$

where $\Phi_m \in \mathbb{Z}[X]$ is the m-th cyclotomic polynomial. In particular, when $m = 2n = 2^{l+1}$ is a power of two, it is well known that

$$\Phi_m(X) = X^{m/2} + 1 = X^n + 1 = X^{2^\ell} + 1.$$

Therefore,

$$\mathcal{R} \stackrel{\text{def}}{=} \mathcal{O}_K / p\mathcal{O}_K = \mathbb{Z}[X]/(p, \Phi_m) = \mathbb{F}_p[X]/(X^n + 1),$$

which emphasises all the more the resemblance with the cyclic ring $\mathbb{F}_q[X]/(X^n - 1)$ used in the code-based setting.

An important difference between $\mathbb{Z}[X]/(X^n+1)$ and $\mathcal{R}$ or $\mathbb{F}_q[X]/(X^n-1)$ is that the former has *Krull dimension* 1, while the latter has *Krull dimension* 0. In other words, the objects involved in lattice-based cryptography can be *lifted* to higher dimensional objects. This is this number theoretic interpretation of $\mathcal{R}$ that allowed to design many search-to-decision reductions for the structured variants of LWE. On the other hand, the situation is quite different in the coding theoretic setting, for which it did not seem possible to do an analogous work, even though the problems were similar. This lack of search-to-decision reduction was even pointed out by the NIST in the report for the second round of the first call for post-quantum cryptographic primitives [AAAC+20].

Contribution of this thesis. The main idea of this chapter is to lift the Decoding Problem to Krull dimension 1 using algebraic function fields, which are finite extensions of the field of the rational functions. Those algebraic function fields bare a strong similarity with algebraic number fields used in structured lattice-based cryptography. In fact, they are two facets of a similar object known as global fields in algebraic number theory.

We introduce a new cryptographic problem which we call the Function Field Decoding Problem (FF-DP) and which is the analogue in positive characteristics of structured variants of LWE, allowing a new number theoretic interpretation of the cyclic ring $\mathbb{F}_q[X]/(X^n - 1)$, more suitable to the code-based setting. With this interpretation in hand, we are able to give the first search-to-decision reduction for the quasi-cyclic variant of the Decoding Problem.

4.2 Algebraic Number Theory in Function Fields

4.2.1 Notions of Algebraic Number Theory

The theory of algebraically structured lattices developed along the notion of algebraic number fields, their ring of integers and their ideals. In order to emphasise the similarities between the two classes of global fields, let us begin with a quick overview of the arithmetic of number fields. Algebraic number fields are finite extensions of the field of rational numbers, *i.e.* they are fields K of the form

$$K \stackrel{\text{def}}{=} \mathbb{Q}[X] \big/ (P(X))$$

where P is an irreducible polynomial (over $\mathbb{Q}$). Since the extension is finite, every element of K is algebraic over $\mathbb{Q}$ and therefore is annihilated by a polynomial with rational coefficients. An element $\alpha \in K$ which is annihilated by a *monic* polynomial with *integer* coefficients is called *integral* over $\mathbb{Z}$. The set of all the integral elements of K form a ring called the *ring of integers* of K, and is usually denoted by $\mathcal{O}_K$:

$$\mathcal{O}_K \stackrel{\text{def}}{=} \{\alpha \in K \mid \exists \mu \in \mathbb{Z}[X] \text{ and } \textit{monic} \text{ such that } \mu(\alpha) = 0\}.$$

The rationale behind this notion of *integral elements* is that $\mathcal{O}_K$ is the correct generalisation of $\mathbb{Z}$ in K: This is the *integral closure* of $\mathbb{Z}$ in K, and as such is a *Dedekind* domain, and has finite quotients. In particular, it has *Krull dimension* 1: any non zero prime ideal is in fact maximal, and the quotient of $\mathcal{O}_K$ by any non zero prime ideal is a finite field. Moreover, if they are not principal ideal domains, nor even factorial, every nonzero proper *ideal* in $\mathcal{O}_K$ admits a unique factorisation into product of nonzero prime ideals. For example, in $K = \mathbb{Q}(\sqrt{-5}) = \mathbb{Q}[X]/(X^2+5)$, it is not hard to prove that $\mathcal{O}_K = \mathbb{Z}[\sqrt{-5}]$. Yet we find that

$$6 = 2 \cdot 3 = (1+\sqrt{-5}) \cdot (1-\sqrt{-5}),$$

and one can check that the elements $2, 3, (1+\sqrt{-5}), (1-\sqrt{-5})$ are irreducible and not associates (see [Lor21, Chapter I, Exercise 4]). On the other hand, the ideal $6\mathcal{O}_K$ uniquely factorises as

$$6\mathcal{O}_K = (2, 1+\sqrt{-5}) \cdot (2, 1+\sqrt{-5}) \cdot (3, 1+\sqrt{-5}) \cdot (3, 2+\sqrt{-5}),$$

where each factor can be proved to be a prime ideal, for example by computing the quotient ring.

The primitive element theorem ensures that for any number field K there exists an element $\zeta \in \overline{\mathbb{Q}}$ such that $K = \mathbb{Q}(\zeta)$. The inclusion $\mathbb{Z}[\zeta] \subset \mathcal{O}_K$ always holds, however $\mathbb{Z}[\zeta]$ might be *too small* and not *integrally closed*. For example, consider the number field $\mathbb{Q}(\sqrt{5}) = \mathbb{Q}[X]/(X^2-5)$, which seems similar to the previous example. However, the domain $\mathbb{Z}[\sqrt{5}]$ does not have the property of unique factorisation of ideals ([Lor21, Chap. I, Sec. 3]). In fact, the element $\frac{1+\sqrt{5}}{2}$ is integral, but not in $\mathbb{Z}[\sqrt{5}]$, and it can be shown that the ring of integers $\mathcal{O}_K$ is exactly $\mathbb{Z}[\frac{1+\sqrt{5}}{2}]$.

A subring of $\mathcal{O} \subset \mathcal{O}_K$ is called an *order* of K. The fundamental property of orders that attracted lattice-based cryptographers is that they are *free abelian groups*, or in other words *free $\mathbb{Z}$-modules*, of rank $\dim_{\mathbb{Q}}(K)$. In particular, they come with a $\mathbb{Z}$-basis since $\mathbb{Z}$ is a principal ideal domain. Moreover, any ideal $\mathcal{I} \subset \mathcal{O}$ can be embedded into a lattice of $\mathbb{R}^n$, *i.e.* a discrete subgroup of $\mathbb{R}^n$, which enables to transport the Euclidean structure of $\mathbb{R}^n$ onto $\mathcal{I}$.

4.2.2 Global Function Fields

In this section, we give the elementary notions underlying function fields in one variables over finite fields. From a geometric point of view, they are the fields of rational functions of a projective curve over a finite field. However, in this manuscript we will mostly consider an arithmetic point of view, more suitable for our cryptographic applications.

4.2.2.1 Algebraic Function Fields in One Variable

Starting from a finite field $\mathbb{F}_q$, define the field of rational functions, or *rational function field*, in the variable T as

$$\mathbb{F}_q(T) \stackrel{\text{def}}{=} \left\{ \frac{P(T)}{Q(T)} \mid P, Q \in \mathbb{F}_q[T] \right\}.$$

This is the field of fractions of the ring of univariate polynomials $\mathbb{F}_q[T]$.

An *algebraic function field in one variable* over $\mathbb{F}_q$ is a finite algebraic extension $K/\mathbb{F}_q(T)$ of degree $n > 0$. Recall that an algebraic extension E/F is called *separable* if the minimal polynomial of every element of E over F only has simple roots. A field is called *perfect* when every algebraic extension is separable. This is in particular the case for finite fields, or fields of characteristics 0. However, this is not true for $\mathbb{F}_q(T)$. Yet, the primitive element theorem

also holds for $\mathbb{F}_q(T)$ and every finite extension $K/\mathbb{F}_q(T)$ is *simple* ([Lor21, Prop. X.1.9]). In particular, an algebraic function field is a field of the form

$$K \stackrel{\text{def}}{=} \mathbb{F}_q(T)[X]\big/(P(X)),$$

where $P \in \mathbb{F}_q(T)[X]$ is an irreducible polynomial of degree n. When the context is clear, we shall simply refer to K as a *function field*, and in the sequel we will only work with *separable* extensions.

The algebraic closure of $\mathbb{F}_q$ in K, namely the field $K \cap \overline{\mathbb{F}_q}$, is referred to as *the field of constants* or *constant field* of K. When $\mathbb{F}_q$ is the full field of constants of K, the extension $K/\mathbb{F}_q(T)$ is called *geometric*. This is equivalent for P to be irreducible even regarded as an element of $\overline{\mathbb{F}_q}(T)[X]$ ([Sti09, Cor. 3.6.8]). In that case, P is said to be *absolutely irreducible*.

Example 4.3.

- *The constant field of* $\mathbb{F}_q(T)$ *is* $\mathbb{F}_q$.
- *If* ζ *is algebraic of degree* $m \geqslant 1$ *over* $\mathbb{F}_q$*, then* $\zeta \notin \mathbb{F}_q(T)$*. Let* $\mu_\zeta \in \mathbb{F}_q[X]$ *be its minimal polynomial (over* $\mathbb{F}_q$*). It can be shown that* μ_ζ *remains irreducible over* $\mathbb{F}_q(T)$*. Then,*

  $$K \stackrel{def}{=} \mathbb{F}_q(T)[\zeta] = \mathbb{F}_q(T)\big/(\mu_\zeta)$$

  is isomorphic to $\mathbb{F}_{q^m}(T)$ *and its constant field is* $\mathbb{F}_{q^m}$.

4.2.2.2 The Number Field - Function Field Analogy

It is well-known for a long time that there is a noticeable analogy between the theory of number fields and that of function field, and many results on number fields have their function field counterparts. Note that actually, many properties that are known for function fields are only conjectures for number fields. The best example is probably the Riemann Hypothesis which was proved by Weil in the 1940s in the function field case [Wei40; Wei48] and is still an open problem in the number field situation. This analogy is summarised in Table 4.1.

Number fields	Function fields
$\mathbb{Q}$	$\mathbb{F}_q(T)$
$\mathbb{Z}$	$\mathbb{F}_q[T]$
Prime numbers $q \in \mathbb{Z}$	Irreducible polynomials $Q \in \mathbb{F}_q[T]$
$K = \mathbb{Q}[X]\big/(P(X))$	$K = \mathbb{F}_q(T)[X]\big/(P(T,X))$
$\mathcal{O}_K$ = Integral closure of $\mathbb{Z}$ *Dedekind* domain	$\mathcal{O}_K$ = Integral closure of $\mathbb{F}_q[T]$ *Dedekind* domain
characteristic 0	**characteristic** > 0

Table 4.1: Number fields and function fields: two facets of global fields.

Starting from the ground, the fields $\mathbb{Q}$ and $\mathbb{F}_q(T)$ are both the field of fractions of euclidean domains with finite quotients, namely $\mathbb{Z}$ and $\mathbb{F}_q[T]$. Their prime ideals are given by prime numbers on the one hand, and irreducible polynomials on the other. Now, if one considers an algebraic function field K, it is natural to introduce $\mathcal{O}_K$ the integral closure of $\mathbb{F}_q[T]$ in K:

$$\mathcal{O}_K \stackrel{\text{def}}{=} \{\alpha \in K \mid \exists \mu \in \mathbb{F}_q[T,X] \text{ and } \textit{monic} \text{ (in } X\text{) such that } \mu(\alpha) = 0\}.$$

This is also a *Dedekind* domain. In particular, any ideal $\mathfrak{I}$ of $\mathcal{O}_K$ has a unique decomposition

$$\mathfrak{I} \stackrel{\text{def}}{=} \mathfrak{p}_1^{e_1} \cdots \mathfrak{p}_\mathfrak{r}^{e_r},$$

where $\mathfrak{p}_i$ are prime ideals of $\mathcal{O}_K$ and e_i are integers.

Let $P \in \mathbb{F}_q(T)[X]$ be a defining polynomial of K, *i.e.* such that

$$K = \mathbb{F}_q(T)[X] \big/ (P(X)).$$

Similarly to the number field situation, the ring

$$\mathbb{F}_q[T,X] \big/ (P(X))$$

is called an *order* of K and is not always integrally closed. However, here there is a geometric characterisation. Indeed, assume that P is absolutely irreducible and let

$$\mathcal{X}_P \stackrel{\text{def}}{=} \{(a,b) \in \overline{\mathbb{F}_q} \mid P(a,b) = 0\}$$

be its zero-locus. Recall that $(a,b) \in \mathcal{X}_P$ is said to be a *singular* point when

$$\frac{\partial P}{\partial T}(a,b) = \frac{\partial P}{\partial X}(a,b) = 0,$$

and $\mathcal{X}_P$ (or simply P) is said to be non-singular or smooth when the set of singular points is empty. Then the ring $\mathbb{F}_q[T,X] \big/ (P(X))$ is a Dedekind domain if and only if $\mathcal{X}$ is non-singular ([Lor21, Cor 2.7]). In particular, from a geometric point of view, $\mathcal{O}_K$ can be regarded as the ring of regular functions on a non-singular plane curve.

Remark. *Note that $\mathcal{X}_P$ needs only to be smooth on the* affine *part, and its* projectivisation *might have singularities.*

Remark 4.4. *Orders in a function field $K/\mathbb{F}_q(T)$ are free $\mathbb{F}_q[T]$-modules. In particular, they (and their quotients !) are also $\mathbb{F}_q$ vector spaces, which induces a more rigid structure than that of orders in number fields.*

In the sequel, we will consider the following setting, represented in the diagram below:

$$\begin{array}{ccc} \mathfrak{P} \subset \mathcal{O}_K & \longrightarrow & K \\ \big| & & \big| \\ \mathfrak{p} \subset \mathbb{F}_q[T] & \longrightarrow & \mathbb{F}_q(T) \end{array}$$

Starting from a prime ideal $\mathfrak{p} \subset \mathbb{F}_q[T]$ (which is nothing but the ideal generated by an irreducible polynomial $Q(T)$ of $\mathbb{F}_q[T]$), we consider the ideal $\mathfrak{P} \stackrel{\text{def}}{=} \mathfrak{p}\mathcal{O}_K$ and its decomposition

$$\mathfrak{P} = \mathfrak{P}_1^{e_1} \cdots \mathfrak{P}_r^{e_r}.$$

The prime ideals $\mathfrak{P}_i$'s are said to *lie above* $\mathfrak{p}$ (or directly above the irreducible polynomial Q). The exponents $e_i's$ are referred to as the *ramification indexes*, and the extension $K/\mathbb{F}_q(T)$ is said to be *unramified* at $\mathfrak{p}$ when all the e_i's are equal to 1. Since $\mathcal{O}_K$ is Dedekind, the $\mathfrak{P}_i$'s are maximal, and the quotients $\mathcal{O}_K/\mathfrak{P}_i$ are finite field extensions of $\mathbb{F}_q[T]/\mathfrak{p}$. In particular, they all have the same characteristics as $\mathbb{F}_q$ (this is one difference with the number field situation). The extension degree

$$f_i \stackrel{\text{def}}{=} \left[\mathcal{O}_K/\mathfrak{P}_i \colon \mathbb{F}_q[T]/\mathfrak{p}\right]$$

is called the *inertia degree* of $\mathfrak{p}$ at $\mathfrak{P}_i$. The inertia degrees and the ramification indexes are related to the degree n of the extension $K/\mathbb{F}_q(T)$ through the fundamental relation:

$$n \stackrel{\text{def}}{=} [K \colon \mathbb{F}_q(T)] = \sum_{i=1}^{r} e_i f_i. \tag{4.1}$$

Example 4.5. *Let $n > 1$ and consider the polynomial $P(T, X) \stackrel{def}{=} X^n + T - 1 \in \mathbb{F}_q[T](X)$. When n is not divisible by the characteristic of $\mathbb{F}_q$, it is a separable polynomial. Moreover, by Eisenstein criterion, P is irreducible. In fact, it is even absolutely irreducible. Define the function field K generated by P, namely the extension field*

$$K \stackrel{def}{=} \mathbb{F}_q(T)[X]/(X^n + T - 1).$$

Its constant field is $\mathbb{F}_q$. Furthermore, the above criterion regarding the partial derivatives shows that P is non-singular. Therefore,

$$\mathcal{O}_K = \mathbb{F}_q[T, X]/(X^n + T - 1).$$

Consider the prime ideal $\mathfrak{p} = (T) \subset \mathbb{F}_q[T]$ generated by the irreducible polynomial T, and let $\mathfrak{P} \stackrel{def}{=} T\mathcal{O}_K$.

$$\begin{array}{ccc} \mathfrak{P} \subset \mathbb{F}_q[T, X]/(X^n + T - 1) & \longrightarrow & \mathbb{F}_q(T)[X]/(X^n + T - 1). \\ \Big| & & \Big| \\ \mathfrak{p} = (T) \subset \mathbb{F}_q[T] & \longrightarrow & \mathbb{F}_q(T) \end{array}$$

Then,

$$\begin{aligned} \mathcal{O}_K/T\mathcal{O}_K &= \mathbb{F}_q[T, X]/(T, X^n + T - 1) \\ &= \left(\mathbb{F}_q[T]/T\right)[X]/(X^n - 1) \\ &= \mathbb{F}_q[X]/X^n - 1. \end{aligned}$$

In particular, $\mathfrak{p}$ *does not ramify in* $\mathcal{O}_K$ *and splits into as many prime ideals as the number of irreducible factors of* $X^n - 1$ *in* $\mathbb{F}_q$*. Furthermore, this examples shows that the cyclic ring* $\mathbb{F}_q[X]/X^n - 1$ *can be regarded as the quotient of a ring of integers in a function field by the ideal generated by an irreducible polynomial which could be called the* modulus*, in the spirit of what is done in the lattice-based cryptography literature.*

Let us end this section with a remark that might be useful one day. The notion of prime ideals in global fields is related to the notions of valuations, absolute values and places. In this chapter, we focus on the so-called *finite places*, but doing so we forget some places which might be of interest: the *infinite places*. In the number field setting, finite places and infinite places are of different natures, and both are used in the study of lattice-based cryptography.

- The *finite places* of a number field K are in one-to-one correspondence with the prime ideals of $\mathcal{O}_K$. They are also called *non-archimedean* places, since they define non-archimedean absolute values, and the completion of K at a finite place (with respect to the metric induced by the corresponding absolute value) is a non-archimedean field. For example, in the case of the rational field $\mathbb{Q}$, the finite places correspond to the prime numbers, and the completions are the p-adic fields $\mathbb{Q}_p$.

- In order to describe the *infinite places*, let us begin with the simple case of $\mathbb{Q}$. Ostrowski's theorem ensures that a non-trivial absolute value is equivalent to either the p-adic absolute values, or the usual real one. Therefore, there is a unique infinite place of $\mathbb{Q}$ and the completion corresponds to the field $\mathbb{R}$ of real numbers. Now, consider a number field K of degree n. It is well-known that it has r_K real embeddings $K \to \mathbb{R}$ and s_K complex (non real) embeddings $K \to \mathbb{C}$ where $n = r_K + 2s_K$ corresponding to each root of the defining polynomial, up to complex conjugation. Each embedding σ gives raise to an archimedean absolute value
$$|x|_\sigma = \begin{cases} |\sigma(x)| & \text{if } \sigma \text{ is a real embedding} \\ |\sigma(x)|^2 & \text{otherwise.} \end{cases}$$
The rationale behind the square for complex embeddings is to ensure the so-called product formula which holds true for any global field:
$$\prod_v |x|_v = 1 \text{ for all } x \in K,$$
where the product runs through all possible absolute values. A generalisation of Ostrowski's theorem ensures that those absolute values are the only archimedean absolute values on K. Therefore, the infinite places of K are in one-to-one correspondence with its complex embeddings (up to complex conjugation).

  In lattice-based cryptography, the lattices based on number fields are usually endowed with the euclidean metric defined through the so-called *canonical embedding*, which takes into account all possible complex embeddings of K in $\mathbb{C}$ (up to complex conjugation). This corresponds exactly to the euclidean metric induced by all the infinite places of K.

On the other hand, in the function field setting, *all* the possible absolute values are non-archimedean and the above asymmetry does not exist. However, one might still consider two kinds of places, though similar in natures and the distinction is somehow arbitrary:

- Similarly to the number field setting, the *finite places* of a function field K correspond to the prime ideals of $\mathcal{O}_K$.

- In order to understand the so-called infinite places, let us begin with the rational function field $\mathbb{F}_q(T)$. Let $x \stackrel{\text{def}}{=} \frac{P(T)}{Q(T)} \in \mathbb{F}_q(T)^\times$ where P and Q are coprime. One can check that $|x| \stackrel{\text{def}}{=} q^{\deg(P)-\deg(Q)}$ defines another non-archimedean absolute value, which is not equivalent to any of the above mentioned finite absolute values. In reality, this absolute value comes from the $\frac{1}{T}$-*adic valuation*. Indeed, let $Y \stackrel{\text{def}}{=} \frac{1}{T}$. Then,

$$x = \frac{P(\frac{1}{Y})}{Q(\frac{1}{Y})} = \frac{Y^{-\deg(P)}P_1(Y)}{Y^{-\deg(Q)}Q_1(Y)} = Y^{\deg(Q)-\deg(P)}\frac{P_1(Y)}{Q_1(Y)}$$

  where $P_1(Y), Q_1(Y)$ are relatively prime to Y (in $\mathbb{F}_q[Y] = \mathbb{F}_q[\frac{1}{T}]$). In other words, $\frac{1}{T}$ plays the same role in $\mathbb{F}_q[\frac{1}{T}]$ as any other (monic) irreducible polynomial in $\mathbb{F}_q[T]$. An analogue of Ostrowski's theorem ensures that this is the only other absolute value ([Vil06, Theorem 2.4.1]).

  Now, let $K/\mathbb{F}_q(T)$ be a function field. The infinite places should correspond to some decomposition of $\frac{1}{T}$ in K. However, $\frac{1}{T} \notin \mathcal{O}_K$, so this notion makes no sense at first glance. Instead, one shall consider another ring of interest in K, namely the integral closure of $\mathbb{F}_q[\frac{1}{T}]$. Since we only want to focus on the infinite place $\frac{1}{T}$, it is enough to consider its localisation at $\frac{1}{T}$: Set

$$\mathbb{F}_q[T]_\infty \stackrel{\text{def}}{=} \mathbb{F}_q\left[\frac{1}{T}\right]_{(1/T)} \stackrel{\text{def}}{=} \left\{\frac{P(T)}{Q(T)} \mid \deg(P) \leqslant \deg(Q)\right\},$$

  and define the *infinite maximal order* $\mathcal{O}_{K,\infty}$ to be its integral closure in K. It is a Dedekind domain, and the infinite places of K are in one-to-one correspondence with the prime ideals of $\mathcal{O}_{K,\infty}$.

Remark. *Another explanation for this discrepancy is the following. In the number field setting, there is only one way to send $\mathbb{Z}$ into $\mathbb{Q}$ so that it respects the ring structure, namely, the inclusion. In the language of categories, $\mathbb{Z}$ is said to be an initial object in the category of rings. As a consequence the choice of $\mathbb{Z}$ to start with is canonical.*

On the other hand, in the function field setting, the choice of $\mathbb{F}_q[T]$ is a bit arbitrary since there are many non equivalent ring homomorphisms between $\mathbb{F}_q[T]$ and $\mathbb{F}_q(T)$. If $K/\mathbb{F}_q(T)$ is a function field, the integral closure of $\mathbb{F}_q[\frac{1}{T}]$ in K is also a Dedekind domain, and a similar framework can be derived with this choice of ground ring. In order to avoid redundancy, we might prefer to use the localisation $\mathbb{F}_q[T]_\infty$ introduced earlier and its integral closure $\mathcal{O}_{K,\infty}$, as represented in the diagram below.

$$\begin{array}{ccccc}
\mathfrak{P} \subset \mathcal{O}_K & \text{———} & K & \text{———} & \mathcal{O}_{K,\infty} \supset \mathfrak{P}_\infty \\
| & & | & & | \\
\mathfrak{p} \subset \mathbb{F}_q[T] & \text{———} & \mathbb{F}_q(T) & \text{———} & \mathbb{F}_q[T]_\infty \supset \mathfrak{p}_\infty
\end{array}$$

This would be all the more interesting, since both $\mathcal{O}_K$ and $\mathcal{O}_{K,\infty}$ are involved in one fundamental object of the theory of algebraic function fields which we do not make use at all in this chapter, namely Riemann-Roch spaces. See [Vil06, Chap. 3] for a description of Riemann-Roch theory in terms of places, or [Hes02] which introduced an algorithm to compute those spaces explicitly.

4.2.2.3 Galois Extensions of Function Fields

In this chapter, and similarly to what happens in lattice-based cryptography, there are special function fields which are easier to work with, namely Galois extensions.

Recall that a finite algebraic field extension L/K is said to be *Galois* when the automorphism group

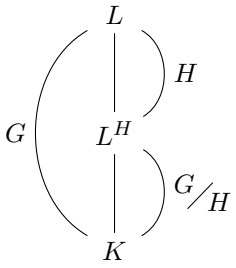
$$\mathsf{Aut}(L/K) \stackrel{\text{def}}{=} \{\sigma\colon L \to L \mid \sigma \text{ is an isomorphism with } \sigma(a) = a \text{ for all } a \in K\}$$

has cardinality exactly $[L : K]$. In that case, $\mathsf{Aut}(L/K)$ is called the *Galois group* of L/K and is denoted by $\mathsf{Gal}(L/K)$. Galois extensions have many properties that do not hold in general field extensions, and L/K is called an *abelian extension* (resp. *cyclic extension*) when $\mathsf{Gal}(L/K)$ is abelian (resp. cyclic). Amongst all the algebraic extensions, abelian extensions are the most well-understood, and in the case of global (and local[i]) fields, they are studied through the so-called *Class Field Theory*.

When L/K is Galois, and if H is a subgroup of $G \stackrel{\text{def}}{=} \mathsf{Gal}(L/K)$, then the sets of invariants

$$L^H \stackrel{\text{def}}{=} \{a \in L \mid \sigma(a) = a \quad \forall \sigma \in H\}$$

is a field called the *fixed field* of H. By definition, $L^G = K$. Furthermore, the extension L/L^H is always Galois with Galois group H. On the other hand, the extension L^H/K may not be Galois in general, but it is the case when H is a normal subgroup of G, and $\mathsf{Gal}(L^H/K) = {}^{G}\!/_{H}$. In particular, this is the case when L/K is abelian.



Let $K/\mathbb{F}_q(T)$ be a Galois function field with ring of integers $\mathcal{O}_K$. The Galois group $G \stackrel{\text{def}}{=} \mathsf{Gal}(K/\mathbb{F}_q(T))$ keeps $\mathcal{O}_K$ globally invariant:

$$\forall \sigma \in \mathsf{Gal}(K/\mathbb{F}_q(T)), \quad \sigma(\mathcal{O}_K) = \mathcal{O}_K.$$

Furthermore, given $\mathfrak{p}$ a prime ideal of $\mathbb{F}_q[T]$, the group G acts transitively on the set $\{\mathfrak{P}_1, \ldots, \mathfrak{P}_r\}$ of prime ideals of $\mathcal{O}_K$ lying above $\mathfrak{p}$:

$$\forall i \neq j, \quad \exists \sigma \in \mathsf{Gal}(K/\mathbb{F}_q(T)), \quad \sigma(\mathfrak{P}_i) = \mathfrak{P}_j.$$

In particular, all the ramification indexes e_i (resp. the inertia degrees f_i) are equal and denoted by e (resp. f):

$$\mathfrak{P} \stackrel{\text{def}}{=} \mathfrak{p}\mathcal{O}_K = (\mathfrak{P}_1 \ldots \mathfrak{P}_r)^e,$$

[i] A local field K is the completion of a number field or an algebraic function field at some non-archimedean place. It is either an algebraic extension of a p-adic field $\mathbb{Q}_p$, or is isomorphic to $\mathbb{F}_q((u))$ the field of Laurent series in the variable u with coefficients in a finite field $\mathbb{F}_q$.

and the fundamental relation 4.1 becomes

$$n \stackrel{\text{def}}{=} [K : \mathbb{F}_q(T)] = efr.$$

Another consequence which will be crucial for the applications, is that the action of G on $\mathcal{O}_K$ is well-defined on the quotient

$$\mathcal{O}_K \big/ \mathfrak{p} = \prod_{i=1}^{r} \mathcal{O}_K \big/ \mathfrak{p}_i^e$$

and simply permutes the factors.

The *decomposition group* of $\mathfrak{P}_i$ is the subgroup of $\mathrm{Gal}(K/\mathbb{F}_q(T))$ which keeps $\mathfrak{P}_i$ globally invariant

$$D_{\mathfrak{P}_i/\mathfrak{p}} \stackrel{\text{def}}{=} \{\sigma \in \mathrm{Gal}(K/\mathbb{F}_q(T)) \mid \sigma(\mathfrak{P}_i) = \mathfrak{P}_i\}. \tag{4.2}$$

In general, the decomposition groups are distinct. However, they are all conjugate under the Galois action: Let $i \neq j$ and let $\sigma \in \mathrm{Gal}(K/\mathbb{F}_q(T))$ such that $\mathfrak{P}_j = \sigma\mathfrak{P}_i$. Then,

$$\sigma^{-1} D_{\mathfrak{P}_j/\mathfrak{p}} \sigma = D_{\mathfrak{P}_i/\mathfrak{p}}. \tag{4.3}$$

Indeed,

$$\begin{aligned} \theta \in D_{\mathfrak{P}_j/\mathfrak{p}} &\iff \theta(\sigma(\mathfrak{P}_i)) = \sigma(\mathfrak{P}_i) \iff (\sigma^{-1}\theta\sigma)(\mathfrak{P}_i) = \mathfrak{P}_i \\ &\iff \sigma^{-1}\theta\sigma \in D_{\mathfrak{P}_i/\mathfrak{p}} \iff \theta \in \sigma D_{\mathfrak{P}_i/\mathfrak{p}} \sigma^{-1}. \end{aligned}$$

In particular, when the extension $K/\mathbb{F}_q(T)$ is abelian, they are all equal and only depend on the ground prime $\mathfrak{p}$.

Let $Q(T) \in \mathbb{F}_q[T]$ be the monic polynomial defining $\mathfrak{p}$, and let $d \stackrel{\text{def}}{=} \deg(Q)$. An element $\sigma \in D_{\mathfrak{P}_i/\mathfrak{p}}$ satisfies $\sigma(\mathcal{O}_K) = \mathcal{O}_K$ and $\sigma(\mathfrak{P}_i) = \mathfrak{P}_i$, and therefore defines an automorphism $\overline{\sigma}$ of

$$\mathbb{F}_{q^{fd}} \stackrel{\text{def}}{=} \mathcal{O}_K \big/ \mathfrak{P}_i,$$

where f is the inertia degree at $\mathfrak{p}$. Furthermore, since $\sigma \in \mathrm{Gal}(K/\mathbb{F}_q(T))$, we have that

$$\overline{\sigma}_{|\mathbb{F}_{q^{\deg(\mathfrak{p})}}} = \mathrm{Id}.$$

In particular, $\overline{\sigma} \in \mathrm{Gal}(\mathbb{F}_{q^{fd}}/\mathbb{F}_{q^d})$, which is a cyclic group of order f.

It is clear that the map

$$\begin{cases} D_{\mathfrak{P}_i/\mathfrak{p}} & \to & \mathrm{Gal}(\mathbb{F}_{q^{fd}}/\mathbb{F}_{q^d}) \\ \sigma & \mapsto & \overline{\sigma} \end{cases}$$

is a surjective group homomorphism, whose kernel is denoted by $I_{\mathfrak{P}_i/\mathfrak{p}}$ and is referred to as the *inertia group* of $\mathfrak{P}_i$ over $\mathfrak{p}$. In particular, this defines the short exact sequence

$$1 \to I_{\mathfrak{P}_i/\mathfrak{p}} \to D_{\mathfrak{P}_i/\mathfrak{p}} \to \mathrm{Gal}(\mathbb{F}_{q^{fd}}/\mathbb{F}_{q^d}) \to 1.$$

It is well-known that when $K/\mathbb{F}_q(T)$ is unramified at $\mathfrak{p}$, then $I_{\mathfrak{P}_i/\mathfrak{p}} = \{\mathrm{Id}\}$, and therefore

$$D_{\mathfrak{P}_i/\mathfrak{p}} \simeq \mathrm{Gal}(\mathbb{F}_{q^{fd}}/\mathbb{F}_{q^d}) \tag{4.4}$$

is cyclic, generated by an automorphism called the *Frobenius element* at $\mathfrak{p}$.

4.2.3 Cyclotomic Function Fields and the Carlitz Module

Recall that the goal is to design a framework which we can use to prove some results on structured code-based cryptography. The idea is that those codes are endowed with the action of some group, and we would like to see this action as the action of a Galois group on some ring of integers of function fields, analogously to the work of [LPR10].

Generally, those groups are abelian, and therefore it makes sense to focus on abelian extensions. In particular, lattice-based cryptography heavily makes use of cyclotomic number fields, which are in some sense the *generic* abelian extensions of the field $\mathbb{Q}$ of rational numbers. Indeed, a famous theorem of Kronecker and Weber shows that any abelian extension of $\mathbb{Q}$ is a subfield of some cyclotomic extension $\mathbb{Q}(\zeta_m)$, where ζ_m is a primitive m-th root of unity. Cyclotomic number fields are all the more interesting to work with that they are very well known, both from a theoretical point of view and also algorithmically.

In this section, we give an introduction to their function field analogues, namely *Carlitz extensions*. A dictionary summarising the similarities between the two objects is given in Table 4.2, and a more detailed presentation can be found in [Ros02, Chap. 12; Vil06, Chap. 12] or in the excellent survey [Con].

Carlitz extensions bare the name of Leonard Carlitz who studied those abelian extensions of rational function fields in the late 1930s, but the analogy with the cyclotomic number fields was not well-known until the work of his student Hayes who proved a function field analogue of the Kronecker-Weber [Hay74]. This explicit class field theory for the rational function field was generalised in the following years with the work of Drinfeld and Goss to yield a complete solution to Hilbert twelfth problem in the function field setting. It is spectacular that in the number field setting such an explicit construction is only known for abelian extensions of $\mathbb{Q}$ (cyclotomic extensions) and imaginary quadratic number fields (via the theory of elliptic curves with complex multiplication).

Recall the recurrent setting in this chapter: starting from a ground prime $\mathfrak{p} \subset \mathbb{F}_q[T]$, we want to consider a finite extension $K/\mathbb{F}_q(T)$ together with its ring of integers $\mathcal{O}_K$, and look at the decomposition of $\mathfrak{P} \overset{\text{def}}{=} \mathfrak{p}\mathcal{O}_K$.

$$\begin{array}{ccc} \mathfrak{P} \subset \mathcal{O}_K & \longrightarrow & K \\ \big| & & \big| \\ \mathfrak{p} \subset \mathbb{F}_q[T] & \longrightarrow & \mathbb{F}_q(T) \end{array}$$

By the Chinese Remainder Theorem, when $\mathfrak{p}$ does not ramify, the quotient $\mathcal{O}_K/\mathfrak{P}$ is a product of finite fields, and for reasons which will be clear in Theorem 4.30, we want them to be as small as possible. Ideally, they should be equal to $\mathbb{F}_q$. However, the constant field k of K is always a subfield of them, so this ideal scenario is not possible if $[k : \mathbb{F}_q] > 1$. Therefore, we want to only consider *geometric extensions*.

4.2.3.1 Roots of unity and torsion

The first idea that comes to mind when one wants to build cyclotomic function fields is to adjoin roots of unity to the field $\mathbb{F}_q(T)$. However, roots of unity are already *algebraic* over $\mathbb{F}_q$, and adjoining them only increases the field of constants, which we want to avoid.

Example 4.6. *As an example, consider the polynomial* T^2+T+1 *over* $\mathbb{F}_2$*. It is irreducible. Let* $\zeta_3 \in \mathbb{F}_4$ *be one of its roots. It is a cube root of* 1*. Now, consider the field extension*

$$K \stackrel{def}{=} \mathbb{F}_2(T)(\zeta_3) = \mathbb{F}_4(T)$$

and let $\mathcal{O}_K$ *be the integral closure of* $\mathbb{F}_2[T]$ *in* K*. The prime ideal* $\mathfrak{p} \stackrel{def}{=} (T^2+T+1)$ *of* $\mathbb{F}_2[T]$ *splits into two prime ideals* $\mathfrak{P}_1$ *and* $\mathfrak{P}_2$ *in* $\mathcal{O}_K$*. But* $\mathcal{O}_K/\mathfrak{P}_1 = \mathcal{O}_K/\mathfrak{P}_2 = \mathbb{F}_4 = \mathbb{F}_2[T]/\mathfrak{p}$ *and we do not win anything by considering the extension* $K/\mathbb{F}_2(T)$*.*

Hence, adjoining roots of unity is not the right idea to obtain our correct analogue. One has to go deeper into the algebraic structure that is adjoined to $\mathbb{Q}$. Indeed, the set of all m–th roots of unity, denoted by $\mu_m \subset \mathbb{C}$, turns out to be an abelian group under multiplication. Moreover, μ_m is in fact *cyclic*, generated by any *primitive* root of unity.

Recall that abelian groups are exactly $\mathbb{Z}$*-modules*. Here the action of $\mathbb{Z}$ is given by exponentiation: $n \in \mathbb{Z}$ acts on $\zeta \in \mu_m$ by $n \cdot \zeta \stackrel{\text{def}}{=} \zeta^n$. This action can in fact be extended to all $\overline{\mathbb{Q}}^\times$. When working with modules over a ring, it is very natural to consider the *torsion elements*, *i.e.* elements of the module that are annihilated by an element of the ring. The torsion elements in the $\mathbb{Z}$–module $\overline{\mathbb{Q}}^\times$ are the $\zeta \in \overline{\mathbb{Q}}^\times$ such that $\zeta^m = 1$ for some $m > 0$; these are precisely the roots of unity. In other words, the cyclotomic number fields are obtained by adjoining to $\mathbb{Q}$ torsions elements of the $\mathbb{Z}$–module $\overline{\mathbb{Q}}^\times$.

Making use of the analogy between number fields and function fields recalled in Section 4.2.2.2, and summed up in Table 4.1, it is more than tempting to adjoin to $\mathbb{F}_q(T)$ the torsion submodule of some well chosen $\mathbb{F}_q[T]$–module $\mathcal{M}$. Note that $\mathbb{F}_q[T]$–modules are in particular $\mathbb{F}_q$–vector spaces and the action of $\mathbb{F}_q[T]$ is $\mathbb{F}_q$-linear. In particular, $\mathcal{M}$ must contain 0. The natural candidate for $\mathcal{M}$ is $\overline{\mathbb{F}_q(T)}$, with $\mathbb{F}_q[T]$ acting by multiplication. However, the torsion submodule

$$\mathcal{M}^{tor} \stackrel{\text{def}}{=} \{x \in \overline{\mathbb{F}_q(T)} \mid a \cdot x = 0 \text{ for some non-zero } a \in \mathbb{F}_q[T]\}$$

is in reality trivial. Thus, we need to *twist* this action of $\mathbb{F}_q[T]$ and define *another* $\mathbb{F}_q[T]$–module structure. Note that in the cyclotomic case, we did not consider the additive abelian group $\overline{\mathbb{Q}}$, but rather the *multiplicative* group $\overline{\mathbb{Q}}^\times$.

Remark 4.7. *This point of view has actually a geometric interpretation. Indeed, to an algebraically closed field* k*, one may associate a geometric object called the* multiplicative group scheme $\mathbb{G}_{m,k}$*, which is an algebraic variety endowed with a group structure, isomorphic* as groups *to* $(k^\times, \cdot)$*. It turns out that the endomorphism ring of* $\mathbb{G}_{m,k}$ *is isomorphic to* $\mathbb{Z}$*, given by exponentiation. In other words, the above action identifies with the action of* $\mathrm{End}(\mathbb{G}_{m,\overline{Q}})$ *on* $\mathbb{G}_{m,\overline{Q}}$*.*

4.2.3.2 Carlitz polynomials

The action of $\mathbb{F}_q[T]$ on $K \stackrel{\text{def}}{=} \overline{\mathbb{F}_q(T)}$ is uniquely determined by the action of T. The previous attempt only considered the endomorphism of multiplication by T:

$$\mu_T : \left\{ \begin{array}{ccc} K & \to & K \\ u & \mapsto & Tu \end{array} \right.$$

However, the set $\mathrm{End}_{\mathbb{F}_q}(K)$ of $\mathbb{F}_q$–endomorphisms of K contains another special element, namely the Frobenius:

$$\mathrm{Frob} : \left\{ \begin{array}{ccc} K & \to & K \\ u & \mapsto & u^q \end{array} \right.$$

Given $M(T) \in \mathbb{F}_q[T]$, the substitution $T \mapsto \mathrm{Frob} + \mu_T$ in M yields a ring homomorphism $\varphi\colon \mathbb{F}_q[T] \to \mathrm{End}_{\mathbb{F}_q}(K)$. A polynomial $M \in \mathbb{F}_q[T]$ will then act on $\alpha \in K$ by $M \cdot \alpha \stackrel{\text{def}}{=} \varphi(M)(\alpha)$.

Remark. *In the literature, the notation α^M can also be found to emphasise the analogy with the action of $\mathbb{Z}$ by exponentiation, but it can be confusing.*

Note that $\varphi(M)$ can be represented by a *linearised* polynomial $[M](X) \in \mathbb{F}_q(T)[X]$, *i.e.* a polynomial whose monomials are only q–th powers of X, namely of the form

$$P(X) = p_0 X + p_1 X^q + \cdots + p_r X^{q^r},$$

where $p_i \in \mathbb{F}_q(T)$.

More precisely, define the Carlitz polynomials by induction and linearity:

- Set $[1](X) \stackrel{\text{def}}{=} X$ and $[T](X) \stackrel{\text{def}}{=} X^q + TX$.
- For $n \geqslant 2$, define

$$[T^n](X) \stackrel{\text{def}}{=} [T]([T^{n-1}](X)) = [T^{n-1}](X)^q + T[T^{n-1}](X).$$

- Then, for a polynomial $M = \sum_{i=0}^{n} a_i T^i \in \mathbb{F}_q[T]$, define $[M](X)$ by enforcing $\mathbb{F}_q$–linearity:

$$[M](X) \stackrel{\text{def}}{=} \sum_{i=0}^{n} a_i [T^i](X).$$

Example 4.8. *We have,*

- $[T^2](X) = [T](X^q + TX) = X^{q^2} + (T^q + T)X^q + T^2 X$
- $[T^2 + T + 1](X) = [T^2](X) + [T](X) + [1](X) = X^{q^2} + (T^q + T + 1)X^q + (T^2 + T + 1)X$

By construction, Carlitz polynomials are additive polynomials, and $\mathbb{F}_q$–linear. Furthermore, for two polynomials $M, N \in \mathbb{F}_q[T]$, $[MN](X) = [M]([N](X)) = [N]([M](X))$. In particular, Carlitz polynomials commute with each other under composition law, which is not the case in general for q–polynomials, which are used for instance to define codes in the rank metric (see Part I).

4.2.3.3 The Carlitz Module

Endowed with this new $\mathbb{F}_q[T]$–module structure, $\overline{\mathbb{F}_q(T)}$ is referred to as the *Carlitz module.*

Definition 4.9

For $M \in \mathbb{F}_q[T]$, $M \neq 0$, let

$$\Lambda_M \stackrel{\text{def}}{=} \left\{ \lambda \in \overline{\mathbb{F}_q(T)} \mid [M](\lambda) = 0 \right\}.$$

This is the submodule of M–torsion of the Carlitz module.

Example 4.10. *The submodule of T–torsion is*

$$\begin{aligned}\Lambda_T &= \{\lambda \in \overline{\mathbb{F}_q(T)} \mid \lambda^q + T\lambda = 0\} \\ &= \{0\} \cup \{\lambda \mid \lambda^{q-1} = -T\}.\end{aligned}$$

Note that Λ_M is a submodule of the Carlitz module: for $\lambda \in \Lambda_M$ and $A \in \mathbb{F}_q[T]$, $[A](\lambda) \in \Lambda_M$. In particular, Λ_M is an $\mathbb{F}_q$–vector space. This is similar to the fact that the set μ_m of m–th roots of unity is a subgroup of $\overline{\mathbb{Q}}^\times$, *i.e.* a $\mathbb{Z}$–module.

Example 4.11. *The module Λ_T defined in Example 4.10 is an $\mathbb{F}_q$–vector space of dimension 1. In particular, for $\lambda \in \Lambda_T$, and $A \in \mathbb{F}_q[T]$, the element $[A](\lambda)$ must be a multiple of λ. In fact the Carlitz action of A on λ is through the constant term of A: writing $A = TB + A(0)$ we have*

$$[A](\lambda) = [TB + A(0)](\lambda) = [B](\underbrace{[T](\lambda)}_{=0}) + A(0)[1](\lambda) = A(0)\lambda.$$

More generally, even if Λ_M is not of dimension 1 over $\mathbb{F}_q$, it is always a *cyclic* $\mathbb{F}_q[T]$–module: as an $\mathbb{F}_q[T]$–module it can be generated by only one element. This is specified in the following theorem.

Theorem 4.12 ([Vil06, Theorem 12.2.17])

There exists $\lambda_0 \in \Lambda_M$ such that

$$\Lambda_M = \{[A](\lambda_0) \mid A \in \mathbb{F}_q[T]/(M)\}$$

and the generators of Λ_M are the $[A](\lambda_0)$ for all A prime to M.

The choice of a generator in the above theorem yields a non canonical isomorphism $\Lambda_M \simeq \mathbb{F}_q[T]/(M)$ as $\mathbb{F}_q[T]$–modules.

Remark 4.13. *This result needs to be related to the cyclotomic case: given the choice of a primitive m–th root of unity, there is a group isomorphism between μ_m and $\mathbb{Z}/m\mathbb{Z}$. Moreover all the m–th roots of unity are of the form ζ^k for $k \in \{0, m-1\}$ and the generators of μ_m are the ζ^k for k prime to m.*

4.2.3.4 Carlitz Extensions

Recall that the cyclotomic number fields are obtained as extensions of $\mathbb{Q}$ generated by the elements of μ_m. In the similar fashion, for a polynomial $M \in \mathbb{F}_q[T]$, let

$$K_M \stackrel{\text{def}}{=} \mathbb{F}_q(T)(\Lambda_M) = \mathbb{F}_q(T)(\lambda_M),$$

where λ_M is a generator of Λ_M. One of the most important facts about the cyclotomic number field $\mathbb{Q}(\zeta_m)$ is that it is a finite Galois extension of $\mathbb{Q}$, with Galois group isomorphic to $(\mathbb{Z}/m\mathbb{Z})^\times$. There is an analogue statement for the Carlitz extensions.

Theorem 4.14 ([Vil06])

Let $M \in \mathbb{F}_q[T]$, $M \neq 0$. Then K_M is a finite Galois extension of $\mathbb{F}_q(T)$, with Galois group isomorphic to $(\mathbb{F}_q[T]/(M))^\times$. The isomorphism is given by

$$\begin{cases} (\mathbb{F}_q[T]/(M))^\times & \longrightarrow & \mathrm{Gal}(K_M/\mathbb{F}_q(T)) \\ A & \longmapsto & \sigma_A, \end{cases}$$

where σ_A is completely determined by $\sigma_A(\lambda_M) = [A](\lambda_M)$.

Remark 4.15. *In particular, Carlitz extensions are* abelian.

Recall that the whole point of this construction is to yield *geometric* extensions. This is ensured by the following non-trivial theorem.

Theorem 4.16 ([Ros02, Cor. of Th. 12.14])

Let $M \in \mathbb{F}_q[T]$, $M \neq 0$. Then $\mathbb{F}_q$ is the full *constant field* of K_M.

Another important fact about cyclotomic extensions is the simple description of their ring of integers. Namely, for $K = \mathbb{Q}(\zeta_m)$, we have $\mathcal{O}_K = \mathbb{Z}[\zeta_m] = \mathbb{Z}[X]/(\Phi_m(X))$ where Φ_m denotes the m–th cyclotomic polynomial. This property also holds for Carlitz extensions.

Theorem 4.17 ([Ros02, Theorem 2.9])

Let $\mathcal{O}_{K_M}$ be the integral closure of $\mathbb{F}_q[T]$ in K_M. Then $\mathcal{O}_{K_M} = \mathbb{F}_q[T][\lambda_M]$. In particular, let $P(T,X) \in \mathbb{F}_q[T][X]$ be the minimal polynomial of λ_M. Then,

$$K_M = \mathbb{F}_q(T)[X]\big/(P(T,X)) \quad \text{and} \quad \mathcal{O}_{K_M} = \mathbb{F}_q[T][X]\big/(P(T,X)).$$

Example 4.18. *Reconsider Example 4.10 and the module* $\Lambda_T = \{0\} \cup \{\lambda \mid \lambda^{q-1} = -T\}$. *The polynomial* $X^{q-1} + T$ *is Eisenstein in* (T) *and therefore is irreducible. Hence,*

$$K_T = \mathbb{F}_q(T)[X]\big/(X^{q-1} + T).$$

Moreover it is Galois, with Galois group $(\mathbb{F}_q[T]/(T))^\times \simeq \mathbb{F}_q^\times$. *A non-zero element* $a \in \mathbb{F}_q^\times$ *will act on* $f(T,X) \in K_T$ *by*

$$a \cdot f(T,X) \overset{def}{=} f(T,[a](X)) = f(T,aX).$$

The integral closure of $\mathbb{F}_q[T]$ *in* K_T *is*

$$\mathcal{O}_{K_T} \overset{def}{=} \mathbb{F}_q[T][X]\big/(X^{q-1} + T)$$

and

$$\mathcal{O}_{K_T}\big/((T+1)\mathcal{O}_{K_T}) = \mathbb{F}_q[T][X]\big/(T+1, X^{q-1}+T) = \mathbb{F}_q[X]\big/(X^{q-1}-1). \tag{4.5}$$

Finally, the following theorem characterises the splitting behaviour of primes in Carlitz ex-

tensions. A very similar result holds for cyclotomic extensions.

Theorem 4.19 ([Ros02, Th. 12.10])

Let $M \in \mathbb{F}_q[T]$, $M \neq 0$, and let $Q \in \mathbb{F}_q[T]$ be a monic, irreducible polynomial. Consider the Carlitz extension K_M and let $\mathcal{O}_{K_M}$ denote its ring of integers. Then,

- If Q divides M, then $Q\mathcal{O}_{K_M}$ is totally ramified.
- Otherwise, let f be the smallest integer f such that $Q^f \equiv 1 \mod M$. Then $Q\mathcal{O}_{K_M}$ is unramified and has inertia degree f. In particular, Q splits completely if and only if $Q \equiv 1 \mod M$.

Note that in Ring–LWE, the prime modulus q is often chosen such that $q \equiv 1 \mod m$ so that it splits completely in the cyclotomic extension $\mathbb{Q}(\zeta_m)$.

Example 4.20. *In the previous example, $T+1 \equiv 1 \mod T$ and therefore $(T+1)$ splits completely in $\mathcal{O}_T$. Indeed,*

$$\mathcal{O}_T \big/ ((T+1)\mathcal{O}_T) = \mathbb{F}_q[X] \big/ (X^{q-1} - 1) = \prod_{\alpha \in \mathbb{F}_q^\times} \mathbb{F}_q[X] \big/ (X - \alpha)$$

is a product of $q-1$ copies of $\mathbb{F}_q$.

The similarities between Carlitz function fields and cyclotomic number fields are summarised in Table 4.2.

$\mathbb{Q}$	$\mathbb{F}_q(T)$
$\mathbb{Z}$	$\mathbb{F}_q[T]$
Prime numbers $q \in \mathbb{Z}$	Irreducible polynomials $Q \in \mathbb{F}_q[T]$
$\mu_m = \langle \zeta \rangle \simeq \mathbb{Z}/m\mathbb{Z}$ (groups)	$\Lambda_M = \langle \lambda \rangle \simeq \mathbb{F}_q[T]/(M)$ (modules)
$d \mid m \Leftrightarrow \mu_d \subset \mu_m$ (subgroups)	$D \mid M \Leftrightarrow \Lambda_D \subset \Lambda_M$ (submodules)
$a \equiv b \mod m \Rightarrow \zeta^a = \zeta^b$	$A \equiv B \mod M \Rightarrow [A](\lambda) = [B](\lambda)$
$K = \mathbb{Q}[\zeta]$ $\mathcal{O}_K = \mathbb{Z}[\zeta]$	$K = \mathbb{F}_q(T)[\lambda]$ $\mathcal{O}_K = \mathbb{F}_q[T][\lambda]$
$\mathrm{Gal}(K/\mathbb{Q}) \simeq (\mathbb{Z}/m\mathbb{Z})^\times$	$\mathrm{Gal}(K/\mathbb{F}_q(T)) \simeq (\mathbb{F}_q[T]/(M))^\times$
Cyclotomic	**Carlitz**

Table 4.2: Analogies between cyclotomic number fields and Carlitz function fields

4.3 The Function Field Decoding Problem

In this section, we introduce a new generic problem that we call the Function Field Decoding Problem (FF-DP) which is the analogue of Ring–LWE in the context of function fields. The main theorem which states the search-to-decision reduction is given in Theorem 4.30, and is proven in Section 4.3.3. In view of the analogy between function fields and number fields described in the previous section, the reduction works similarly as that of [LPR10]. Section 4.4 will discuss instantiations in the code-based setting.

4.3.1 Search and decision problems.

Consider a function field $K/\mathbb{F}_q(T)$ with constant field $\mathbb{F}_q$ and ring of integers $\mathcal{O}_K$, and let $\mathfrak{p} \subset \mathbb{F}_q[T]$ be a prime ideal, generated by an irreducible polynomial Q which is called the *modulus*. Denote by $\mathfrak{P} \stackrel{\text{def}}{=} Q\mathcal{O}_K$ be the ideal of $\mathcal{O}_K$ lying above Q, and recall that $\mathcal{O}_K/\mathfrak{P}$ is a finite ring. FF-DP is parameterised by an element $\mathbf{s} \in \mathcal{O}_K/\mathfrak{P}$ called the *secret* and by a probability distribution ψ defined over $\mathcal{O}_K/\mathfrak{P}$ called the *error distribution*.

Definition 4.21 (FF-DP Distribution)

A sample $(\mathbf{a}, \mathbf{b}) \in \mathcal{O}_K/\mathfrak{P} \times \mathcal{O}_K/\mathfrak{P}$ is distributed according to the FF-DP distribution modulo $\mathfrak{P}$ with secret $\mathbf{s}$ and error distribution ψ if

- $\mathbf{a}$ is uniformly distributed over $\mathcal{O}_K/\mathfrak{P}$,
- $\mathbf{b} = \mathbf{a}\mathbf{s} + \mathbf{e} \in \mathcal{O}_K/\mathfrak{P}$ where $\mathbf{e}$ is distributed according to ψ.

We denote by $(\mathbf{a}, \mathbf{b}) \leftarrow \mathcal{F}_{\mathbf{s},\psi}$ a sample drawn with respect to this distribution.

The aim of the search version of the FF-DP problem is to recover the secret $\mathbf{s}$ given samples drawn from $\mathcal{F}_{\mathbf{s},\psi}$. This is formalised in the following problem.

Problem 4.22 (FF-DP, Search version)

Let $\mathbf{s} \in \mathcal{O}_K/\mathfrak{P}$, and let ψ be a probability distribution over $\mathcal{O}_K/\mathfrak{P}$. An instance of FF-DP problem consists in an oracle giving access to independent samples $(\mathbf{a}, \mathbf{b}) \leftarrow \mathcal{F}_{\mathbf{s},\psi}$. The goal is to recover $\mathbf{s}$.

Example 4.23. *Let $n, \ell > 1$ be integers, and let Ψ be a probability distribution over $\mathbb{F}_q[X]/(X^n - 1)$. Recall from Problem 4.1 that the search version of QC-DP(ℓ, Ψ) corresponds to finding a polynomial $\mathbf{m} \in \mathbb{F}_q[X]/(X^n - 1)$ given access to ℓ samples of the form*

$$(\mathbf{a}, \mathbf{m} \cdot \mathbf{a} + \mathbf{e}) \in \mathbb{F}_q[X]/(X^n - 1) \times \mathbb{F}_q[X]/(X^n - 1),$$

where $\mathbf{a}$ is uniformly distributed in $\mathbb{F}_q[X]/(X^n - 1)$ and $\mathbf{e} \leftarrow \Psi$.

Furthermore, Example 4.5 shows that

$$\mathbb{F}_q[X]\big/_{X^n-1} \simeq \mathcal{O}_K\big/_{Q\mathcal{O}_K},$$

where $Q(T) \stackrel{def}{=} T \in \mathbb{F}_q[T]$ *is an irreducible polynomial, and*

$$\mathcal{O}_K \stackrel{def}{=} \mathbb{F}_q[T,X]\big/_{(X^n+T-1)}$$

is the ring of integers of the algebraic function field

$$K \stackrel{def}{=} \mathbb{F}_q(T)[X]\big/_{(X^n+T-1)}.$$

In other words, FF-DP is a generalisation of the decoding problem of quasi-cyclic codes, when considering arbitrary function fields and irreducible moduli.

For cryptographic applications, we are also interested in the *decision* version of this problem. The goal is now to distinguish between the FF-DP distribution and the uniform distribution over $\mathcal{O}_K/\mathfrak{p} \times \mathcal{O}_K/\mathfrak{p}$.

Problem 4.24 (FF-DP, Decision version)

Let $\mathbf{s}$ be drawn uniformly at random in $\mathcal{O}_K\big/_{\mathfrak{p}}$ and let ψ be a probability distribution over $\mathcal{O}_K\big/_{\mathfrak{p}}$, and consider the following two distributions:

- Let $\mathcal{D}_0 \colon (\mathbf{a}, \mathbf{y}^{\text{unif}})$ be the uniform distribution over $\mathcal{O}_K\big/_{\mathfrak{p}} \times \mathcal{O}_K\big/_{\mathfrak{p}}$,
- let $\mathcal{D}_1 \stackrel{\text{def}}{=} \mathcal{F}_{\mathbf{s},\psi} \colon (\mathbf{a}, \mathbf{as}+\mathbf{e})$ be the FF-DP distribution with secret $\mathbf{s}$ and error distribution ψ.

Given access to an oracle $\mathcal{O}_b$ providing samples from distribution $\mathcal{D}_b$ where $b \leftarrow \{0,1\}$ is a uniformly random bit, the goal is to recover b.

Remark 4.25. *In the decision version, it may be more convenient for some applications to have the secret* $\mathbf{s}$ *drawn from the error distribution* ψ*, instead of the uniform distribution over* $\mathcal{O}_K\big/_{\mathfrak{p}}$*. In the lattice-based setting, this version is sometimes called LWE with* short secret *or LWE in* Hermite normal form*. However, both decision problems are easily proved to be computationally equivalent, see [Lyu11, Lemma 3]. The proof applies* mutatis mutandis *to FF-DP.*

Remark 4.26. *Following the previous remark, note that in the code-based setting, FF-DP with short secret corresponds to the Decoding Problem, but stated in terms of parity-check matrices (in systematic form). For example, consider the most common case of random* $[2n,n]$*, double-circulant code* $\mathscr{C}$*, with a parity-check matrix* $\mathbf{H}$ *in systematic form as used in BIKE and HQC cryptosystems. In other words,* $\mathbf{H}$ *is of the form*

$$\mathbf{H} \stackrel{def}{=} \begin{pmatrix} I_n & \mathbf{h} \\ & \circlearrowleft \end{pmatrix}.$$

Let $\mathbf{e} = (\mathbf{e}^{(1)}, \mathbf{e}^{(2)}) \in \mathbb{F}_q^{2n}$ *be an error vector with* $\mathbf{e}^{(i)} \leftarrow \Psi$ *following the same distribution* Ψ*. For*

example, in both BIKE and HQC, $\mathbf{e}$ *is such that* $|\mathbf{e}^{(1)}| = |\mathbf{e}^{(2)}| = \omega$ *with small weight* ω*. Now, a syndrome is of the form*

$$\mathbf{H}\mathbf{e}^\top = \mathbf{e}^{(1)} + \begin{pmatrix} \mathbf{h} \\ \circlearrowleft \end{pmatrix} \cdot \mathbf{e}^{(2)\top},$$

or in the polynomial representation

$$\mathbf{e}^{(1)}(X) + \mathbf{h}(X) \cdot \mathbf{e}^{(2)}(X) \in \mathbb{F}_q[X] \big/ (X^n - 1).$$

In other words, a random $[2n, n]$ *double-circulant quasi-cyclic code yields one sample*

$$(\mathbf{h}, \mathbf{e}^{(1)} + \mathbf{h} \cdot \mathbf{e}^{(2)})$$

where $\mathbf{e}^{(2)}$ *can now be thought of as a secret with the same distribution as the error term* $\mathbf{e}^{(1)}$*.*

The generalisation to a higher rate code is straightforward.

A module version. Instead of considering one secret $\mathbf{s} \in \mathcal{O}_K/\mathfrak{P}$, we could use multiple secrets $(\mathbf{s}_1, \ldots, \mathbf{s}_d) \in (\mathcal{O}_K/\mathfrak{P})^d$. This generalisation has been considered in lattice-based cryptography under the terminology Module-LWE [LS15], where the secret can be thought as an element of $\mathcal{O}_K^d$ which is a free $\mathcal{O}_K$-module of rank d, before a reduction modulo $\mathfrak{P}$ on each component. This would yield the following definition.

Definition 4.27 (MFF-DP Distribution)

Let $d \geqslant 1$ be an integer. A sample $(\mathbf{a}, \mathbf{b}) \in (\mathcal{O}_K/\mathfrak{P})^d \times \mathcal{O}_K/\mathfrak{P}$ is distributed according to the MFF-DP distribution modulo $\mathfrak{P}$ with secret $\mathbf{s} \stackrel{\text{def}}{=} (\mathbf{s}_1, \ldots, \mathbf{s}_d) \in (\mathcal{O}_K/\mathfrak{P})^d$ and error distribution ψ over $\mathcal{O}_K/\mathfrak{P}$ if

- $\mathbf{a}$ is uniformly distributed over $(\mathcal{O}_K/\mathfrak{P})^d$,
- $\mathbf{b} = \sum_{i=1}^d \mathbf{a}_i \mathbf{s}_i + \mathbf{e} \in \mathcal{O}_K/\mathfrak{P}$ where $\mathbf{e}$ is distributed according to ψ.

The search and decision problems associated to MFF-DP can be defined as a natural generalisation of Problems 4.22 and 4.24.

Problem 4.28 (MFF-DP, Search version)

Let $\mathbf{s} \in (\mathcal{O}_K/\mathfrak{P})^d$ be a collection of elements of $\mathcal{O}_K/\mathfrak{P}$ called the secrets, and let ψ be a probability distribution over $\mathcal{O}_K/\mathfrak{P}$. An instance of the MFF-DP problem consists in an oracle giving access to independent samples $(\mathbf{a}, \mathbf{b})$ from the MFF-DP distribution with secrets $\mathbf{s}$ and error distribution ψ. The goal is to recover $\mathbf{s}$.

Problem 4.29 (MFF-DP, Decision version)

Let $\mathbf{s}$ be drawn uniformly at random in $(\mathcal{O}_K/\mathfrak{P})^d$ and let ψ be a probability distribution over $\mathcal{O}_K/\mathfrak{P}$. Define $\mathscr{D}_0$ to be the uniform distribution over $(\mathcal{O}_K/\mathfrak{P})^d \times \mathcal{O}_K/\mathfrak{P}$, and $\mathscr{D}_1$ to be the MFF-DP distribution with secrets $\mathbf{s}$ and error distribution ψ. Furthermore, let b be a uniform element of $\{0,1\}$. Given access to an oracle $\mathscr{O}_b$ providing samples from distribution $\mathscr{D}_b$, the goal of the decision MFF-DP is to recover b.

4.3.2 Search to decision reduction

There is an obvious reduction from the decision to the search version of FF-DP. Indeed, if there exists an algorithm $\mathscr{A}$ that given access to the $\mathcal{F}_{\mathbf{s},\psi}$ distribution is able to recover the secret $\mathbf{s}$, then it yields to a distinguisher between $\mathcal{F}_{\mathbf{s},\psi}$ and the uniform distribution. The converse reduction needs more work. However, due to the strong analogy between function and number fields, our proof is in fact essentially the same as in [LPR10; Lyu11]. More precisely, we have the following theorem.

Theorem 4.30 (Search to decision reduction for FF-DP)

Let $K/\mathbb{F}_q(T)$ be a function field with *field of constants* $\mathbb{F}_q$, and denote by $\mathcal{O}_K$ its ring of integers. Let $\mathfrak{p} \subset \mathbb{F}_q[T]$ be a prime ideal generated by an irreducible modulus $Q(T)$, and set

$$\mathfrak{P} \stackrel{\text{def}}{=} Q\mathcal{O}_K.$$

Denote by $f \stackrel{\text{def}}{=} f(\mathfrak{P}/\mathfrak{p})$ its inertia degree and by $e \stackrel{\text{def}}{=} e(\mathfrak{P}/\mathfrak{p})$ its ramification index. Let ψ be a probability distribution over $\mathcal{O}_K/\mathfrak{P}$. Let $\mathbf{s} \in \mathcal{O}_K/\mathfrak{P}$.
Assume that

1. $K/\mathbb{F}_q(T)$ is a *Galois* algebraic extension of degree n;
2. $\mathfrak{p}$ *does not ramify, i.e.* $e = 1$;
3. ψ is *closed under the action of* $\mathrm{Gal}(K/\mathbb{F}_q(T))$, *i.e.* if $\mathbf{e} \leftarrow \psi$, then for any $\sigma \in \mathrm{Gal}(K/\mathbb{F}_q(T))$, it holds that $\sigma(\mathbf{e})$ is still distributed according to ψ.

Suppose that we have oracle access to $\mathcal{F}_{\mathbf{s},\psi}$ and there exists a distinguisher between the uniform distribution over $\mathcal{O}_K/\mathfrak{P}$ and the FF-DP distribution with uniform secret and error distribution ψ, running in time t and having an advantage ε.
Then there exists an algorithm that recovers $\mathbf{s} \in \mathcal{O}_K/\mathfrak{P}$ (with an overwhelming probability in n) in time

$$O\left(\frac{n^4}{f^3} \times \frac{1}{\varepsilon^2} \times q^{f\deg(Q)} \times t\right).$$

Remark 4.31. *In the theorem, we ask that the irreducible modulus Q do not ramify in $\mathcal{O}_K$. In reality, this is not really restrictive since ramification happens only for finitely many irreducible*

polynomials in $\mathbb{F}_q[T]$ *(see for example [Lor21, Proposition 5.2]).*

Remark 4.32. *We have assumed implicitly in the statement of the theorem that we have an efficient access to the Galois group of* $K/\mathbb{F}_q(T)$ *and its action can be computed in polynomial time, which is a reasonable assumption to make in practice.*

Remark 4.33. *There are many degrees of freedom in the previous statement: choice of the function field* K *(and on the degree* n*), choice of the irreducible polynomial* Q *(and on* f *as well as* $\deg(Q)$*). For our instantiations, we will often choose the "modulus"* Q *to be a linear polynomial (*$\deg(Q) = 1$*) and* K *to be an abelian extension, i.e.* K *will be a subfield of a cyclotomic function field.*

Remark 4.34. *Due to the continuity of error distributions used in lattice-based cryptography, a technical tool called the* smoothing parameter *was introduced by Micciancio and Regev in [MR04]. It characterises how a Gaussian distribution is close to uniform, both modulo the lattice, and is ubiquitously used in reductions. However, in the function field setting, we do not need to introduce such a tool because the error distribution is discrete and already defined on the quotient* $\mathcal{O}_K/\mathfrak{P}$*.*

Remark 4.35. *In [LS15, Section 4.3], Langlois and Stehlé proved a search to decision reduction for the module version of LWE. The idea is to use the distinguisher in order to retrieve the secrets one by one. Their proof applies* mutatis mutandis *to MFF-DP (Problems 4.28 and 4.29), resulting in a time overhead of* d*, where* d *denotes the rank of the underlying module,* i.e. *the number of secrets. The main change is in the* guess and search *step (Step 3 in the proof presented in Section 4.3.3) where the randomisation is applied on only one component of* $\mathbf{a}$ *to recover one secret, and repeating the process* d *times (one for each secret). More precisely, for MFF-DP, the running time claimed in Theorem 4.30 should be replaced with*

$$O\left(d \times \frac{n^4}{f^3} \times \frac{1}{\varepsilon^2} \times q^{f\deg(Q)} \times t\right).$$

4.3.3 Search to Decision Reductions: Proof of Theorem 4.30

The proof of Theorem 4.30 will be similar to the one given by Lyubashevsky, Peikert and Regev in [LPR10] for Ring-LWE and lattices. Let $\mathcal{A}$ be an algorithm running in time t which is able to distinguish with advantage ε between the uniform distribution over $\mathcal{O}_K/\mathfrak{P}$ and the FF-DP distribution $\mathcal{F}_{\mathbf{s},\psi}$ with uniform secret $\mathbf{s}$ and error distribution ψ. Let $r \stackrel{\text{def}}{=} \frac{n}{f}$ and consider an ordering $(\mathfrak{P}_i)_{1\leqslant i\leqslant r}$ of the prime ideals above $\mathfrak{p}$. Let

$$\mathfrak{P} \stackrel{\text{def}}{=} \mathfrak{P}_1 \cdots \mathfrak{P}_r$$

be the factorisation of $\mathfrak{P}$ in $\mathcal{O}_K$. Since $\mathfrak{p}$ does not ramify, the Chinese Remainder Theorem (CRT) ensures that the ring

$$\mathcal{O}_K/\mathfrak{P} \stackrel{\text{def}}{=} \prod_{i=1}^{r} \mathcal{O}_K/\mathfrak{P}_i$$

is actually a product of *finite fields*. In particular, none of the factors have zero divisors.

Our goal is to design a procedure to recover the secret $\mathbf{s}$ with the help of our distinguisher $\mathscr{A}$. At a high level, it consists in reducing the search domain to a single factor $\mathcal{O}_K/\mathfrak{P}_{i_0}$, before doing an *exhaustive search* in order to recover the secret $\mathbf{s} \mod \mathfrak{P}_i$. The transitive action of $\mathsf{Gal}(K/\mathbb{F}_q(T))$ on the factors allows to recover the other parts of the secret. This uses crucially

that the error distribution is invariant under this action. More formally, the proof uses four steps.

Step 1: Worst to Average Case. Recall that in the definition of Problem 4.24 the secret $\mathbf{s}$ is supposed to be *uniformly* distributed over $\mathcal{O}_K/\mathfrak{p}$, while in the search version the secret is *fixed*. In other words, the decision problem is somehow an *average* case problem, while the search version should work in *any* case. Fortunately, this can easily be addressed by randomising the secret. Indeed, for any sample $(\mathbf{a}, \mathbf{b}) \leftarrow \mathcal{F}_{\mathbf{s},\psi}$ with fixed secret $\mathbf{s}$, if $\mathbf{s}' \leftarrow \mathcal{O}_K/\mathfrak{p}$, then $(\mathbf{a}, \mathbf{b} + \mathbf{a}\mathbf{s}')$ is now a sample from $\mathcal{F}_{\mathbf{s}+\mathbf{s}',\psi}$ with secret $\mathbf{s}+\mathbf{s}'$ which is uniformly distributed over $\mathcal{O}_K/\mathfrak{p}$.

Step 2: Hybrid argument. The goal of this step is to prove that $\mathscr{A}$ can not only distinguish between the uniform and $\mathcal{F}_{\mathbf{s},\psi}$, but also between two somewhat *closer* distributions, at the cost of reducing the advantage by a factor $r = \frac{n}{f}$. For this purpose, let us introduce some intermediate distributions over $\mathcal{O}_K/\mathfrak{p} \times \mathcal{O}_K/\mathfrak{p}$ which we call $\mathscr{H}_i$, for $i \in \{0, \dots, r\}$.

Definition 4.36 (Hybrid Distributions)

For $i \in \{0, \dots, r\}$, a sample $(\mathbf{a}, \mathbf{b})$ is said to be distributed according to the hybrid distribution $\mathscr{H}_i$ if it is of the form $(\mathbf{a}', \mathbf{b}' + \mathbf{h})$ where:

- $(\mathbf{a}', \mathbf{b}') \leftarrow \mathcal{F}_{\mathbf{s},\psi}$ is distributed according to the usual FF-DP distribution with secret $\mathbf{s}$ and noise ψ;
- $\mathbf{h} \in \mathcal{O}_K/\mathfrak{p}$ is
  - uniformly distributed modulo $\mathfrak{p}_j$ for $j \leqslant i$
  - 0 modulo the other factors.

In other words, through the isomorphism

$$\mathcal{O}_K/\mathfrak{p} \simeq \mathcal{O}_K/\mathfrak{p}_1 \times \mathcal{O}_K/\mathfrak{p}_r$$

such an element h is of the form

$$h \stackrel{\text{def}}{=} (r_1, \dots, r_i, 0, \dots, 0)$$

where r_i is uniformly distributed in $\mathcal{O}_K/\mathfrak{p}_i$. It can easily be constructed using the Chinese Remainder Theorem.

In particular, for $i = 0$, it holds that $\mathbf{h} = 0$ and $\mathscr{H}_0 = \mathcal{F}_{\mathbf{s},\psi}$. On the other hand, when $i = r$, the element $\mathbf{h}$ is uniformly distributed over $\mathcal{O}_K/\mathfrak{p}$, therefore $\mathscr{H}_r$ is *exactly* the uniform distribution over $\mathcal{O}_K/\mathfrak{p}$.

Remark 4.37. *For the reader familiar with the reduction of [LPR10] for Ring-LWE, there is a difference between the code-based and the lattice-based settings. Indeed, in the latter it is necessary to introduce a technical tool, namely the* smoothing parameter *in order to cope with*

the discretisation of the Gaussian which is a priori *defined over the torus*[ii]

$$\mathbb{T} \stackrel{def}{=} (K \otimes \mathbb{R}) \big/ \mathcal{O}_K.$$

In particular, the standard deviation of the Gaussian needs to be large enough for the reduction to work, see [LPR10, Lemma 5.13]. On the other hand, in our setting everything is discrete, and such a tool is not needed. In other words, the code-based situation is somehow nicer *than the lattice-based.*

The following lemma proves that $\mathscr{A}$ can distinguish between two consecutive hybrid distributions, though with a slightly smaller advantage.

Lemma 4.38 (Hybrid argument)

There exists i_0 such that $\mathrm{Adv}_{\mathscr{A}}(\mathscr{H}_{i_0}, \mathscr{H}_{i_0+1}) \geqslant \frac{\varepsilon}{r}$.

Proof. By definition, $\mathrm{Adv}_{\mathscr{A}}(\mathscr{H}_0, \mathscr{H}_r) = \varepsilon$. Furthermore, the following equality holds:

$$\mathrm{Adv}_{\mathscr{A}}(\mathscr{H}_0, \mathscr{H}_r) = \sum_{i=0}^{r-1} \mathrm{Adv}_{\mathscr{A}}(\mathscr{H}_i, \mathscr{H}_{i+1}).$$

Therefore, it exists $i_0 \in \{0, \ldots, r-1\}$ such that $\mathrm{Adv}_{\mathscr{A}}(\mathscr{H}_{i_0}, \mathscr{H}_{i_0+1}) \geqslant \frac{\mathrm{Adv}_{\mathscr{A}}(\mathscr{H}_0, \mathscr{H}_r)}{r} = \frac{\varepsilon}{r}$. □

Remark 4.39. *This hybrid argument has shown the existence of an i_0 such that $\mathscr{A}$ has an advantage ε/r for distinguishing distributions $\mathscr{H}_{i_0}$ and $\mathscr{H}_{i_0+1}$. In what follows, everything is analysed as if we knew this index i_0. In practice we can run $\mathscr{A}$ concurrently with all the r instances $(\mathscr{H}_i, \mathscr{H}_{i+1})$'s. Computations on the right index i_0 will output the secret $\mathbf{s}$ (which can be verified) as it will be explained afterwards. Therefore, our reduction will output $\mathbf{s}$ with a "resource overhead" given by at most a factor r. This technique is known as* Dovetailing.

Step 3: Guess and search. Let i_0 be such as in Lemma 4.38. The idea is to perform an exhaustive search in $\mathcal{O}_K \big/ \mathfrak{p}_{i_0+1}$ and to use $\mathscr{A}$ to recover $\mathbf{s} \mod \mathfrak{p}_{i_0+1}$.

Lemma 4.40

Let $\mathscr{A}$ be a distinguisher with advantage δ between hybrid distributions $\mathscr{H}_{i_0}$ and $\mathscr{H}_{i_0+1}$, with secret $\mathbf{s}$, running in time t. Then there exists an algorithm $\mathscr{B}$ that recovers $\mathbf{s} \mod \mathfrak{p}_{i_0+1}$ with overwhelming probability in n in time $O\left(q^{f \deg(Q)} \times \frac{n}{\delta^2} \times t\right)$.

[ii] More precisely, in the lattice-based setting, the secret is defined in a larger ring than $\mathcal{O}_K$, namely its dual $\mathcal{O}_K^\vee$, and the torus is defined modulo this dual ring.

Proof. Our algorithm will proceed with a *guess and search* technique using the distinguisher $\mathscr{A}$ in hand. The idea is to *guess* the value of $\mathbf{s} \mod \mathfrak{P}_{i_0+1}$ and transform any sample $(\mathbf{a}, \mathbf{b}) \leftarrow \mathcal{F}_{\mathbf{s},\psi}$ into a sample of $\mathscr{H}_{i_0}$ if the guess is correct, and into a sample of $\mathscr{H}_{i_0+1}$ if the guess is wrong.

Transformation: Let

$$\widehat{\mathbf{s}} \stackrel{\text{def}}{=} \mathbf{s} \mod \mathfrak{P}_{i_0+1},$$

and let $\mathbf{g}_{i_0+1} \in \mathcal{O}_K/\mathfrak{P}_{i_0+1}$ be our guess value for $\widehat{\mathbf{s}}$. Let us consider now the following operations

- Let $\mathbf{g} \in \mathcal{O}_K/\mathfrak{P}$ be such that

$$\mathbf{g} \equiv \begin{cases} \mathbf{g}_{i_0+1} & \mod \mathfrak{P}_{i_0+1} \\ 0 & \mod \mathfrak{P}_j \quad \text{for } j \neq i_0+1 \end{cases}$$

- For $1 \leqslant j \leqslant i_0$, sample $\mathbf{h}_j \leftarrow \mathcal{O}_K/\mathfrak{P}_j$ and let $\mathbf{h} \in \mathcal{O}_K/\mathfrak{P}$ be such that

$$\mathbf{h} \equiv \begin{cases} \mathbf{h}_j & \mod \mathfrak{P}_j \quad \text{for } 1 \leqslant j \leqslant i_0 \\ 0 & \mod \mathfrak{P}_j \quad \text{for } j \geqslant i_0+1 \end{cases}$$

- Sample $\mathbf{v}_{i_0+1} \leftarrow \mathcal{O}_K/\mathfrak{P}_{i_0+1}$ and let $\mathbf{v} \in \mathcal{O}_K/\mathfrak{P}$ be such that

$$\mathbf{v} \equiv \begin{cases} \mathbf{v}_{i_0+1} & \mod \mathfrak{P}_{i_0+1} \\ 0 & \mod \mathfrak{P}_j \quad \text{for } j \neq i_0+1 \end{cases}$$

All those operations can be done via the CRT. Now, for each sample $(\mathbf{a}, \mathbf{b} \stackrel{\text{def}}{=} \mathbf{as}+\mathbf{e}) \leftarrow \mathcal{F}_{\mathbf{s},\psi}$, set $(\mathbf{a}', \mathbf{b}')$ with

$$\mathbf{a}' \stackrel{\text{def}}{=} \mathbf{a}+\mathbf{v} \quad \text{and} \quad \mathbf{b}' \stackrel{\text{def}}{=} \mathbf{b}+\mathbf{h}+\mathbf{vg}.$$

Note that for each sample $(\mathbf{a}, \mathbf{b})$, the corresponding $\mathbf{a}'$ is still uniformly distributed over $\mathcal{O}_K/\mathfrak{P}$ and $\mathbf{b}' = \mathbf{a}'\mathbf{s}+\mathbf{e}+\mathbf{h}'$ with $\mathbf{h}' \stackrel{\text{def}}{=} \mathbf{h}+(\mathbf{g}-\mathbf{s})\mathbf{v}$. Furthermore, it holds that

$$\begin{cases} \mathbf{h}' \equiv \mathbf{h}_j & \mod \mathfrak{P}_j \quad \text{for } j \leqslant i_0 \\ \mathbf{h}' \equiv (\mathbf{g}_{i_0+1}-\widehat{\mathbf{s}})\mathbf{v}_{i_0+1} & \mod \mathfrak{P}_{i_0+1} \\ \mathbf{h}' \equiv 0 & \mod \mathfrak{P}_j \quad \text{for } j > i_0+1. \end{cases}$$

In particular, $\mathbf{h}'$ is uniformly distributed modulo $\mathfrak{P}_j$ for $j \leqslant i_0$ and 0 modulo $\mathfrak{P}_j$ for $j > i_0+1$. Now, if the guess is correct, meaning that $\mathbf{g}_{i_0+1} = \widehat{\mathbf{s}}$, then $\mathbf{h}' \equiv 0 \mod \mathfrak{P}_{i_0+1}$, hence $(\mathbf{a}', \mathbf{b}')$ is distributed according to $\mathscr{H}_{i_0}$. On the other hand, if the guess is incorrect, then $(\mathbf{g}_{i_0+1}-\widehat{\mathbf{s}}) \neq 0$ in $\mathcal{O}_K/\mathfrak{P}_{i_0+1}$, which is a *field* by hypothesis. Therefore, since $\mathbf{v}_{i_0+1}$ is uniformly distributed in $\mathcal{O}_K/\mathfrak{P}_{i_0+1}$, so is $(\mathbf{g}_{i_0+1}-\widehat{\mathbf{s}})\mathbf{v}_{i_0+1}$. In particular, $\mathbf{h}'$ is also uniformly distributed modulo $\mathfrak{P}_{i_0+1}$. Hence, $(\mathbf{a}', \mathbf{b}')$ is distributed according to $\mathscr{H}_{i_0+1}$.

We can now define the algorithm $\mathscr{B}$. It proceeds as follows: for each $(\mathbf{a}, \mathbf{b}) \leftarrow \mathcal{F}_{\mathbf{s},\psi}$, it

applies the previous transformation to get a sample $(\mathbf{a}', \mathbf{b}')$, and then uses the distinguisher $\mathscr{A}$. Repeating the procedure m times (for each guess $\mathbf{g}_{i_0+1}$), for m large enough, and doing a majority voting allows to recover $\widehat{\mathbf{s}}$ with overwhelming probability (See Section 1.2.1).

More precisely, by Proposition 1.24, if we run $\mathscr{A}$ with $m \geqslant \ln(\frac{1}{\mu})\frac{1}{2\delta^2}$ samples, then the majority voting strategy allows us to tell whether they are distributed according to $\mathscr{H}_{i_0}$ or $\mathscr{H}_{i_0+1}$ with probability at least $1-\mu$. In other words, by setting $\mu = 2^{-\Theta(n)}$, it is enough to choose m as $\Theta\left(\frac{n}{\delta^2}\right)$ to be able to decide if our guess $\widehat{\mathbf{s}} = \mathbf{s} \mod \mathfrak{P}_{i_0+1}$ is correct or not with probability at least $1 - 2^{-\Theta(n)}$.

Finally, for recovering $\mathbf{s} \mod \mathfrak{P}_{i_0+1}$, it suffices to try all the possible guesses $\widehat{\mathbf{s}} \in \mathcal{O}_K/\mathfrak{P}_{i_0+1}$. Since the size of $\mathcal{O}_K/\mathfrak{P}_{i_0+1}$ is given by $q^{f \deg(Q)}$, this yields the claimed time complexity. □

Step 4: Action of the Galois group. Until Step 3, we are able to recover the secret $\mathbf{s}$ modulo one of the factors. In order to recover the full secret, we use the Galois group $G \stackrel{\text{def}}{=} \mathrm{Gal}(K/\mathbb{F}_q(T))$. This last part is *crucial* for the reduction to work. Recall that G acts transitively on the set of prime ideals above $\mathfrak{p}$, *i.e.* for every $i \neq j$, there exists $\sigma \in G$ such that $\sigma(\mathfrak{P}_i) = \mathfrak{P}_j$.

Lemma 4.41

Fix $\mathbf{s} \in \mathcal{O}_K/\mathfrak{P}$. Let $1 \leqslant i \leqslant r$ and let $\mathscr{A}$ be an algorithm running in time t, and recovering $\mathbf{s} \mod \mathfrak{P}_i$ by making queries to an oracle for $\mathcal{F}_{\mathbf{s},\psi}$.
Then there exists an algorithm $\mathscr{B}$ running in time $O(t \times r)$ that recovers the full secret $\mathbf{s}$.

Proof. We build $\mathscr{B}$ as follows: for every factor $\mathfrak{P}_j$ of $\mathfrak{P}$, it chooses $\sigma \in \mathrm{Gal}(K/\mathbb{F}_q(T))$ such that $\sigma(\mathfrak{P}_j) = \mathfrak{P}_i$. Then, for each sample $(\mathbf{a}, \mathbf{b}) \leftarrow \mathcal{F}_{\mathbf{s},\psi}$, it runs $\mathscr{A}$ on the input $(\sigma(\mathbf{a}), \sigma(\mathbf{b}))$ to recover an element $\mathbf{s}_j$ and stores $\sigma^{-1}(\mathbf{s}_j)$.

Note that $\mathrm{Gal}(K/\mathbb{F}_q(T))$ keeps the uniform distribution over $\mathcal{O}_K/\mathfrak{P}$. In particular, for every sample $(\mathbf{a}, \mathbf{b}) \leftarrow \mathcal{F}_{\mathbf{s},\psi}$, the corresponding $\sigma(\mathbf{a})$ is also uniformly distributed over $\mathcal{O}_K/\mathfrak{P}$. Furthermore, $\mathbf{b} = \mathbf{as} + \mathbf{e}$ with $\mathbf{e} \leftarrow \psi$. Therefore, $\sigma(\mathbf{b}) = \sigma(\mathbf{a})\sigma(\mathbf{s}) + \sigma(\mathbf{e})$. But ψ is Galois invariant by assumption, and hence $\sigma(\mathbf{e})$ is also distributed according to ψ. In particular, $(\sigma(\mathbf{a}), \sigma(\mathbf{b}))$ is a valid sample of $\mathcal{F}_{\sigma(\mathbf{s}),\psi}$.

Now, our algorithm $\mathscr{A}$ is able to recover $\mathbf{s}_j \stackrel{\text{def}}{=} \sigma(\mathbf{s}) \mod \mathfrak{P}_i$ in time t, and

$$\sigma^{-1}(\mathbf{s}_j) = \sigma^{-1}(\sigma(\mathbf{s}) \mod \mathfrak{P}_i) = \mathbf{s} \mod \sigma^{-1}(\mathfrak{P}_i) = \mathbf{s} \mod \mathfrak{P}_j.$$

Therefore, we are able to recover $\mathbf{s} \mod \mathfrak{P}_j$ for any $1 \leqslant j \leqslant r$. To compute the full secret $\mathbf{s}$ it remains to use the Chinese Remainder Theorem. The running time of this full procedure is given by a $O(t \times r)$ which concludes the proof. □

Combining the steps all together gives the full reduction with the claimed complexity. Indeed, at the end of Step 2, we know that $\mathscr{A}$ distinguishes between $\mathscr{H}_{i_0}$ and $\mathscr{H}_{i_0+1}$ with advantage

$$\delta \stackrel{\text{def}}{=} \frac{\varepsilon}{r} = \varepsilon \times \frac{f}{n}.$$

Then, Lemma 4.40 yields an algorithm $\mathscr{B}$ recovering $\mathbf{s} \mod \mathfrak{P}_{i_0+1}$ in time

$$O\left(q^{f\deg(Q)} \times \frac{n}{\delta^2} \times t\right) = O\left(\frac{n^3}{f^2} \times q^{f\deg(Q)} \times \frac{1}{\varepsilon^2} \times t\right).$$

Finally, Step 4 only adds a factor $r = \frac{n}{f}$ overhead by Lemma 4.41, which gives the final claimed complexity

$$O\left(\frac{n^4}{f^3} \times q^{f\deg(Q)} \times \frac{1}{\varepsilon^2} \times t\right).$$

□

4.4 Instantiations

In this last section, we discuss instantiations of the FF-DP problem, of Theorem 4.30, and their implications for cryptography. Another major application will be presented in Part III of this manuscript.

4.4.1 Decoding of Quasi-Cyclic Codes

Let n be an integer, coprime to q. Recall from the introductory Examples 4.5 and 4.23 that the cyclic ring

$$\mathbb{F}_q[X]/(X^n - 1) = \mathcal{O}_K/T\mathcal{O}_K$$

where

$$K \stackrel{\text{def}}{=} \mathbb{F}_q(T)[X]/X^n + T - 1.$$

In other words, $\mathbb{F}_q[X]/(X^n - 1)$ can be represented as the quotient of a Dedekind domain of Krull dimension 1, which allows to interpret the decoding problem of quasi-cyclic codes, with ℓ blocks, as the FF-DP problem with function field K and irreducible modulus $T \in \mathbb{F}_q[T]$. For the error distribution, the most usual one which is the uniform distribution over the regular words of weight ℓt, *i.e.* in the language of FF-DP, ψ is the uniform distribution over the polynomials in $\mathbb{F}_q[X]/(X^n - 1)$ of Hamming weight t. This is the error distribution used in BIKE and HQC candidates in the 4th round of the on-ramp NIST competition for encryption schemes. This is not the only possible choice, and many other distributions could be considered. For example, one may consider a Bernoulli distribution over $\mathbb{F}_q[X]/(X^n - 1)$[iii].

It remains to check if the hypothesis of Theorem 4.30 are satisfied. First, notice that T does not ramify in $\mathcal{O}_K$. However, the extension $K/\mathbb{F}_q(T)$ is *not Galois* in general. Such extensions are studied through the theory of *Kummer extensions*, and they will be Galois when $\mathbb{F}_q(T)$ contains n distinct roots of 1. Since they are all algebraic over $\mathbb{F}_q$, this means that $\mathbb{F}_q$ should contain all the n-th roots of unity, *i.e.* when $n|q-1$, which seems fairly restrictive.

One may then wonder if there exists another extension $K'/\mathbb{F}_q(T)$ which could work. However, the following simple proposition dashes any hope.

[iii] An element $P \in \mathbb{F}_q[X]/(X^n - 1)$ is distributed according to a Bernoulli distribution with parameter p if its coefficients are independently distributed according to a q-ary Bernoulli distribution with parameter p.

Proposition 4.42

Let n be coprime to q. There exists a Galois extension $K/\mathbb{F}_q(T)$ and an irreducible polynomial $Q \in \mathbb{F}_q[T]$ such that

$$\mathbb{F}_q[X]/(X^n-1) \simeq \mathcal{O}_K/Q\mathcal{O}_K$$

if and only if $\mathbb{F}_q$ contains all n-th roots of unity.

Proof.

- When $\mathbb{F}_q$ contains all n-th roots of unity, the previous discussion shows that

  $$K \stackrel{\text{def}}{=} \mathbb{F}_q(T)[X]/X^n+T-1,$$

  is a suitable choice, together with $Q=T$.

- Conversely, let K be such a putative Galois extension and let $\mathfrak{p} \stackrel{\text{def}}{=} Q\mathcal{O}_K$. The idea is to look at the factorisation of $\mathfrak{p}$ into prime ideals of $\mathcal{O}_K$. Indeed, if K were Galois, then all the inertia degrees and ramification indexes would be equal, and we would have

  $$\mathfrak{p} = (\mathfrak{p}_1 \cdots \mathfrak{p}_r)^e$$

  for some integer e. This implies that

  $$\mathbb{F}_q[X]/(X^n-1) = \mathcal{O}_K/Q\mathcal{O}_K \simeq \mathcal{O}_K/\mathfrak{p}_1^e \times \cdots \times \mathcal{O}_K/\mathfrak{p}_r^e.$$

  Note that e cannot be greater than 1 since $\mathbb{F}_q[X]/(X^n-1)$ does not contain any nilpotent element, and the above isomorphism is a decomposition of $\mathbb{F}_q[X]/(X^n-1)$ as a product of fields.

  Now, let

  $$X^n-1 = (X-1)P_2(X)\cdots P_s(X)$$

  be the factorisation of X^n-1 into irreducible polynomials of $\mathbb{F}_q[X]$. The Chinese Remainder Theorem entails that

  $$\mathbb{F}_q[X]/(X^n-1) \simeq \mathbb{F}_q[X]/X-1 \times \mathbb{F}_q[X]/P_2(X) \times \ldots \mathbb{F}_q[X]/P_s(X),$$

  which is another decomposition of $\mathbb{F}_q[X]/(X^n-1)$ into product of fields. Artin-Wedderburn theorem (see [DK12, Theorem 2.5.1]) ensures that those two decompositions are actually equal, up to permutation of the factors. In other words, without loss of generality, we may assume that

  $$\mathcal{O}_K/\mathfrak{p}_i = \mathbb{F}_q[X]/P_i(X) = \mathbb{F}_{q^{\deg(P_i)}},$$

where we set $P_1(X) \stackrel{\text{def}}{=} X - 1$. On the other hand, let $f_i \stackrel{\text{def}}{=} f(\mathfrak{P}_i/Q)$ be the inertia degree of Q at $\mathfrak{P}_i$. By definition, we have that

$$\mathcal{O}_K \big/ \mathfrak{P}_i = \mathbb{F}_{q^{f_i \deg(Q)}}.$$

In particular, this equality for $i = 1$ implies that $\deg(Q) = 1$. Moreover, if the extension $K/\mathbb{F}_q(T)$ is Galois, then all the f_i's are equal to some integer f and the finite fields $\mathbb{F}_{q^{\deg(P_i)}}$ must all be equal to $\mathbb{F}_{q^f}$. In particular, the P_i are all of the same degree f. Since $(X-1)$ is one of them, this means that f must be equal to 1. In other words, $\mathbb{F}_q[X] \big/ (X^n - 1)$ can only be realised as the quotient of the ring of integers $\mathcal{O}_K$ of some Galois extension $K/\mathbb{F}_q(T)$ if $X^n - 1$ splits into linear factors, *i.e.* if $\mathbb{F}_q$ contains all n-th roots of unity.

□

However, this proposition does not mean that Theorem 4.30 is meaningless. Indeed, consider one example which should work, and set $n \stackrel{\text{def}}{=} q - 1$. Indeed, recall by Example 4.18 that

$$\mathbb{F}_q[X] \big/ X^{q-1} - 1 \simeq \mathcal{O}_K \big/ (T+1)\mathcal{O}_K$$

where

$$K \stackrel{\text{def}}{=} \mathbb{F}_q(T)[\Lambda_T] = \mathbb{F}_q(T)[X] \big/ X^{q-1} + T$$

is the Carlitz extension with respect to T.

Remark 4.43. *One could also consider Example 4.23 with $n = q - 1$, by noticing that*

$$\mathbb{F}_q(T)[X] \big/ X^{q-1} + T - 1$$

is nothing but the Carlitz extension with respect to the polynomial $T - 1$.

Therefore, instantiating the FF-DP problem with this function field K, irreducible modulus $Q(T) \stackrel{\text{def}}{=} T + 1$ and prime ideal $\mathfrak{P} \stackrel{\text{def}}{=} (T+1)\mathcal{O}_K$, Theorem 4.30 immediately yields a search-to-decision reduction when the error distribution ψ is invariant under the action of $\mathsf{Gal}(K/\mathbb{F}_q(T))$. But recall from Theorem 4.14 and Example 4.18 that

$$\mathsf{Gal}(K/\mathbb{F}_q(T)) \stackrel{\text{def}}{=} \left(\mathbb{F}_q[T] \big/ (T)\right)^\times = \mathbb{F}_q^\times,$$

and an element $b \in \mathbb{F}_q^\times$ acts on $f(T,X) \in K$ by

$$b \cdot f(T,X) = f(T,[b](X)) = f(T,bX).$$

This action induces an action of $\mathbb{F}_q^\times$ on the quotient

$$\mathcal{O}_K \big/ (T+1) \simeq \mathbb{F}_q[X] \big/ X^{q-1} - 1$$

by

$$b \cdot m(X) \stackrel{\text{def}}{=} m(bX).$$

The remarkable point is that this action *does not* change the Hamming support of m, and in

particular it does not affect its Hamming weight. Therefore, whenever ψ only depends on the support of the error (*e.g.* the uniform on all polynomials of weight t, or a Bernoulli), then it is Galois invariant. Theorem 4.30 translates in this setting into the following theorem:

Theorem 4.44

Let ψ be a probability distribution over

$$\mathcal{R} \stackrel{\text{def}}{=} \mathbb{F}_q[X] \Big/ X^{q-1} - 1$$

invariant by the action of $\mathbb{F}_q^\times$ (*e.g.* the uniform distribution over polynomials of fixed Hamming weight, or the q–ary Bernoulli). Let $\mathbf{m} \leftarrow \mathcal{R}$ be drawn uniformly at random and consider the following two distributions

- $\mathcal{D}_0 : (\mathbf{a}, \mathbf{y}^{\text{unif}})$ uniformly distributed over $\mathcal{R}^2$,
- $\mathcal{D}_1 : (\mathbf{a}, \mathbf{a} \cdot \mathbf{m} + \mathbf{e})$ where $\mathbf{a} \leftarrow \mathcal{R}$ and $\mathbf{e} \leftarrow \psi$

Assume that there exists an algorithm which distinguish between $\mathcal{D}_0$ and $\mathcal{D}_1$ in time t with advantage ε.

Then, there exists an algorithm which recovers $\mathbf{m}$ with overwhelming probability (in q) in time

$$O\left(q^5 \times \frac{1}{\varepsilon^2} \times t\right).$$

Remark 4.45. *With this reduction in hand, one may wonder how meaningful it is. Indeed, this would not say anything if the search version was easy. In fact, it is still an open problem in coding theory to decode random quasi-cyclic codes, even with this block length $q-1$. In fact, quasi-cyclic codes are a particular case of so-called* quasi-group codes*, which will be introduced in Chapter 7 (Definition 7.7). Finding efficient decoding algorithms for those algebraically structured codes is still an open question after more than 50 years of research, and is even listed as one of the main open questions in algebraic coding theory in the recent Encyclopedia of Coding Theory [Wil21, Problem 16.10.5]. This fact will motivate the introduction of the more general Quasi-Abelian Decoding Problem in Chapter 7 (Problem 7.20).*

Remark 4.46. *The search to decision reduction given via FF-DP can actually be understood directly with the ring*

$$\mathbb{F}_q[X] \Big/ X^{q-1} - 1 \simeq \prod_{a \in \mathbb{F}_q^\times} \mathbb{F}_q[X] \Big/ (X - a).$$

Indeed, it is clear that the action $b \cdot m(X) \stackrel{def}{=} m(bX)$ actually maps the factor $\mathbb{F}_q[X] \Big/ X - a$ onto $\mathbb{F}_q[X] \Big/ X - b^{-1}a$. However, without the Carlitz interpretation this action seems completely unnatural.

Note that the proof of this theorem actually necessitates quite a lot of samples. In other words, the code which we decode has a rate which tends to 0, *i.e.* this theorem is more in the LPN regime. This motivates the introduction of structured variants of the LPN problem.

4.4.2 The Ring-LPN problem

Similarly to the LWE problem, structured variants of LPN have been defined ([HKLP+12; DP12]), and they are related to the decoding problem of structured codes. Some instantiations have been proven to be insecure ([BL12]). Nonetheless, such structured variants have regained interest recently due to their applications to secure multiparty computation [BCGI+20b]. In particular, in Chapter 7, we investigate a large family of instantiations, based on the decoding of codes over group algebras.

Definition 4.47 (Ring–LPN distribution)

Fix a positive integer r, a public polynomial $P(X) \in \mathbb{F}_q[X]$ of degree r and let $\mathbf{s} \in \mathcal{R} \stackrel{\text{def}}{=} \mathbb{F}_q[X]/(P(X))$ be a uniformly distributed polynomial. Let ψ be a probability distribution over the ring $\mathcal{R}$. A sample $(\mathbf{a}, \mathbf{b})$ is distributed according to the Ring-LPN distribution with secret $\mathbf{s}$ if

- $\mathbf{a} \leftarrow \mathcal{R}$ is drawn uniformly at random;
- $\mathbf{b} \stackrel{\text{def}}{=} \mathbf{as} + \mathbf{e}$ where $\mathbf{e} \leftarrow \psi$

A sample drawn according to this distribution will be denoted $(\mathbf{a}, \mathbf{as}+\mathbf{e}) \leftarrow \mathscr{D}^{\mathsf{RLPN}}_{\mathbf{s},\psi}$.

When $P(X) \stackrel{\text{def}}{=} X^r - 1$ and ψ is the uniform distribution over polynomials of Hamming weight t, this is exactly the decisional QC-DP distribution. In the literature, other noise distributions ψ have been considered. The most common other one is the Bernoulli distribution with respect to an $\mathbb{F}_q$ basis of $\mathcal{R}$, *e.g.* the canonical monomial basis $(X^i)_{0 \leqslant i \leqslant r-1}$. Other basis are discussed below.

This distribution leads to a search and a decisional version of the so-called Ring-LPN problem

Problem 4.48 (Search Ring-LPN)

Let $P \in \mathbb{F}_q[X]$ be a degree r polynomial, and let $\mathcal{R} \stackrel{\text{def}}{=} \mathbb{F}_q[X]/(P(X))$. Let ψ be a probability distribution over $\mathcal{R}$ and let $\mathbf{y}^{(1)}, \ldots, \mathbf{y}^{(\ell)}$ be of the form

$$\mathbf{y}_i \stackrel{\text{def}}{=} \mathbf{a}^{(i)}\mathbf{m} + \mathbf{e}^{(i)} \in \mathcal{R}$$

for some *fixed* $\mathbf{m} \in \mathcal{R}$, where $\mathbf{a}^{(i)} \leftarrow \mathcal{R}$ is uniformly distributed, and $\mathbf{e}^{(i)} \leftarrow \psi$.

Given the ℓ samples $(\mathbf{a}^{(i)}, \mathbf{y}^{(i)})$, the goal is to recover $\mathbf{m}$.

Problem 4.49 (Decisional Ring-LPN)

Let $\mathcal{R}$ and ψ be defined as in the previous definition. Let $\mathbf{m}$ be drawn uniformly at random in $\mathcal{R}$ and consider the following two distributions

- $\mathcal{D}_0 : (\mathbf{a}, \mathbf{y}^{\text{unif}})$ uniformly distributed over $\mathcal{R}^2$,
- $\mathcal{D}_1 : (\mathbf{a}, \mathbf{a}\cdot\mathbf{m}+\mathbf{e})$ where $\mathbf{a} \leftarrow \mathcal{R}$, and $\mathbf{e} \leftarrow \psi$, *i.e.* the $\mathscr{D}^{\mathsf{RLPN}}_{\mathbf{m},\psi}$ distribution.

Given oracle access to distribution $\mathcal{D}_b$ where $b \leftarrow \{0,1\}$, the goal is to recover b.

Remark 4.50. *Obviously, the problem of distinguishing samples from $\mathscr{D}^{\mathsf{RLPN}}_{\mathbf{m},\psi}$ should be related to the decisional version of the decoding problem of structured codes whose basis is block-wise defined as*

$$\begin{pmatrix} \mathbf{A}_1 & \cdots & \mathbf{A}_m \end{pmatrix}$$

where, for any $i \in \{1,\ldots,n\}$, the matrix $\mathbf{A}_i$ represents (in the canonical basis) the multiplication by some random element $\mathbf{a}_i \in \mathbb{F}_q[X]/(P(X))$.

4.4.3 Application of FF-DP to Ring-LPN

4.4.3.1 When the polynomial $P(X)$ splits totally in $\mathbb{F}_q$

When the polynomial $P(X) \stackrel{\text{def}}{=} X^{q-1} - 1 = \prod_{a\in\mathbb{F}_q^\times}(X-a)$, Theorem 4.44 already gives an application of FF-DP. The key point is that the roots of P are exactly the elements of $\mathbb{F}_q^\times$ which identifies as the Galois group of a Carlitz extension. In reality, this can be generalised to the case where the roots form a strict subgroup H of $\mathbb{F}_q^\times$, or even a coset of said H.

Theorem 4.51

Let H be a subgroup of $\mathbb{F}_q^\times$ and let $P(X) \stackrel{\text{def}}{=} \prod_{a\in H}(X-a)$. Then, there exists a Galois function field $L/\mathbb{F}_q(T)$, with ring of integers $\mathcal{O}_L$ and field of constants $\mathbb{F}_q$, such that $H = \mathrm{Gal}(L/\mathbb{F}_q(T))$ and

$$\mathcal{O}_L/(T+1)\mathcal{O}_L \simeq \mathbb{F}_q[X]/(P(X)).$$

Moreover, the action of H keeps invariant the Hamming supports.

Proof. Let $K \stackrel{\text{def}}{=} K_T$ be the Carlitz extension with respect to the T torsion, *i.e.*

$$K \stackrel{\text{def}}{=} \mathbb{F}_q(T)[\Lambda_T] = \mathbb{F}_q(T)[X]/X^{q-1}+T.$$

Since $G \stackrel{\text{def}}{=} \mathrm{Gal}(K/\mathbb{F}_q(T)) = \mathbb{F}_q^\times$ is *cyclic*, it has a *unique* subgroup N of cardinality $\frac{q-1}{|H|}$. Let $L \stackrel{\text{def}}{=} K^N$ be the fixed field of N. Since G is abelian, N is normal and therefore the extension $L/\mathbb{F}_q(T)$ is *Galois*, of Galois group G/N. Using again the cyclicity of G, this group identifies as H. In particular, L is an extension of degree $|H|$. It holds that its ring of integers $\mathcal{O}_L$ is exactly $\mathcal{O}_K^N$ the elements of $\mathcal{O}_K$ fixed by N, and since $(T+1)$ totally splits in $\mathcal{O}_K$, it also splits totally

in $\mathcal{O}_L$ and we have

$$\mathcal{O}_L \big/ (T+1)\mathcal{O}_L \simeq \underbrace{\mathbb{F}_q \times \cdots \mathbb{F}_q}_{|H| \text{ factors}} \simeq \mathbb{F}_q[X] \big/ (P(X)).$$

It is readily verified that the action of $H \stackrel{\text{def}}{=} \mathsf{Gal}(L/\mathbb{F}_q(T))$ only permutes the factors, but keeps the *supports* invariant. □

Therefore, this is possible to instantiate FF-DP with this function field L to yield a theorem analogous to Theorem 4.44.

Remark 4.52. *When the roots of $P(X)$ do not form a subgroup of G, but rather a* coset *bH instead, i.e.*

$$P(X) = \prod_{\alpha \in bH} (X - \alpha),$$

then it suffices to perform a translation by b prior: $X \mapsto bX$ also keeps all the distributions invariant (including the uniform), and $\mathbb{F}_q[X] \big/ (P(X))$ is mapped onto $\mathbb{F}_q[X] \big/ (\pi(X))$ where

$$\pi(X) \stackrel{def}{=} \prod_{\alpha \in H} (X - \alpha),$$

which yields the desired result by seeing the action of H as arising from a Galois action.

4.4.3.2 When P splits into irreducible polynomials with the same degree

As already noticed, if it is possible to interpret Ring-LPN as a particular case of FF-DP, all the instantiations do not necessarily fit into the framework developed in this Chapter for the search-to-decision reduction. In particular, not all rings can arise from a Galois function field. A necessary condition is that all the inertia degree should be equal, *i.e.* the ring $\mathcal{R} \stackrel{\text{def}}{=} \mathbb{F}_q[X] \big/ (P(X))$ should be a direct product of identical finite fields. In other words, it is necessary that P factorises as a product of irreducible polynomials with the same degree. In fact, this condition is sufficient. Over small fields, and in particular for $q = 2$, this is particularly interesting since asking for P to split completely is very restrictive.

For example, the instantiation of Ring-LPN over $\mathbb{F}_2$ has been considered in [HKLP+12] to design an authentication protocol named LAPIN. In this setting, the polynomial P splits into a product of m distinct irreducible polynomials

$$P(X) = P_1(X) \cdots P_m(X).$$

Assume that all the P_i's have the same degree, denoted by f. Therefore,

$$\mathcal{R} \stackrel{\text{def}}{=} \mathbb{F}_2[X] \big/ P(X) \simeq \mathbb{F}_{2^f} \times \cdots \times \mathbb{F}_{2^f}$$

is a product of m copies of $\mathbb{F}_{2^f}$.

Now, consider a function field K which is a Galois extension of $\mathbb{F}_2(T)$ with Galois group G and denote by $\mathcal{O}_K$ its ring of integers. Suppose that the ideal (T) of $\mathbb{F}_2[T]$ is unramified in $\mathcal{O}_K$, with inertia degree f. Then $T\mathcal{O}_K$ splits into a product of prime ideals:

$$T\mathcal{O}_K = \mathfrak{p}_1 \cdots \mathfrak{p}_m \qquad \text{and} \qquad \mathcal{O}_K \big/ T\mathcal{O}_K \simeq \prod_{i=1}^{m} \mathcal{O}_K \big/ \mathfrak{p}_i,$$

where, here again, the right–hand side is a product of m copies of $\mathbb{F}_{2^f}$.

Next, the idea is now to apply Theorem 4.30 in this setting. However, there is here a difficulty since for our search-to-decision reduction to hold, the noise should arise from a Galois invariant distribution. Thus, if we want the noise distribution to be Galois invariant we need to have a Galois invariant $\mathbb{F}_2$–basis of the algebra $\mathcal{O}_K/T\mathcal{O}_K$. One may wonder if such a basis even exists in general, and indeed it does. This can be deduced from a theorem of Noether asserting the existence of *local normal integral bases* at non ramified places [Noe32; Cha96]. However, here we give a constructive proof. Since this result also holds for larger finite fields, from now on, the underlying field is not supposed to be $\mathbb{F}_2$ anymore.

Proposition 4.53

Let $K/\mathbb{F}_q(T)$ be a Galois function field, with ring of integers $\mathcal{O}_K$. Let $Q \in \mathbb{F}_q[T]$ be an irreducible polynomial of degree 1, unramified in $\mathcal{O}_K$, and with inertia degree f. Let $\mathfrak{p}_1, \ldots, \mathfrak{p}_m$ be the prime ideals above Q.
Then $G \stackrel{\text{def}}{=} \mathrm{Gal}(K/\mathbb{F}_q(T))$ acts on the $|G|$-dimensional algebra $\mathcal{O}_K/Q\mathcal{O}_K$ and we can construct an element $\mathbf{x}$ such that $(\sigma(\mathbf{x}))_{\sigma\in G}$ forms an $\mathbb{F}_q$–basis of $\mathcal{O}_K/Q\mathcal{O}_K$.

Proof. By definition,

$$\mathcal{O}_K/\mathfrak{p}_1 \simeq \mathbb{F}_{q^f}.$$

Denote by $D_{\mathfrak{p}_1/Q}$ the decomposition group of $\mathfrak{p}_1$ above Q. Recall from Section 4.2.2.3, and more precisely from Equation (4.4), that

$$D_{\mathfrak{p}_1/Q} \simeq \mathrm{Gal}(\mathbb{F}_{q^f}/\mathbb{F}_q),$$

This entails in particular that $|D_P| = f$.

According to the Chinese Remainder Theorem,

$$\mathcal{O}_K/Q\mathcal{O}_K \simeq \mathcal{O}_K/\mathfrak{p}_1 \times \cdots \times \mathcal{O}_K/\mathfrak{p}_m.$$

Next, from the Normal Basis Theorem for finite fields (see for instance [LN97, Thm. 2.35]), there exists $\mathbf{a} \in \mathcal{O}_K/\mathfrak{p}_1$ such that $(\sigma(\mathbf{a}))_{\sigma\in D_{\mathfrak{p}_1/Q}}$ is an $\mathbb{F}_q$–basis of $\mathcal{O}_K/\mathfrak{p}_1$. Now, let

$$\mathbf{b} \stackrel{\text{def}}{=} (\mathbf{a}, 0, \ldots, 0) \in \prod_{i=1}^{m} \mathcal{O}_K/\mathfrak{p}_i \simeq \mathcal{O}_K/Q\mathcal{O}_K.$$

We claim that $(\sigma(\mathbf{b}))_{\sigma\in G}$ is an $\mathbb{F}_q$–basis of $\mathcal{O}_K/Q\mathcal{O}_K$. Indeed, denote by V the $\mathbb{F}_q$–span of $\{\sigma(\mathbf{b}) \mid \sigma \in G\}$ and suppose that V is a proper subspace of $\mathcal{O}_K/Q\mathcal{O}_K$. Then, there exists $i_0 \in \{1, \ldots, m\}$ such that

$$V \cap \mathcal{O}_K/\mathfrak{p}_{i_0} \subsetneq \mathcal{O}_K/\mathfrak{p}_{i_0},$$

where we denote by $\mathcal{O}_K/\mathfrak{p}_{i_0}$ the subspace $\{0\} \times \cdots \times \{0\} \times \mathcal{O}_K/\mathfrak{p}_{i_0} \times \{0\} \times \cdots \times \{0\}$ of

$\prod_j \mathcal{O}_K/\mathfrak{p}_j$. In particular,

$$\dim_{\mathbb{F}_q}\left(V \cap \mathcal{O}_K/\mathfrak{p}_{i_0}\right) < f.$$

Since G acts transitively on the $\mathfrak{p}_j$'s, there exists $\sigma_0 \in G$ such that $\sigma_0(\mathfrak{p}_1) = \mathfrak{p}_{i_0}$. By definition, we have

$$\mathbf{b} \in V \cap \mathcal{O}_K/\mathfrak{p}_1,$$

therefore

$$\sigma_0(\mathbf{b}) \in V \cap \mathcal{O}_K/\mathfrak{p}_{i_0}$$

and so does $(\sigma\sigma_0)(\mathbf{b})$ for any $\sigma \in D_{\mathfrak{p}_{i_0}/Q}$. Since

$$\left|D_{\mathfrak{p}_{i_0}/Q}\right| = f > \dim_{\mathbb{F}_q}\left(V \cap \mathcal{O}_K/\mathfrak{p}_{i_0}\right),$$

there exist nonzero elements $(\lambda_\sigma)_{\sigma \in D_{\mathfrak{p}_{i_0}/Q}} \in \mathbb{F}_q^f$ such that

$$\sum_{\sigma \in D_{\mathfrak{p}_{i_0}/Q}} \lambda_\sigma \cdot (\sigma\sigma_0)(\mathbf{b}) = 0. \tag{4.6}$$

Applying σ_0^{-1} to (4.6), we get

$$\sum_{\sigma \in D_{\mathfrak{p}_{i_0}/Q}} \lambda_\sigma \cdot (\sigma_0^{-1}\sigma\sigma_0)(\mathbf{b}) = 0.$$

By Equation (4.3), we have $\sigma_0^{-1} D_{\mathfrak{p}_{i_0}/Q} \sigma_0 = D_{\mathfrak{p}_1/Q}$ and the above sum is exactly

$$\sum_{\sigma \in D_{\mathfrak{p}_1/Q}} \lambda_\sigma \cdot \sigma(\mathbf{b}) = 0 \in \mathcal{O}_K/\mathfrak{p}_1.$$

Since by construction $(\sigma(\mathbf{b}))_{\sigma \in D_{\mathfrak{p}_1/Q}}$ form a basis of $\mathcal{O}_K/\mathfrak{p}_1$, we deduce that the λ_σ's are all zero. A contradiction. □

The previous proposition asserts the existence of a *normal* $\mathbb{F}_q$–basis of the space $\mathcal{O}_K/Q\mathcal{O}_K$, *i.e.* a Galois invariant basis. For any such basis, $(\mathbf{b}_\sigma)_{\sigma \in G}$ one can define a Galois noise distribution by sampling linear combinations of elements of this basis whose coefficients are independent Bernoulli random variables. This Ring–LPN distribution is hence defined as pairs $(\mathbf{a}, \mathbf{b}) \in \mathcal{O}_K/Q\mathcal{O}_K \times \mathcal{O}_K/Q\mathcal{O}_K$ such that $\mathbf{a}$ is drawn uniformly at random and $\mathbf{b} = \mathbf{as} + \mathbf{e}$ where $\mathbf{e}$ is a noise term drawn from the previously described distribution.

Definition 4.54 (Galois modulus)

Let r and f be positive integers. A polynomial $P(X) \in \mathbb{F}_q[X]$ of degree r is called a Galois modulus of inertia f if there exists a Galois function field $K/\mathbb{F}_q(T)$ and an unramified polynomial $Q(T) \in \mathbb{F}_q[T]$ of degree one such that

$$\mathbb{F}_q[X]/(P(X)) \simeq \mathcal{O}_K \big/ Q\mathcal{O}_K$$

and the ideal $Q\mathcal{O}_K$ has inertia degree f.

In particular, P factorises in $\mathbb{F}_q[X]$ as a product of distinct irreducible polynomials of same degree f.

Carlitz extensions permit to easily exhibit many Galois moduli of given inertia f. Indeed, let $M(T) \in \mathbb{F}_q[T]$ be any divisor of $T^f - 1$ which vanishes at least at one primitive f–th root of unity, and let K_M be the Carlitz extension with respect to the M–torsion. Set

$$r \stackrel{\text{def}}{=} \frac{[K_M : \mathbb{F}_q(T)]}{f} = \frac{\left|\left(\mathbb{F}_q[X] \big/ (M(X))\right)^\times\right|}{f}.$$

Then, any polynomial $P(X) \in \mathbb{F}_q[X]$ which is a product of r distinct irreducible polynomials of degree f is a Galois modulus. Indeed, since the multiplicative order of T modulo $M(T)$ is f, Theorem 4.19 entails that (T) does not ramify and has inertia degree f. In particular,

$$\mathcal{O}_{K_M} \big/ T\mathcal{O}_{K_M} \simeq \underbrace{\mathbb{F}_{q^f} \times \cdots \times \mathbb{F}_{q^f}}_{r \text{ copies}} \simeq \mathbb{F}_q[X] \big/ (P(X)).$$

Example 4.55. *The polynomial* $P(X) \stackrel{def}{=} X^{63} + X^7 + 1 \in \mathbb{F}_2[X]$ *is a* Galois *modulus of inertia* 9. *First, note that*

$$\begin{aligned} P(X) = {} & (X^9 + X^5 + X^4 + X + 1)(X^9 + X^6 + X^5 + X^2 + 1)(X^9 + X^6 + X^5 + X^4 + X^3 + X^2 + 1) \\ & (X^9 + X^7 + X^4 + X^2 + 1)(X^9 + X^7 + X^5 + X + 1)(X^9 + X^7 + X^6 + X^3 + X^2 + X + 1) \\ & (X^9 + X^7 + X^6 + X^4 + X^3 + X + 1). \end{aligned}$$

Now, let $M(T) \stackrel{def}{=} T^6 + T^3 + 1$ *and consider* K_M *the Carlitz extension of* M*–torsion with ring of integers* $\mathcal{O}_{K_M}$. *Then*

$$T^9 \equiv 1 \mod M$$

and 9 *is the smallest integer that has this property. By Theorem 4.19, the ideal* $T\mathcal{O}_M$ *splits into* $\frac{63}{9} = 7$ *ideals* $\mathfrak{p}_1, \ldots, \mathfrak{p}_7$ *and has inertia* 9. *Hence,*

$$\mathbb{F}_2[X] \big/ (P(X)) \simeq \mathcal{O}_{K_M} \big/ (T\mathcal{O}_{K_M}).$$

Remark 4.56. *The polynomial* $P(X)$ *of Example 4.55 is also* lightness-preserving *in the sense of [DP12, Def 2.22] which can be used to instantiate Ring-LPN.*

From now on, let $P(X)$ be a Galois modulus of degree r and inertia f, and let $\mathscr{B} \stackrel{\text{def}}{=} (\sigma(\mathbf{c}))_{\sigma \in G_P}$ denote a normal basis, where G_P is the Galois group of the related function field. Its existence is ensured by Proposition 4.53, but $\mathscr{B}$ need not be exactly the normal basis constructed in the proof. This is discussed further, after the statement of Theorem 4.58.

Definition 4.57 (Normal Ring–LPN distributionl)

A sample $(\mathbf{a}, \mathbf{b})$ is distributed according to the *Normal* Ring-LPN distribution relatively to basis $\mathscr{B}$, with secret $\mathbf{s}$ if

- $\mathbf{a}$ is drawn uniformly at random over $\mathbb{F}_q[X]/(P(X))$;
- $\mathbf{b} \stackrel{\text{def}}{=} \mathbf{a}\mathbf{s} + \mathbf{e}$, where $\mathbf{e}(X) \stackrel{\text{def}}{=} \sum_{\sigma \in G_P} e_\sigma \sigma(\mathbf{c})(X) \in \mathbb{F}_q[X]/(P(X))$ has coefficients e_i's which are independent q–ary Bernoulli random variables with parameter p.

All the formalism developed in this Chapter allows to give the following theorem:

Theorem 4.58

The decision Ring–LPN is equivalent to its search version for the normal Ring–LPN distribution.

However, in order for this theorem to be meaningful, the search version needs to remain hard. It should be emphasised that the Galois invariant basis constructed in the proof of Proposition 4.53 yields a noise which is partially cancelled when applying the projection

$$\mathcal{O}_K \big/ Q\mathcal{O}_K \to \mathcal{O}_K \big/ \mathfrak{p}_i.$$

This is an example of *folding attacks* as discussed in Section 1.3.4.2, as well as with more details in Section 7.3.3.5. Therefore, this choice of normal basis might yield insecure instances. On the other hand, Proposition 4.53 is only an existence result and actually, it turns out that a random element of $\mathcal{O}_K \big/ Q\mathcal{O}_K$ generates a normal basis with a high probability. Indeed, the existence of such a normal basis can be reformulated as $\mathcal{O}_K \big/ Q\mathcal{O}_K$ being a free $\mathbb{F}_q[G]$–module of rank 1, where $\mathbb{F}_q[G]$ is the group algebra of G with coefficients in $\mathbb{F}_q$ (group algebras will play an important role in Chapter 7. See Section 7.2.1 for a precise definition). An element $\mathbf{a} \in \mathcal{O}_K \big/ Q\mathcal{O}_K$ generates a normal basis if and only if

$$\mathcal{O}_K \big/ q\mathcal{O}_K \simeq \mathbb{F}_q[G] \cdot a$$

as $\mathbb{F}_q[G]$–module. Now, any other element of $\mathbb{F}_q[G]^\times \mathbf{a}$ is also a generator of a normal basis. Consequently, the probability that a uniformly random element of $\mathcal{O}_K \big/ Q\mathcal{O}_K$ is a generator of a normal basis is

$$\frac{\left|\mathbb{F}_q[G]^\times\right|}{\left|\mathbb{F}_q[G]\right|}.$$

For instance, consider the case of a cyclic group of order N prime to q, *i.e.*

$$G \stackrel{\text{def}}{=} \mathbb{Z} \big/ N\mathbb{Z}.$$

Then $X^N - 1$ splits into a product of distinct irreducible factors $u_1 \cdots u_r$ and

$$\mathbb{F}_q[G] \simeq \mathbb{F}_q[X]/(X^N - 1) \simeq \prod_i \mathbb{F}_q[X]/(u_i(X)).$$

In this context, the probability that a uniformly random element of $\mathcal{O}_K/Q\mathcal{O}_K$ generates a normal basis is

$$\frac{\prod_{i=1}^{r}(q^{\deg u_i} - 1)}{q^N}.$$

This is a starting point for the structured variants which will be considered in Chapter 7 with applications to secure multiparty computation.

Chapter 5

Taking More Advice from Lattice-Based Cryptography: The OCP Framework

Contribution of this thesis. We have already seen so far, code-based cryptography appears to lag behind lattice-based cryptography in terms of theoretical reductions, especially regarding problems involving structured codes. In Chapter 4, we began to reduce this gap by adapting a reduction technique from the world of structured lattices which involves number fields, to that of structured codes, by means of function fields. However, not all the lattice-based reductions have been explored, and the most recent one involve the so-called OHCP technique.

In this new Chapter based on [BCD23], we revisit OHCP in the context of code-based cryptography, to provide the first *direct* worst-case to average-case, search-to-decision reduction for the Decoding Problem. Finally, we discuss potential applications to structured variants.

Outline of the current chapter

5.1 Motivations

Chapter 4 showed that the approach used in lattice-based cryptography in the past fifteen years is actually very general and may be adapted to other metrics, especially the Hamming metric.

However, similarly to what happens with lattices, the function field approach introduced in the previous chapter is still frustrating for at least two reasons: first, the reduction is more suitable in an LPN regime, where the rate of the underlying code is very low, or equivalently when the algorithms solving either the search or the decisional version are allowed to query a lot of samples. In particular, in the reduction for FF-DP, if ε is the advantage of the algorithm solving the decisional problem, we need at least $1/\varepsilon^2$ samples in the search version in order to succeed with a good enough probability. Second, and maybe the most important one, the arithmetic conditions in Theorem 4.30 make the reduction useless for the codes used in actual code-based cryptosystems such as BIKE or HQC. Indeed, those schemes consider quasi-cyclic codes of block length n such that 2 is primitive modulo n, or equivalently such that X^n-1 has only two factors in $\mathbb{F}_2[X]$: The trivial $(X-1)$ and $(X^{n-1}+\cdots+1)$. In particular, even if the ring $\mathbb{F}_2[X]/(X^n-1)$ can be seen as a quotient of the ring of integers $\mathcal{O}_K$ of some function field K by some ideal $\mathfrak{p} \subset \mathbb{F}_2[T]$ as shows in Example 4.5, the extension $K/\mathbb{F}_2(T)$ cannot be Galois since the primes above $\mathfrak{p}$ do not have the same inertia degrees. Note that this is not inherent to the design of the scheme, but it was the choice made by the authors of those cryptosystems, probably in order to limit the power of an attacker who may be tempted to use so-called *folding attacks* to reduce the problem modulo one of the factors.

Remark 5.1. *In Chapter 7 we give another application of the framework developed in Chapter 4 for another kind of structured codes, namely quasi-abelian codes. They are used in order to build programmable Pseudorandom Correlation Generators, an important tool for modern secure multiparty computation. In Section 7.3.3.5 we also discuss more in-depth those folding attacks in this general setting.*

Nevertheless, this limitation of arithmetic nature is not quite satisfying and it would be better if we could remove it. It turns out that for lattice-based cryptography, up until recently, the known reductions faced similar difficulties. However, in the breakthrough paper [PRS17], Peikert, Regev and Stephens-Davidowitz introduced a powerful tool for reductions called the OHCP framework (for *Oracle with Hidden Center Problem*), and which is now the modern way of dealing with reductions in lattice based-cryptography, especially in the context of structured lattices (*e.g.* module-lattices). Since then, it has been used in many reductions [RSW18; BJRW20b; PS21b] to name a few.

Yet, no matter how general it is, this OHCP technique seems to be inherent to the euclidean metric at first glance, it does not seem clear how one can extend it to other metrics, even less for the Hamming metric.

Motivated by the similarity between codes and lattices, we revisit in this chapter this OHCP technique and adapt it for use in the Hamming metric. More precisely, we extract the very substance of this tool which actually lie in a technical subproblem called the *Oracle Comparison Problem* (OCP), where one has access to two oracles $\mathcal{O}_0$ and $\mathcal{O}_1$ whose acceptance probability on some input x are a *shift* of one another: if $f_i(x) \stackrel{\text{def}}{=} \mathbb{P}(\mathcal{O}_i \text{ accepts on input } x)$, then $f_1(x) = f_0(x + \text{some additive shift})$; and the goal is to detect which one is in advance. In reality, this innocent looking problem is the core of the OHCP technique. Indeed, the reduction of [PRS17] is constituted by two parts: an inner algorithm solving the above OCP problem, and an outer algorithm using this OCP to build the actual reduction. It is often the outer algorithm which is called OHCP. The idea is that we can use an algorithm distinguishing between the LWE and uniform distributions to build a new oracle whose acceptance probability on some input $\mathbf{x}$ depends only on *the distance* between $\mathbf{x}$ and the secret error term $\mathbf{e}$ in a given LWE sample $(\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle + \mathbf{e})$ (which can be a plain LWE sample, or a structured variant), hence the name *Hidden Center*. The outer algorithm then uses a solver OCP in order to cleverly change $\mathbf{x}$ so that it converges towards $\mathbf{e}$.

For adapting this technique to the Hamming metric, we introduce a different outer algorithm which we call the *Oracle with Hidden Support Problem* (OHSP), and which will be able to recover the *Support* of the error $\mathbf{t}$ in a given instance of the Decoding Problem $(\mathbf{G}, \mathbf{m}\cdot\mathbf{G}+\mathbf{t})$, by making use of a solver for OCP. Since in the Hamming metric the support determines the error term, our outer algorithm is therefore able to recover $\mathbf{t}$, *i.e.* to decode the noisy codeword, given only an adversary against the decisional decoding problem.

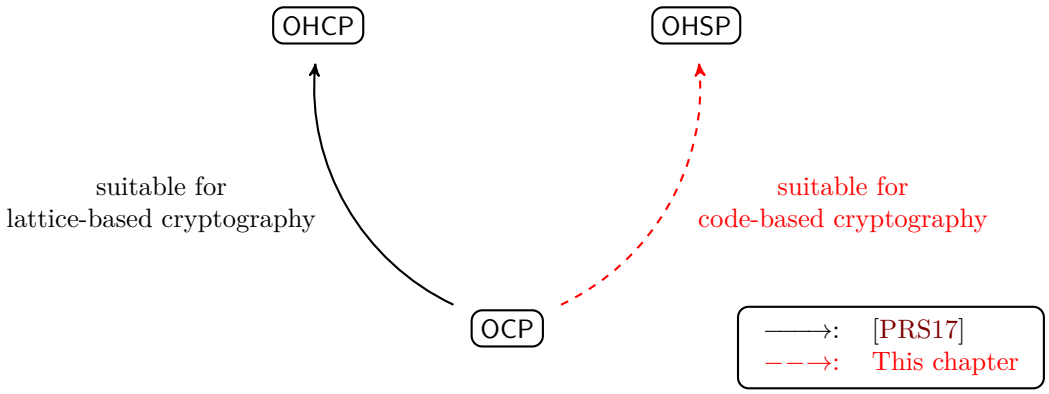


Figure 5.1: Relationships between OCP, OHCP and OHSP.

Contrary to previous reduction techniques used in the coding theoretic setting such as [FS96] which uses the Goldreich-Levin theorem (see Section 1.2.3), OCP-based techniques are typically suited for *worst-case to average-case* reductions. Indeed, starting from a fixed, given instance of the decoding problem, the first step of every OCP-based reduction is to somehow randomise the instance. In particular, with this technique, we get the first direct reduction from a worst-case search decoding problem where the input instance is *fixed*, to an average-case decisional decoding problem, where the input is a random code. Informally, the reduction we obtain can be stated as follows (for binary codes):

Theorem 5.2 (Informal)

Let $n, k, t \in \mathbb{N}$ and let $D < 1/2$ be such that

$$\frac{k}{n} = \frac{1}{n^D} \text{ and } \frac{t}{n} = \frac{\log(n)^2}{n^{1-D}}.$$

Suppose that there exists an algorithm which distinguishes with polynomial advantage between $(\mathbf{A}, \mathbf{sA}+\mathbf{e})$ and $(\mathbf{A}, \mathbf{y})$ where $\mathbf{A}$ is a random binary $k \times n$ matrix, $\mathbf{y}$ is a random binary vector of length n and $\mathbf{e}$ is a random binary vector of Hamming weight

$$\frac{n}{2}\left(1 - \frac{1}{n^{D(1+o(1))}}\right).$$

Then there exists an algorithm which solves the worst-case decoding problem for input codes of length n, dimension k and decoding distance t.

There is a little caveat though. Indeed, the reduction crucially relies on a technical tool called

the *smoothing parameter*, which has widely been used in lattice-based cryptography for years. Basically, it quantifies the level of noise needed so that a syndrome $\mathbf{He}^\top$ of a *fixed* code with parity check matrix $\mathbf{H}$ is statistically close to the uniform distribution. Note that on average, *i.e.* when the matrix $\mathbf{H}$ is uniformly random, such a result is given by the so-called left-over hash lemma (see for instance [Deb23, Section 2.5]). There is a caveat though here. Indeed, in the code-based setting, such smoothing bounds have only been obtained for *balanced* binary codes, *i.e.* for codes which do not have codewords of too small weight, neither of too large weight, while when working on average, we can obtain similar bounds without that hypothesis[i].

In this chapter, we also discuss an average-case to average-case reduction, which does not make use of those smoothing bounds, but obtain similar code-rate and noise-level. It has to be noted that the parameter of the reduction are worse than that of previous average-case to average-case reductions since OCP is typically made to handle the worst-case.

Finally, we come back to our initial goal of handling structured codes, as it has been done for structured lattices. If this technique does not suffice in the coding theoretic setting, in this chapter we identify what is left to be done.

5.2 OCP-based search-to-decision reductions

Notations. In this Chapter, we will write $X \leftarrow \mathrm{Ber}(\omega)$ for a real parameter $\omega \in \mathbb{R}_+$, when X is a (binary) Bernoulli random variable such that

$$\mathbb{P}(X = 1) = \frac{1}{2}\left(1 - 2^{-\omega}\right).$$

The parameter ω is also called the log-*bias* of X. The rationale behind this notation is two-fold: first, we will need to focus our attention at large noise-rate, namely at the neighbourhood of $1/2$. This corresponds to $\omega \to \infty$. But it has another advantage: it is readily seen that when $X \leftarrow \mathrm{Ber}(\omega_1)$ and $Y \leftarrow \mathrm{Ber}(\omega_2)$ are two independent Bernoulli random variables, then

$$X + Y \leftarrow \mathrm{Ber}(\omega_1 + \omega_2), \quad \text{(where the sum is done in } \mathbb{F}_2\text{)}.$$

As usual, if N is some positive integer, we denote by $\mathrm{Ber}(\omega)^{\otimes N}$ the probability distribution which outputs vectors of length N whose entries are independent and identically distributed according to $\mathrm{Ber}(\omega)$. In particular, if $\mathbf{x} \leftarrow \mathrm{Ber}(\omega)^{\otimes N}$, then $|\mathbf{x}|$ is concentrated around its expected value and

$$|\mathbf{x}| \approx \mathbb{E}(|\mathbf{x}|) = \frac{N}{2}\left(1 - 2^{-\omega}\right).$$

5.2.1 A high level intuition

Let $\mathcal{A}$ be an algorithm running in time T which can distinguish noisy codewords at some targeted Hamming distance, from uniform random vectors of the ambient space. $\mathcal{A}$ takes as input a pair $(\mathbf{A}, \mathbf{y})$ where $\mathbf{A} \leftarrow \mathbb{F}_2^{k \times N}$ is a uniformly random binary matrix, and $\mathbf{y} \in \mathbb{F}_2^N$ is distributed according to some distribution $\mathcal{D}$, and has to output

- "1" if $\mathbf{y}$ is a codeword of the code $\mathscr{C}$ generated by $\mathbf{A}$ corrupted by an error $\mathbf{e}$ whose entries are independent Bernoulli random variables of some parameter $\omega \in \mathbb{R}_+$, that is

$$\mathbf{y} \stackrel{\text{def}}{=} \mathbf{sA} + \mathbf{e},$$

[i] That being said, a random code will be balanced will overwhelming probability.

for some $\mathbf{s} \in \mathbb{F}_2^k$, and $\mathbf{e} \leftarrow \mathrm{Ber}(\omega)^{\otimes N}$.

- "0" if $\mathbf{y}$ is uniformly distributed in $\mathbb{F}_2^N$.
- Otherwise, $\mathcal{A}$'s behaviour is undefined.

Moreover, in order to be relevant in a cryptographic context, we also assume that $\mathcal{A}$ is not perfect, and might give false answers. This is quantified by its distinguishing advantage

$$\mathrm{Adv}(\mathcal{A}) \stackrel{\text{def}}{=} \varepsilon(k,N,\omega) \stackrel{\text{def}}{=} \frac{1}{2}\left(\mathbb{P}_{\mathbf{A},\mathbf{s},\mathbf{e}}(\mathcal{A}(\mathbf{A},\mathbf{sA}+\mathbf{e})=1) - \mathbb{P}_{\mathbf{A},\mathbf{y}}(\mathcal{A}(\mathbf{A},\mathbf{y})=1)\right),$$

$$(i)\ \mathbf{A} \leftarrow \mathbb{F}_2^{k\times N}, \quad (ii)\ \mathbf{s} \leftarrow \mathbb{F}_2^k, \quad (iii)\ \mathbf{y} \leftarrow \mathbb{F}_2^N \quad \text{and} \quad (iv)\ \mathbf{e} \leftarrow \mathrm{Ber}(\omega)^{\otimes N}. \tag{5.1}$$

In the sequel, we may drop the dependencies in (k,N,ω) when it is clear from the context.

Starting from a given, *fixed*, noisy codeword $\mathbf{y}$ of a *fixed* code $\mathscr{C}$ generated by some matrix $\mathbf{G} \in \mathbb{F}_2^{k\times n}$, that is

$$\mathbf{y} \stackrel{\text{def}}{=} \mathbf{mG} + \mathbf{t},$$

where $\mathbf{m} \in \mathbb{F}_2^k$ is fixed, and $\mathbf{t} \in \mathbb{F}_2^n$ has Hamming weight t ; we want to use Algorithm $\mathcal{A}$ to get some insight on the *Hamming support* of $\mathbf{t}$. Recall that in the reduction from Chapter 4 (in Section 4.3.3) one of the main steps consists in a *hybrid argument* to turn our distinguisher between noisy codewords and the uniform, into a distinguisher between two consecutive *hybrid* distributions $\mathcal{D}_i$ and $\mathcal{D}_{i+1}$, which randomise part of the vector $\mathbf{y}$, before using a *guess-and-search* technique. More precisely, we made a guess for the secret, applied a transformation so that the samples are distributed according to $\mathcal{D}_i$ if the guess is correct, and $\mathcal{D}_{i+1}$ otherwise, and use the distinguisher to make our decision.

At a high level, the reduction of this chapter will follow a similar path, but using a *continuous* version of the aforementioned hybrid argument. More precisely, we will turn our distinguisher $\mathcal{A}$ into a distinguisher between two closer distributions $\mathcal{D}$ and $\mathcal{E}$, and then for each $i \in \{1,\dots,n\}$ we will apply a transformation such that the samples are distributed according to $\mathcal{D}$ if $i \in \mathrm{Supp}(\mathbf{t})$, and to $\mathcal{E}$ otherwise, before applying the distinguisher to make the decision. The main difficulty is that now, the frontier between $\mathcal{D}$ and $\mathcal{E}$ is not clear cut anymore due to the "continuous deformation"; and basically making the decision will exactly mean solving this so-called Oracle Comparison Problem.

5.2.2 Outline of the reduction

The reduction will follow from a sequence of lemmas, which we now sketch before giving formal proofs.

First, we need to define the "intermediate" distribution in our continuous hybrid argument. The starting and ending points seem clear: at one end we should have the distribution of the decoding problem, and on the other hand we should have the uniform. However for the reduction to work, we will in reality need to vary the length of the codes at stake. Therefore, we will step into LPN territory. Recall that the distribution of the decoding problem of a random $[N,k]$ code can be seen as the collection of N samples of the form $(\mathbf{a}, \langle \mathbf{a},\mathbf{s}\rangle + e)$ where $\mathbf{a} \leftarrow \mathbb{F}_2^k$ and $e \leftarrow \mathrm{Ber}(\omega)$. This is made precised in Step 1 below. When the value of the parameter ω matters, we will call such a sample an $\mathrm{LPN}(\omega)$ sample. Moreover, by definition when $e \leftarrow \mathrm{Ber}(\omega)$, then

$$\mathbb{P}(e=1) = \frac{1}{2}\left(1-2^{-\omega}\right) \xrightarrow[\omega\to\infty]{} \frac{1}{2}.$$

In other words, LPN(∞) is nothing but the uniform distribution over $\mathbb{F}_2^k \times \mathbb{F}_2$, and a distinguisher which behaves differently when fed with LPN(ω) and LPN(∞) samples, must change its acceptance probability as ω grows. Detecting this change of behaviour is the main goal of the Oracle Comparison Problem. We are now ready to describe the reduction.

Step 1. (*From distinguishing LPN samples to distinguishing noisy codewords*). We start from an algorithm $\mathcal{A}$ that distinguishes, with advantage ε, between a noisy codeword $\mathbf{c}+\mathbf{e}$ (by outputting 1) and a uniform $\mathbf{y} \in \mathbb{F}_2^N$ (by outputting 0) with $\mathbf{c}$ drawn uniformly at random from some random binary $[N,k]$-code $\mathscr{C}$, and $\mathbf{e} \leftarrow \mathrm{Ber}(\omega)^{\otimes N}$. This algorithm can easily be turned into an algorithm $\mathcal{A}'$ distinguishing (with the same advantage ε) oracles

$$\mathcal{O}(\omega) : (\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle + e) \quad \text{and} \quad \mathcal{O}(\infty) : (\mathbf{a}, u) \tag{5.2}$$

where $\mathbf{s} \in \mathbb{F}_2^k$, $e \leftarrow \mathrm{Ber}(\omega)$, $\mathbf{a} \leftarrow \mathbb{F}_2^k$ and $u \leftarrow \mathbb{F}_2$. Indeed, given one of the above oracles $\mathcal{O}$, in order to design $\mathcal{A}'$, it is enough to perform N queries $(\mathbf{a}_i, b_i)$ to $\mathcal{O}$ and gather them to generate the pair $(\mathbf{A}, \mathbf{b})$ where the columns of $\mathbf{A}$ are the $\mathbf{a}_i^\top$'s and $\mathbf{b} = (b_1, \ldots, b_N)$. Then, we feed $\mathcal{A}$ with the generated pair $(\mathbf{A}, \mathbf{b})$ to make our decision. Defining such an algorithm $\mathcal{A}'$ solving the above LPN-decisional problem with at most N queries may seem at first sight tautological, but for our reduction it is more convenient to emphasise this point.

From now on, we may as well assume that our algorithm $\mathcal{A}$ actually distinguishes LPN(ω) from LPN(∞) with some advantage ε, *and* querying the underlying oracles *at most* N times.

Step 2. (*Transforming a fixed noisy codeword into LPN-samples*). The starting point of the reduction consists in noticing that, from any input of a decoding problem, we can build some LPN-oracle. Given

$$\Big(\mathbf{G}, \mathbf{y} \stackrel{\text{def}}{=} \mathbf{m}\mathbf{G} + \mathbf{t}\Big) \in \mathbb{F}_2^{k \times n} \times \mathbb{F}_2^n,$$

we can design the following oracle $\mathcal{O}_0$ which samples $\mathbf{r}$ according to $\mathrm{Ber}(\omega_0)^{\otimes n}$, and then computes $\mathbf{r}\mathbf{G}^\top$ and

$$\langle \mathbf{y}, \mathbf{r} \rangle = \langle \mathbf{m}\mathbf{G} + \mathbf{t}, \mathbf{r} \rangle = \langle \mathbf{m}, \mathbf{r}\mathbf{G}^\top \rangle + \langle \mathbf{t}, \mathbf{r} \rangle. \tag{5.3}$$

The oracle $\mathcal{O}_0$ outputs LPN–like samples of the form:

$$\mathcal{O}_0 : \big(\mathbf{a}', \langle \mathbf{s}, \mathbf{a}' \rangle + e'\big) \quad \text{where} \quad \begin{cases} \mathbf{s} \stackrel{\text{def}}{=} \mathbf{m} \\ \mathbf{a}' \stackrel{\text{def}}{=} \mathbf{r}\mathbf{G}^\top \\ e' \stackrel{\text{def}}{=} \langle \mathbf{t}, \mathbf{r} \rangle. \end{cases} \tag{5.4}$$

The following simple lemma shows that the random variable e' follows a Bernoulli distribution $\omega \stackrel{\text{def}}{=} \omega_0 |\mathbf{t}| = \omega_0 t$. This is basically an application of the so-called piling-up lemma.

Lemma 5.3

Let $\mathbf{r} \leftarrow \mathrm{Ber}(\alpha)^{\otimes n}$, then for any $\mathbf{z} \in \mathbb{F}_2^n$ we have

$$\langle \mathbf{z}, \mathbf{r} \rangle \leftarrow \mathrm{Ber}(|\mathbf{z}|\alpha).$$

Proof. Let $z \stackrel{\text{def}}{=} |\mathbf{z}|$ and $p \stackrel{\text{def}}{=} \frac{1}{2}\left(1-2^{-\alpha}\right)$. By definition of $\mathbf{r} \leftarrow \mathrm{Ber}(\alpha)^{\otimes n}$ we have the following computation

$$\begin{aligned}
\mathbb{P}_{\mathbf{r}}\left(\langle \mathbf{z},\mathbf{r}\rangle = 1\right) &= \sum_{j \text{ odd}} \binom{z}{j} p^j (1-p)^{z-j} \\
&= \frac{1}{2}\left(\sum_j \binom{z}{j} p^j (1-p)^{z-j} - \sum_j (-1)^j \binom{z}{j} p^j (1-p)^{z-j}\right) \\
&= \frac{1}{2}\left(1-(1-2p)^z\right) \\
&= \frac{1}{2}\left(1-2^{-z\alpha}\right),
\end{aligned}$$

which concludes the proof. □

However, Oracle $\mathcal{O}_0$ does not output *true* LPN(ω) samples since $\mathbf{a}' \stackrel{\text{def}}{=} \mathbf{r}\mathbf{G}^\top$ is *a priori* not uniformly distributed. Furthermore, it is correlated to the noise term $e' \stackrel{\text{def}}{=} \langle \mathbf{t},\mathbf{r}\rangle$. Nonetheless, using the Data Processing Inequality (recalled in Equation (1.2)), replacing the sample $(\mathbf{r}\mathbf{G}^\top, \langle \mathbf{m},\mathbf{r}\mathbf{G}^\top\rangle + \langle \mathbf{t},\mathbf{r}\rangle)$ by a genuine LPN sample $(\mathbf{a}, \langle \mathbf{a},\mathbf{m}\rangle + e)$ changes the probabilities by at most the additive term

$$\Delta\left(\left(\mathbf{r}\mathbf{G}^\top, \langle \mathbf{r},\mathbf{t}\rangle\right), (\mathbf{a},e)\right), \quad \text{where} \quad e \leftarrow \mathrm{Ber}(\omega) \quad \text{and is independent from } \mathbf{a}.$$

In the instantiations of the reduction, we need choose the distribution of $\mathbf{r}$ such that this statistical distance is negligible. Such a distribution is known as a *smoothing distribution* in the literature. For our applications, we will carefully choose ω_0 using the smoothing bounds recently developed in the coding theoretic setting in [BLVW19; DST19b; DDRT23; DR22].

Now, one may wonder how we can use $\mathcal{O}_0$ with our algorithm $\mathcal{A}'$ distinguishing between LPN-distributions to solve our underlying decoding problem. It is the aim of the next step.

Step 3. (*Solving the Oracle Comparison Problem* (OCP)). OCP has been introduced by Peikert, Regev and Stephens-Davidowitz in [PRS17]. Intuitively, given access to two oracles $\mathcal{O}_1$ and $\mathcal{O}_2$ whose acceptance probability are just a "shift" of one another, the goal of OCP is to tell which one is in advance, and which one lags behind (see Figure 5.2). This is made more precise in Lemma 5.8.

The first core idea of the reduction is to notice that in order to build the oracle $\mathcal{O}_0$ of (5.4), we have computed $\langle \mathbf{y},\mathbf{r}\rangle$ with $\mathbf{r} \leftarrow \mathrm{Ber}(\omega_0)^{\otimes n}$, leading to an LPN-like sample with parameter $\omega \stackrel{\text{def}}{=} \omega_0|\mathbf{t}|$. We could instead have considered an inner product of the form $\langle \mathbf{y}+\mathbf{z},\mathbf{r}\rangle$ for some fixed $\mathbf{z} \in \mathbb{F}_2^n$ of our choice. This has the following consequence: our new oracle provides LPN-samples with Bernoulli noise of parameter $\omega_0|\mathbf{t}+\mathbf{z}|$. In particular, if $\mathbf{z} = \mathbf{t}$, then the noise is completely cancelled. In other words, the very idea of the reduction is to hint our algorithm towards correctly guessing this value. In the context of lattice-based cryptography, this is done via a *guided random walk*, where one starts from some $\mathbf{z}_0$ and iteratively updates it by choosing a direction, and querying the OCP solver to decide if

this path is worth taking, or not. This is known as the *Oracle with Hidden Center Problem* (OHCP) ([PRS17, Definition 4.3; RSW18]). This approach really uses properties of the Euclidean metric.

In the code-based setting, we will make use of the existence of *supports*. Indeed, note that for $i \in \{1,\dots,n\}$, if

$$\mathbf{z} = \mathbf{v}_i \stackrel{\text{def}}{=} (0,\dots,0,1,0,\dots,0)$$

is a unit vector where the non zero entry happens at position i, *i.e.* the i-th element of the canonical basis of $\mathbb{F}_2^n$, then

$$|\mathbf{t}+\mathbf{z}| = \begin{cases} |\mathbf{t}|+1 & \text{if } i \notin \mathrm{Supp}(\mathbf{t}) \\ |\mathbf{t}|-1 & \text{if } i \in \mathrm{Supp}(\mathbf{t}). \end{cases} \tag{5.5}$$

Let oracle $\mathcal{O}^{\mathbf{v}_i}$ be defined similarly to $\mathcal{O}_0$, by outputting $\langle \mathbf{y}+\mathbf{v}_i, \mathbf{r}\rangle$ instead of $\langle \mathbf{y},\mathbf{r}\rangle$. In other words, $\mathcal{O}_0$ outputs LPN-like samples of parameter $\omega = \omega_0 t$, while $\mathcal{O}^{\mathbf{v}_i}$ outputs LPN-like samples of parameter $\omega_0(t+1)$ or $\omega_0(t-1)$, depending whether $i \in \mathrm{Supp}(\mathbf{t})$ or not. Because our algorithm $\mathcal{A}$ distinguishes $\mathrm{LPN}(\omega) = \mathrm{LPN}(\omega_0 t)$ from $\mathrm{LPN}(\infty)$ with advantage ε, the probability that it outputs 1 when being fed with $\mathcal{O}_0$ is roughly $\frac{1}{2}+\varepsilon$, and since the noise level in $\mathcal{O}^{\mathbf{v}_i}$ is different, we may expect that the acceptance rate of $\mathcal{A}$ on inputs from $\mathcal{O}^{\mathbf{v}_i}$ slightly differs ; a behaviour that could be detected via statistical methods. Unfortunately, the success probability $\frac{1}{2}+\varepsilon$ may be the same in all these cases since the noise levels are in reality very close. Therefore, we need to somehow *amplify* this discrepancy. This brings us to the second core idea of the reduction: choosing a smoothing distribution which depends on a second parameter x.

More precisely, instead of defining $\mathcal{O}_0$ and $\mathcal{O}^{\mathbf{v}_i}$ by sampling $\mathbf{r}$ according to $\mathrm{Ber}(\omega_0)^{\otimes n}$, we choose $\mathbf{r} \leftarrow \mathrm{Ber}(2^x\omega_0)^{\otimes n}$ for $x \in \mathbb{R}_+$. The LPN-noise now follows the following distributions

$$\mathrm{Ber}(2^x\omega_0 t) \text{ in } \mathcal{O}_0 \quad \text{and} \quad \begin{cases} \mathrm{Ber}\left(2^x\omega_0(t-1)\right) & \text{if } t_i = 1 \\ \mathrm{Ber}\left(2^x\omega_0(t+1)\right) & \text{if } t_i = 0 \end{cases} \text{ in } \mathcal{O}^{\mathbf{v}_i}. \tag{5.6}$$

Letting $x \to \infty$, the above distributions go to $\mathrm{Ber}(\infty)$. In this situation, $\mathcal{A}$ should reject with probability $1/2+\varepsilon$, that is to say $\mathcal{A}$ should only accept with probability $1/2-\varepsilon$. In other words, if we feed $\mathcal{A}$ with samples from $\mathcal{O}_0(x)$ and make x large enough, the acceptance probability of $\mathcal{A}$ will drastically drop. Say that this change of behaviour happens for x greater than some x_0, that is starting from a noise $\mathrm{Ber}(2^{x_0}\omega_0 t)$.

Assume now that we feed $\mathcal{A}$ with $\mathcal{O}^{\mathbf{v}_i}$ instead of $\mathcal{O}_0$. We can also play with the value x and look at the acceptance probability of $\mathcal{A}$. We know that the change in the behaviour will happen when the noise reaches $\mathrm{Ber}(2^{x_0}\omega_0 t)$. Therefore, in that case, we will observe a difference at some $x_0' \geqslant 0$ such that

$$\begin{cases} 2^{x_0'}\omega_0(t-1) = 2^{x_0}\omega_0 t \iff x_0' = x_0 + \log\left(\frac{t}{t-1}\right) > x_0 & \text{if } i \in \mathrm{Supp}(\mathbf{t}), \\ 2^{x_0'}\omega_0(t+1) = 2^{x_0}\omega_0 t \iff x_0' = x_0 + \log\left(\frac{t}{t+1}\right) < x_0 & \text{if } i \notin \mathrm{Supp}(\mathbf{t}). \end{cases}$$

It turns out that with classical statistical methods, we can now detect this difference in the acceptance probability of $\mathcal{A}$. The idea is just to estimate when $\mathcal{A}$ changes its behaviour

given as input $\mathcal{O}_0$ and $\mathcal{O}^{\mathbf{v}_i}$. Depending whether $i \in \mathrm{Supp}(\mathbf{t})$ or not, this change of behaviour will happen for a smaller x than with input $\mathcal{O}_0$, or a bigger x. This yields the claimed reduction: for $i \in \{1, \ldots, n\}$ we are able to decide whether $i \in \mathrm{Supp}(\mathbf{t})$ or not, *i.e.* we are able to recover the *hidden support* of the error, and hence to solve the decoding problem. In other words, we turned a "distinguishing decoding" algorithm into a "search decoding" algorithm.

This is summarised in the very general Theorem 5.4 below.

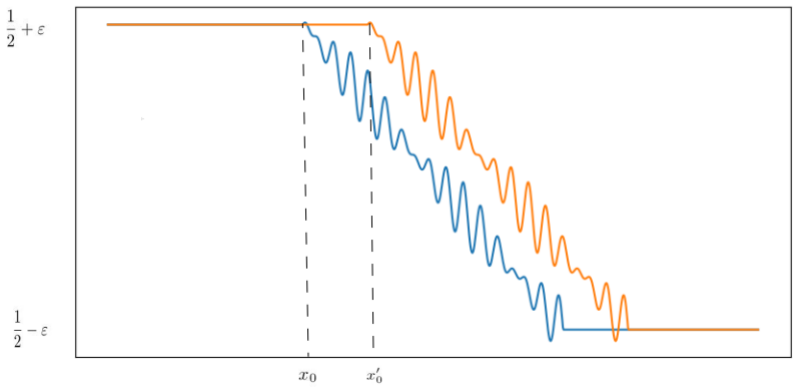


Figure 5.2: Illustration of Step 3 in the case $i \in \mathrm{Supp}(\mathbf{t})$: we plot the acceptance probability of our algorithm as a function of the parameter x.

Theorem 5.4

Let $N,n,t \in \mathbb{N}$ and $k \in \{0,\dots,\min(N,n)\}$. Let $(\mathbf{G},\mathbf{mG}+\mathbf{t})$ be a *fixed* instance of the Decoding Problem, with $\mathbf{G} \in \mathbb{F}_2^{k\times n}$, $\mathbf{m} \in \mathbb{F}_2^k$, and $|\mathbf{t}| = t \in \{0,\dots,n\}$.

Suppose that there exists an algorithm $\mathcal{A}$ which distinguishes in time T distribution $(\mathbf{a},\mathbf{sa}+\mathbf{e})$ from $(\mathbf{a},\mathbf{y})$ with advantage $\varepsilon(k,N,\omega)$ where $\mathbf{a},\mathbf{s},\mathbf{e},\mathbf{y}$ satisfy (5.1) and $\omega \in \mathbb{R}_+$ is such that

$$\omega = \Omega(1) \quad \text{and} \quad \omega = O(n). \tag{5.7}$$

Let $\omega_0, \alpha \in \mathbb{R}_+$ be such that

$$t\,\omega_0 = \omega \quad \text{and} \quad \alpha \stackrel{\text{def}}{=} \max\left(\frac{1}{\varepsilon(k,N,\omega)},N,n\right). \tag{5.8}$$

Then, there exists an algorithm which takes as input $(\mathbf{G},\mathbf{mG}+\mathbf{t})$ and which outputs $\mathbf{t}$ in time $T\,\mathsf{poly}(\alpha)$ with probability (over its internal randomness and **not** the choice of $\mathbf{G}$, $\mathbf{m}$ and $\mathbf{t}$ which are fixed) bigger than

$$1 - 2^{-\Omega(n)} - N\,\mathsf{poly}(\alpha) \max_{x\geqslant 0} \Delta\Big(\Big(\mathbf{r}(x)\mathbf{G}^\top, \langle \mathbf{r}(x),\mathbf{t}\rangle\Big),(\mathbf{a},e(x))\Big), \tag{5.9}$$

where $\mathbf{a} \leftarrow \mathbb{F}_2^k$, $\mathbf{r}(x) \leftarrow \mathrm{Ber}(2^x\omega_0)^{\otimes n}$ and $e(x) \leftarrow \mathrm{Ber}(2^x\omega_0 t)$ with $x \geqslant 0$.

Remark 5.5. *Theorem 5.4 is given with as many parameters as possible for flexibility. In practice, and for our instantiations (Theorem 5.16 for the average-case to average-case reduction; or Theorem 5.21 for the worst-case to average-case), we will restrict ourselves to $N = n$, i.e. we only allow to query n LPN samples for the distinguishing problem, where n is the length of the code we want to decode. However, it can still be used for larger values of N, in an LPN context.*

Moreover, the dependency in x has been made explicit in the theorem, but here again in practice the maximum is reached when $x = 0$: the higher is the noise, the closer are our LPN-like samples to genuine LPN.

5.2.3 Proof of the reduction

The proof will exactly follow the aforementioned steps. In reality, the only thing that needs to be made more precise is the third, and last, one ; namely the Oracle Comparison Technique. From now on, let $(\mathbf{G},\mathbf{mG}+\mathbf{t})$ be the instance of the decoding problem we wish to solve, and let $\mathcal{A}$ be our distinguisher.

For $\mathbf{z} \in \mathbb{F}_2^n$, consider the oracles $\mathcal{O}^{\mathbf{z}}(\cdot)$ and $\mathcal{O}^{\mathbf{z}}_{\text{ideal}}(\cdot)$ defined in Figure 5.3. As mentioned in Step 2 above, $\mathcal{O}^{\mathbf{z}}(\cdot)$ is the actual oracle we can build from our instance $(\mathbf{G},\mathbf{mG}+\mathbf{t})$, and which outputs our LPN-like samples. However, since $\mathbf{rG}^\top$ is not really uniformly distributed, and is correlated to $\langle \mathbf{z}+\mathbf{t},\mathbf{r}\rangle$, the real oracle $\mathcal{O}^{\mathbf{z}}(\cdot)$ is quite difficult to analyse. Instead, we replace it with the idealised version $\mathcal{O}^{\mathbf{z}}_{\text{ideal}}(\cdot)$ having exactly the same noise term (namely $\mathrm{Ber}(2^x\omega_0|\mathbf{z}+\mathbf{t}|)$) but independent from the first entry. This induces at the end of the day an additional term corresponding to the statistical distance between a sequence of N queries to $\mathcal{O}^{\mathbf{z}}(\cdot)$ and N queries to $\mathcal{O}^{\mathbf{z}}_{\text{ideal}}(\cdot)$, namely $N \times \Delta\big(\big(\mathbf{rG}^\top,\langle \mathbf{r},\mathbf{t}\rangle\big),(\mathbf{a},e)\big)$.

Oracle $\mathcal{O}^{\mathbf{z}}(x)$:
  Input: $x \in \mathbb{R}$
    Sample: $\mathbf{r} \leftarrow \mathrm{Ber}(2^x\omega_0)^{\otimes n}$
  Return: $\left(\mathbf{r}\mathbf{G}^\top, \langle \mathbf{y}+\mathbf{z}, \mathbf{r}\rangle\right)$

Oracle $\mathcal{O}^{\mathbf{z}}_{\mathrm{ideal}}(x)$:
  Input: $x \in \mathbb{R}$
    Sample $\mathbf{r} \leftarrow \mathrm{Ber}(2^x\omega_0)^{\otimes n}$
      and $\mathbf{a} \leftarrow \mathbb{F}_2^k$
  Return: $(\mathbf{a}, \langle \mathbf{a}, \mathbf{m}\rangle + \langle \mathbf{z}+\mathbf{t}, \mathbf{r}\rangle)$

Figure 5.3: Oracles $\mathcal{O}^{\mathbf{z}}(x)$ and $\mathcal{O}^{\mathbf{z}}_{\mathrm{ideal}}(x)$

Let us introduce the following function

$$p : x \in \mathbb{R} \longmapsto \mathbb{P}\left(\mathcal{A}(\mathcal{O}^{\mathbf{0}}_{\mathrm{ideal}}(x)) = 1\right) \tag{5.10}$$

i.e. the acceptance probability of $\mathcal{A}$ when being fed with the ideal oracle, and recall that $(\mathbf{v}_i)_{1\leqslant i\leqslant n}$ denotes the canonical basis of $\mathbb{F}_2^n$. From Lemma 5.3, we notice that

$$\begin{aligned} p\left(x + \log\frac{|\mathbf{t}+\mathbf{v}_i|}{|\mathbf{t}|}\right) &= \mathbb{P}\left(\mathcal{A}\left(\mathcal{O}^{\mathbf{0}}_{\mathrm{ideal}}\left(x + \log\frac{|\mathbf{t}+\mathbf{v}_i|}{|\mathbf{t}|}\right)\right) = 1\right) \\ &= \mathbb{P}\left(\mathcal{A}\left(\mathcal{O}^{\mathbf{v}_i}_{\mathrm{ideal}}(x)\right) = 1\right) \end{aligned} \tag{5.11}$$

where the last equality follows from the fact that $\mathcal{O}^{\mathbf{0}}_{\mathrm{ideal}}\left(x + \log\frac{|\mathbf{t}+\mathbf{v}_i|}{|\mathbf{t}|})\right)$ outputs proper LPN samples with Bernoulli noise of parameter

$$2^{x+\log\frac{|\mathbf{t}+\mathbf{v}_i|}{|\mathbf{t}|}}\omega_0|\mathbf{t}| = 2^x\omega_0\,|\mathbf{t}+\mathbf{v}_i|\,.$$

In other words, the probability that $\mathcal{A}$ outputs 1 when fed with $\mathcal{O}^{\mathbf{v}_i}_{\mathrm{ideal}}(x)$ is the probability that $\mathcal{A}$ outputs 1 when fed with $\mathcal{O}^{\mathbf{0}}_{\mathrm{ideal}}$ on x shifted by

$$\log\left(\frac{|\mathbf{t}+\mathbf{v}_i|}{|\mathbf{t}|}\right) = \begin{cases} \log(1+1/t) > 0 & \text{if } i \notin \mathrm{Supp}(\mathbf{t}) \\ \leqslant 0 & \text{if } i \in \mathrm{Supp}(\mathbf{t}). \end{cases} \tag{5.12}$$

Remark 5.6. *In practice, we will mostly consider codes defined over the finite field $\mathbb{F}_2$, and in that case the discrepancy between the two situations is clearer since we always have*

$$\log\left(\frac{|\mathbf{t}+\mathbf{v}_i|}{|\mathbf{t}|}\right) = \log(1-1/t) < 0 \text{ when } i \in \mathrm{Supp}(\mathbf{t}),$$

but the approach is actually more general and can be applied for larger finite fields $\mathbb{F}_q$ where this equality is not true anymore. In fact, when $i \in \mathrm{Supp}(\mathbf{t})$, then adding $\mathbf{v}_i$ will most likely not *change the error distribution compared to $\mathcal{O}^{\mathbf{0}}_{\mathrm{ideal}}$.*

Remark 5.7. *Having introduced the ideal oracles is crucial here, since (5.11) would not necessarily hold with the real oracles. Indeed, outputs $(\mathbf{a}, b)$ of $\mathcal{O}^{\mathbf{0}}(x)$* both *have a distribution which depends on x. Hence, changing x in the real oracle $\mathcal{O}^{\mathbf{0}}(\cdot)$ might change the distribution of the first component as well.*

As previously hinted, the core idea of the reduction is to feed $\mathcal{A}$ with oracles $\mathcal{O}^{\mathbf{0}}_{\mathrm{ideal}}(x)$ and $\mathcal{O}^{\mathbf{v}_i}_{\mathrm{ideal}}(x)$, making x varying over $[0, +\infty)$ and decide whether $i \in \mathrm{Supp}(\mathbf{t})$ or not. More pre-

cisely, we rely on statistical estimations of this probability when x ranges over a *discretisation* of $[0, X_{\max}]$ for some upper bound $X_{\max}$ to be chosen later. A trade-off should be made between $X_{\max}$ and the discretisation step. Indeed, for the empirical estimation to be close enough to the actual probability function p, we need to choose $X_{\max}$ as large as possible, and the discretisation step should be small enough. On the other hand, the running time of our algorithm is directly affected by those choices, which need to be made so that the running time is still polynomial in the input (*i.e.* polynomial in the length n of the code generated by $\mathbf{G}$). In particular, $X_{\max}$ should not be too large, and the step should not be too small either.

Let $\mathcal{O}(\omega)$ be the LPN(ω) oracle from (5.2). In other words, $\mathcal{A}$ can distinguish $\mathcal{O}(\omega_0 t) = \mathcal{O}^{\mathbf{0}}_{\text{ideal}}(0)$ from $\mathcal{O}(\infty) = \mathcal{O}^{\mathbf{0}}_{\text{ideal}}(\infty)$. The goal of the reduction is to determine the first input x which induces a change in the acceptance behaviour of $\mathcal{A}$. We will compare both values and depending which one is bigger, we will decide if $i \in \text{Supp}(\mathbf{t})$ or not.

The following technical lemma shows how two oracles depending on a parameter x can be distinguished if the acceptance probability of one is a shift of the other. The proof is essentially the same as that of [PRS17].

Lemma 5.8

Let $s_1, s_2 \in \mathbb{R}$ and $p : \mathbb{R} \to [0,1]$. We suppose that there exists $\alpha > 0$ and $p_\infty \in [0,1]$ such that p verifies the following assumptions

(*i*) $p(s_1) - p_\infty \geqslant \frac{1}{\alpha}$;

(*ii*) $\forall x \in \mathbb{R}_+$, $|p(x) - p_\infty| \leqslant \alpha 2^{-\frac{x}{\alpha}}$;

(*iii*) p is α-lipschitz.

Let $\mathcal{O}_{s_1}$ and $\mathcal{O}_{s_2}$ be two oracles that output 0 or 1 and such that

$$\forall x \in \mathbb{R}, \quad \mathbb{P}\left(\mathcal{O}_{s_1}(x) = 1\right) = p(s_1 + x) \quad \text{and} \quad \mathbb{P}\left(\mathcal{O}_{s_2}(x) = 1\right) = p(s_2 + x).$$

We suppose that a call to one of the above oracle costs a time T. Furthermore, s_1 and s_2 are such that

$$\text{either} \quad \text{(I)} \quad s_1 \leqslant s_2 \quad \text{or} \quad \text{(II)} \quad s_1 \geqslant s_2 + \frac{1}{\alpha}.$$

Then, there exists an algorithm, running in time $T\,\mathsf{poly}(\alpha)$, taking as inputs $(\mathcal{O}_{s_1}, \mathcal{O}_{s_2})$, querying them $\mathsf{poly}(\alpha)$ times and which can decide whether (I) or (II) holds, with a success probability $\geqslant 1 - e^{-\alpha}$ (over the outputs of the oracles $\mathcal{O}_{s_i}$'s).

Proof. The fundamental idea of the proof is to introduce the following function

$$h(s) \stackrel{\text{def}}{=} \max_{x \geqslant 0} (1 + x)\,|p(s + x) - p_\infty|\,.$$

Estimating this function thanks to the oracles $\mathcal{O}_{s_1}$ and $\mathcal{O}_{s_2}$ (by using classical statistical methods) will discriminate both considered cases, namely if $s_1 \leqslant s_2$ or $s_1 \geqslant s_2 + \frac{1}{\alpha}$. We will show in the second part of the proof how to estimate h evaluated at s_1 and s_2. Let us first show that h discriminates cases $s_1 \leqslant s_2$ or $s_1 \geqslant s_2 + \frac{1}{\alpha}$.

- Case $s_1 \leqslant s_2$: we have

$$h(s_2) \leqslant h(s_1). \tag{5.13}$$

By definition,

$$\begin{aligned} h(s_2) &= \max_{x \geqslant 0} (1+x)\,|p(s_2+x) - p_\infty| \\ &= \max_{x \geqslant 0} (1+x)\,|p(s_1 + (x + s_2 - s_1)) - p_\infty| \\ &\leqslant \max_{x \geqslant 0} (1 + (x + s_2 - s_1))\,|p(s_1 + (x + s_2 - s_1)) - p_\infty| \\ &= \max_{y \geqslant s_2 - s_1 \geqslant 0} (1+y)\,|p(s_1+y) - p_\infty|, \end{aligned}$$

which shows Equation (5.13).

- Case $s_1 \geqslant s_2 + \frac{1}{\alpha}$: we have

$$h(s_1) < h(s_2) - P\left(\frac{1}{\alpha}\right). \tag{5.14}$$

for some polynomial P. For this case let us define

$$\widehat{x}(s) \stackrel{\text{def}}{=} \min \operatorname*{argmax}_{x \geqslant 0} (1+x)\,|p(s+x) - p_\infty|.$$

It is the smallest value $x \in [0, +\infty)$ at which h reaches its maximum. Notice that

$$\begin{aligned} h(s_1) &= (1 + \widehat{x}(s_1))\,|p(s_1 + \widehat{x}(s_1)) - p_\infty| \\ &\leqslant (1 + \widehat{x}(s_1) + s_1 - s_2)\,|p(s_1 + \widehat{x}(s_1)) - p_\infty| \\ &= (1 + (\widehat{x}(s_1) + s_1 - s_2))\,|p(s_2 + (\widehat{x}(s_1) + s_1 - s_2)) - p_\infty| \\ &\leqslant h(s_2), \end{aligned}$$

where in the last line we used that $\widehat{x}(s_1) + s_1 - s_2 \geqslant 0$ as $\widehat{x}(s_1) \geqslant 0$ and $s_1 - s_2 \geqslant 0$ by assumption. Therefore, from the above last inequality,

$$\begin{aligned} h(s_2) &\geqslant (1 + (\widehat{x}(s_1) + s_1 - s_2))\,|p(s_2 + (\widehat{x}(s_1) + s_1 - s_2)) - p_\infty| \\ &= \left(1 + \frac{s_1 - s_2}{1 + \widehat{x}(s_1)}\right) h(s_1), \end{aligned}$$

which shows that

$$h(s_2) - h(s_1) \geqslant \frac{s_1 - s_2}{1 + \widehat{x}(s_1)} h(s_1) \geqslant \frac{1}{\alpha}\,\frac{h(s_1)}{1 + \widehat{x}(s_1)}. \tag{5.15}$$

But,

$$h(s_1) = \max_{x \geqslant 0} (1+x)\,|p(s_1+x) - p_\infty| \geqslant |p(s_1) - p_\infty| \geqslant \frac{1}{\alpha}, \tag{5.16}$$

where in the last inequality we used assumption *(i)* on p. Plugging this in (5.15) leads to

$$h(s_2) - h(s_1) \geqslant \frac{1}{\alpha^2} \frac{1}{1+\widehat{x}(s_1)}. \tag{5.17}$$

Let us now bound $\widehat{x}(s_1)$ from above. We have by assumption *(ii)* about p,

$$\begin{aligned} h(s_1) &= (1+\widehat{x}(s_1))\,|p(s_1+\widehat{x}(s_1)) - p_\infty| \\ &\leqslant \alpha(1+\widehat{x}(s_1))2^{-\frac{s_1+\widehat{x}(s_1)}{\alpha}} \\ &\leqslant \alpha(1+\widehat{x}(s_1))2^{-\frac{\widehat{x}(s_1)}{\alpha}} \end{aligned}$$

but $h(s_1) \geqslant \frac{1}{\alpha}$ according to (5.16). Therefore,

$$-\frac{\widehat{x}(s_1)}{\alpha} + \log(1+\widehat{x}(s_1)) \geqslant \log\left(\frac{1}{\alpha^2}\right) \Longrightarrow \widehat{x}(s_1) \leqslant 2\alpha\log(\alpha) + \alpha\log(1+\widehat{x}(s_1)).$$

Consequently,

$$\widehat{x}(s_1) \leqslant C\alpha\log(\alpha),$$

for some constant C. Plugging this in (5.17) shows (5.14).

To summarise, considering cases $s_1 \leqslant s_2$ or $s_1 \geqslant s_2 + \frac{1}{\alpha}$, we have $h(s_2) \leqslant h(s_1)$ or $h(s_2) > h(s_1) + P\left(\frac{1}{\alpha}\right)$. Therefore, in order to distinguish both cases it is enough to find an approximation of $h(s_1)$ and $h(s_2)$ (by at most a $P\left(\frac{1}{\alpha}\right)/2$ factor). However, one may wonder how to find these estimations, since s_1 and s_2 are unknown. Recall that we have access to the oracles $\mathcal{O}_{s_1}$ and $\mathcal{O}_{s_2}$ which are such that

$$\mathbb{P}\left(\mathcal{O}_{s_i}(x) = 1\right) = p(s_i + x).$$

The idea of the proof is then to estimate $\mathbb{P}\left(\mathcal{O}_{s_i}(x) = 1\right)$ by running $\mathcal{O}_{s_i}(x)$ many times (one call to it costs a time T) and repeating this process for many different values of x. It will give an approximation of the graph of the map $x \mapsto p(s_i + x)$ and therefore an estimate of $h(s_i)$. All of this can be achieved by using the most basic statistical tool: empirical estimators of the expectation. The procedure is described in Algorithm 5.8.

Algorithm 5.8 : Estimator of $h(s_i)$

Parameters : N_{iter}, $x_{\max}$ and δ
Input : $\mathcal{O}_{s_i}$
Output : $\overline{h}(s_i) \in \mathbb{R}_+$ be the estimation of $h(s_i)$

for $j = 0, \dots, X_{\max} \overset{def}{=} \lfloor \frac{x_{\max}}{\delta} \rfloor$ **do**
  $\overline{p}(j) = 0$
  for $\ell = 0, \dots, N_{\text{iter}} - 1$ **do**
    $b = \mathcal{O}_{s_i}(\delta j)$
    $\overline{p}(j) = \overline{p}(j) + \frac{b}{N_{\text{iter}}}$ ▷ We compute here the empirical value of $\mathbb{P}(\mathcal{O}_{s_i}(\delta j) = 1)$
return $\overline{h}(s_i) = \max_j (1 + \delta j) \, |\overline{p}(j) - \overline{p}(X_{\max})|$

Parameters N_{iter}, $x_{\max}$ and δ of Algorithm 5.8 will be chosen later. Notice that one call to this algorithm costs a time given by

$$X_{\max} \, N_{\text{iter}} \, T, \text{ where, } \quad \left(X_{\max} \overset{\text{def}}{=} \left\lfloor \frac{x_{\max}}{\delta} \right\rfloor \right),$$

as one call to $\mathcal{O}_{s_i}$ costs T. Furthermore, $X_{\max} N_{\text{iter}}$ is the number of queries to the oracle $\mathcal{O}_{s_i}$. Our aim is to show that for well chosen parameters

$$\mathbb{P}\left(|\overline{h}(s_i) - h(s_i)| \geqslant \frac{P\left(\frac{1}{\alpha}\right)}{2} \right) \leqslant e^{-\alpha}, \tag{5.18}$$

where the probability is computed over $\mathcal{O}_{s_i}$ which is itself used to compute $\overline{h}(s_i)$. This will conclude the proof, since in the case $s_1 \leqslant s_2$ we will have $\overline{h}(s_1) - \overline{h}(s_2) \geqslant -P\left(\frac{1}{\alpha}\right)$ and in the other case $\overline{h}(s_1) - \overline{h}(s_2) < -P\left(\frac{1}{\alpha}\right)$ with probability $\geqslant 1 - e^{-\alpha}$.

To prove this statement, let us first show that for all $\chi \in [0,1]$ and all $j \in \{0, \dots, X_{\max}\}$,

$$\left| \, |\overline{p}(j) - \overline{p}(X_{\max})| - |(p(s_i + (j+\chi)\delta) - p_\infty)| \, \right| \leqslant 2Y + \alpha 2^{-\frac{s_i + X_{\max}\delta}{\alpha}} \tag{5.19}$$

holds with probability larger than $1 - 3X_{\max} e^{-2N_{\text{iter}} Y^2}$.

Notice that $\overline{p}(j)$ is the empirical expectation of $\mathcal{O}_{s_i}(\delta j)$ where

$$\mathbb{E}(\mathcal{O}_{s_i}(\delta j)) = \mathbb{P}(\mathcal{O}_{s_i}(\delta j) = 1) = p(s_i + \delta j)$$

(the oracle only outputs 1 or 0). Therefore, by Chernoff's bound, for some $Y \geqslant 0$, we have

$$\begin{aligned} \mathbb{P}(|\overline{p}(j) - p(s_i + \delta j)| \geqslant Y) &= \mathbb{P}(|\overline{p}(j) - \mathbb{P}(\mathcal{O}_{s_i}(\delta j) = 1)| \geqslant Y) \\ &\leqslant 2e^{-2N_{\text{iter}} Y^2}. \end{aligned} \tag{5.20}$$

Next, from the union bound,

$$\mathbb{P}(\forall j \in \{0, \dots, X_{\max}\}, \ |\overline{p}(j) - p(s_i + \delta j)| \leqslant Y) \geqslant 1 - 2X_{\max} \, e^{-2N_{\text{iter}} Y^2}. \tag{5.21}$$

Let us now make the following computation for $\chi \in [0,1]$,

$$\begin{aligned}|\overline{p}(j) - p(s_i + (j+\chi)\delta)| &\leqslant |\overline{p}(j) - p(s_i + j\delta)| + |p(s_i + j\delta) - p(s_i + (j+\chi)\delta)| \\ &\leqslant |\overline{p}(j) - p(s_i + j\delta)| + \alpha\delta,\end{aligned}$$

where in the last line we used assumption *(iii)* that p is α-lipschitz together with $\chi \in [0,1]$. According to Equation (5.21),

$$\begin{aligned}\mathbb{P}\Big(\forall j \in \{0,\dots,X_{\max}\},\ |\overline{p}(X_{\max}) - p(s_i + (j+\chi)\delta)| \leqslant Y + \alpha\delta\Big) \\ \geqslant 1 - 2X_{\max}\, e^{-2N_{\text{iter}}Y^2}.\end{aligned} \tag{5.22}$$

Furthermore,

$$\begin{aligned}|\overline{p}(X_{\max}) - p_\infty| &\leqslant |\overline{p}(X_{\max}) - p(s_i + X_{\max}\delta)| + |p(s_i + X_{\max}\delta) - p_\infty| \\ &\leqslant |\overline{p}(X_{\max}) - p(s_i + X_{\max}\delta)| + \alpha 2^{-\frac{s_i + x_{\max}}{\alpha}},\end{aligned}$$

where we used assumption *(ii)* on p. According to (5.20),

$$\mathbb{P}\left(|\overline{p}(X_{\max}) - p_\infty| \leqslant Y + \alpha 2^{-\frac{s_i + x_{\max}}{\alpha}}\right) \geqslant 1 - 2e^{-2N_{\text{iter}}Y^2}. \tag{5.23}$$

Notice now by triangle inequalities,

$$\begin{aligned}\Big|\ |\overline{p}(j) - \overline{p}(X_{\max})| - |(p(s + (j+\chi)\delta) - p_\infty)|\ \Big| \\ \leqslant |\overline{p}(j) - \overline{p}(X_{\max}) - (p(s + (j+\chi)\delta) - p_\infty)| \\ \leqslant |\overline{p}(j) - p(s + (j+\chi)\delta)| + |\overline{p}(X_{\max}) - p_\infty|.\end{aligned}$$

Therefore, combining the union bound with (5.22) and (5.23) leads to our claim given in (5.19). Let us define now,

$$\overline{q}(j) \overset{\text{def}}{=} (1 + j\delta)\,|\overline{p}(j) - \overline{p}(X_{\max})| \quad \text{and} \quad q(x) \overset{\text{def}}{=} (1+x)\,|p(s_i + x) - p_\infty|.$$

We have the following computation,

$$\begin{aligned}&|\overline{q}(j) - q((j+\chi)\delta)| \\ &\quad = \Big|(1+j\delta)\,|\overline{p}(j) - \overline{p}(X_{\max})| - (1 + (j+\chi)\delta)\,|p(s_i + (j+\chi)\delta) - p_\infty|\Big| \\ &\quad \leqslant (1+j\delta)\Big|\ |\overline{p}(j) - \overline{p}(X_{\max})| - |p(s_i + (j+\chi)\delta) - p_\infty|\ \Big| \\ &\qquad + \chi\delta\,|p(s_i + (j+\chi)\delta) - p_\infty| \\ &\quad \leqslant (1 + x_{\max})\Big|\ |\overline{p}(j) - \overline{p}(X_{\max})| - |p(s_i + (j+\chi)\delta) - p_\infty|\ \Big| + \alpha\delta 2^{-\frac{s_i + (j+\chi)\delta}{\alpha}},\end{aligned}$$

where in the last line, we used assumption *(ii)* on p together with $\chi \leqslant 1$. Therefore, according to (5.19), for all j and $\chi \in [0,1]$,

$$|\overline{q}(j) - q((j+\chi)\delta)| \leqslant 2(1 + x_{\max})Y + \alpha 2^{-\frac{s_i + x_{\max}}{\alpha}} + \alpha\delta \tag{5.24}$$

with probability $\geqslant 1 - 3X_{\max} e^{-2N_{\text{iter}} Y^2}$. Notice now that, by definition of $\overline{h}(s_i)$ and $h(s)$, we have

$$\begin{aligned}\left|\overline{h}(s_i) - h(s_i)\right| &\leqslant \left| \max_j \overline{q}(j) - \max_{\substack{j \in]\!]0, X_{\max}]\!] \\ \chi \in [0,1] \\ (j+\chi)\delta \leqslant x_{\max}}} q((j+\chi)\delta) \right| + \max_{t \geqslant x_{\max}} q(t) \\ &\leqslant \max_{j,\chi} |\overline{q}(j) - q((j+\chi)\delta)| + \alpha(1 + x_{\max}) 2^{-\frac{x_{\max}}{\alpha}},\end{aligned}$$

where in the last line, we used assumption *(ii)*. Therefore, by plugging (5.24) in the above equations, we have with probability $\geqslant 1 - 3X_{\max} e^{-2NY^2}$,

$$\begin{aligned}\left|\overline{h}(s_i) - h(s_i)\right| &\leqslant 2Y + \alpha 2^{-\frac{s_i + x_{\max}}{\alpha}} + \alpha\delta + \alpha(1 + x_{\max}) 2^{-\frac{x_{\max}}{\alpha}} \\ &\leqslant 2Y + 2\alpha(1 + x_{\max}) 2^{-\frac{x_{\max}}{\alpha}} + \alpha\delta.\end{aligned}$$

Now, let us choose parameters such that

$$x_{\max} = -\alpha \log \frac{P\left(\frac{1}{\alpha}\right)}{6\alpha(1 + x_{\max})}, \quad Y = \frac{P\left(\frac{1}{\alpha}\right)}{12} \quad \text{and} \quad \delta = \frac{P\left(\frac{1}{\alpha}\right)}{6\alpha}.$$

Plugging this in Equation leads to

$$\left|\overline{h}(s_i) - h(s_i)\right| \leqslant \frac{P\left(\frac{1}{\alpha}\right)}{2}.$$

Furthermore, by choosing N_{iter} as

$$-2N_{\text{iter}} Y^2 = -\alpha + \log\left(\frac{1}{3X_{\max}}\right) \iff N_{\text{iter}} = \frac{\alpha + \log(3X_{\max})}{2Y^2}.$$

Then the above inequality is true with probability $\geqslant 1 - e^{-\alpha}$. Recall that the cost of our algorithm is given by

$$X_{\max}\, N_{\text{iter}}\, T = X_{\max} \left(\frac{\alpha + \log(3X_{\max})}{P\left(\frac{1}{\alpha}\right)^2 / 72} \right) T = \mathsf{poly}(\alpha) T$$

as $X_{\max} = \left\lfloor \frac{x_{\max}}{\delta} \right\rfloor = \mathsf{poly}(\alpha)$. This concludes the proof. □

Equipped with this statement, we are almost ready to prove Theorem 5.4. However, it still remains to verify that the function p given in (5.10) satisfies the assumption of the lemma for some parameters α and p_∞.

Lemma 5.9

We use the notation of Theorem 5.4. Let p be the function defined in (5.10), and let

$$p_\infty \stackrel{\text{def}}{=} \mathbb{P}\left(\mathcal{A}\left(\mathcal{O}^{\mathbf{0}}_{\text{ideal}}(\infty)\right)=1\right) \tag{5.25}$$

Then, we have

(*i*) $p(0)-p_\infty \geqslant \frac{1}{\alpha}$;

(*ii*) $|p(x)-p_\infty| \leqslant \alpha 2^{-\frac{x}{\alpha}}$;

(*iii*) p is α-lipschitz;

with α satisfying

$$\alpha = C\max\left(\frac{1}{\varepsilon}, N, n\right) \tag{5.26}$$

for some large enough constant C and where ε is the distinguishing advantage of $\mathcal{A}$.

Proof. Let us first prove (*i*). Following the discussion in Step 1, let $\mathcal{O}(\omega) \stackrel{\text{def}}{=} \mathcal{O}^{\mathbf{0}}_{\text{ideal}}(0)$ and $\mathcal{O}(\infty) \stackrel{\text{def}}{=} \mathcal{O}^{\mathbf{0}}_{\text{ideal}}(\infty)$ (defined in (5.2)). By definition of p and the distinguishing advantage ε,

$$\begin{aligned} p(0)-p_\infty &= \mathbb{P}\left(\mathcal{A}(\mathcal{O}(\omega))=1\right) - \mathbb{P}\left(\mathcal{A}(\mathcal{O}(\infty))=1\right) \\ &= 2\varepsilon \\ &\geqslant \frac{1}{\alpha}, \end{aligned}$$

where in the last line we used the assumption on α given in Equation (5.26).

Let us prove (*ii*). Using the data processing inequality (1.2) together with (1.3), for $X \leftarrow \text{Ber}(2^x\omega_0 t)$ and $Y \leftarrow \text{Ber}(\infty)$, we have

$$\begin{aligned} |p(x)-p(\infty)| &\leqslant N\,\Delta(X,Y) \\ &= N\,2^{-2^x\omega_0 t}. \end{aligned}$$

Notice now that

$$N2^{-2^x\omega_0 t} \leqslant \alpha 2^{-\frac{x}{\alpha}} \iff \log(N) - 2^x\omega_0 t \leqslant -\frac{x}{\alpha} + \log(\alpha),$$

and the last equality is verified for all $x \geqslant 0$ since, from (5.7), we know that $\omega_0 t = \omega = \Omega(1)$ and $\alpha \geqslant CN$ for some large enough constant C. It proves item (*ii*).

We are now ready to finish the proof by proving item (*iii*). In the same manner as

before, for $X \leftarrow \mathrm{Ber}(2^x\omega_0 t)$ and $Y \leftarrow \mathrm{Ber}(2^y\omega_0 y)$ and for all $x, y \geqslant 0$, we have

$$\begin{aligned}|p(x) - p(y)| &\leqslant N\Delta(X,Y)\\ &= N\left|2^{-2^x\omega_0 t} - 2^{-2^y\omega_0 t}\right|\\ &\leqslant N\omega_0 t\,|x-y|\,,\end{aligned}$$

where the last inequality follows from the mean value theorem. Notice now that $N\omega_0 t \leqslant \alpha$ as $N\omega_0 t = N\omega = O(Nn)$. This concludes the proof. □

We are now ready to prove Theorem 5.4.[ii]

Proof of Theorem 5.4.

Let $(\mathbf{G}, \mathbf{mG}+\mathbf{t})$ be an instance of the decoding problem, and let $i \in \{1,\dots,n\}$. The goal is to decide whether $i \in \mathrm{Supp}(\mathbf{t})$ using our distinguisher $\mathcal{A}$ between the uniform, and noisy codewords.

Define the following (parameterised) oracles

$$\mathcal{O}_{s_1}(x) \stackrel{\text{def}}{=} \mathcal{A}(\mathcal{O}^{\mathbf{v}_i}_{\text{ideal}}(x)) \quad \text{and } \mathcal{O}_{s_2}(x) \stackrel{\text{def}}{=} \mathcal{A}(\mathcal{O}^{\mathbf{0}}_{\text{ideal}}(x)).$$

The idea is to run the procedure of Lemma 5.8. By definition of p,

$$\mathbb{P}(\mathcal{O}_{s_2}(x) = 1) = p(x) = p(x + s_2) \text{ with } s_2 = 0$$

and by Equations (5.11) and (5.12)

$$\mathbb{P}(\mathcal{O}_{s_1}(x) = 1) = \mathbb{P}(\mathcal{A}\left(\mathcal{O}^{\mathbf{v}_i}_{\text{ideal}}(x)\right) = 1) = p\left(x + \log\left(\frac{|\mathbf{t}+\mathbf{v}_i|}{\mathbf{t}}\right)\right) = p(x+s_1).$$

with

$$s_1 = \begin{cases} \log(1+1/t) > 0 & \text{if } i \notin \mathrm{Supp}(\mathbf{t})\\ \leqslant 0 & \text{if } i \in \mathrm{Supp}(\mathbf{t}).\end{cases}$$

Therefore, either

$$s_1 \leqslant s_2 = 0 \quad \text{if } i \in \mathrm{Supp}(\mathbf{t})$$

or,

$$s_1 - \frac{1}{t+1} = \log\left(1+\frac{1}{t}\right) - \frac{1}{t+1} \geqslant 0 = s_2 \quad \text{if } i \notin \mathrm{Supp}(\mathbf{t}).$$

i.e.

$$s_1 \geqslant s_2 + \frac{1}{t+1} \quad \text{if } i \notin \mathrm{Supp}(\mathbf{t}).$$

Consequently, to apply Lemma 5.8 it suffices to take $\alpha \geqslant t+1$. But the function p has also to verify items *(i)*, *(ii)* and *(iii)* of the lemma. According to Lemma 5.9, all these assumptions are met if we choose α as a $\Theta\left(\max\left(\frac{1}{\varepsilon}, N, n\right)\right)$ (recall that $t \leqslant n$). Notice that $\mathsf{poly}(\alpha) = \mathsf{poly}\left(\max\left(\frac{1}{\varepsilon}, N, n\right)\right)$.

Running the procedure of Lemma 5.8 for any $i \in \{1,\dots,n\}$ will output the support of $\mathbf{t}$,

[ii] The following proof slightly differs from that of [BCD23] in order to handle non binary fields. See the discussion from Section 5.2.4

namely $\{i \in \{1,\dots,n\}, t_i \neq 0\}$ with probability

$$\geqslant \left(1-e^{-\alpha}\right)^n = \left(1-e^{-\Omega(n)}\right)^n = 1-2^{-\Omega(n)}.$$

and in time $T\mathsf{poly}(\alpha)$.

However, recall that in practice, we do *not* have access to those ideal oracles, and we have to run this procedure with $\mathcal{A}(\mathcal{O}^{\mathbf{0}}(x))$ and $\mathcal{A}(\mathcal{O}^{\mathbf{v}_i}(x))$ instead. This induces a loss in the success probability corresponding to the statistical distance between the ideal and the real oracles. Recall that by Lemma 5.8, the procedure makes $\mathsf{poly}(\alpha)$ queries to the oracles $\mathcal{A}(\mathcal{O}^{\mathbf{0}}(x))$ and $\mathcal{A}(\mathcal{O}^{\mathbf{v}_i}(x))$, and for each of them, $\mathcal{A}$ makes N queries to its input oracles (to build the instance of the decisional decoding problem). In other words, the actual procedure will recover the support of $\mathbf{t}$ in the same time and with probability

$$\geqslant 1-2^{-\Omega(n)} - N\mathsf{poly}(\alpha) \max_{x\geqslant 0} \Delta\Big(\Big(\mathbf{r}(x)\mathbf{G}^\top, \langle \mathbf{r}(x), \mathbf{t}\rangle\Big), (\mathbf{a}, e(x))\Big)$$

This concludes the proof. □

5.2.4 Oracle with Hidden Support Problem

The above procedure will recover the support of the error. Mimicking the situation in lattice-based cryptography, we can tell that this procedure actually solves the following Oracle with Hidden Support Problem (OHSP), which could be formally defined as follows

Problem 5.10 (Oracle with Hidden Support)

Consider a fixed subset $S \subset \{1,\dots,n\}$ called *support*. An instance of OHSP consists in a randomised oracle $\mathcal{O} : 2^{\{1,\dots,n\}} \times \mathbb{R}_+ \to \{0,1\}$ whose acceptance probability on input (I,s) is of the form $p(s+\log(\sigma_I))$ where

$$\sigma_I \leqslant \frac{|I \cup S|}{|S|},$$

with equality when $I \cap S = \emptyset$, for some unknown function p. The goal is to output the hidden support S.

Looking closely, the above procedure actually builds iteratively the support S by deciding for any $i \in \{1,\dots,n\}$ if $i \in S$. More precisely, oracle $\mathcal{O}^{\mathbf{z}}_{\text{ideal}}(\cdot)$ from Figure 5.3 can actually be seen as an oracle with hidden support $S \stackrel{\text{def}}{=} \text{Supp}(\mathbf{t})$ and $I \stackrel{\text{def}}{=} \text{Supp}(\mathbf{z})$. For our reduction we always consider $\mathbf{z}$ to be unitary, *i.e.* I is simply the one point set $\{1\}$.

In the case of the binary field $\mathbb{F}_2$, recovering the support and recovering the actual error is a tautology, and since we focus in the binary case, this OHSP problem makes more sense for larger fields. However, for $\mathbb{F}_q$ with $q \geqslant 3$, recovering the support is not actually the same as recovering the error. Fortunately, once the support is known, it suffices to solve a linear system to find the actual error, which only induces an additional $\mathsf{poly}(n)$ factor in the running time of the reduction.

There is a little caveat, though: we need to properly define the *smoothing* distribution, *i.e.* the distribution of $\mathbf{r}$ in the oracles of Figure 5.3. In other words, to instantiate the reduction with

the Hamming metric over larger fields $\mathbb{F}_q$, there only needs to derive proper smoothing bounds for a good distribution.

Towards other metrics. This reduction shows that the OCP framework can be extended from the Euclidean metric to the Hamming metric, by basically changing the outer problem OHCP into OHSP. In both cases, we really make use of the underlying metric to solve those problems, however OCP is completely oblivious to that.

This point of view can be seen as a starting point towards search-to-decision reductions for other metrics relevant for cryptography such as the rank metric (Chapter 2) or the less studied Lee metric [Lee58] which has recently found its way into the Hash-and-sign signature scheme FuLeeca [RBKM+23] submitted to NIST second call on post-quantum signatures.

More precisely, we need to add more arrows to Figure 5.1 to get Figure 5.4.

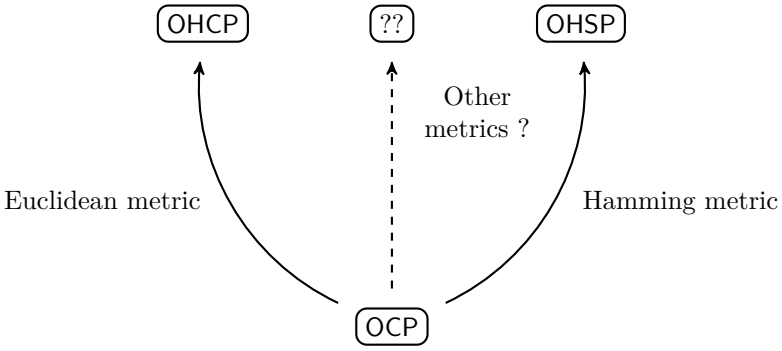


Figure 5.4: Towards reductions for other metrics ?

The most natural next metric is probably the rank metric, since the notion of (row) support completely characterises the error, as for the Hamming metric. Therefore, it might be possible to solve a rank metric version of OHSP. There is still a lot of work to do though: if the decisional decoding problem in the rank metric is obviously well defined (see for example [BBCG+18]), an intermediate problem *à la LWE/LPN* is not clear in the rank metric. Moreover, as we will see in Section 5.3 instantiating the reduction necessitates to introduce good smoothing distributions, which itself needs a rank metric analogue of smoothing bounds in the worst-case situation.

For the Lee metric, the situation is even less clear since this metric has been less studied in a cryptographic context.

5.3 Instantiations in the coding theoretic setting

In order to instantiate Theorem 5.4, we need to quantify how small is the statistical distance between the ideal LPN samples, and the LPN-like samples given by the oracles. More precisely, we want to ensure that the additional term

$$N\mathsf{poly}(\alpha)\Delta\Big(\Big(\mathbf{r}(x)\mathbf{G}^\top, \langle \mathbf{r}(x), \mathbf{t}\rangle\Big), (\mathbf{a}, e(x))\Big)$$

in Equation (5.9) is negligible, where

$$\mathbf{G} \in \mathbb{F}_2^{k\times n},\ |\mathbf{t}| = t,\ \mathbf{r}(x) \leftarrow \mathrm{Ber}(2^x\omega_0)^{\otimes n},\ e \leftarrow \mathrm{Ber}(2^x\omega_0 t).$$

In other words, we want to understand for which parameters ω_0 and x the distribution of $\mathbf{r}(x)$ *smoothes* the dual of the code generated by $\mathbf{G}$, and kills the correlation.

We will consider two situations:

Average-case to Average-case. When the instance $(\mathbf{G}, \mathbf{mG} + \mathbf{t})$ of the initial decoding problem is sampled *on average*, that is when $\mathbf{G}$ and $\mathbf{m}$ are uniformly distributed in their respective domains. In this case, the parameters can be estimated using a variation on the famous leftover hash lemma [BDKP+11] (see Lemma 5.11). This yields a completely different search-to-decision reduction than that of [FS96], and gives insight on the best possible trade-off that our reduction can offer.

Worst-case to Average-case. The OCP framework is fundamentally a tool for giving reductions in the worst-case. This is the case which has been considered in lattice-based cryptography. However, this situation is much more complex than the average-case situation, since we need to provide a reduction which works for *any* given instance, even the worst ones. The main ingredient to control the aforementioned statistical distance is the so-called smoothing bounds, which have been obtained in a series of paper [BLVW19; YZ21; DDRT23; DR22] for various smoothing distributions.

5.3.1 Average-case to Average-case Reduction

The parameters of the reduction in the average-case to average-case regime will be derived from a series of technical lemmas. We start by the following variation on the leftover hash lemma. It is similar to [DST19b, Lemma 3] which did not consider the second component, and in particular its proof follows the same path.

When computing the probability of en event $\mathcal{E}$ which may depend en multiple parameters, we put in index where we take the randomness. Recall that for an integer $\rho \in \{1, \dots, n\}$, we denote by $\mathcal{S}_\rho$ the Hamming sphere of radius ρ centred at $\mathbf{0}$.

Lemma 5.11

Consider two finite sets E and F. Let $\mathcal{H} \stackrel{\text{def}}{=} (h_i)_{i \in I}$ be a finite family of functions from E to F, and let $T \subset E$. Let $\langle \cdot, \cdot \rangle$ be a map from $E \times E$ to $\{0,1\}$, and consider the following probability

$$p \stackrel{\text{def}}{=} \mathbb{P}_{t,r}\left(\langle r, t \rangle = 1\right) \tag{5.27}$$

where $t \leftarrow T$ and r is distributed according to some distribution $\mathcal{D}$ over E.

Denote by η the "collision bias", defined as

$$\mathbb{P}_{h,t,r_0,r_1}\Big(h(r_0) = h(r_1), \langle t, r_0 \rangle = \langle t, r_1 \rangle\Big) = \frac{1}{|F|}(p^2 + (1-p)^2 + \eta) \tag{5.28}$$

where $h \leftarrow \mathcal{H}$, $t \leftarrow T$ are distributed uniformly at random in $\mathcal{H}$ and T respectively; and the r_i's are independent and distributed according to $\mathcal{D}$.

Let Y be the random variable (u, e) where $u \leftarrow F$ is uniform over F and $e \in \{0,1\}$ is a Bernoulli random variable of success probability p, independent from u. Let $Y(h,t)$ be the random variable $(h(r), \langle r, t \rangle)$ when r is distributed according to $\mathcal{D}$. Then,

$$\mathbb{E}_{h,t}\left(\Delta(Y(h,t), Y)\right) \leqslant \sqrt{\eta}. \tag{5.29}$$

Before proving this lemma, let us stop a moment here. Note that when $(u_0, e_0), (u_1, e_1)$ are independent and such that $u_i \leftarrow F$ is uniform over F and e_i is a Bernoulli of success probability p, then

$$\mathbb{P}_{u_0,e_0,u_1,e_1,t}\Big(u_0 = u_1, \langle t, r_0 \rangle = \langle t, r_1 \rangle\Big) = \frac{1}{|F|}(p^2 + (1-p)^2)$$

exactly corresponds to $\eta = 0$. In this situation, we obviously have that $Y(h,t) = Y$ and the statistical distance is 0. In other words, η quantifies how far h is from decorrelating and uniforming everything. The intuition is that when h is a good hash function, the statistical distance should be small.

For our instantiations, we will choose $E = \mathbb{F}_2^n$, $F = \mathbb{F}_2^k$ and $\mathcal{H}$ is the set of functions defined by

$$h(r) \stackrel{\text{def}}{=} r\mathbf{G}^\top,$$

where $\mathbf{G}$ ranges over a given family of $\mathbb{F}_2^{k \times n}$ matrices (with or without structure). Furthermore, $\langle \cdot, \cdot \rangle$ denotes the usual inner product of $\mathbb{F}_2^n$.

Proof of Lemma 5.11. By definition of the statistical distance we have

$$\begin{aligned}
\mathbb{E}_{h,t}\left(\Delta(Y(h,t),Y)\right) &= \sum_{\substack{h\in\mathcal{H}\\ t\in T}} \frac{1}{|\mathcal{H}|\cdot|T|}\,\Delta\left((h(r),\langle r,t\rangle),(u,e)\right)\\
&= \frac{1}{2}\sum_{\substack{h\in\mathcal{H}\\ t\in T}} \frac{1}{|\mathcal{H}|\cdot|T|} \sum_{\substack{f\in F\\ b\in\{0,1\}}} \left|\mathbb{P}_r(h(r)=f,\langle r,t\rangle=b) - \frac{\mathbb{P}(e=b)}{|F|}\right|\\
&= \frac{1}{2}\sum_{\substack{h,t\\ f,b}} \left|\mathbb{P}_{h_0,t_0,r}\left(h_0=h,t_0=t,h_0(r)=f,\langle r,t_0\rangle=b\right) - \frac{\mathbb{P}(e=b)}{|\mathcal{H}|\cdot|T|\cdot|F|}\right|\\
&= \frac{1}{2}\sum_{\substack{h,t\\ f,b}} \left|q_{h,t,f,b} - \frac{\mathbb{P}(e=b)}{|\mathcal{H}|\cdot|T|\cdot|F|}\right| \qquad (5.30)
\end{aligned}$$

where for $h\in\mathcal{H}, t\in T, b\in\{0,1\}$ and $f\in F$, we define

$$q_{h,t,f,b} \stackrel{\text{def}}{=} \mathbb{P}_{h_0,t_0,r}\left(h_0=h,t_0=t,h_0(r)=f,\langle r,t_0\rangle=b\right)$$

with (h_0,t_0) uniformly distributed over $\mathcal{H}\times T$ and r distributed according to $\mathcal{D}$. Using the Cauchy-Schwarz inequality, we get

$$\sum_{\substack{h,t\\ f,b}} \left|q_{h,t,f,b} - \frac{\mathbb{P}(e=b)}{|\mathcal{H}|\cdot|T|\cdot|F|}\right| \leqslant \sqrt{\sum_{\substack{h,t\\ f,b}} \left(q_{h,t,f,b} - \frac{\mathbb{P}(e=b)}{|\mathcal{H}|\cdot|T|\cdot|F|}\right)^2}\sqrt{|\mathcal{B}|\cdot|\mathcal{H}|\cdot|T|\cdot|F|}. \qquad (5.31)$$

Unfolding the computations, we get

$$\begin{aligned}
\sum_{\substack{h,t\\ f,b}} \left(q_{h,t,f,b} - \frac{\mathbb{P}(e=b)}{|\mathcal{H}|\cdot|T|\cdot|F|}\right)^2 &= \sum_{\substack{h,t\\ f,b}} \left(q^2_{h,t,f,b} - 2\mathbb{P}(e=b)\frac{q_{h,t,f,b}}{|\mathcal{H}|\cdot|T|\cdot|F|} + \frac{\mathbb{P}(e=b)^2}{|\mathcal{H}|^2\cdot|T|^2\cdot|F|^2}\right)\\
&= \sum_{\substack{h,t\\ f,b}} q^2_{h,t,f,b} - \frac{1}{|\mathcal{H}|\cdot|T|\cdot|F|}\sum_b \mathbb{P}(e=b)\left(2\sum_{\substack{h,t\\ f}} q_{h,t,f,b} - \mathbb{P}(e=b)\right) \qquad (5.32)
\end{aligned}$$

Let us now observe that

$$\begin{aligned}
\sum_{\substack{h,t\\ f}} q_{h,t,f,b} &= \sum_{\substack{h,t\\ f}} \mathbb{P}_{h_0,t_0,r}\left(h_0=h,t_0=t,h_0(r)=f,\langle r,t_0\rangle=b\right)\\
&= \mathbb{P}_{t_0,r}\left(\langle r,t_0\rangle=b\right)\\
&= \mathbb{P}(e=b)
\end{aligned}$$

where in the last line we used the fact that e is exactly a Bernoulli random variable with success probability p (defined in Equation (5.27)). Plugging this in Equation (5.32) we

obtain

$$\sum_{\substack{h,t\\f,b}}\left(q_{h,t,f,b}-\frac{\mathbb{P}(e=b)}{|\mathcal{H}|\cdot|T|\cdot|F|}\right)^2=\sum_{\substack{h,t\\f,b}}q_{h,t,f,b}^2-\frac{\mathbb{P}(e=0)^2+\mathbb{P}(e=1)^2}{|\mathcal{H}|\cdot|T|\cdot|F|}$$
$$=\sum_{\substack{h,t\\f,b}}q_{h,t,f,b}^2-\frac{p^2+(1-p)^2}{|\mathcal{H}|\cdot|T|\cdot|F|} \tag{5.33}$$

Consider now for $i\in\{0,1\}$ independent random variables h_i, t_i and r_i that are drawn uniformly at random in $\mathcal{H}$, T and according to $\mathcal{D}$ respectively. We continue this computation by noticing now that

$$\begin{aligned}\sum_{h,t,f,b}q_{h,t,f,b}^2&=\sum_{h,f}\mathbb{P}_{h_0,t_0,r_0}\left(h_0=h,t_0=t,h_0(r)=f,\langle r_0,t_0\rangle=b\right)\\&\qquad\mathbb{P}_{h_1,t_1,r_1}\left(h_1=h,t_1=t,,h_1(r_1)=f,\langle r_1,t_1\rangle=b\right)\\&=\mathbb{P}_{h_0,h_1,t_0,t_1,r_0,r_1}\left(h_0=h_1,t_0=t_1,h_0(r_0)=h_1(r_1),\langle t_0,r_0\rangle=\langle t_1,r_1\rangle\right)\\&=\frac{\mathbb{P}_{h_0,t_0,r_0,r_1}\left(h_0(r_0)=h_0(r_1),\langle t_0,r_0\rangle=\langle t_0,r_1\rangle\right)}{|\mathcal{H}|\cdot|T|}\\&\leqslant\frac{p^2+(1-p)^2+\eta}{|\mathcal{H}|\cdot|T|\cdot|F|}.\end{aligned} \tag{5.34}$$

where in the last line we used the definition of η given in Equation (5.28). By substituting for $\sum_{h,f}q_{h,f}^2$ the expression obtained in (5.34) into (5.33) and then back into (5.31) we finally obtain

$$\begin{aligned}\sum_{\substack{h,t\\f,b}}\left|q_{h,t,f,b}-\frac{\mathbb{P}(e=0)^2+\mathbb{P}(e=1)^2}{|\mathcal{H}|\cdot|T|\cdot|F|}\right|&\leqslant\sqrt{\frac{p^2+(1-p)^2+\eta}{|\mathcal{H}|\cdot|T|\cdot|F|}-\frac{p^2+(1-p)^2}{|\mathcal{H}|\cdot|T|\cdot|F|}}\sqrt{2\,|\mathcal{H}|\cdot|T|\cdot|F|}\\&=\sqrt{\frac{\eta}{|\mathcal{H}|\cdot|T|\cdot|F|}}\sqrt{2\,|\mathcal{H}|\cdot|T|\cdot|F|}\\&=\sqrt{2\eta}.\end{aligned}$$

which concludes the proof. □

With Lemma 5.11 in hand, we can determine a set of parameters useful for the reduction.

Lemma 5.12 (Parameters for the average to average reduction)

Let $\beta,\eta\in(0,1)$, $k\leqslant n\in\mathbb{N}$, $t\in\{1,\dots,n\}$ and $\omega_0\in\mathbb{R}_+$ be such that

$$\omega_0\geqslant-\log_2\left(1-2\,\frac{1+\eta}{1-\beta}\,h^{-1}\left(\frac{k}{n}\right)\right) \tag{5.35}$$

where $h^{-1}:[0,1]\to[0,\frac{1}{2}]$ denotes the inverse of the binary entropy function h. Then, for

all $x \geqslant 0$,

$$\mathbb{E}_{\mathbf{G},\mathbf{t}}\left(\Delta\left(\left(\mathbf{r}(x)\mathbf{G}^\top, \langle \mathbf{r}(x), \mathbf{t}\rangle\right), (\mathbf{a}, e(x))\right)\right) = 2^{-\Omega(n)}$$

where $\mathbf{a} \leftarrow \mathbb{F}_2^k$, $\mathbf{r}(x) \leftarrow \mathrm{Ber}(2^x\omega_0)^{\otimes n}$, $e(x) \leftarrow \mathrm{Ber}(2^x\omega_0 t)$, $\mathbf{G} \leftarrow \mathbb{F}_2^{k\times n}$ and $\mathbf{t}$ is uniformly distributed amongst the words of $\mathbb{F}_2^n$ of Hamming weight t.

The proof of Lemma 5.12 will proceed in two steps: first, we begin to prove that a similar result holds when $\mathbf{r}$ is replaced by the uniform distribution over the words of weight $r(x) \stackrel{\text{def}}{=} \frac{n}{2}\left(1-2^{-2^x\omega_0}\right)(1-\beta)$; and then apply the following proposition from [DR22] which shows that the Bernoulli distribution inherits the smoothing properties of the uniform distribution over a Hamming sphere.

Proposition 5.13 ([DR22, Proposition 6.7])

Let $\mathbf{t} \in \mathbb{F}_2^n$, $\beta > 0$, $\omega \in \mathbb{R}_+$ and $p \stackrel{\text{def}}{=} \frac{1}{2}\left(1-2^{-\omega}\right)$. Let $\mathbf{G} \in \mathbb{F}_2^{k\times n}$ be the generator matrix of an $[n,k]$-code. Then,

$$\Delta\left(\left(\mathbf{r}\mathbf{G}^\top, \langle \mathbf{r}, \mathbf{t}\rangle\right), (\mathbf{a}, e)\right) \leqslant \sum_{r=(1-\beta)np}^{(1+\beta)np} \Delta\left(\left(\mathbf{r}_r\mathbf{G}^\top, \langle \mathbf{r}_r, \mathbf{t}\rangle\right), (\mathbf{a}, e_r)\right) + 2^{-\Omega(n)}$$

where $\mathbf{r} \leftarrow \mathrm{Ber}(\omega)^{\otimes n}$, $\mathbf{a} \leftarrow \mathbb{F}_2^k$, $e \leftarrow \mathrm{Ber}(\omega|\mathbf{t}|)$, $\mathbf{r}_r \leftarrow \mathcal{S}_r$ and the e_r's are distributed as the $\langle \mathbf{r}_r, \mathbf{t}\rangle$'s.

Remark 5.14. *Equation (5.35) is equivalent to*

$$\frac{r}{n} \stackrel{def}{=} \frac{1}{2}\left(1-2^{-\omega_0}\right)(1-\beta) \geqslant (1+\eta)h^{-1}\left(\frac{k}{n}\right). \tag{5.36}$$

That is to say, we require the least index in the sum in Proposition 5.13 to be above the Gilbert-Varshamov bound. This is a necessary condition for the statistical distances to be negligible. In other words, those bounds are the best possible we can achieve with this kind of reduction.

Proof of Lemma 5.12. In order to ease the reading, let us drop the dependency in x (the maximum of the statistical distance is reached for $x = 0$; taking $x \geqslant 0$ can only decrease this statistical distance as it increases the noise). Let $r \stackrel{\text{def}}{=} \frac{n}{2}\left(1-2^{-\omega_0}\right)(1-\beta)$ and $\mathbf{r} \leftarrow \mathcal{S}_r$. Our aim is to show that the result holds for this distribution. By Lemma 5.11, it suffices to

compute the following collision probability (where $\mathbf{r}_0, \mathbf{r}_1 \leftarrow \mathcal{S}_r$, $\mathbf{G} \leftarrow \mathbb{F}_2^{k \times n}$ and $\mathbf{t} \leftarrow \mathcal{S}_t$)

$$\begin{aligned}
&\mathbb{P}_{\mathbf{r}_0,\mathbf{r}_1,\mathbf{G},\mathbf{t}}\left(\mathbf{r}_0\mathbf{G}^\top = \mathbf{r}_1\mathbf{G}^\top, \langle \mathbf{t}, \mathbf{r}_0\rangle = \langle \mathbf{t}, \mathbf{r}_1\rangle\right) \\
&= \mathbb{P}_{\mathbf{r}_0,\mathbf{r}_1,\mathbf{G},\mathbf{t}}\left((\mathbf{r}_0 - \mathbf{r}_1)\,\mathbf{G}^\top = \mathbf{0}, \langle \mathbf{t}, \mathbf{r}_0 - \mathbf{r}_1\rangle = 0\right) \\
&= \sum_{\mathbf{r}\neq\mathbf{0}} \mathbb{P}_{\mathbf{G}}\left(\mathbf{r}\mathbf{G}^\top = \mathbf{0}\right) \mathbb{P}_{\mathbf{t}}\left(\langle \mathbf{t}, \mathbf{r}\rangle = 0\right) \mathbb{P}_{\mathbf{r}_0,\mathbf{r}_1}\left(\mathbf{r}_0 - \mathbf{r}_1 = \mathbf{r}\right) + \mathbb{P}_{\mathbf{r}_0,\mathbf{r}_1}\left(\mathbf{r}_0 = \mathbf{r}_1\right) \\
&= \frac{1}{2^k} \sum_{\mathbf{r}\neq\mathbf{0}} \mathbb{P}_{\mathbf{t}}\left(\langle \mathbf{t}, \mathbf{r}\rangle = 0\right) \mathbb{P}_{\mathbf{r}_0,\mathbf{r}_1}\left(\mathbf{r}_0 - \mathbf{r}_1 = \mathbf{r}\right) + \mathbb{P}_{\mathbf{r}_0,\mathbf{r}_1}\left(\mathbf{r}_0 = \mathbf{r}_1\right) \\
&\leqslant \frac{1}{2^k}\left(\mathbb{P}_{\mathbf{t},\mathbf{r}_0,\mathbf{r}_1}\left(\langle \mathbf{t}, \mathbf{r}_0 - \mathbf{r}_1\rangle = 0\right) + 2^k \mathbb{P}_{\mathbf{r}_0,\mathbf{r}_1}\left(\mathbf{r}_0 = \mathbf{r}_1\right)\right) \\
&= \frac{1}{2^k}\left(p^2 + (1-p)^2 + \frac{2^k}{\binom{n}{r}}\right)
\end{aligned}$$

where $p \stackrel{\text{def}}{=} \mathbb{P}_{\mathbf{r},\mathbf{t}}\left(\langle \mathbf{t}, \mathbf{r}\rangle = 1\right)$ and we used in the inequality the law of total probability. By Lemma 5.11,

$$\mathbb{E}_{\mathbf{G},\mathbf{t}}\left(\Delta\left(\left(\mathbf{r}_r\mathbf{G}^\top, \langle \mathbf{r}_r, \mathbf{t}\rangle\right), (\mathbf{a}, e_r)\right)\right) \leqslant \sqrt{\frac{2^k}{\binom{n}{r}}}.$$

Recall that $\binom{n}{r} = 2^{nh(r/n)(1+o(1))}$ where h denotes the binary entropy function. By Equation (5.36), r verifies $(1+\eta)h^{-1}\left(\frac{k}{n}\right) \leqslant \frac{r}{n} \leqslant 1/2$. Therefore, since h is a strictly increasing function, the above upper-bound is a $2^{-\Omega(n)}$. This yields the claimed result. □

We can now give the full reduction, in the average-case to average-case situation. Recall that in Theorem 5.4, the considered (search) decoding problem is fixed once and for all. However, the above lemma tells us that, on average over the choice of $\mathbf{G}$ and $\mathbf{t}$, the considered statistical distance is negligible. In reality, we can prove that it holds for *almost* all choices by a simple application of Markov's inequality.

Lemma 5.15

Let $k \leqslant n \in \mathbb{N}, t \in \{0,\dots,n\}, \omega_0 \in \mathbb{R}_+$. For a matrix $\mathbf{G} \in \mathbb{F}_2^{k\times n}$ and a vector $\mathbf{t} \in \mathbb{F}_2^n$ of Hamming weight t, let

$$X(\mathbf{G},\mathbf{t}) \stackrel{\text{def}}{=} \Delta\Big(\Big(\mathbf{r}\mathbf{G}^\top, \langle \mathbf{r},\mathbf{t}\rangle\Big), (\mathbf{a}, e)\Big), \quad \text{where } \mathbf{r} \leftarrow \mathrm{Ber}(\omega_0 t)^{\otimes n} \text{ and } e \leftarrow \mathrm{Ber}(\omega_0 t).$$

For independent and uniformly distributed $\mathbf{G}_u \leftarrow \mathbb{F}_2^{k\times n}$ and $\mathbf{t}_u \leftarrow \mathcal{S}_t$, denote by γ the following expected value

$$\gamma \stackrel{\text{def}}{=} \mathbb{E}_{\mathbf{G}_u,\mathbf{t}_u}\left(X(\mathbf{G}_u,\mathbf{t}_u)\right).$$

Then,

$$\frac{\left|\{(\mathbf{G},\mathbf{t}) \in \mathbb{F}_2^{k\times n} \times \mathcal{S}_t \mid X(\mathbf{G},\mathbf{t}) \geqslant \sqrt{\gamma}\}\right|}{2^{kn}\binom{n}{t}} \leqslant \sqrt{\gamma}.$$

Proof. Since $\mathbf{G}_u$ and $\mathbf{t}_u$ are independent, and uniformly distributed over their respective domains, this proportion is nothing but

$$\mathbb{P}_{\mathbf{G}_u,\mathbf{t}_u}(X(\mathbf{G}_u,\mathbf{t}_u) \geqslant \sqrt{\gamma}).$$

By Markov's inequality, we have

$$\mathbb{P}_{\mathbf{G}_u,\mathbf{t}_u}(X(\mathbf{G}_u,\mathbf{t}_u) \geqslant \sqrt{\gamma}) \leqslant \frac{\mathbb{E}_{\mathbf{G}_u,\mathbf{t}_u}(X(\mathbf{G}_u,\mathbf{t}_u))}{\sqrt{\gamma}} \leqslant \sqrt{\gamma},$$

which concludes the proof. □

We are now ready to instantiate the search-to-decision reduction in the average-case to average-case regime. However, we must be very careful, and not all parameters allowed by the previous lemmas are relevant for cryptographic applications. Indeed, we need to ensure that the decision problem is not *too hard* while the search version needs to be not *too easy*.

Discussion on the parameters. Notice that the noise of the decision decoding problem of the reduction is distributed as $\mathrm{Ber}(\omega_0 t)$ with ω_0 given in Equation (5.35). If one chooses k,n such that $\frac{k}{n} = \Theta(1)$, one would obtain a noise distributed as $\mathrm{Ber}(\omega_0 t) = \mathrm{Ber}(\Theta(t))$. In that case, it seems that we need to choose t as a $O(\log_2(n))$ to reach a noise rate $\frac{1}{2}(1-2^{-\omega_0 t}) = \frac{1}{2} - \frac{1}{\mathsf{poly}(n)}$ in the decision decoding problem. Otherwise, we would reduce the decoding problem into a decision decoding problem with a noise rate *exponentially or sub-exponentially* close to $1/2$; an *extremely hard* problem which is not very satisfying. On the other hand, choosing $t = O(\log_2(n))$ is a real disaster for the reduction: decoding a code of length n at distance $O(\log_2(n))$ can be done in polynomial time (using for instance Prange algorithm [Pra62]). That is, we would be reducing an *easy* worst-case search decoding problem to an average-case decision decoding problem; which says nothing about the hardness of the decision version. We therefore conclude that the only way to reach an error rate $\frac{1}{2}(1-2^{-\omega_0 t}) = \frac{1}{2} - \frac{1}{\mathsf{poly}(n)}$ is to decrease as much as possible ω_0 given in Equation (5.35). In particular, this leads us to choose $\frac{k}{n} = o(1)$, since in

that case $\omega_0 = -\log_2(1-o(1)) = o(1)$. More precisely, for these parameters, ω_0 verifies

$$\omega_0 = -\log_2\left(1-\Theta\left(h^{-1}\left(\frac{k}{n}\right)\right)\right) \approx \frac{1}{\log_2\left(\frac{n}{k}\right)}\,\frac{k}{n}$$

where we used the expansion $h^{-1}(\varepsilon) \underset{\varepsilon\to 0}{\approx} \frac{\varepsilon}{\log_2(1/\varepsilon)}$. Therefore, to reach the noise rate $\frac{1}{2} - \frac{1}{\mathsf{poly}(n)}$ we need to choose parameters such that

$$\frac{k}{n} = o(1) \quad \text{and} \quad \omega_0 t = \frac{1}{\log_2\left(\frac{n}{k}\right)}\,\frac{k}{n}\,t = O\left(\log_2(n)\right). \tag{5.37}$$

Notice that necessarily in the above choice of parameters, we need t to be sublinear in n, since otherwise k would be too small, allowing an exhaustive search to decode in polynomial time. Fortunately, in that case the reduction is non-trivial. The cost of Prange's algorithm [Pra62] (which is asymptotically the best known decoding algorithm when the decoding distance t is sublinear in the length of the input code, see [CS16]) is given by

$$2^{\Theta\left(t\,\frac{k}{n}\right)} = 2^{\Theta(\log_2(n)\log_2(n/k))} = n^{\Theta(\log_2(n/k))},$$

which is super-polynomial.

In what follows we focus our attention to a noise rate $\frac{1}{2} - \frac{1}{\mathsf{poly}(n)}$ in the decision problem, that is to say we propose parameters where the rate $\frac{k}{n}$ of the codes considered in the reduction verifies $\frac{k}{n} = o(1)$.

Theorem 5.16 (Average-case to average-case, search-to-decision reduction)

Let $\beta, \eta \in (0,1)$, $C > 0$ and $n, k, t \in \mathbb{N}$ be such that

$$\frac{k}{n} = o(1) \quad \text{and} \quad \frac{2}{\ln(2)}\,\frac{1+\eta}{1-\beta}\,\frac{1}{\log_2\left(\frac{n}{k}\right)}\,\frac{k}{n}\,t = C\log_2(n). \tag{5.38}$$

Furthermore, let

$$\omega_0 = -\log_2\left(1 - 2\,\frac{1+\eta}{1-\beta}\,h^{-1}\left(\frac{k}{n}\right)\right) \quad \textit{i.e.} \quad \frac{1-\beta}{2}\left(1-2^{-\omega_0}\right) = (1+\eta)h^{-1}\left(\frac{k}{n}\right). \tag{5.39}$$

Suppose that there exists an algorithm $\mathcal{A}$, with advantage $\varepsilon = \frac{1}{\mathsf{poly}(n)}$, which distinguishes in time T distributions $(\mathbf{A}, \mathbf{sA}+\mathbf{e})$ and $(\mathbf{A}, \mathbf{y})$ with

$$\mathbf{A} \leftarrow \mathbb{F}_2^{k\times n},\ \mathbf{s} \leftarrow \mathbb{F}_2^{k},\ \mathbf{y} \leftarrow \mathbb{F}_2^{n} \ \text{ and } \ \mathbf{e} \leftarrow \mathrm{Ber}\left(\omega_0 t\right)^{\otimes n}.$$

Then, there exists an algorithm running in time $T\mathsf{poly}(n)$, which takes as inputs $\mathbf{G} \in \mathbb{F}_2^{k\times n}$, $\mathbf{mG}+\mathbf{t}$ where $\mathbf{m} \in \mathbb{F}_2^k$, $\mathbf{t} \in \mathcal{S}_t^n$, and outputs $\mathbf{t}$ (or equivalently $\mathbf{m}$) with probability at least $1-2^{-\Omega(n)}$ over a uniform choice of $\mathbf{G}$ and $\mathbf{t}$.

Remark 5.17. *With the above parameter choice, we have*

$$\omega_0 t = C\log_2(n)(1+o(1))$$

i.e. the error rate in the decision *problem is*

$$\frac{1}{2}(1-2^{-\omega_0 t}) = \frac{1}{2} - \frac{1}{\mathsf{poly}(n)}.$$

Remark 5.18. *Note that in the above reduction, the length of the code used in the decisional problem is* the same *than the length of the code in the search problem.*

Proof. We use the notations of Theorem 5.4 and Lemma 5.15. Let $\mathbf{G} \leftarrow \mathbb{F}_2^{k\times n}$ and $\mathbf{t} \leftarrow \mathcal{S}_t$. Notice that, since $\frac{k}{n} = o(1)$, the following computation holds

$$\begin{aligned}\omega_0 t &= -\log_2\left(1 - 2\,\frac{1+\eta}{1-\beta}\,h^{-1}\left(\frac{k}{n}\right)\right)t \\ &= \frac{2}{\ln(2)}\,\frac{1+\eta}{1-\beta}\,h^{-1}\left(\frac{k}{n}\right)t(1+o(1)) \\ &= \frac{2}{\ln(2)}\,\frac{1+\eta}{1-\beta}\,\frac{1}{\log_2\left(\frac{n}{k}\right)}\,\frac{k}{n}\,t\,(1+o(1)),\end{aligned}$$

where we used the expansion $h^{-1}(x) = \frac{x}{\log_2(1/x)}(1+o(1))$. Therefore, by Equation (5.38), we have

$$\omega_0 t = C\log_2(n)(1+o(1)).$$

Let $\mathcal{B}$ be the algorithm given by Theorem 5.4, which outputs some $\mathbf{t}'$ in time $T\mathsf{poly}(\alpha)$, such that[a]

$$\mathbb{P}(\mathbf{t}' = \mathbf{t}) = 1 - 2^{-\Omega(n)} - n\,\mathsf{poly}(\alpha)X(\mathbf{G},\mathbf{t}),$$

where

$$X(\mathbf{G},\mathbf{t}) \stackrel{\text{def}}{=} \Delta\Big(\Big(\mathbf{r}\mathbf{G}^\top, \langle \mathbf{r},\mathbf{t}\rangle\Big), (\mathbf{a},e)\Big), \qquad \text{where } \mathbf{r} \leftarrow \mathrm{Ber}(\omega_0 t)^{\otimes n} \text{ and } e \leftarrow \mathrm{Ber}(\omega_0 t)$$

has been defined in Lemma 5.15. Equation (5.39) and Lemma 5.12 entail that

$$\mathbb{E}(X(\mathbf{G},\mathbf{t})) = 2^{-\Omega(n)}.$$

Since $\alpha = \max\left(\frac{1}{\varepsilon}, n\right) = \mathsf{poly}(n)$, Lemma 5.15 entails that $\mathcal{B}$ outputs $\mathbf{t}$ with probability $1-2^{-\Omega(n)}$ for a proportion $1-2^{-\Omega(n)}$ of the possible instances $(\mathbf{G},\mathbf{t})$. Moreover, the success probability of $\mathcal{B}$ is independent from $\mathbf{G}$ and $\mathbf{t}$. Therefore, the probability that $\mathcal{B}(\mathbf{G},\mathbf{t})$ outputs 1 will be greater than $(1-2^{-\Omega(n)})(1-2^{-\Omega(n)}) = 1-2^{-\Omega(n)}$, which concludes the proof. □

[a] We dropped the max here because it is reached when $x = 0$: the higher is the noise, the closer our distribution is from the genuine LPN.

5.3.2 Worst-case to Average-case Reduction

We will now deal with the worst-case to average-case reduction. Recall that in Theorem 5.4, we need to set the statistical distance between our produced samples and genuine LPN samples to be negligible. For a worst-case hardness we need it to be negligible for *any* fixed code, *i.e.* for *any* matrix $\mathbf{G}$. To this end we will use smoothing bounds as given in [DR22, Proposition

7.6]. However, this bound is only stated when $\mathbf{G}$ generates an $[n,k]$-code which is balanced (A similar assumption was made in [BLVW19]), *i.e.* which does not have codewords of extremely large weights.

Definition 5.19 (Balanced code)

An $[n,k]$-code is δ-balanced if its minimum distance is at least δn and all the codewords have Hamming weight at most $(1-\delta)n$. That is, for all $\mathbf{x} \in \mathcal{C} \setminus \{0\}$,

$$\delta n \leqslant |\mathbf{x}| \leqslant (1-\delta)n.$$

In the worst-case to average-case search-to-decision reduction we will restrict "worst" instances to δ-balanced codes. A natural choice for δ is given by the relative *Gilbert-Varshamov* bound $h^{-1}\left(1-\frac{k}{n}\right)$. However, for the same reasons as for the average-case to average-case reduction, in order to reach a noise rate $\frac{1}{2}-\frac{1}{\mathsf{poly}(n)}$ in the decision problem, we will choose parameters k,n so that $\frac{k}{n}=o(1)$. In order to reach a negligible statistical distance we will use the following proposition.

Proposition 5.20 ([DR22, Proposition 7.6])

Let $\mathbf{G} \in \mathbb{F}_2^{k\times n}$ be the generator matrix of an $[n,k]$-code which is δ-balanced with $\frac{1}{2} \geqslant \delta \geqslant h^{-1}\left(1-\frac{k}{n}\right) \geqslant C$ for some constant $C>0$. Let $\mathbf{t} \in \mathbb{F}_2^n$ and suppose that $\frac{|\mathbf{t}|}{n}=o(1)$.

Let $\beta,\eta>0$ and $\omega \in \mathbb{R}_+$ be such that

$$(1-\beta)\frac{1}{2}(1-2^{-\omega}) \geqslant (1+\eta)h^{-1}\left(2\frac{k}{n}+D\frac{|\mathbf{t}|}{n}\right)$$

for some large enough constant D. Then,

$$\Delta\Big(\Big(\mathbf{r}\mathbf{G}^\top, \langle \mathbf{r},\mathbf{t}\rangle\Big), (\mathbf{a},e)\Big) = 2^{-\Omega(n)}$$

where $\mathbf{r} \leftarrow \mathrm{Ber}(\omega)^{\otimes n}$, $\mathbf{a} \leftarrow \mathbb{F}_2^k$ and $e \leftarrow \mathrm{Ber}(\omega|\mathbf{t}|)$.

This proposition leads to the following instantiation of our worst-case to average-case, search-to-decision reduction:

Theorem 5.21 (Worst-case to average-case, search-to-decision reduction)

Let $\beta, \eta \in (0,1)$, $C > 0$ and $n, k, t \in \mathbb{N}$ be such that

$$\frac{k}{n} = o(1), \quad \frac{t}{n} = o\left(\frac{k}{n}\right) \quad \text{and} \quad \frac{4}{\ln(2)} \frac{1+\eta}{1-\beta} \frac{1}{\log_2\left(\frac{n}{k}\right)} \frac{k}{n} t = C \log_2(n). \tag{5.40}$$

Furthermore, for some large enough constant D let

$$\omega_0 \stackrel{\text{def}}{=} -\log_2\left(1 - 2\frac{1+\eta}{1-\beta} h^{-1}\left(2\frac{k}{n} + D\frac{t}{n}\right)\right) \tag{5.41}$$

Suppose that there exists an algorithm $\mathcal{A}$, with advantage $\varepsilon = \frac{1}{\mathsf{poly}(n)}$, which distinguishes in time T distributions $(\mathbf{A}, \mathbf{sA}+\mathbf{e})$ and $(\mathbf{A}, \mathbf{y})$ with

$$\mathbf{A} \leftarrow \mathbb{F}_2^{k \times n},\ \mathbf{s} \leftarrow \mathbb{F}_2^k,\ \mathbf{y} \leftarrow \mathbb{F}_2^n \ \text{ and } \ \mathbf{e} \leftarrow \mathrm{Ber}(\omega_0 t)^{\otimes n} \quad \text{where } \ \omega_0 t = C \log_2(n)(1 + o(1)).$$

Then, there exists an algorithm running in time $T\mathsf{poly}(n)$, which takes as input $(\mathbf{G}, \mathbf{mG}+\mathbf{t})$ where $\mathbf{G} \in \mathbb{F}_2^{k \times n}$ is a *(fixed)* generator matrix of a δ-balanced $[n,k]$ code with

$$\delta \geqslant h^{-1}\left(1 - \frac{k}{n}\right) = \frac{1}{2} - \sqrt{\frac{k}{n}}(1 + o(1)),$$

and $\mathbf{t}$ of Hamming weight t, and outputs $\mathbf{t}$ (or equivalently $\mathbf{m}$) with probability at least $1 - 2^{-\Omega(n)}$ (where the probability is *not* taken over the choice of $\mathbf{m}$, $\mathbf{G}$ and $\mathbf{t}$).

Proof. We use the notations of Theorem 5.4 and Proposition 5.20. Notice that, since $k/n = o(1)$ and $t/n = o(k/n)$, the following computation holds

$$\begin{aligned} \omega_0 t &= -\log_2\left(1 - 2\frac{1+\eta}{1-\beta} h^{-1}\left(2\frac{k}{n} + D\frac{t}{n}\right)\right) t \\ &= \frac{4}{\ln(2)} \frac{1+\eta}{1-\beta} \frac{1}{\log_2\left(\frac{n}{k}\right)} \frac{k}{n} t\,(1 + o(1)) \end{aligned}$$

where we used the expansion $h^{-1}(x) = \frac{x}{\log_2(1/x)}(1 + o(1))$. Therefore, by Equation (5.40), we have

$$\omega_0 t = C \log_2(n)(1 + o(1)),$$

i.e.

$$\frac{1}{2}\left(1 - 2^{-\omega_0 t}\right) = \frac{1}{2}\left(1 - \frac{1}{n^{C(1+o(1))}}\right).$$

Let $\mathcal{B}$ be the algorithm given by Theorem 5.4. It will output $\mathbf{t}'$ in time $T\mathsf{poly}(\alpha)$. By Theorem 5.4[a],

$$\mathbb{P}(\mathbf{t}' = \mathbf{t}) = 1 - 2^{-\Omega(n)} - n\,\mathsf{poly}(\alpha)\Delta\Big(\Big(\mathbf{r}\mathbf{G}^\top, \langle \mathbf{r}, \mathbf{t} \rangle\Big), (\mathbf{a}, e)\Big),$$

where $\mathbf{r} \leftarrow \mathrm{Ber}(\omega_0)^{\otimes n}$.

Note that Equation (5.41) is equivalent to

$$(1-\beta)\frac{1}{2}\left(1-2^{-\omega_0}\right) = 2(1+\eta)h^{-1}\left(\frac{k}{n}\right)$$

Therefore, Proposition 5.20 entails that the above statistical distance is $1-2^{-\Omega(n)}$. In other words, since $\alpha = \max\left(\frac{1}{\varepsilon}, n\right) = \mathsf{poly}(n)$, Algorithm $\mathcal{B}$ will output $\mathbf{t}$ with probability $1-2^{-\Omega(n)}$, and this probability is independent from $\mathbf{G}$. This concludes the proof. □

[a] We also drop the max here, since it is reached for $x = 0$: the higher is the noise, the closer our distribution is from genuine LPN

A concrete set of parameters. The above theorem is a little bit abstract, and one can wonder which cryptographically relevant parameters can be achieved. For instance, one use the following set of parameters:

$$\frac{k}{n} = \frac{1}{n^D} \quad \text{and} \quad \frac{t}{n} = \frac{\log_2(n)^2}{n^{1-D}}$$

with $D < 1/2$. Theorem 5.21 shows that solving the average-case decision decoding problem of codes with length n, dimension n^{1-D}, and at decoding distance $\frac{1}{2} - O\left(\frac{1}{n^{D\ln(2)/4}}\right)$ is at least as hard as decoding a given *fixed* δ-balanced code (with $\delta \geqslant h^{-1}(1-\frac{1}{n^D})$) at distance $n^D \log_2(n)^2$.

Remark 5.22. *Asking the code in the worst-case search problem to be δ-balanced can seem a bit disappointing. However, as noticed in [BLVW19; YZ21] and even [BF02] (though not with the same terminology),* most *of the codes are δ-balanced, and no generic decoding algorithm is known to take advantage of this property.*

5.4 Discussion on structured codes

Let us finally turn towards the original goal of this work, namely giving search-to-decision reductions for structured codes, such as quasi-cyclic codes. It turns out that a major obstacle arises when working with codes, instead of euclidean lattices. As a consequence, there is still work to do in order to derive a good reduction for code-based cryptography.

5.4.1 Applying the OCP-based reduction

Recall that an $[n\ell, nk]$-quasi-cyclic codes have a generator matrix formed by k rows of ℓ circulant $n \times n$ matrices. Via the polynomial representation (See Section 1.3.1.1) they can be represented by a $k \times \ell$ matrix

$$\mathbf{G} \stackrel{\text{def}}{=} \begin{pmatrix} a_{1,1} & \cdots & a_{1,\ell} \\ \vdots & \ddots & \vdots \\ a_{k,1} & \cdots & a_{k,\ell} \end{pmatrix} \in \left(\mathbb{F}_2[X]/(X^n-1)\right)^{k\times\ell},$$

so that a noisy codeword is of the form

$$\mathbf{y} = \mathbf{m}\mathbf{G} + \mathbf{e} = (\langle \mathbf{a}_1, \mathbf{m}\rangle + e_1, \ldots, \langle \mathbf{a}_\ell, \mathbf{m}\rangle + e_\ell) \in \left(\mathbb{F}_2[X]/(X^n-1)\right)^{\ell}$$

where $\mathbf{a}_i \stackrel{\text{def}}{=} (a_{i,1}, \ldots, a_{i,\ell})$ is the i-th column of $\mathbf{G}$.

In order to instantiate the reduction, we need in particular to find a distribution $\mathcal{D}$ which *smoothes* the dual of the code generated by $\mathbf{G}$, that is such that for $\mathbf{r} \leftarrow \mathcal{D}$,

$$\Delta\left(\mathbf{r}\mathbf{G}^\top, \mathbf{u}^{\text{unif}}\right), \quad \text{where } \mathbf{u}^{\text{unif}} \leftarrow \left(\mathbb{F}_2[X]/(X^n-1)\right)^k$$

is negligible. Write $\mathbf{r} \stackrel{\text{def}}{=} (r_1, \ldots, r_\ell)$ so that the i-th component of $\mathbf{r}\mathbf{G}^\top$ is equal to

$$\rho_i \stackrel{\text{def}}{=} r_1 a_{i,1} + r_2 a_{i,2} + \cdots + r_\ell a_{i,\ell} \mod (X^n - 1).$$

In particular, it is an element of the ideal $\mathcal{J}_i$ generated by $(a_{i,j})_{1 \leqslant j \leqslant \ell}$. Similarly to the situation in Section 1.3.1.2, When this ideal $\mathcal{J}_i$ is too small, then ρ_i is far from being uniformly distributed in $\mathbb{F}_2[X]/(X^n-1)$, and there is no hope to give a good upper bound on the above statistical distance. Therefore, all the $\mathcal{J}_i$'s need to be large enough, and ideally they should be equal to the full ring. This can easily be enforced by restricting the decoding problem to codes in systematic form. In particular, for such codes, $a_{i,i} = 1$ (for $i \leqslant k$) and the $\mathcal{J}_i$'s are all equal to $\mathbb{F}_2[X]/(X^n-1)$. This will be our setting for the rest of this section. Moreover, for simplicity we will restrict ourselves to the case $k = 1$ and $\ell = 2$, *i.e.* to double-circulant codes, but virtually all the sequel can be extended to the general setting.

Notation. In the sequel, an element of $\mathbb{F}_2[X]/(X^n-1)$ will be written in bold font. In particular, we consider matrices of the form $(1 \mid \mathbf{a}) \in \left(\mathbb{F}_2[X]/(X^n-1)\right)^{1\times 2}$.

Remark 5.23. *It can be readily seen that a double circulant code generated by a matrix of the form* $(1 \mid \mathbf{a})$ *has a parity-check matrix of the same form.*

Discussion on the worst-case reduction. The first thing to notice is that, since the worst-case to average-case reduction given by Theorem 5.21 works for *any* (balanced) code, it would also work when the input code is quasi-cyclic (as long as it is balanced). However, there is a caveat here: if for the plain decoding problem the condition of being balanced is not really restrictive, in the case of quasi-cyclic codes it is devastating because a binary quasi-cyclic code will typically have codewords of very large weights. In fact, half of the binary quasi-cyclic codes have the all-1 codeword. This can be proved via the following simple lemma:

Lemma 5.24

Let $\mathbf{x} = (\mathbf{x}_0 \mid \mathbf{x}_1) \in \mathbb{F}_2^{2n}$, and denote by $\langle \mathbf{x}_1 \rangle$ the ideal generated by $\mathbf{x}_1$ in $\mathbb{F}_2[X]/(X^n-1)$. Let $\mathbf{a} \leftarrow \mathbb{F}_2[X]/(X^n-1)$ and consider $\mathbf{H} = (1 \mid \mathbf{a})$. Then,

$$\mathbb{P}_{\mathbf{H}}(\mathbf{x}\mathbf{H}^\top = 0) = \begin{cases} \dfrac{1}{|\langle \mathbf{x}_1 \rangle|} & \text{if } \mathbf{x}_0 \in \langle \mathbf{x}_1 \rangle \\ 0 & \text{else} \end{cases} = \frac{1}{|\langle \mathbf{x}_1 \rangle|} = 2^{n-\delta_{\mathbf{x}_1}} \mathbb{1}_{\mathbf{x}_0 \in \langle \Delta_{\mathbf{x}_1} \rangle},$$

where $\Delta_{\mathbf{x}_1} \stackrel{\text{def}}{=} \gcd(\mathbf{x}_1, X^n - 1)$ and $\delta_{\mathbf{x}_1} \stackrel{\text{def}}{=} \deg \Delta_{\mathbf{x}_1}$.

Proof. Let $\mathscr{C}$ be the code with parity-check matrix $\mathbf{H}$, and let us compute the syndrome

$$\mathbf{x}\mathbf{H}^\top = \mathbf{x}_0 + \mathbf{a}\cdot\mathbf{x}_1 \mod (X^n-1).$$

In other words,

$$\mathbf{x}\in\mathscr{C} \quad \text{if and only if} \quad \mathbf{x}_0 = a\cdot\mathbf{x}_1.$$

Since $\mathbf{a}$ is uniformly distributed over $\mathbb{F}_2[X]/(X^n-1)$, the right hand side is uniformly distributed in the ideal generated by $\mathbf{x}_1$. Therefore,

$$\mathbb{P}_{\mathbf{C}}(\mathbf{x}\in\mathscr{C}) = \mathbb{P}_{\mathbf{a}}(\mathbf{a}\cdot\mathbf{x}_1) = \mathbf{x}_0 = \frac{1}{|\langle\mathbf{x}_1\rangle|}\mathbb{1}_{\mathbf{x}_0\in\langle\mathbf{x}_1\rangle}.$$

Since $\mathbb{F}_2[X]$ is a principal ideal ring, so is its quotient $\mathbb{F}_2[X]/(X^n-1)$, and the canonical projection $\pi:\mathbb{F}_2[X]\to\mathbb{F}_2[X]/(X^n-1)$ induces a one-to-one correspondence between ideals of $\mathbb{F}_2[X]/(X^n-1)$ and ideals of $\mathbb{F}_2[X]$ containing (X^n-1). In particular, identifying $\mathbf{x}_1$ with its lift of degree at most $n-1$, $\pi^{-1}(\langle\mathbf{x}_1\rangle)$ is the (principal) ideal which contains both $\mathbf{x}_1$ and (X^n-1), *i.e.* it is the ideal generated by $\Delta_{\mathbf{x}_1}\stackrel{\text{def}}{=}\gcd(\mathbf{x}_1, X^n-1)$, of degree $\delta_{\mathbf{x}_1}\leqslant n-1$ (assuming $\mathbf{x}_1\neq 0$).

In other words, in the quotient ring, $\langle\mathbf{x}_1\rangle = \langle\Delta_{\mathbf{x}_1}\rangle$, and it is readily seen that it has dimension $n-\delta_{\mathbf{x}_1}$ as $\mathbb{F}_2$ vector space. In particular, $|\langle\mathbf{x}_1\rangle| = 2^{n-\delta_{\mathbf{x}_1}}$, which concludes the proof. □

Now, let $\mathbf{x}\stackrel{\text{def}}{=}(1,\dots,1)\in\mathbb{F}_2^{2n}$ be the all-one vector. Its polynomial representation is (ϕ,ϕ) where

$$\phi(X)\stackrel{\text{def}}{=}1+X+\cdots+X^{n-1}.$$

Lemma 5.24 entails that for a random double circulant code $\mathscr{C}$, then

$$\mathbb{P}_{\mathscr{C}}(\mathbf{x}\in\mathbf{C}) = \frac{1}{|\langle\phi\rangle|} = \frac{1}{2}.$$

Hence, half of the double-circulant codes have the all-one vector. In particular, smoothing a quasi-cyclic code is still an interesting open question. Therefore we are forced to only consider the average-case situation.

Discussion on the average-case reduction. Let $t\in\{0,\dots,n\}$ and let $(\mathbf{G},\mathbf{y})$ be an instance of the average-case (search) quasi-cyclic decoding problem, where $\mathbf{G}\stackrel{\text{def}}{=}(1\mid\mathbf{a})$ is a uniformly random double-circulant matrix, and $\mathbf{y}\stackrel{\text{def}}{=}\mathbf{m}\mathbf{G}+\mathbf{t}$ where $\mathbf{t}=(\mathbf{t}_0\mid\mathbf{t}_1)\leftarrow\mathcal{S}_t\times\mathcal{S}_t$ is a uniformly random regular error vector of weight $2t$.

Our goal is to prove the hardness of the decisional version of the quasi-cyclic decoding problem, an instance of which consists in samples of the form $(\mathbf{a}',\mathbf{b}')$ where $\mathbf{a}'\leftarrow\mathbb{F}_2[X]/(X^n-1)$ is uniformly distributed, and $\mathbf{b}'$ is either also uniformly distributed, or of the form $\mathbf{m}'\cdot\mathbf{a}'+\mathbf{e}'$, where $\mathbf{m}'$ is the *same* secret for every sample, and $\mathbf{e}'$ follows some error distribution $\mathcal{D}$ over $\mathbb{F}_2[X]/(X^n-1)$, *e.g.* a Bernoulli (*i.e.* such that each coefficient of $\mathbf{e}'$ is independently distributed according to some Bernoulli distribution over $\mathbb{F}_2$), or the uniform distribution over

polynomials of given Hamming weight.

If one wants to use a reduction following the OCP technique as for the plain decoding problem, the first step consists in introducing a distribution Ψ which *smoothes* the input, that is such that for $\mathbf{e}' \leftarrow \Psi$ we can build samples close enough from the target distribution recalled above by computing $\langle \mathbf{y}, \mathbf{r} \rangle$ for some application

$$\langle \cdot, \cdot \rangle : \left(\mathbb{F}_2[X] \big/ (X^n - 1) \right)^\ell \times \left(\mathbb{F}_2[X] \big/ (X^n - 1) \right)^\ell \to \mathbb{F}_2[X] \big/ (X^n - 1).$$

In the quasi-cyclic situation, it is natural to consider the following inner product[iii]

$$\langle \mathbf{x}, \mathbf{x}' \rangle \stackrel{\text{def}}{=} \sum_{i=1}^{\ell} x_i \cdot x'_i,$$

where $\mathbf{x} \stackrel{\text{def}}{=} (x_1, \ldots, x_\ell)$ and $\mathbf{x}' \stackrel{\text{def}}{=} (x'_1, \ldots, x'_\ell)$. Now, given $\mathbf{r}$ distributed according to a Bernoulli distribution over $\mathbb{F}_2[X] \big/ (X^n - 1)$, we can compute (in our case, we simplified to $\ell = 2$ but this can be more general)

$$\langle \mathbf{y}, \mathbf{r} \rangle = \mathbf{m} \left(\sum_{i=1}^{\ell} \mathbf{a}_i \mathbf{r}_i \right) + \underbrace{\sum_{i=1}^{\ell} \mathbf{t}_i \mathbf{r}_j}_{\text{LPN noise}}, \tag{5.42}$$

where $\mathbf{r}$ is a Bernoulli random variable over $\mathbb{F}_2[X] \big/ (X^n - 1)$ of parameter $\omega_0 \in \mathbb{R}_+$.

In order to apply the same reduction strategy as for the plain decoding problem, we need to analyse the distribution of the produced samples. More precisely, we need to analyse how far away they are from genuine "Ring-LPN" samples. However, we immediately face one difficulty regarding the noise distribution in Equation (5.42).

Indeed, let $\tau \in \mathbb{F}_2[X] \big/ (X^n - 1)$ of Hamming weight t, and let $\rho \in \mathbb{F}_2[X] \big/ (X^n - 1)$ distributed according to a Bernoulli of parameter ω_0, *i.e.*

$$\tau = \sum_{i=0}^{n-1} \tau_i X^i, \quad \text{with } |\{i \mid \tau_i \neq 0\}| = t,$$

and

$$\rho = \sum_{i=0}^{n-1} \rho_i X^i, \quad \text{with } \rho_i \leftarrow \text{Ber}(\omega_0).$$

Then,

$$\tau\rho = \sum_{k=0}^{n-1} \sum_{\substack{i+j \equiv k \\ \text{mod } n}} \tau_i \rho_j X^k. \tag{5.43}$$

Each coefficient of this product is a sum of exactly t independent $\text{Ber}(\omega_0)$ random variables. Therefore, each coefficient of $\tau\rho$ is a $\text{Ber}(\omega_0 t)$ random variable, and each coefficient of $\langle t, r \rangle$ is a $\text{Ber}(2\omega_0 t)$ random variable. At first glance, it seems that we can do the analysis as for the plain decoding problem. However, there is a strong caveat here: the coefficients are *not* independent.

[iii] This is not the only choice, and changing it might help in getting the reduction

In particular, this inner product is *not* a Bernoulli random variable over $\mathbb{F}_2[X]/(X^n-1)$, even though it would have the correct Hamming weight on average.

It turns out that this distribution is very difficult to analyse: it is not even radial as it depends on the actual Hamming support of $\mathbf{t}$, and not only on its weight. This fact was already emphasised in the HQC submission to the NIST competition [AABB+22b], when studying the Decoding Failure Rate (DFR) of this scheme. In their analysis, the authors replaced this weird distribution by an actual Bernoulli distribution and made experiments to support their modelisation. However, such an approach is not enough for a theoretical reduction. In other words, for applying the OCP reduction, we lack a so-called *random self-reducibility* for structured codes.

In the world of Euclidean lattices, the reductions do not face the same problem since the error distribution is not directly defined over $\mathcal{O}_K/q\mathcal{O}_K$ (the analogue of $\mathbb{F}_2[X]/(X^n-1)$), nor over $\mathcal{O}_K$ or even $K_{\mathbb{R}} \stackrel{\text{def}}{=} K \otimes_{\mathbb{Q}} \mathbb{R}$, but through the Minkowski embedding. Due to the properties of this embedding, the noise then affect each coordinate independently. Moreover, the reduction from [RSW18, Section 4] benefits from the fact that the Vandermonde matrix, which maps the so-called coefficient embedding to the Minkowski embedding, does not distort the noise too much. Note that this is nothing else but a (discrete) Fourier transform. In the code-based setting, such a Fourier-based approach takes an exaggerated toll on the noise distribution: there exist an uncertainty principle, and a sparse error vector is mapped to a dense error vector. Moreover, in the code-based setting we cannot define the error distribution through the Fourier transform. Indeed, consider a double circulant parity-check matrix represented by a polynomial h. A syndrome of a regular error with respect to some distribution $\mathcal{D}$ is then of the form $\sigma \stackrel{\text{def}}{=} e_1 + h \cdot e_2 \in \mathbb{F}_2[X]/(X^n-1)$, where $e_1, e_2 \leftarrow \mathcal{D}$. Usually, $\mathcal{D}$ outputs vectors of small Hamming weight, but let us assume, as it is the case in lattice-based cryptography, that this sparsity is defined after Fourier transform, that is through the embedding

$$\Phi\colon \begin{cases} \mathbb{F}_2[X]/(X^n-1) & \hookrightarrow & \mathbb{F}_{2^m} \times \cdots \times \mathbb{F}_{2^m} \\ P & \mapsto & (P(\omega_1), \ldots, P(\omega_n)) \end{cases}$$

where ω_i are the n-th roots of unity in some extension field $\mathbb{F}_{2^m}$. In other words, assume that $|\Phi(e_i)| = t$ is small. Note that Φ is a ring homomorphism:

$$\Phi(\sigma) = \Phi(h) \star \Phi(e_1) + \Phi(e_2),$$

where $\star$ denotes the component-wise product. In particular, $|\Phi(h) \star \Phi(e_1)| \leqslant |\Phi(e_1)| = t$, and therefore $|\Phi(\sigma)| \leqslant 2t$. On the other hand, since Φ is linear, a uniformly random element of $\mathbb{F}_2[X]/(X^n-1)$ will be mapped to a uniformly random element of its image, which is a product of subfields of $\mathbb{F}_{2^m}$ whose degrees are exactly those appearing in the factorisation of X^n-1 in $\mathbb{F}_2$. More precisely, $\Phi(P)_i = 0$ with probability $1/2^d$ where d is the degree of the minimal polynomial of ω_i. In general, n is chosen such that 2 is primitive modulo n, and therefore X^n-1 has only two irreducible factors[iv]. Hence,

$$\mathbb{P}\left(\Phi(P)_i = 0\right) = \begin{cases} 1/2 & \text{if } \omega_i{=}1 \\ 1/2^{n-1} & \text{otherwise.} \end{cases}$$

[iv] A famous conjecture of Artin asserts, among other things, that there are infinitely many such n; and this is the choice made for example in BIKE and HQC.

In other words, Φ maps a random element of $\mathbb{F}_2[X]/(X^n-1)$ to a weight n or $n-1$ vector with overwhelming probability, while it maps σ to a sparse vector of weight $2t$. For $t < n/2$ this provides an obvious distinguisher between a syndrome of the regular distribution associated to $\mathcal{D}$ and the uniform. In particular, we cannot hope to design a search-to-decision reduction for this error distribution, unless the search problem is easy.

There may exist a good choice of a distribution $\mathcal{D}$ which avoids such biases, but this is not clear at the moment of writing this chapter.

5.4.2 Possible future research direction

As emphasised above, the main issue to derive a good search-to-decision reduction in the OCP framework is this lack of random self-reducibility: given an instance of the decoding problem $(\mathbf{G}, \mathbf{mG}+\mathbf{t})$, we do not know how to obtain a fresh instance $\mathbf{xH}+\mathbf{e}$ when $\mathbf{G}$ and $\mathbf{H}$ are supposed to generate structured codes.

In the case of structured lattices, a new tool to derive worst-case to average-case reductions has been introduced in [BDPW20]: namely random walks on the so-called Arakelov class group, which is a group structure on the space of ideal lattices (up to isometry). Indeed, inspired by elliptic curve cryptography where it was shown in [JMV09] that the discrete logarithm problem on a given elliptic curve is as hard as the discrete logarithm on *any* isogenous curve, they manage to "distort" a given ideal lattice into a random ideal lattice by doing a random walk on this Arakelov class group, while keeping the problems of finding short vectors in those lattices more or less equivalent. One of the key observations is that this random walk converges quickly and they manage to bound the number of steps needed to get a "random enough" ideal lattice. Interestingly, the study of this random walk involves Fourier analysis, and hence representation theory, on this compact abelian group.

This approach has been generalised in [DK22] to some module lattices of higher rank, using representation theory on more involved groups, with connections to the very active research domain known as the *Langlands correspondence*.

At the moment of this writing, it is not clear how such an approach could be adapted to the code-based setting, but this random walk technique seems worth exploring, building upon the function field approach introduced in Chapter 4. In the function field setting, the Arakelov class group can be somehow considered as an analogue of the Jacobian variety associated with the function field, even though in the number field setting there is a discrepancy between the finite and infinite places. Nevertheless, in the function field setting, the aforementioned Langlands correspondence is more well-known, and this theory involves Drinfeld modules, which are a generalisation of the Carlitz module defined in Chapter 4 (see for instance [Vil06, Chapter 13] for an introduction to Drinfeld modules).

Part III

Secure Multiparty Computation

Chapter 6

A Short Introduction to Secure Multiparty Computation

This short chapter is an introduction to the topic of secure multiparty computation. It serves as a motivation for Chapter 7 which is one of the contributions of this thesis.

Outline of the current chapter

6.1 Introduction

One of the main goals of the cryptographer is to be able to compute some function over secret data. One may consider two main use cases:

- *Outsourced computing*, where a client delegates its computation to a third-party, *e.g.* a powerful remote server, without compromising the data;
- *Multi-Party computation* (MPC), where multiple clients that do not trust each other want to jointly compute a public function on all of their private inputs.

Outsourced computing can make use of *Fully Homomorphic Encryption* (FHE), which allows the third-party to perform the computation over the client's encrypted data and get the encryption of the result. The client can finally decrypt it with its own secret key. This advanced functionality has been conceptualised by Rivest, Adleman and Dertouzos in [RAD+78], but the first complete scheme was proposed in the breakthrough work of Gentry [Gen09], which uses structured lattice-based cryptography. Since then on, a lot of progress have been made, yet FHE remains quite costly and might not be suitable for all applications.

On the other hand, in Multi-Party Computation each participant to the computation owns part of the data, and their goal is to cooperate to compute a function without leaking anything else than the output of the function. MPC has been envisioned by Yao in the early 1980s ([Yao82]), and the first generic solution for secure multi-party computation with more than two players appeared in [GMW87]. Since then, many protocols have been developed, and for all of them the bottleneck in the efficiency relies in the *communication* between all the participants. A presentation of those protocols is out of scope of this manuscript. In order to cope with that, it was observed in several works (*e.g.* [Bea91; IPS08; DPSZ12]) that when all the parties share some random elements *correlated* in a useful way, it was possible to design significantly more efficient MPC protocols. The idea is to push as much as possible the necessary communication between the parties in a *preprocessing phase* in which they get those correlated random elements, before doing the relevant computation. This is model of computation is known as the *preprocessing model* or the *correlated randomness model*. However, this leaves open the question of efficiently generating them: this is the main motivation of [BCCD23] which is one of the contributions of this thesis, and which will be covered in Chapter 7. The reader interested in a more in-depth presentation of those protocols can refer to [CDN15; EKR18].

Remark 6.1. *MPC can also be used to build zero-knowledge proofs of knowledge, using the so-called* MPC in the Head *technique introduced by Ishai, Kushilevitz, Ostrovsky and Sahai in [IKOS07]. Such proofs can then be turned into a signature scheme via the famous Fiat-Shamir transform. This paradigm has been used to design the Picnic signature scheme [CDGK+20] which reached round 3 of the NIST first call for post-quantum algorithms, but was not selected for standardisation. On the other hand, this technique has received a lot of interest in the past year since the work of Feneuil, Joux and Rivain [FJR22]. Due to the linearity, and the compact representation of sparse vectors, this technique seems particularly suitable to be used with codes. At the time of writing those lines, NIST did not yet release the candidates for standardisation for the second call for post-quantum signatures, but it declared that out of the 50 submissions, 7 are based on MPC in the Head, and the security of 5 of them reduces to code-based problems.*

Remark 6.2. *Note that FHE also provides a solution to secure two-party computation: one party, say Alice, can encrypt her input and broadcast it, keeping the secret key for herself. The other party, say Bob, can then homomorphically evaluate the function on Alice's ciphertext and his own input and send back the encrypted output to Alice. However, allowing more than two parties is not evident at all, especially when some parties are corrupted. Nonetheless, some solutions have been proposed using so-called* Threshold FHE, *or* multi-key FHE *[CDN01; AJLT+12; MW16]. However for now, those solutions are still much less efficient than pure MPC protocols.*

6.2 Secure Computation in the Preprocessing Model

From now on, let $\mathbb{F}_q$ denote some finite field. Given n participants $P_1, \dots, P_n$ referred to as the *parties* or the *players*, each of them owning a secret input x_i, their goal is to engage in a protocol in order to compute a value

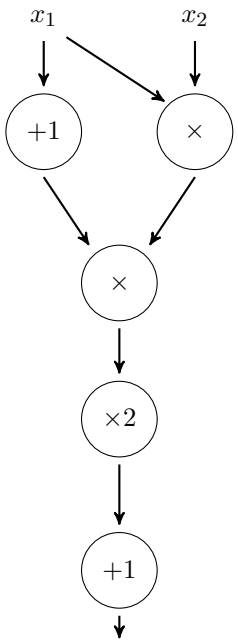
$$y \stackrel{\text{def}}{=} f(x_1, \dots, x_n) \in \mathbb{F}_q$$

such that no player learns anything else about the x_i's. Since a function $f : \mathbb{F}_q^n \mapsto \mathbb{F}_q$ can be written as a polynomial in its inputs, it can be represented as an arithmetic circuit. More formally, an arithmetic circuit is a directed acyclic graph whose internal nodes are called *gates* and the edges are called *wires*. There are n input values corresponding to the x_i's, and one output value, namely y. There are four different internal gates:

- the addition gate, which has two input wires, and outputs the sum of its inputs;
- the multiplication gate, which has two input wires, and outputs the multiplication of its inputs;
- the scalar addition gate, which has only one input wire, is labelled by a constant $\alpha \in \mathbb{F}_q$, and outputs the sum of its input and α.
- the scalar multiplication gate, which has only one input wire, is labelled by a constant $\beta \in \mathbb{F}_q$, and outputs the multiplication by β.

Each gate may have any number of output wires, since the result of an internal operation can be reused by several other internal operations.

Example 6.3. *The function* $f(x_1, x_2) \stackrel{def}{=} 2x_1^2x_2 + 2x_1x_2 + 1$ *over* $\mathbb{F}_5$ *can be represented by the following arithmetic circuit.*



In order to jointly compute the function f, all the players must *communicate* between each other, while revealing no information. This can be done using *additive secret sharing*.

6.2.1 Additive Secret Sharing

Secret sharing is an important tool in MPC, put forth by Shamir in [Sha79]. The goal is to split a secret data s_i owned by one party P_i into pieces $(s_i^{(j)})_{1 \leqslant j \leqslant n}$ called *shares* such that for any

strict subset $B \subsetneq \{1,\dots,n\}$, the knowledge of $(s_i^{(j)})_{j\in B}$ gives *no* information about s_i. On the other hand, we ask that the secret s_i can be reconstructed from all the shares $s_i^{(j)}$.

For example, when $s_i \in G$ for some finite group G, this can be achieved as follows: P_i picks $s_i^{(j)}$ uniformly at random in G for $j \in \{1,\dots,n-1\}$ and sets $s_i^{(n)} \stackrel{\text{def}}{=} (s_i^{(n-1)})^{-1}\dots(s_i^{(1)})^{-1}s_i$. Then, they distribute $s_i^{(j)}$ to party P_j (and keeps $s_i^{(i)}$). In other words, $s_i^{(n-1)}\dots s_i^{(1)}$ acts as a *one-time pad* on the secret. It follows that any vector of $n-1$ shares is uniformly distributed in G^{n-1} and is independent from the secret s_i, while $s_i^{(1)}\dots s_i^{(n)} = s_i$. In the sequel, G will usually be abelian. In that case, we use an additive notation, hence the name *additive secret sharing*.

Remark 6.4. *Sometimes, we want to introduce a threshold $0 \leqslant t < n$, such that any number $N \leqslant t$ of shares do not reveal any information on s_i, whereas any number $N' \geqslant t+1$ uniquely determine the secret. This can be achieved via polynomial interpolation (see [Sha79; CDN15]), but will not be needed in the sequel. The scheme presented above corresponds to a threshold $t = n-1$.*

When keeping track of which party owns which share does not matter, the operation of sharing a secret s is often denoted by $[\![s]\!]$. Opening $[\![s]\!]$ consists in revealing all the shares to reconstruct s.

Assume that $G = (\mathbb{F}_q, +)$ is the additive group of a finite field $\mathbb{F}_q$. Then, it is readily seen that for any $x, y \in \mathbb{F}_q$, the sum of two random shares of x and y is a valid random share of the sum $x+y$:

$$[\![x]\!] + [\![y]\!] = [\![x+y]\!].$$

Similarly, if $\alpha \in \mathbb{F}_q$ is a constant known to any party, then $\alpha[\![x]\!] = [\![\alpha x]\!]$. In other words, additive secret sharing in $\mathbb{F}_q$ is linear. Furthermore, addition by a public constant $\beta \in \mathbb{F}_q$ can also easily be achieved: it suffices that only one party adds β to their share. On the other hand, multiplication is trickier, and computing a share $[\![xy]\!]$ necessitates a protocol with communication between all the parties (see [EKR18, Section 3.3]).

This easy construction has in reality a lot of applications for computing an arithmetic circuit. Indeed, imagine that two parties P_1 and P_2 want to jointly compute a function $f(x_1, x_2)$ of their respective private inputs x_1, x_2. They can first secret share the x_i's and then, each addition, scalar multiplication, scalar addition can be done locally on the shares. The multiplication gates can be computed via a secure multiplication protocol. At the end of the computation, the parties can open their shares to reveal the output of the function, while revealing nothing on the x_i's. This can be generalised to $n > 2$ parties.

Note that in the above protocol, the only computation that requires communication between the parties is the multiplication gates. Usually, local computations are much less costly than communication which is therefore the main bottleneck in MPC protocols.

6.2.2 Beaver Multiplication Triples

In order to improve on the communication complexity, a line of work initiated by Beaver in [Bea91] proposed to split the computation into two phases: First, the parties preprocess a lot of multiplications of random values before engaging in the protocol, and then they use these precomputed values to speed up the computation.

Definition 6.5 (Beaver triples)

A triple $(u,v,w) \in \mathbb{F}_q^3$ is said to be a *Beaver multiplication triple*, or simply a *Beaver triple*, when u and v are uniformly random elements in $\mathbb{F}_q$, and that $w \stackrel{\text{def}}{=} u \cdot v$ is their product.

Assume that a trusted third-party called the *dealer* creates a lot of Beaver triples $(u,v,w) \in \mathbb{F}_q^3$ unknown to the parties, and distributes additive shares $([\![u]\!],[\![v]\!],[\![w]\!])$ to them. This phase is referred to as the *preprocessing phase*.

Then, the parties can enter the *online phase* of the protocol. Addition, scalar addition and scalar multiplication gates do not present any difficulty. But now, computing multiplication gates can be done quite efficiently:

Assume that we are given shares $[\![x]\!]$ and $[\![y]\!]$ and we want to compute a share of the product $[\![xy]\!]$. It suffices to take the next unused random multiplication triple $([\![u]\!],[\![v]\!],[\![w]\!])$ and to compute

$$[\![x]\!] + [\![u]\!] = [\![x+u]\!] \qquad [\![y]\!] + [\![v]\!] = [\![y+v]\!].$$

Then, all the parties open their respective shares, which reveals $\mathbf{d} \stackrel{\text{def}}{=} x+u$ and $\mathbf{e} \stackrel{\text{def}}{=} y+v$, but since u and v have been drawn uniformly at random in $\mathbb{F}_q$ and are unknown to any of the parties, $\mathbf{d}$ and $\mathbf{e}$ reveal nothing on x and y. We put in boldface the elements that are publicly known. Now, note that

$$\begin{aligned} xy &= (x+u-u)(y+v-v) \\ &\stackrel{\text{def}}{=} (\mathbf{d}-u)(\mathbf{e}-v) \\ &= \mathbf{de} - \mathbf{d}v - u\mathbf{e} + w, \qquad \text{with } w = uv \text{ by definition.} \end{aligned}$$

By linearity, a share $[\![xy]\!]$ can then be locally computed by the parties by linearity

$$[\![xy]\!] \stackrel{\text{def}}{=} \mathbf{de} - [\![v]\!]\mathbf{d} - [\![u]\!]\mathbf{e} + [\![w]\!].$$

This protocol boils down to opening two secret sharing by multiplication gates plus some cheap local computations. The only limitation now is the generation of a very large number of multiplication triples. Indeed, since they are used as a one-time pad, they cannot be reused in the execution of the protocol. Just to give an idea, a useful function f may have more than 2^{30} multiplication gates.

This technique has been widely considered for secure multiparty computation, and Damgård, Pastro, Smart and Zakarias introduced in [DPSZ12] an efficient protocol often called SPDZ-protocol[i] by a clever rearrangement of the initials of the authors, which allows to securely compute a function even when *any number* $t \leqslant n-1$ of the participants deviate from the protocol specification. This setting is known as *dishonest majority*. The SPDZ-protocol makes use of a variant called *authenticated multiplication triples*, namely an additive sharing

$$([\![u]\!],[\![v]\!],[\![uv]\!]),([\![\Delta u]\!],[\![\Delta v]\!],[\![\Delta uv]\!]),$$

where Δ is a global random element of $\mathbb{F}_q$, fixed for every share, of which the parties only know a secret sharing $[\![\Delta]\!]$. See [EKR18, Section 6.6.2] for a pedagogical presentation of SPDZ.

[i] pronounced "Speeds"

6.2.3 Oblivious Linear Evaluation

As we have seen in the previous section, MPC protocols can often make use of many random multiplication triples. In other words, before engaging in the protocol, the parties share a long *correlated random* list of elements of $\mathbb{F}_q$. It remains to answer how such a list can be generated efficiently.

In order to perform MPC, other useful correlations have been considered in the literature. In the sequel we will only consider one of them, namely *Oblivious Linear Evaluation* or OLE for short.

Definition 6.6 (OLE **Correlation**)

A tuple $(\mathrm{U},\mathrm{X},\mathrm{V},\mathrm{Y}) \in \mathcal{R}^4$ defined over a finite ring $\mathcal{R}$ is said to have the OLE *correlation* when $\mathrm{U},\mathrm{V},\mathrm{X}$ are uniformly, and independently, distributed in $\mathcal{R}$ and they are subject to the following correlation

$$\mathrm{U}\cdot\mathrm{V} = \mathrm{X} + \mathrm{Y}. \tag{6.1}$$

Such a tuple with the OLE correlation will be simply referred to as *an* OLE. An OLE $(\mathrm{U},\mathrm{V},\mathrm{X},\mathrm{Y})$ is said to be shared between two parties P_1 and P_2 if P_1 owns U and X while P_2 owns V and Y. In other words, they both own one factor of a product, and an additive share of said product.

Remark. *As in the previous section, $\mathcal{R}$ will often be a finite field $\mathbb{F}_q$. However, this will not always be the case and we need to define* OLE*'s over more general rings. Nonetheless, they will always be $\mathbb{F}_q$-algebras.*

The rationale behind the name *Oblivious Linear Evaluation* comes from the following fact: consider the 2-party task of evaluating an affine function

$$f : \left\{ \begin{array}{ccc} \mathcal{R} & \longrightarrow & \mathcal{R} \\ x & \longmapsto & ax+b \end{array} \right.$$

owned by one party, say Alice, on a private input owned by the second party, say Bob. In other words, Alice owns the coefficients a and b, while Bob owns x and gets $f(x) \stackrel{\text{def}}{=} ax+b$. This evaluation is said to be *oblivious* when Alice learns nothing about x while Bob learns nothing about a and b. Now, it is readily seen that $(a,x,-b,ax+b)$ satisfies Equation (6.1).

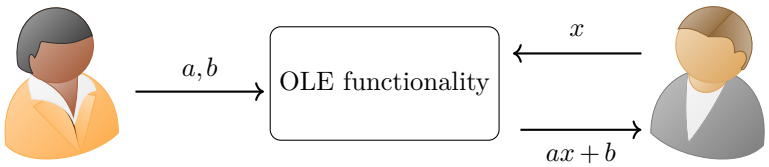


Figure 6.1: Oblivious Linear Evaluation

Remark 6.7. *Note that* OLE*'s generalise so-called* oblivious transfers *(*OT*'s) which are another fundamental tool used in MPC. In the* OT *functionality, Alice owns two values $a_0, a_1 \in \mathbb{F}_{2^m}$ while Bob owns a secret bit $\beta \in \mathbb{F}_2$. At the end of the protocol, Bob learns a_β and nothing about $a_{1-\beta}$ while Alice learns nothing about β. This corresponds to an* OLE *of $f(x) \stackrel{def}{=} (a_1 - a_0)x + a_0$ over $\mathbb{F}_2$.*

Relation to Beaver's multiplication triples. One of the main reasons why we are interested in OLE is that they are related to multiplication triples. Indeed, assume two players Alice and Bob share two OLE's (U_1, V_1, X_1, Y_1) and (U_2, V_2, X_2, Y_2) defined over a finite field $\mathbb{F}_q$, *i.e.* Alice owns U_i, X_i while Bob owns V_i, Y_i. Then, set

$$C_A \stackrel{\text{def}}{=} X_1 + U_1 U_2 + X_2$$

which Alice can locally compute, and

$$C_B \stackrel{\text{def}}{=} Y_1 + V_1 V_2 + Y_2$$

which Bob can locally compute. Note that C_A and C_B are uniformly distributed over $\mathbb{F}_q$. Moreover,

$$\begin{aligned} C \stackrel{\text{def}}{=} C_A + C_B &= (X_1 + Y_1) + U_1 U_2 + V_1 V_2 + (X_2 + Y_2) \\ &= U_1 V_1 + U_1 U_2 + V_1 V_2 + U_2 V_2 \\ &= (U_1 + V_2)(U_2 + V_1). \end{aligned}$$

Setting $A \stackrel{\text{def}}{=} U_1 + V_2$ and $B \stackrel{\text{def}}{=} U_2 + V_1$, *i.e.* such that U_i and V_i be additive shares of A and B, then we have that $C = AB$, *i.e.* (A, B, C) is a multiplication triple of which Alice and Bob can compute an additive sharing. All in all, we have the following proposition:

Proposition 6.8

Given *two* OLE's over $\mathbb{F}_q$ shared by two parties P_1, P_2, both parties can compute *one* Beaver multiplication triple.

As a consequence, it suffices to distribute to the parties twice as many OLE's as the number of multiplication gates in the circuit representation of a function to be able to securely compute it. However, distributing those OLE's is not an easy problem. In particular, usual approaches such as introduced in [IKNP03; DPSZ12; KPR18] to generate N correlated random elements still require to communicate $\Omega(N)$ elements, which is quite costly, and actually forms the bottleneck of many MPC protocols. This state-of-affairs recently changed when Boyle *et al* [BCGI18; BCGI+19] introduced a new tool called *Pseudorandom Correlation Generator* (PCG) which allows to generate a large list of OLE's starting from short *correlated seeds*, similarly to a more common pseudorandom generator which generates randomness from a short seed. The idea is that using a PCG, the communication between the parties can be limited to the generation of the correlated seeds, which the parties can locally expand into a list of OLE's without further communication. In other words, beyond the generation of the seeds, this preprocessing phase is *silent*. This will be detailed in Section 6.4.

Remark 6.9. *As is,* Pseudorandom Correlation Generators *are limited to the important, but a still rather limited, setting of two-party computation. However, they can be extended to* N*-party computation thanks to so-called property of* programmability*.*

In Chapter 7, we build a programmable PCG *making use of another class of algebraically structured codes, namely* Quasi-Abelian Codes*.*

6.3 Function Secret Sharing

6.3.1 Generalities

In Section 6.2 we gave a quick introduction to additive secret sharing and its use in MPC. In this section, we introduce a somewhat generalised object, namely *Function Secret Sharing* (FSS), where the goal is to split, not a single value s, but rather a function $f : \mathcal{S} \to \mathbb{F}_q$ for some finite set $\mathcal{S}$. Informally, an 2-party *Function Secret Sharing* has two algorithms: Gen which is a Probabilistic Polynomial Time algorithm that given f outputs a pair of *short* keys (k_0, k_1), and Eval which given a key k_i and an element $x \in \mathcal{S}$ outputs a field element $f_i(x) \in \mathbb{F}_q$ such that

$$\forall x \in \mathcal{S}, \qquad f(x) = f_0(x) + f_1(x),$$

a property called *correctness*, and so that taken separately k_i (or equivalently the function f_i) computationally hides f to ensure *secrecy*. Denoting by $N \stackrel{\text{def}}{=} |\mathcal{S}|$, and choosing an ordering of $\mathcal{S}$, we may (and will) assume without loss of generality that $\mathcal{S} = \{0, \dots, N-1\}$.

The goal of this section is not to give an in-depth description of what is doable, but rather to give a high-level presentation of FSS for one specific class of functions which will be used in Chapter 7, namely *point functions*. The interested reader can read [BGI16], [BCGI+20b, Section5] or [BCGI22] for further reference.

Definition 6.10 (Point Function)

Given two values $\alpha \in \{0, \dots, N-1\}$ and $\beta \in \mathbb{F}_q$, the *point function* $f_{\alpha,\beta} : \{0, \dots, N-1\} \to \mathbb{F}_q$ is defined as

$$f_{\alpha,\beta}(x) \stackrel{\text{def}}{=} \beta \cdot \mathbb{1}_{x=\alpha} \stackrel{\text{def}}{=} \begin{cases} \beta & \text{if } x = \alpha \\ 0 & \text{otherwise .} \end{cases}$$

In other words, a point function takes only one non-zero value β, at the specific input $x = \alpha$. Function Secret Sharing was initially introduced in [GI14] for the class of point functions under the name *Distributed Point Function* (DPF), and later generalised in [BGI16]. This reference also gives the most efficient construction for a DPF.

Remark 6.11. *The point function $f_{\alpha,\beta}$ can be represented by a vector of $\mathbb{F}_q^N$ of Hamming weight 1, i.e. a vector of the form*

$$(0, \dots, 0, \beta, 0, \dots, 0),$$

where β is in the α-th position. Just to give an idea, for our applications we can think of N to be about 2^{30}. In other words, a DPF *scheme is a way of efficiently* compressing *an additive secret sharing of a* long *vector of Hamming weight 1. Note that a traditional secret sharing is far from being compact, since each additive share should be indistinguishable from a random element of $\mathbb{F}_q^N$.*

This can easily be generalised to a compact secret sharing of t-sparse vectors, i.e. vectors of Hamming weight t. Indeed, such a vector can be decomposed as the sum of t vectors of Hamming weight 1. Therefore, a share of a t-sparse vector is nothing but the sum the corresponding shares of the t weight-1 vectors. In other words, this boils down to running the protocol t times.

6.3.2 Distributed Point Functions (DPF)

In this section we give an overview of the construction of [BGI16] for a 2-party DPF. In order to simplify the presentation, let us assume that N is of the form 2^n. The construction can be easily adapted to the general case. Using a binary representation, we may assume that $f_{\alpha,\beta}$ is actually defined over $\{0,1\}^n$. Moreover, let us also assume that $\beta = 1$. The generalisation to any $\beta \in \mathbb{F}_q$ is given in Remark 6.14.

Recall that a Pseudorandom Generator (PRG) is a function

$$G : \mathbb{F}_q^n \to \mathbb{F}_q^m \quad \text{with } m > n$$

which takes as input a small seed, and stretches it into a long string of length m which is computationally indistinguishable from true randomness.

6.3.2.1 GGM-tree

The design of [BGI16] makes use of any PRG $G : \mathbb{F}_q^\lambda \to \mathbb{F}_q^{2\lambda+2}$, where λ is a security parameter, in a tree-based construction. Starting from a seed $\mathbf{s}$ of size λ, define the following binary tree representing all the elements of $\{0,\dots,N-1\}$ written in binary (with n bits), and whose nodes are labelled by a tuple (σ,τ) where $\sigma \in \mathbb{F}_q^\lambda$ and $t \in \mathbb{F}_q$ is an additional *control element* which will only be used in the actual construction.

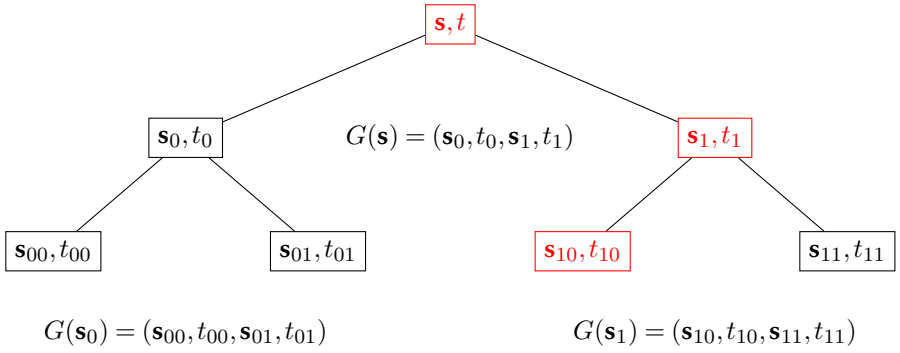


Figure 6.2: Illustration of the tree-based construction, with 2 levels and a Pseudorandom Generator $G : \mathbb{F}_q^\lambda \to \mathbb{F}_q^{2\lambda+2}$. The red path represents the element $\alpha = (1,0) \in \mathcal{S} \stackrel{\text{def}}{=} \{0,1\}^2$.

- The root of the tree has a label of the form $(\mathbf{s},t)$ where $\mathbf{s} \in \mathbb{F}_q^\lambda$ is the seed and $t \in \mathbb{F}_q$ will be specified later.
- Let ν denote an internal node, with a label of the form $(\mathbf{s}^{(\nu)}, t^{(\nu)})$. Write

$$G(\mathbf{s}^{(\nu)}) \stackrel{\text{def}}{=} (\mathbf{s}_0^{(\nu)}, t_0^{(\nu)}, \mathbf{s}_1^{(\nu)}, t_1^{(\nu)})$$

  where $\mathbf{s}_i^{(\nu)} \in \mathbb{F}_q^\lambda$ are considered as fresh seeds, and $t_i^{(\nu)}$ are new control elements. Then, ν has two children ν_0 and ν_1 with respective labels $(\mathbf{s}_0^{(\nu)}, t_0^{(\nu)})$ and $(\mathbf{s}_1^{(\nu)}, t_1^{(\nu)})$.

Such a tree-based construction from a PRG is called a GGM-tree, named after Goldreich, Goldwasser and Micali who used it to design a so-called Pseudorandom Function in [GGM84].

Remark 6.12. *Note that in this construction, the label of a given node is entirely determined by the seed of its parent.*

6.3.2.2 The construction

Back to our goal of sharing some point function $f_{\alpha,\beta} : \{0,1\}^n \to \mathbb{F}_q$, the idea will be to represent the shares f_0 and f_1 as two GGM-like trees specified by the label of their roots, included in their respective keys k_0 and k_1.

Remark 6.13. *Given a node ν, it will be convenient to see its labels on the two trees as an additive secret sharing $[\![\mathbf{s}]\!], [\![t]\!]$ of an underlying meta-label.*

For each leaf $x \in \mathcal{S}$, define its *evaluation path* to be the path from the root to x, and define $f_i(x)$ to be the control element corresponding to x. In other words, $\mathsf{Eval}(i, k_i, x)$ will compute the labels of all the nodes on the evaluation path of x using k_i and G, and will output the last control element. For correctness of the sharing scheme, it is necessary that the two control elements (one on each tree) corresponding to a given leaf $x \neq \alpha$ be opposite to each other. On the other hand, they must sum up to 1 for $x = \alpha$. In order to achieve that, we will maintain the following invariant:

(i) For each node on the evaluation path of α, the control elements sum up to 1 and the two seeds are indistinguishable from being independent and uniformly distributed in $\mathbb{F}_q^\lambda$.

(ii) For each other node, the two full labels are opposite.

In other words, the labels of each node outside the evaluation path of α should be seen as an additive secret sharing $[\![\mathbf{0}]\!], [\![0]\!]$ of $0^{\lambda+1}$, while all the nodes in the evaluation path should be seen as an additive secret sharing $[\![\mathbf{s'}]\!], [\![1]\!]$ of the control element $1 \in \mathbb{F}_q$ and a random seed $\mathbf{s'} \in \mathbb{F}_q^\lambda$ (different at each level).

Note that this invariant can easily be satisfied by the roots, which are always on the evaluation path of α: it suffices to include the full labels in the keys k_i to initialise the scheme. Let us see how this invariant can be maintained on the full tree.

First, notice that a 2-party secret sharing scheme satisfies the following property: If G is a PRG, then from a small share $[\![\mathbf{0}]\!]$ of 0^λ, one can compute a share of a longer 0-string $0^{2\lambda+2}$ by applying G. Indeed, assume one of the parties, say Alice, owns $\mathbf{s}^{(a)} \in \mathbb{F}_q^\lambda$ and the other, say Bob, owns $\mathbf{s}^{(b)} \in \mathbb{F}_q^\lambda$ such that

$$\mathbf{s}^{(a)} + \mathbf{s}^{(b)} = \mathbf{0}, \quad i.e. \quad \mathbf{s}^{(b)} = -\mathbf{s}^{(a)}.$$

Now, $G(\mathbf{s}^{(b)}) = G(-\mathbf{s}^{(a)})$ in $\mathbb{F}_q^{2\lambda+2}$, *i.e.*

$$-G(-\mathbf{s}^{(a)}) + G(\mathbf{s}^{(b)}) = \mathbf{0}.$$

On the other hand, from a share $[\![\mathbf{s}]\!] = (\mathbf{s}^{(a)}, \mathbf{s}^{(b)})$ of a pseudorandom string, $G([\![\mathbf{s}]\!]) \stackrel{\text{def}}{=} (G(\mathbf{s}^{(a)}), G(\mathbf{s}^{(b)}))$ forms a sharing of a longer string $\mathbf{S}$, which is pseudorandom even given the knowledge of one share of $\mathbf{s}$.

Now, assume that the invariant is satisfied by a node ν. One can distinguish two situations:

- If ν is outside of the evaluation path of α, then its labels form a secret sharing of $(\mathbf{0}, 0)$, and the above fact shows that this can be extended to both its children.

- On the other hand, if ν is on the evaluation path, the invariant can be maintained on the only child which is on the evaluation path by carefully choosing the control element (see below).

Therefore, the main difficulty arises when one wants to maintain the invariant upon leaving the evaluation path of α. In order to do that, the idea is to go down along this evaluation path, and *correct* at each level the node which deviates from it. But this correction needs to happen without divulging α to any of the parties. We will proceed by induction. Consider a level i such that the node ν on the evaluation path of α satisfies the invariant. Without loss of generality, let us assume that $\alpha_{i+1} = 1$, *i.e.* the next node on the evaluation path is the right child ν_1 of ν, while its left child ν_0 leaves the path. The case where $\alpha_{i+1} = 0$ is symmetric.

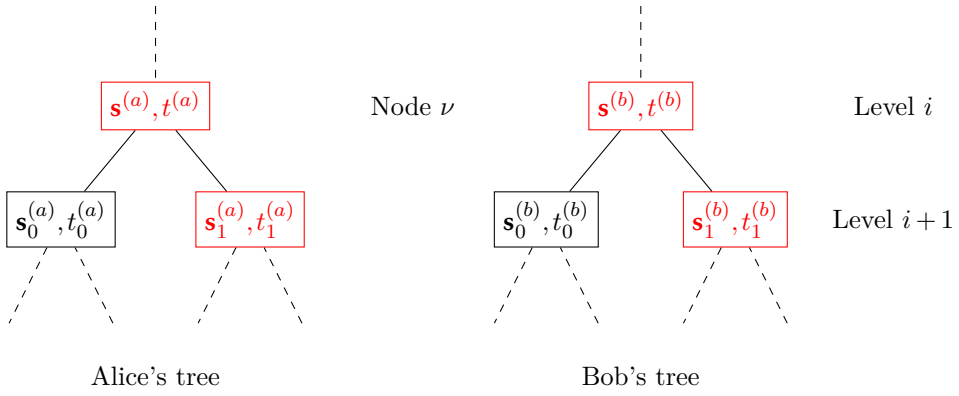


Figure 6.3: Part of the trees corresponding to node ν and its children, before correction.

By hypothesis, the labels of ν on both trees form a secret sharing $(\llbracket \mathbf{s} \rrbracket, \llbracket 1 \rrbracket)$ where $\mathbf{s}$ is a pseudorandom seed, *i.e.* with the notations of Figure 6.3,

$$\begin{cases} \mathbf{s}^{(a)} + \mathbf{s}^{(b)} = \mathbf{s} \\ t^{(a)} + t^{(b)} = 1. \end{cases}$$

As mentioned above, applying G on their seeds allows Alice and Bob to compute an additive sharing of the pseudorandom string $\mathbf{S} \stackrel{\text{def}}{=} -G(-\mathbf{s}^{(a)}) + G(\mathbf{s}^{(b)})$. Now, let $\mathbf{W}^{(i+1)} \in \mathbb{F}_q^{2\lambda+2}$ be publicly known. Then each party can compute an additive sharing $\llbracket \mathbf{S} - \mathbf{W}^{(i+1)} \rrbracket$ by noticing that

$$(-G(-\mathbf{s}^{(a)}) - t^{(a)}\mathbf{W}^{(i+1)}) + (G(\mathbf{s}^{(b)}) - t^{(b)}\mathbf{W}^{(i+1)}) = \underbrace{(-G(-\mathbf{s}^{(a)}) + G(\mathbf{s}^{(b)}))}_{\stackrel{\text{def}}{=}\mathbf{S}} - \underbrace{(t^{(a)} + t^{(b)})}_{=1}\mathbf{W}^{(i+1)} = \mathbf{S} - \mathbf{W}^{(i+1)}$$

Therefore, setting $\mathbf{W}^{(i+1)}$ to be of the form

$$\mathbf{W}^{(i+1)} \stackrel{\text{def}}{=} \mathbf{S} - (\mathbf{0}, 0, \widehat{\mathbf{s}}^{(i+1)}, 1)$$

where $\widehat{\mathbf{s}}^{(i+1)} \leftarrow \mathbb{F}_q^\lambda$ is a new random seed allows the parties to compute an additive sharing of $(\mathbf{0}, 0, \widehat{\mathbf{s}}^{(i+1)}, 1)$, which we define to be the labels of the two children of ν. Moreover, the

pseudorandomness of $\mathbf{S}$ ensures that $\mathbf{W}^{(i+1)}$ is pseudorandom, and in particular does not leak the bit α_{i+1}.

All in all, each key is constituted by

- A pseudorandom seed $\mathbf{s} \in \mathbb{F}_q^\lambda$, and a random additive sharing of 1, which form together the label of the root;
- the collection of the n correction words $\mathbf{W}^{(i)}$ for $i \in \{1,\dots,n\}$.

Note that the $\mathbf{W}^{(i)}$'s can easily be computed by Algorithm Gen which setups the keys (and therefore knows them both) by following the evaluation path of α on both trees and applying the previous iterative construction.

There is still an issue to address here. Indeed, by definition the function secret sharing scheme should computationally hide $f_{\alpha,\beta}$. In other words, the parties are oblivious to the evaluation path of α and cannot know to which node they need to add the correction word. In reality, this is not a problem because they can simply add it to *every node* without changing anything: when ν is not on the evaluation path of α, its labels $(\mathbf{s}^{(a)},t^{(a)})$ and $(\mathbf{s}^{(b)},t^{(b)})$ verify

$$\begin{cases} \mathbf{s}^{(a)} + \mathbf{s}^{(b)} = \mathbf{0} \\ t^{(a)} + t^{(b)} = 0 \end{cases}$$

Therefore

$$(-G(-\mathbf{s}^{(a)}) + t^{(a)}\mathbf{W}) + (G(\mathbf{s}^{(b)}) + t^{(b)}\mathbf{W}) = (-G(\mathbf{s}^{(b)}) + G(\mathbf{s}^{b})) + (t^{(a)} + t^{(b)})\mathbf{W} = \mathbf{0}$$

is still an additive sharing of $\mathbf{0}$.

Remark 6.14. *When $\beta \neq 1$, it suffices to use an additional correction word. Let $(\mathbf{s}^{(a)},t^{(a)})$ and $(\mathbf{s}^{(b)},t^{(b)})$ be the two labels of the leaf corresponding to α. By construction they form a random additive secret sharing $(\llbracket \mathbf{s} \rrbracket, \llbracket 1 \rrbracket)$ where $\mathbf{s}$ is pseudorandom. Similarly to the above construction, set*

$$\mathbf{S} \overset{def}{=} -G(-\mathbf{s}^{(a)}) + G(\mathbf{s}^{(b)}) \quad \text{and} \quad \mathbf{W}^{(n+1)} \overset{def}{=} \mathbf{S} - (\mathbf{0},0,\mathbf{0},\beta).$$

Since $\mathbf{S}$ is pseudorandom, $\mathbf{W}^{(n+1)}$ computationally hides β.

Now, in order to evaluate its share on input $x \in \{0,1\}^n$, each party can follow the path until the leaf corresponding to x, whose control elements form an additive sharing $\llbracket t \rrbracket$ of some $t \in \mathbb{F}_q$.

- *If $x \neq \alpha$, then $t = 0$ and the correction does not do anything: the parties will get shares of the form $(\llbracket \mathbf{0} \rrbracket, \llbracket 0 \rrbracket, \llbracket \mathbf{0} \rrbracket, \llbracket 0 \rrbracket)$.*
- *On the other hand, if $x = \alpha$, then $t = 1$ and the correction yields shares of the form $(\llbracket \mathbf{0} \rrbracket, \llbracket 0 \rrbracket, \llbracket \mathbf{0} \rrbracket, \llbracket \beta \rrbracket)$.*

In both situations, the last component is a share of $f(x)$.

As shown in [BGI16], the above construction can be further optimised to get a key size of at most $\lceil \log N \rceil \cdot (\lambda + 2) + \lambda + \lceil \log q \rceil$ bits, N is the size of the evaluation domain $\mathcal{S}$ of the point function $f_{\alpha,\beta}$, *i.e.* the length of the unitary vectors we want to share.

This construction can be extended to $m > 2$ parties (see [BCGI22]), but the construction is not as efficient as for 2-party.

6.4 Pseudorandom Correlation Generators (PCG)

6.4.1 Generalities

Recall from Section 6.2 that secure computation can often make use of *preprocessing phase* where the parties first run an input-independent protocol to share a lot of random elements having a useful correlation, such as OLE's or (authenticated) Beaver multiplication triples, before engaging in the actual computation of the arithmetic circuit and decreasing the communication complexity. Nonetheless, usual approaches such as introduced in [IKNP03; DPSZ12; KPR18] to generate N correlated random elements still require to communicate $\Omega(N)$ elements, which is quite costly, and actually forms the bottleneck of many MPC protocols. The situation changed recently with the breakthrough results of Boyle *et al* [BCGI18; BCGI+19] who introduced a new tool called a *Pseudorandom Correlation Generator* (PCG) which allows to expand short correlated seeds into a long list of random elements with a target correlation. As mentioned at the end of Section 6.2, the idea is that using a PCG, the communication between the parties can be limited to the generation of the correlated seeds, which the parties can locally expand into a list of OLE's without further communication. In other words, beyond the generation of the seeds, this preprocessing phase is *silent*.

Definition 6.15 (Pseudorandom Correlation Generator (PCG))

Given a target correlation $\mathcal{C}$, a Pseudorandom Correlation Generator is a pair of algorithms (Gen, Expand) where

- Gen is a probabilistic polynomial time algorithm which outputs a pair of *short* correlated random keys (k_0, k_1)
- $\mathsf{Expand}(k_i)$ is a polynomial time algorithm which outputs a long vector R_i such that R_0, R_1 are both pseudorandom, but subject to correlation $\mathcal{C}$.

The above definition requires that k_0 and R_0 be pseudorandom, *i.e.* they should not reveal anything about k_1 and R_1 (except perhaps the correlation), and reciprocally.

Remark 6.16. *When the target correlation $\mathcal{C}$ is simply the equality, then a Pseudorandom Correlation Generator is nothing else than a Pseudorandom Generator where Algorithm* Gen *outputs two equal keys.*

Example 6.17. *A* PCG *for producing N* OLE*'s over $\mathbb{F}_q$ would generate two short keys k_0, k_1 (say of size logarithmic in N) , such that*

$$\mathsf{Expand}(k_0) = ((u_i, x_i))_{i=1,\dots,N} \in (\mathbb{F}_q^2)^N \quad \text{and} \quad \mathsf{Expand}(k_1) = ((v_i, y_i))_{i=1,\dots,N} \in (\mathbb{F}_q^2)^N$$

are pseudorandom, and such that (u_i, x_i, v_i, y_i) has the OLE *correlation (Definition 6.6) for each i, i.e.*

$$\forall i \in \{1, \dots, N\}, \quad u_i \cdot v_i = x_i + y_i.$$

Using a PCG, the preprocessing phase can now be described as follows:

1. First, the parties securely distribute the short seeds, using a small amount of work, which is often independent from the circuit size.

2. Then, the parties can locally stretch their seeds into long correlated vectors, without further communication.

Since most of the preprocessing is pushed *offline*, this model of secure computation is often called the *silent preprocessing model*.

6.4.2 Programmable PCG's

In the above presentation, we restricted ourselves to the case of 2-party Pseudorandom Correlation Generators such as for the OLE correlation. While this is enables 2-party secure computation, it is *a priori* not clear how to extend it to the m-party setting, with $m > 2$.

Recall that the goal is in the end produce Beaver triples as per Definition 6.5. In Section 6.2.3 we explained how two OLE's can be transformed into one beaver triple shared between two parties. In fact, this can be extended to the m-party setting by computing two OLE's between all the parties, as long as we can manage to get the *crossed-terms*.

More precisely, consider m player $P_1, \ldots, P_m$ wanting to perform a secure computation. It all boils down to generating m-parties Beaver triples of the form $(u_i, v_i, w_i)_{1 \leqslant i \leqslant m}$ such that

$$\left(\sum_{i=1}^{m} u_i\right) \cdot \left(\sum_{i=1}^{m} v_i\right) = \sum_{i=1}^{m} w_i.$$

The idea is two get two OLE's *for each pair* of players $\{P_i, P_j\}$ with $i \neq j$. Indeed, assume that each party P_i get assigned two random values u_i, v_i and are able to get shares of the products $u_i v_j$ for $i \neq j$: *i.e.* for each pair $\{P_i, P_j\}$, party P_i gets $\gamma_{i,j}$ and $\delta_{j,i}$ such that

$$u_i v_j = \gamma_{i,j} + \delta_{i,j}$$

and

$$u_j v_i = \gamma_{j,i} + \delta_{j,i}.$$

Each party P_i can then compute

$$w_i \stackrel{\text{def}}{=} u_i v_i + \sum_{j \neq i} \gamma_{i,j} + \sum_{j \neq i} \delta_{j,i}.$$

Finally, notice that

$$\sum_{i=1}^{m} w_i = \sum_{i=1}^{m} u_i v_i + \sum_{i=1}^{m} \sum_{j \neq i} \gamma_{i,j} + \sum_{i=1}^{m} \sum_{j \neq i} \delta_{j,i} \tag{6.2}$$

$$= \sum_{i=1}^{m} u_i v_i + \sum_{i=1}^{m} \sum_{j \neq i} (\gamma_{i,j} + \delta_{i,j}) \quad \text{(after rearranging the terms)} \tag{6.3}$$

$$= \sum_{i=1}^{m} \sum_{j=1}^{m} u_i v_j = \left(\sum_{i=1}^{m} u_i\right) \left(\sum_{j=1}^{m} v_j\right). \tag{6.4}$$

The main difficulty here is that *a priori*, the OLE's generated by a PCG are *independent*. Fortunately, [BCGI+19] introduced the notion of *programmable* PCG's, that can be instantiated to get a correlation with the desired factors. More formally, a programmable PCG for OLE is a PCG such that:

- Algorithm $\mathsf{Gen}(a',b')$ takes two additional random seeds a',b' and outputs two correlated keys (k_0,k_1).
- Algorithm $\mathsf{Expand}(k_0)$ outputs (a,x) and $\mathsf{Expand}(k_1)$ outputs (b,y) such that

  $$a \cdot b = x + y \quad (\mathsf{OLE}\text{ correlation})$$

  and there exist *deterministic* (and efficiently computable) functions f_0 and f_1 such that

  $$a = f_0(a') \quad \text{and} \quad b = f_1(b') \quad (\text{programmability})$$

- To ensure *security*, the distribution

  $$\left\{ (k_1, b', f_0(a')) \right\} \quad \text{where } a', b' \text{ are independent random seeds, and } (\cdot, k_1) \leftarrow \mathsf{Gen}(a', b')$$

  is computationally indistinguishable from the distribution

  $$\left\{ (k_1, b', f_0(\tilde{a})) \right\} \quad \text{where } a', b', \tilde{a} \text{ are independent random seeds, and } (\cdot, k_1) \leftarrow \mathsf{Gen}(\tilde{a}, b')$$

  Obviously, a symmetric indistinguishability statement should hold.

Now, with a programmable PCG the cross terms can be handled as follows: in the setup phase, the trusted dealer

1. samples $a'_1, \ldots, a'_m$ and $b'_1, \ldots, b'_m$ uniformly at random,
2. generates keys $(k_0^{i,j}, k_1^{i,j}) \leftarrow \mathsf{Gen}(a'_i, b'_j)$ for $i \neq j$,
3. and gives to party P_{i_0} the following key k_{i_0}:

$$k_{i_0} \stackrel{\text{def}}{=} \left(\{k_0^{i_0,j}\}_{j \neq i_0}, \{k_1^{j,i_0}\}_{j \neq i_0} \right)$$

One can easily check that yields exactly the wanted setting with $u_i \stackrel{\text{def}}{=} f_0(a'_i)$ and $v_i \stackrel{\text{def}}{=} f_1(b'_i)$.

The indistinguishability hypothesis in the definition is there to ensure that reusing one of the factors does not leak too much information in the protocol.

Remark 6.18. *In reality, in the general framework of [BCGI+19, Theorem 41] to turn a programmable* PCG *for a given bilinear correlation, into an m-party* PCG *requires additional randomness to ensure security, which can be added to the keys as seeds to some Pseudorandom Generator. We ignored them in the above presentation.*

6.4.3 A template for generating OLE's

Since the introduction of the notion of Pseudorandom Correlation Generators, some works have presented different ways to build random OT's over $\mathbb{F}_2$. However, those PCG's are not programmable and are not known to extend to the m-party setting. On the other hand, [BCGI+20b] presented a template to design programmable PCG's for the OLE correlation. However, they could only instantiate their construction over fields $\mathbb{F}_q$ larger than the number of OLE's to be

produced. Nonetheless, this protocol is general enough as long as some underlying rings have the right property. In particular, this protocol is used in [BCCD23] which is one contribution of this thesis and which will be presented in Chapter 7.

One OLE to rule them all. Let $\mathcal{R}$ be a finite ring, isomorphic to a product of N copies of $\mathbb{F}_q$. The idea of [BCGI+20b] is that by applying this isomorphism, it suffices to build *only one* OLE over the ring $\mathcal{R}$ to produce N OLE's over $\mathbb{F}_q$: if $\mathbf{U},\mathbf{V},\mathbf{X} \in \mathcal{R}$ are pseudorandom $\mathcal{R}$ and $\mathbf{Y} = \mathbf{U}\cdot\mathbf{V} - \mathbf{X}$, then this relation will be satisfied by any component over $\mathbb{F}_q$. In other words, $(\mathbf{U},\mathbf{X})$ and $(\mathbf{V},\mathbf{Y})$ can be distributed the two parties, and the local expansion simply consists in computing and applying the isomorphism. It only remains to see how to describe them succinctly.

The construction. Their idea was to set $\mathcal{R} \stackrel{\text{def}}{=} \mathbb{F}_q[X]/(F(X))$ where F is a polynomial of degree N totally splitting in $\mathbb{F}_q$. In particular, it entails that $q \geqslant N$ for this to work. Now, assume that a trusted dealer samples an element $\mathbf{a} \in \mathcal{R}$ uniformly at random, as well as some *sparse* polynomials $\mathbf{e}_u.\mathbf{f}_u$ and $\mathbf{e}_v,\mathbf{f}_v$ of degree at most N, where the sparsity means that they only have at most t non-zero coefficients. Denote also by $\mathbf{a}$ a lift of degree at most N in $\mathbb{F}_q[X]$. Let

$$\mathbf{U} \stackrel{\text{def}}{=} \mathbf{a}\cdot\mathbf{e}_u + \mathbf{f}_u \in \mathbb{F}_q[X] \quad \text{and} \quad \mathbf{V} \stackrel{\text{def}}{=} \mathbf{a}\cdot\mathbf{e}_v + \mathbf{f}_v \in \mathbb{F}_q[X]$$

Then, the product

$$\mathbf{U}\cdot\mathbf{V} = \mathbf{a}^2(\mathbf{e}_u\mathbf{e}_v) + \mathbf{a}(\mathbf{e}_u\mathbf{f}_v + \mathbf{e}_v\mathbf{f}_u) + \mathbf{f}_u\mathbf{f}_v \in \mathbb{F}_q[X]$$

is now a linear combination of t^2-sparse polynomials. In other words, the polynomial $\mathbf{U}$ (resp. $\mathbf{V}$) can be succinctly described by

- a short seed which can be expanded into $\mathbf{a}$ using a pseudorandom generator,
- and by the $2t$ non-zero coefficients of $\mathbf{e}_u$ and $\mathbf{f}_u$ (resp. $\mathbf{e}_v$ and $\mathbf{f}_v$)

Moreover, since the cross products are t^2-sparse polynomials, they can be described by the t^2-sparse vector of their coefficients, which can be distributed to the parties using Function Secret Sharing for a sum of t^2 point functions, as described in Section 6.3.

Wrapping up, the final PCG looks as follows:

- Algorithm Gen, which is the setup phase of the protocol, samples a random seed $\sigma_{\mathbf{a}}$, as well as random monomials $(X^{i_{1,j}},\dots,X^{i_{t,j}})$ where $0 \leqslant i_{\ell,j} < N$ are distinct and random coefficient vectors $(\gamma^{i_{1,j}},\dots,\gamma^{i_{t,j}}) \in (\mathbb{F}_q^\times)^t$ for $j \in \{0,\dots,3\}$ which define the t-sparse polynomials $\mathbf{e}_u,\mathbf{e}_v,\mathbf{f}_u,\mathbf{f}_v$.

  It then generates FSS keys $(K_0^{i,j},K_1^{i,j})$ for $i,j \in \{0,\dots,3\}$ corresponding to the four cross products, and outputs

$$k_0 \stackrel{\text{def}}{=} \left(\sigma_{\mathbf{a}}, \{K_0^{i,j}\}, \{X^{i_{1,j}},\dots,X^{i_{t,j}}\}_{j=0,2}, \{\gamma^{i_{1,j}},\dots,\gamma^{i_{t,j}}\}_{j=0,2}\right)$$
$$k_1 \stackrel{\text{def}}{=} \left(\sigma_{\mathbf{a}}, \{K_1^{i,j}\}, \{X^{i_{1,j}},\dots,X^{i_{t,j}}\}_{j=1,3}, \{\gamma^{i_{1,j}},\dots,\gamma^{i_{t,j}}\}_{j=1,3}\right).$$

- Algorithm Expand performed by each party, can now take one of the keys, say k_0, compute $\mathbf{a}$ with the seed $\sigma_{\mathbf{a}}$, compute $\mathbf{e}_u,\mathbf{f}_u$ with the monomials and coefficients and set

$$\mathbf{U} \stackrel{\text{def}}{=} \mathbf{a}\cdot\mathbf{e}_u + \mathbf{f}_u. \tag{6.5}$$

With the keys $K_0^{i,j}$, it can compute additive shares of the cross products $[\![\mathbf{e}_u\mathbf{e}_v]\!]$, $[\![\mathbf{e}_u\mathbf{f}_v]\!]$, $[\![\mathbf{f}_u\mathbf{e}_v]\!]$, $[\![\mathbf{f}_u\mathbf{f}_v]\!]$, and set

$$\mathbf{X} \stackrel{\text{def}}{=} \mathbf{a}^2[\![\mathbf{e}_u\mathbf{e}_v]\!] + \mathbf{a}\,([\![\mathbf{e}_u\mathbf{f}_v]\!] + [\![\mathbf{f}_u\mathbf{e}_v]\!]) + [\![\mathbf{f}_u\mathbf{f}_v]\!], \tag{6.6}$$

where we identify a polynomial with the vector of its coefficients. Equations 6.5 and 6.6 define $\mathbf{U}$ and $\mathbf{X}$ as polynomials in $\mathbb{F}_q[X]$, which can then be reduced modulo $F(X)$. We also denote them by $\mathbf{U},\mathbf{X}$ after reduction.

Finally, it computes the Chinese isomorphism

$$\varphi\colon \mathbb{F}_q[X]/(F(X)) \to \mathbb{F}_q \times \cdots \times \mathbb{F}_q,$$

and outputs $\varphi(\mathbf{U}), \varphi(\mathbf{X})$.

The expansion of k_1 can be similarly described to output $\varphi(\mathbf{V}), \varphi(\mathbf{Y})$. Note that, by construction the equation

$$\mathbf{U}\cdot\mathbf{V} \stackrel{\text{def}}{=} \mathbf{X} + \mathbf{Y}$$

holds in $\mathbb{F}_q[X]$, and therefore in $\mathcal{R}$. Since φ is a ring isomorphism, we have

$$\varphi(\mathbf{U}) \star \varphi(\mathbf{V}) = \varphi(\mathbf{X}) + \varphi(\mathbf{Y}) \in \mathbb{F}_q^N$$

where the product $\star$ is taken coefficient by coefficient. Moreover, it maps the uniform distribution in $\mathbb{F}_q[X]/(F(X))$ to the uniform distribution in $\mathbb{F}_q^N$. In particular, for the OLE's over $\mathbb{F}_q$ to be pseudorandom, it suffices to have $\mathbf{U}$ and $\mathbf{V}$ pseudorandom. Note that $\mathbf{X}$ and $\mathbf{Y}$ are already pseudorandom by the properties of a secure FSS scheme for a sum of t^2 point functions.

Remark 6.19. A priori, *the cross-products are only t^2-sparse when seen as elements of $\mathbb{F}_q[X]$, but the reduction modulo $F(X)$ can mess this up. In Chapter 7 we use* group algebras*, for which it is possible to define the products and the sparsity directly in the ring $\mathcal{R}$, which results in a* slightly *more efficient scheme.*

Note that this PCG is programmable, since the vectors of monomials and coefficients define deterministically $\mathbf{U}$ and $\mathbf{V}$. In other words, reusing them allows to reuse the factors to define multi-party multiplication triples as explained in Section 6.4.2. It turns out that this property is *black-box* in the ring $\mathcal{R}$ as long as there is a notion of sparsity compatible with multiplication. This will extend naturally to the setting of Chapter 7.

Pseudorandomness. It only remains to see why $\mathbf{U}$ and $\mathbf{V}$ are pseudorandom. Note that this is exactly an instance of the (decisional version of) Ring-LPN, as per Definition 4.47. At the time of writing [BCGI+20b], this *totally-split* Ring-LPN was a rather new hypothesis in cryptography, and they assessed the security by studying different kinds of attacks, such as folding attacks. However, those attacks are for the search version, and the link between the search and decisional versions was not clear at the time of [BCGI+20b].

Let us consider a different approach, making use of the function field framework developed in Chapter 4 and the FF-DP problem. Indeed, since $F(X)$ is totally split in $\mathbb{F}_q$, this corresponds exactly to the situation described in the aforementioned chapter, and the main task boils down to finding a Galois function field $K/\mathbb{F}_q(T)$ such that

$$\mathbb{F}_q[X]/(F(X)) \simeq \mathcal{O}_K/\mathfrak{p}.$$

This is exactly was is done in Section 4.4.3 with the polynomial $F(X) \stackrel{\text{def}}{=} X^{q-1} - 1$ and the ring

$$\mathbb{F}_q[X] \Big/ (X^{q-1} - 1)$$

which identifies as the quotient of the ring of integers of a Carlitz extension of $\mathbb{F}_q(T)$. With this choice of ring, the decisional version is therefore equivalent to the search version.

Remark 6.20. *In Chapter 7 we identify a weakness that was overlook in [BCGI+20b]: not all choices of totally split polynomials are suitable. In particular, $F(X) \stackrel{def}{=} X^q - X$ yields an insecure instance of the problem: reduction modulo X yields an easy distinguisher. This is similar to* evaluation at one *attacks on some instantiations of Polynomial-LWE [CIV16].*

A note on efficiency.

- For MPC applications, we also care on the efficiency of the PCG's built. With this template, we need to have an efficient multiplication in the ring $\mathcal{R}$. As a consequence, it was proposed in [BCGI+20b] to use a ring of the form

  $$\mathcal{R} \simeq \mathbb{F}_p[X] \Big/ (X^{2^\ell} + 1),$$

  where p is a prime of the form $2^{\ell+1} + 1$[ii]. Indeed, this kind of rings has been used heavily in lattice-based cryptography, as well as for pairing-friendly elliptic curves, and FFT-based multiplication in $\mathcal{R}$, which is often called NTT for Number Theoretic Transform in this situation, has been heavily optimised. It turns out that this ring identifies as the quotient of the ring of integers of a cyclotomic number field by the ideal generated by the prime p. In other words, the choice

  $$\mathcal{R} \stackrel{\text{def}}{=} \mathbb{F}_q[X] \Big/ (X^{q-1} - 1)$$

  is yet another example of this number field - function field analogy.

- Boyle *et al* also propose to use a variant of Ring-LPN where the error distribution is supposed to be regular, *i.e.* when the error vector can be split into t pieces of size $\approx N/t$ such that each piece has exactly one non-zero coefficient. This is not known to yield much insecurity, while on the other hand improving the sharing of the cross-products by a factor of t. This will extend *mutatis mutandis* to the setting of Chapter 7.

[ii] In [BCGI+20b], p is chosen to be a 128 bit prime

Chapter 7

Pseudorandom Correlation Generators from the Quasi-Abelian Decoding Problem

As explained in Chapter 6, and more particularly in Section 6.4.3, the state-of-the-art construction of *programmable Pseudorandom Correlation Generators* (PCG's) for the OLE correlation is due to Boyle *et al* [BCGI+20b]. However, it only works over large fields, and the security is not well understood.

Contributions of this thesis. In this chapter, based on [BCCD23] and published at CRYPTO 2023, we revisit this approach using structured codes. In particular, we give stronger foundations to the security, and by introducing *quasi-abelian codes* we extend the construction to any finite field $\mathbb{F}_q$ with $q > 2$. This is the first time that such structured codes are considered in this generality for cryptographic applications. Finally, at the end of the chapter we propose some directions towards a complete solution when $q = 2$.

Outline of the current chapter

7.1 Introduction

Recall from Chapter 6 that the recent *silent preprocessing* model introduced in [BCGI+19] consists in using Pseudorandom Correlation Generators (PCG's) to generate and distribute a small amount of correlated (pseudo)random seeds to all the participants, who can then expand them into long strings having a useful correlation such as Oblivious Transfers (OT's) over the binary field $\mathbb{F}_2$, or their generalisation to larger fields, namely Oblivious Linear Evaluations (OLE's).

Furthermore, some special PCG's endowed with an additional *programmability* property can be used to provide useful m-party correlations with $m > 2$, enabling multi-party computation.

Two lines of work have been able to provide PCG's for the OLE correlation, but are different in nature:

- Initiated with [BCGI+19], which proposed a protocol based on the hardness of decoding some classes of random codes, some follow-up work such as [CRR21] and [BCGI+22] proposed different choices of codes to obtain very efficient constructions ale to generate millions of random OT's per second on one core of a standard laptop. However, those PCG's are inherently *non-programmable*, and therefore getting to the m-party setting induces a very large communication overhead, and this is limited to computations over the binary field $\mathbb{F}_2$.
- On the other hand, [BCGI+20b] introduced a general template for building *programmable* PCG's for the OLE correlation, but their construction only allows to create OLE's over very large finite fields, larger than the number of OLE's one wants to generate. For example, in their paper they proposed to instantiate their construction over prime fields $\mathbb{F}_p$ with 128-bit primes. Moreover, this construction relies on some Ring-LPN assumption which lacked strong foundational background.

In this Chapter, based on [BCCD23] which is one of the contributions of this thesis, we present a way to overcome both limitations of [BCGI+20b]. Basically, we would like to use the template from [BCGI+20b] and recalled in Section 6.4.3. In order to do that, we need an effective ring $\mathcal{R}$, isomorphic to a product of many copies of $\mathbb{F}_q$ for any small q, and such that we can define a Ring-LPN problem as secure as possible. Furthermore, for this to be used in practice, we need the multiplication of $\mathcal{R}$ to be efficiently implementable.

A natural idea to generalise the construction of [BCGI+20b] based on univariate polynomial rings, is to consider multivariate polynomial rings. For example, the ring

$$\mathcal{R} \stackrel{\text{def}}{=} \mathbb{F}_2[X_1,\ldots,X_t] \Big/ (X_1^2 - X_1,\ldots,X_t^2 - X_t)$$

of boolean functions is isomorphic to a product of 2^t copies of $\mathbb{F}_2$, and the multiplication of two boolean functions can be implemented very efficiently. However, as we will see later in this chapter, this choice of ring, as natural as it looks, leads to an insecure instantiation.

This leads us to consider other classes of structured codes, endowed with the action by permutation of more general abelian groups, namely *quasi-abelian codes*, which were introduced in [Was77]. As a result, we are able to build programmable PCG's for the OLE correlation for any finite field as small as $\mathbb{F}_3$, and propose a research direction towards building programmable PCG's for OT over $\mathbb{F}_2$.

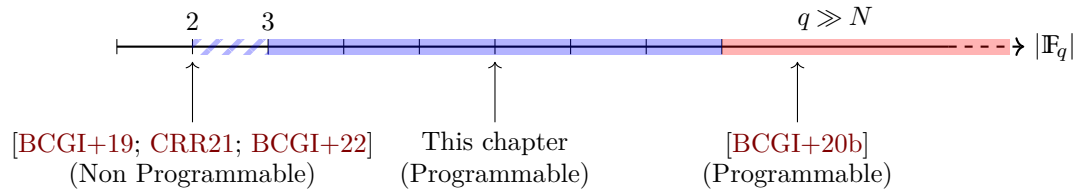


Figure 7.1: Landscape picture of PCG constructions for the OLE correlation

7.2 Group Algebras and Quasi-Abelian Codes

7.2.1 Group Algebras

Let $\mathbb{F}_q$ be the finite field with q elements, and let G be a finite group. The *group algebra* of G with coefficients in $\mathbb{F}_q$, denoted by $\mathbb{F}_q[G]$, is the free algebra generated by G:

$$\mathbb{F}_q[G] \stackrel{\text{def}}{=} \left\{ \sum_{g \in G} a_g g \;\middle|\; a_g \in \mathbb{F}_q \right\},$$

endowed with a natural $\mathbb{F}_q$-vector space structure, and the product given by the convolution

$$\left(\sum_{g \in G} a_g g\right)\left(\sum_{g \in G} b_g g\right) \stackrel{\text{def}}{=} \sum_{g \in G}\left(\sum_{h \in G} a_h b_{h^{-1}g}\right) g.$$

Remark 7.1. *An element of $\mathbb{F}_q[G]$ can also be considered as a function from G to $\mathbb{F}_q$.*

Although in this definition the group G is not required to be abelian, we will make this assumption for the applications of this chapter. In this case, $\mathbb{F}_q[G]$ is commutative.

Remark 7.2. *Group algebras are deeply related to (linear) representations of G (see for instance [Dor71; DK12]). Indeed, a linear representation of G is an $\mathbb{F}_q$-vector space, together with a group homomorphism*

$$\rho : G \to \mathrm{GL}(V).$$

Note that for any representation (V, ρ), we can define a natural action of G on V by

$$g \cdot v \stackrel{def}{=} \rho(g)(v),$$

which can be extended by linearity to $\mathbb{F}_q[G]$*. In other words,* (V, ρ) *is naturally endowed with a structure of a (left)* $\mathbb{F}_q[G]$*-module (this works similarly on the right hand side). On the other hand, any* $\mathbb{F}_q[G]$*-module is endowed with an* $\mathbb{F}_q$*-vector space structure, and the action of* G *is linear. Therefore, it defines an* $\mathbb{F}_q$*-representation of* G*. Additionally, morphisms of representations are in bijection with morphism of group algebras. In other words, there is an* equivalence of categories *between the category of* $\mathbb{F}_q[G]$*-modules, and that of* $\mathbb{F}_q$*-linear representations of* G*.*

For example, $\mathbb{F}_q[G]$ *seen as a (left or right) module over itself corresponds to the (left or right)* regular representation *of* G*.*

Fix an ordering $g_0, \dots, g_{n-1}$ where $n \stackrel{\text{def}}{=} |G|$. Then, the group algebra $\mathbb{F}_q[G]$ is isomorphic (as $\mathbb{F}_q$-vector spaces) to $\mathbb{F}_q^n$ via $\varphi : \sum_{i=0}^{n-1} a_i g_i \mapsto (g_0, \dots, g_{n-1})$. This allows to transport the Hamming metric from $\mathbb{F}_q^n$ to $\mathbb{F}_q[G]$ by defining the weight of $\mathbf{a} \in \mathbb{F}_q[G]$ by that of $\varphi(\mathbf{a})$. Note that the isomorphism depends on the chosen order. However, changing it only induces a permutation of the coordinates, and therefore the Hamming metric is well defined on $\mathbb{F}_q[G]$. In particular, the natural action of G on $\mathbb{F}_q[G]$ is by isometries.

Remark 7.3. *Many groups, especially* abelian *groups come with a canonical ordering, so in general there will be natural way of writing the elements of* $\mathbb{F}_q[G]$*.*

Example 7.4. *In order to understand further the notion of the algebra of an abelian group, let us start with the simplest case of cyclic groups.*

- *When* $G \stackrel{def}{=} \{1\}$ *is the trivial group, then the group algebra* $\mathbb{F}_q[G]$ *is exactly the finite field* $\mathbb{F}_q$*.*
- *When* $G \stackrel{def}{=} \mathbb{Z}/n\mathbb{Z}$ *is the cyclic group with* n *elements, then an element* $a \in \mathbb{F}_q[G]$ *is of the form*

$$a \stackrel{def}{=} \sum_{i=0}^{n-1} a_i \cdot i.$$

Note that the action of G *corresponds to the cyclic shift. In other words,*

$$\mathbb{F}_q\left[\mathbb{Z}/n\mathbb{Z}\right] \simeq \mathbb{F}_q[X]/(X^n - 1)$$

via $i \mapsto X^i$*, extended by* $\mathbb{F}_q$*-linearity.*

Example 7.4 shows that the cyclic group algebra can be identified to the quotient of a univariate polynomial ring. The following proposition which can be shows that this fact extends to general abelian groups, at the price of using more variables.

Proposition 7.5

Let G_1, G_2 be two finite groups. Then

$$\mathbb{F}_q[G_1 \times G_2] \simeq \mathbb{F}_q[G_1] \otimes_{\mathbb{F}_q} \mathbb{F}_q[G_2].$$

Now, consider any abelian group. It is isomorphic to a product of cyclic groups

$$G \simeq \mathbb{Z}/n_1\mathbb{Z} \times \cdots \times \mathbb{Z}/n_t\mathbb{Z}$$

where the n_i are not necessarily distinct. Therefore,

$$\begin{aligned}\mathbb{F}_q[G] &\simeq \mathbb{F}_q\left[\mathbb{Z}/n_1\mathbb{Z}\right] \otimes_{\mathbb{F}_q} \cdots \otimes_{\mathbb{F}_q} \mathbb{F}_q\left[\mathbb{Z}/n_t\mathbb{Z}\right] \simeq \mathbb{F}_q[X]/X^{n_1}-1 \otimes_{\mathbb{F}_q} \cdots \otimes_{\mathbb{F}_q} \mathbb{F}_q[X]/X^{n_t}-1 \\ &\simeq \mathbb{F}_q[X_1,\ldots,X_t]/(X^{n_1}-1,\ldots,X^{n_t}-1).\end{aligned} \tag{7.1}$$

Remark 7.6. *The above isomorphism can actually be made explicit by* $(k_1,\ldots,k_t) \mapsto X_1^{k_1}\ldots X_t^{k_t}$, *extended by linearity.*

7.2.2 Quasi-Abelian Codes

Let G be a finite group of size n, with an implicit ordering $g_0,\ldots,g_{n-1}$. Let $\ell > 0$ be any positive integer, and consider the free $\mathbb{F}_q[G]$-module of rank ℓ defined by

$$(\mathbb{F}_q[G])^\ell \stackrel{\text{def}}{=} \mathbb{F}_q[G] \oplus \cdots \oplus \mathbb{F}_q[G] = \Big\{(a_1,\ldots,a_\ell) \mid a_i \in \mathbb{F}_q[G]\Big\}.$$

Definition 7.7 (Quasi-group code)

A *quasi-group code* of G of index ℓ (or ℓ-quasi-G code for short, or even quasi-G code when the index does not matter) is *any* (right) $\mathbb{F}_q[G]$-submodule of $(\mathbb{F}_q[G])^\ell$. When G is abelian, a quasi-G code is called a *quasi-abelian* code.

Remark 7.8. *Considering right modules in the above definition is mostly a matter of conventions. It does not really matter in this chapter since everything will be abelian in the end.*

More precisely, given a matrix

$$\mathbf{\Gamma} = \begin{pmatrix} \gamma_{1,1} & \cdots & \gamma_{1,\ell} \\ \vdots & \ddots & \vdots \\ \gamma_{k,1} & \cdots & \gamma_{k,\ell} \end{pmatrix} \in (\mathbb{F}_q[G])^{k\times\ell},$$

the quasi-G code generated by $\mathbf{\Gamma}$ is

$$\mathscr{C} \stackrel{\text{def}}{=} \{\mathbf{m}\mathbf{\Gamma} = (\mathbf{m}\mathbf{\Gamma}_1,\ldots,\mathbf{m}\mathbf{\Gamma}_\ell) \mid \mathbf{m} = (m_1,\ldots,m_k) \in (\mathbb{F}_q[G])^k\},$$

where $\mathbf{\Gamma}_i$ denotes the column $\begin{pmatrix}\gamma_{1,i}\\ \vdots \\ \gamma_{k,i}\end{pmatrix}$ and $\mathbf{m}\mathbf{\Gamma}_i = m_1\gamma_{1,i}+\cdots+m_k\gamma_{k,i} \in \mathbb{F}_q[G]$. The matrix $\mathbf{\Gamma}$ is said to be *systematic* if it is of the form $\mathbf{\Gamma} = \big(I_k \mid \mathbf{\Gamma}'\big)$, where $\mathbf{\Gamma}' \in (\mathbb{F}_q[G])^{k\times(\ell-k)}$ and $I_k \in (\mathbb{F}_q[G])^{k\times k}$ is the diagonal matrix with values 1_G. In other words, a quasi-group code $\mathscr{C}$ is nothing else than a code defined over the group algebra $\mathbb{F}_q[G]$. We give more concrete instantiations in Example 7.10. Note that in particular such a code is $\mathbb{F}_q$-linear. More precisely, for $a \stackrel{\text{def}}{=} \sum_{i=0}^{n-1} a_i g_i \in \mathbb{F}_q[G]$,

represented by the vector $\varphi(a) \stackrel{\text{def}}{=} (a_0, \ldots, a_{n-1}) \in \mathbb{F}_q^n$, consider the square matrix

$$M_a \stackrel{\text{def}}{=} \begin{pmatrix} \varphi(a \cdot g_0) \\ \vdots \\ \varphi(a \cdot g_{n-1}) \end{pmatrix} \in \mathbb{F}_q^{n \times n},$$

where each row is the vector representation of a shift of a by some element $g_i \in G$. The matrix M_a represents the multiplication-by-a map $m \mapsto a \cdot m$ in the basis given by G. In particular, for $a, m \in \mathbb{F}_q[G]$, the vector representation of the product $m \cdot a$ is the vector-matrix product

$$\varphi(m \cdot a) = \varphi(m) M_a = (m_0, \ldots, m_{n-1}) \begin{pmatrix} \varphi(a \cdot g_0) \\ \vdots \\ \varphi(a \cdot g_{n-1}) \end{pmatrix}.$$

Therefore, given an ℓ-quasi-G code $\mathscr{C}$, one can *unroll* its generator matrix $\mathbf{\Gamma} \in \mathbb{F}_q[G]^{k \times \ell}$ into a matrix of $\mathbb{F}_q^{kn \times \ell n}$ formed out by $k \times \ell$ square blocks of size n, representing the multiplication by the corresponding element $\gamma_{i,j}$.

$$\mathbf{\Gamma} = \begin{pmatrix} \gamma_{1,1} & \cdots & \gamma_{1,\ell} \\ \vdots & \ddots & \vdots \\ \gamma_{k,1} & \cdots & \gamma_{k,\ell} \end{pmatrix} \in \mathbb{F}_q[G]^{k \times \ell}$$

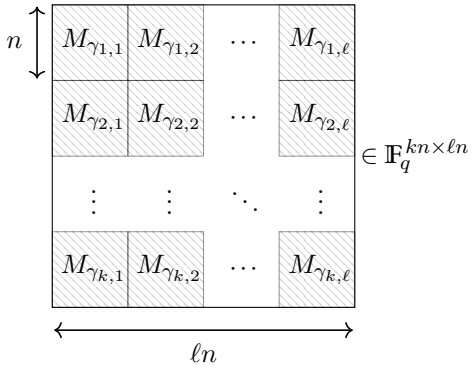


Figure 7.2: Generator matrix of a quasi-G code.

Remark 7.9. *Figure 7.2 illustrates what the generator matrix of a quasi-G code looks like, when unrolled as a matrix with coefficients in $\mathbb{F}_q$. In particular, we can see a striking resemblance with the structured matrix defining the module-LWE problem in lattice-based cryptography (see [Pel19, Figure 0.7]).*

We shall pause here for a moment. Let G be a finite abelian group. In this section, we defined a quasi-abelian code as an $\mathbb{F}_q[G]$-submodule of $\mathbb{F}_q[G]^\ell$. This is particularly suitable to define *random* quasi-abelian codes. More generally, we could consider instead abstract $\mathbb{F}_q[G]$-modules of finite presentation. By the structure theorem for finite abelian groups, G is isomorphic to a direct product of cyclic-groups

$$G \simeq \mathbb{Z}/n_1\mathbb{Z} \times \cdots \times \mathbb{Z}/n_r\mathbb{Z},$$

where the n_i are not necessarily distinct. As we have seen in Equation (7.1), this entails that

$$\mathbb{F}_q[G] \simeq \mathbb{F}_q\left[\mathbb{Z}/n_1\mathbb{Z}\right] \otimes_{\mathbb{F}_q} \cdots \otimes_{\mathbb{F}_q} \mathbb{F}_q\left[\mathbb{Z}/n_r\mathbb{Z}\right].$$

In particular, $\mathbb{F}_q[G]$ can also be considered as a module over *any* factor $\mathbb{F}_q[\mathbb{Z}/n_i\mathbb{Z}]$, and therefore a quasi-abelian code is a quasi-cyclic code with block length corresponding to any invariant factor n_i. However, beware: a *random* quasi-abelian code, is *not* a *random* quasi-cyclic code. See Example 7.10:(iii).

Example 7.10. *Let us continue with Example 7.4.*

(i) When $G \stackrel{def}{=} \{1\}$ *is the trivial group, then* any *linear code is a quasi-*G *code.*

(ii) When $G \stackrel{def}{=} \mathbb{Z}/n\mathbb{Z}$ *is the cyclic group with* n *elements and* q *is coprime to* n*, any element of* $\mathbb{F}_q[G] \simeq \mathbb{F}_q[X]/(X^n-1)$ *is a polynomial of degree at most* n *defined by the vector of its coefficients, and any product* $m(X)\cdot a(X) \in \mathbb{F}_q[G]$ *can be represented by the* circulant *vector-matrix product*

$$\begin{pmatrix} m_0 & m_1 & \dots & m_{n-1} \end{pmatrix} \begin{pmatrix} a_0 & a_1 & \dots & a_{n-1} \\ a_{n-1} & a_0 & \dots & a_{n-2} \\ \vdots & & & \vdots \\ a_1 & a_{n-1} & \dots & a_0 \end{pmatrix} \in \mathbb{F}_q^n.$$

For simplicity, assume that $k=1$ *and* $\ell=2$*. Then, a quasi-*$\mathbb{Z}/n\mathbb{Z}$ *code of index* 2 *is defined over* $\mathbb{F}_q$ *by a double-circulant generator matrix*

$$\left(\begin{array}{cccc|cccc} a_0 & a_1 & \dots & a_{n-1} & b_0 & b_1 & \dots & b_{n-1} \\ a_{n-1} & a_0 & \dots & a_{n-2} & b_{n-1} & b_0 & \dots & b_{n-2} \\ \vdots & & & \vdots & \vdots & & & \vdots \\ a_1 & a_{n-1} & \dots & a_0 & b_1 & b_{n-1} & \dots & b_0 \end{array}\right).$$

In other words, a quasi-$\mathbb{Z}/n\mathbb{Z}$ *code is nothing else than a usual* quasi-cyclic *code with block length* n*.*

(iii) Let us do an example with an abelian, non cyclic group $G \stackrel{def}{=} \mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}/2\mathbb{Z}$*. Ordering* G *with the lexicographic order, an element of* $\mathbb{F}_q[G]$ *is of the form*

$$a = a_0(0,0) + a_1(0,1) + a_2(1,0) + a_3(1,1).$$

The action of G *on* a *is given by*

$$\begin{aligned} (0,1)\cdot a &= a_1(0,0) + a_0(0,1) + a_3(1,0) + a_2(1,1) \\ (1,0)\cdot a &= a_2(0,0) + a_3(0,1) + a_0(1,0) + a_1(1,1) \\ (1,1)\cdot a &= a_3(0,0) + a_2(0,1) + a_1(1,0) + a_0(1,1) \end{aligned}$$

Therefore, a quasi-G *code will have a generator matrix formed out by blocks of the form*

$$\begin{pmatrix} a_0 & a_1 & a_2 & a_3 \\ a_1 & a_0 & a_3 & a_2 \\ a_2 & a_3 & a_0 & a_1 \\ a_3 & a_2 & a_1 & a_0 \end{pmatrix} \in \mathbb{F}_q^{4\times 4}.$$

Note that by the tensor product structure, this block itself is formed by circulant *(2-by-2 in*

this case) matrices:

$$\left(\begin{array}{cc|cc} a_0 & a_1 & a_2 & a_3 \\ a_1 & a_0 & a_3 & a_2 \\ \hline a_2 & a_3 & a_0 & a_1 \\ a_3 & a_2 & a_1 & a_0 \end{array}\right) \in \mathbb{F}_q^{4\times 4}.$$

This entails that in particular a quasi-G code is a quasi-cyclic code with block length 2*. But, since the blocks depend on each other, a random quasi-G code is* not *a random quasi-cyclic code with block length* 2*.*

7.2.3 Duality for Quasi-Group Codes and Parity-Check Matrices

In the previous section we adopted the generator matrix point of view, but as we have seen in this manuscript, it may be more suitable in cryptography to consider codes defined by their parity-check matrices. This point of view naturally extends to the quasi-group setting.

Let G be a finite group. The algebra $\mathbb{F}_q[G]$ is naturally endowed with an inner product $\langle \cdot, \cdot \rangle$ defined by

$$\left\langle \sum_{g\in G} a_g g, \sum_{g\in G} b_g g \right\rangle \stackrel{\text{def}}{=} \sum_{g\in G} a_g b_g \in \mathbb{F}_q,$$

Note that it coincides with the standard inner product in $\mathbb{F}_q^n$ (when replacing an element of the group algebra by the vector of its coefficients), and that it does not depend on the chosen order. This inner product can easily be extended to the free module $(\mathbb{F}_q[G])^\ell$:

$$\langle (a_1, \ldots, a_\ell), (b_1, \ldots, b_\ell) \rangle \stackrel{\text{def}}{=} \sum_{i=1}^{\ell} \langle a_i, b_i \rangle.$$

The dual $\mathscr{C}^\perp$ of a quasi-group code $\mathscr{C}$ can be naturally defined as the usual dual in terms of linear codes, but it can be formulated in terms of the above inner product

$$\mathscr{C}^\perp \stackrel{\text{def}}{=} \left\{ x \in (\mathbb{F}_q[G])^\ell \mid \langle x, c \rangle = 0 \quad \forall c \in \mathscr{C} \right\}.$$

Proposition 7.11

Let G be a finite group, and let $\mathscr{C}$ be an ℓ-quasi-G code. Then $\mathscr{C}^\perp$ is also an ℓ-quasi-G code.

Proof. It suffices to prove that $\mathscr{C}^\perp$ is stable under the action of $\mathbb{F}_q[G]$ on the right, *i.e.* for any $x \in \mathscr{C}^\perp$ and $a \in \mathbb{F}_q[G]$, we have $x \cdot a \in \mathscr{C}^\perp$.

For any $a \stackrel{\text{def}}{=} \sum_{g\in G} a_g g \in \mathbb{F}_q[G]$, let

$$\overline{a} \stackrel{\text{def}}{=} \sum_{g\in G} a_g g^{-1} \quad \text{and} \quad \sigma(a) \stackrel{\text{def}}{=} a_{1_G} \in \mathbb{F}_q$$

where 1_G denotes the identity element of G. The map $a \mapsto \overline{a}$ is an anti-homomorphism of

order 2, *i.e.* it is bijective, involutive, and reverses the order of multiplication:

$$\overline{ab} = \overline{b} \cdot \overline{a} \quad \text{and} \quad \overline{(\overline{a})} = a.$$

Furthermore, σ is a linear form, and for any $a, b \in \mathbb{F}_q[G]$ it is easily seen that

$$\sigma(a\overline{b}) = \sum_{h \in G} a_h b_h = \langle a, b \rangle.$$

Now, let $x \stackrel{\text{def}}{=} (x_1, \ldots, x_\ell) \in \mathscr{C}^\perp$ and $a \in \mathbb{F}_q[G]$. For any $c \stackrel{\text{def}}{=} (c_1, \ldots, c_\ell) \in \mathscr{C}$ it holds that

$$\langle x \cdot a, c \rangle = \sum_{i=1}^{\ell} \sigma\Big((x_i a)\overline{c_i}\Big) = \sum_{i=1}^{\ell} \sigma\Big(x_i \overline{(c_i \overline{a})}\Big) = \sum_{i=1}^{\ell} \langle x, c \cdot \overline{a} \rangle = 0,$$

where the last equality holds since $\mathscr{C}$ is a (right) $\mathbb{F}_q[G]$-module, and therefore $c \cdot \overline{a} \in \mathscr{C}$. □

Proposition 7.11 entails that a quasi-G code has also a structured parity-check matrix, formed by blocks representing multiplications by elements of the group algebra.

Example 7.12. *Let $a \in \mathbb{F}_q[G]$ and let $\mathscr{C}$ be the 2-quasi-G code with generator matrix $\mathbf{\Gamma} \stackrel{def}{=} (1 \mid a)$. Then $\mathscr{C}$ admits a parity-check matrix of the form $\mathbf{H} \stackrel{def}{=} (\overline{a} \mid -1)$.*

7.2.4 Fast Encoding of Quasi-Abelian codes

Recall that for the applications to MPC, we need that the product in our ring be efficiently implementable. In terms of codes, this means that we want to be able to efficiently do the *encoding*, which can be done through Fast Fourier Transform (FFT) algorithms. From now on, let us fix a group G of cardinality n such that $\gcd(n, q) = 1$. By Maschke's theorem, this entails that $\mathbb{F}_q[G]$ is semisimple, *i.e.* is isomorphic to a product of matrix algebras. We further assume that G is abelian, so that the aforementioned product is actually a product of finite field extensions of $\mathbb{F}_q$:

$$\mathbb{F}_q[G] \simeq \mathbb{F}_{q^{\ell_1}} \times \cdots \times \mathbb{F}_{q^{\ell_r}}.$$

The strategy of FFT algorithms for computing a product $a \cdot b$ in $\mathbb{F}_q[G]$ can be reduced to three parts

1. First, compute the forward isomorphism $\varphi : \mathbb{F}_q[G] \to \mathbb{F}_{q^{\ell_1}} \times \cdots \times \mathbb{F}_{q^{\ell_r}}$.
2. Then, compute the product $\varphi(a) \star \varphi(b)$, componentwise.
3. Finally, compute the inverse map φ^{-1}.

Remark 7.13. *The forward isomorphism φ is often called the* Discrete Fourier Transform *of G in $\mathbb{F}_q$.*

More precisely, we speak about *Fast* Fourier Transform algorithms when Steps 1 and 3 can be done efficiently, say in $O(n \times \mathsf{polylog}(n))$ operations in $\mathbb{F}_q$, as opposed to the quadratic *naive* approach. This operation is all the more efficient when $\ell_i = 1$ for all i. This happens when $\mathbb{F}_q$ contains a primitive d-th root of unity, where $d = \exp(G)$ is the *exponent* of G, *i.e.* when the order of any element of G divides d.

We will describe FFT algorithms in terms only of the structure of the group G, and the finite field $\mathbb{F}_q$. This will encompass the usual FFT algorithm introduced by Cooley and Tuckey in [CT65], or the Number Theoretic Transform (NTT), which we have already mentioned in this manuscript. For a detailed presentation, the interested reader can refer to [Obe07].

Definition 7.14 (Composition series)

Let G be a finite group. A finite sequence $(G_0, \dots, G_r)$ of subgroups of G is called a *composition series* of G of length r when

1. $G_0 = \{1_G\}$ and $G_r = G$;
2. for $i \in \{0, \dots, r-1\}$, each G_i is a *strict* normal subgroup of G_{i+1};
3. the quotient G_{i+1}/G_i is a *simple* group, *i.e.* which does not have any non trivial normal subgroup.

The quotients G_{i+1}/G_i are called the *factors* of the series.

Remark 7.15. *Condition* (3) *in Definition 7.14 is equivalent to the series to be maximal, that is no subgroup can be added to the sequence while maintaining a sequence of normal subgroups.*

A group may have multiple composition series, but the Jordan-Hölder theorem (see for example [Lan12, Theorem 3.5]) states that they are all equal, up to permutation of the factors. Recall that a finite simple group is abelian if and only if it is a cyclic group of prime order $\mathbb{Z}/p\mathbb{Z}$. In particular, each factor G_{i+1}/G_i of a composition series of an abelian group G is of the form $\mathbb{Z}/p_i\mathbb{Z}$, and Jordan-Hölder theorem entails then that the sequence of the p_i's is uniquely determined by G.

Example 7.16. *For an abelian group G of cardinality n, the primes of the Jordan-Hölder series verify $n = p_1 \dots p_n$.*

- *If p is prime, then $\mathbb{Z}/p\mathbb{Z}$ is already a simple group, and the Jordan-Hölder series is nothing else than $\{1\} \triangleleft G$.*
- *The cyclic group $\mathbb{Z}/12\mathbb{Z}$ has three composition series,*

$$\{0\} \triangleleft \mathbb{Z}/2\mathbb{Z} \triangleleft \mathbb{Z}/4\mathbb{Z} \triangleleft \mathbb{Z}/12\mathbb{Z} \qquad \text{with factors } \left(\mathbb{Z}/2\mathbb{Z}, \mathbb{Z}/2\mathbb{Z}, \mathbb{Z}/3\mathbb{Z}\right)$$

$$\{0\} \triangleleft \mathbb{Z}/2\mathbb{Z} \triangleleft \mathbb{Z}/6\mathbb{Z} \triangleleft \mathbb{Z}/12\mathbb{Z} \qquad \text{with factors } \left(\mathbb{Z}/2\mathbb{Z}, \mathbb{Z}/3\mathbb{Z}, \mathbb{Z}/2\mathbb{Z}\right)$$

$$\{0\} \triangleleft \mathbb{Z}/3\mathbb{Z} \triangleleft \mathbb{Z}/6\mathbb{Z} \triangleleft \mathbb{Z}/12\mathbb{Z} \qquad \text{with factors } \left(\mathbb{Z}/3\mathbb{Z}, \mathbb{Z}/2\mathbb{Z}, \mathbb{Z}/2\mathbb{Z}\right)$$

- *If $G = \left(\mathbb{Z}/2\mathbb{Z}\right)^n$, the composition series are of the form*

$$\{0\}^n \triangleleft \mathbb{Z}/2\mathbb{Z} \times \{0\}^{n-1} \triangleleft \cdots \triangleleft \left(\mathbb{Z}/2\mathbb{Z}\right)^{n-1} \times \{0\} \triangleleft \left(\mathbb{Z}/2\mathbb{Z}\right)^n.$$

The divide-and-conquer approach in FFT algorithms can be understood in terms of the Jordan-Hölder series of G.

Proposition 7.17 ([Obe07, Section 5])

Let G be a finite abelian group of size n with $\gcd(n,q)=1$, and exponent d. Assume that $\mathbb{F}_q$ contains a primitive d-th root of unity. Let $p_1,\dots,p_r$ denote all the primes (possibly non distinct) appearing in the Jordan-Hölder series of G. Then the Discrete Fourier Transform (and its inverse) in $\mathbb{F}_q[G]$ can be computed in $O(n\times(p_1+\cdots+p_r))$ operations in $\mathbb{F}_q$.

Example 7.18. *Proposition 7.17 encompasses well-known* FFT*'s from the literature.*

- *The usual* FFT *corresponds to* $G=\mathbb{Z}/2^t\mathbb{Z}$. *In this case, a composition series is given by*

$$G_0=\{0\}\lhd\cdots\lhd G_i=2^{t-i}\mathbb{Z}/2^t\mathbb{Z}\lhd\cdots\lhd G_t=G=\mathbb{Z}/2^t\mathbb{Z},$$

  and each factor G_{i+1}/G_i *is isomorphic to* $\mathbb{Z}/2\mathbb{Z}$, *and with the above proposition we recover the usual complexity in* $O(2^t\times t)=O(n\log(n))$.

- *Consider the finite field* $\mathbb{F}_3$ *and the group* $G=(\mathbb{Z}/2\mathbb{Z})^t$. *We have*

$$\mathbb{F}_3[G]\simeq\mathbb{F}_3[X_1,\dots,X_t]/(X_1^2-1,\dots,X_t^2-1).$$

  A composition series of G *is given by*

$$G_0=\{0\}^t\lhd\cdots\lhd G_i=(\mathbb{Z}/2\mathbb{Z})^i\times\{0\}^{t-i}\lhd\cdots\lhd G_t=G=(\mathbb{Z}/2\mathbb{Z})^t,$$

  and the FFT *can also be computed in time* $O(2^t\times t)=O(n\log(n))$. *This is nothing else than a* t*-dimensional* FFT *in* $\mathbb{F}_3$.

In general, a primitive d-th root of unity will be contained in a strict extension $\mathbb{F}_{q^\ell}$, and the FFT can be computed in $\mathbb{F}_{q^\ell}$. However, van der Hoeven and Larrieu showed in [HL17] that one may exploit the symmetries given by the Frobenius automorphism of $\mathbb{F}_{q^\ell}/\mathbb{F}_q$ to regain the factor ℓ lost in considering the extension. This approach is now known as the *Frobenius* FFT.

Remark 7.19. *Proposition 7.17 is* asymptotic, *although efficient implementations exist for several groups and fields. They are particularly efficient when* G *admits a Jordan-Hölder composition series with groups of exponent* 2, *such as in Example 7.18. For a more precise description of Multivariate* FFT *algorithms, see [HLS13, Section 2.2].*

7.3 Building PCG's for OLE's from Quasi-Abelian Codes

We now have all the material to instantiate the template of [BCGI+20b], recalled in Section 6.4.3, with codes over group algebras.

7.3.1 The Quasi-Abelian Decoding Problem

Let G be a finite abelian group. In this section, we will adopt the point of view of parity-check matrices. Moreover, for a reason that will appear clear later, we restrict our definition to *systematic* parity-check matrices. In other words, given an integer k, a random $\ell-$quasi-G code

will be defined as the code of parity-check matrix $\mathbf{H} = (\mathbf{H}' \mid \mathrm{I}_{\ell-k})$ where $\mathbf{H}' \leftarrow\in \mathbb{F}_q[G]^{(\ell-k)\times k}$ is uniformly distributed and $\mathrm{I}_{\ell-k}$ is the diagonal matrix with coefficients 1_G. Let Ψ be a probability distribution over $\mathbb{F}_q[G]$. In general, Ψ will be parameterised with an integer $t \in \{1, \dots, n\}$ such that $\mathbb{E}(|x|) = t$ when $x \leftarrow \Psi$, where $|x|$ denotes the Hamming weight of x with respect to any ordering of G. For example, Ψ can be a Bernoulli distribution, or the uniform over the words of fixed Hamming weight t.

In this section, we introduce the variant of the Decoding Problem with respect to quasi-abelian codes, namely the *Quasi-Abelian Decoding Problem* (QADP). It will be more convenient to describe it with the language of parity-check matrices and syndromes. Recall that when $\mathcal{E}$ is a finite set, we write $X \leftarrow \mathcal{E}$ for denoting a random variable X uniformly distributed in $\mathcal{E}$.

Problem 7.20 (QADP(G, ℓ, k, Ψ), search version)

Data. $\mathbf{H} \stackrel{\text{def}}{=} (\mathbf{H}' \mid \mathrm{I}_{\ell-k})$ where $\mathbf{H}' \leftarrow \mathbb{F}_q[G]^{(\ell-k)\times k}$, and $\mathbf{s}^\top \stackrel{\text{def}}{=} \mathbf{H}\mathbf{e}^\top \in \mathbb{F}_q[G]^{\ell-k}$, where $\mathbf{e}_i \leftarrow \Psi$.

Goal. Recover $\mathbf{e}$.

Remark 7.21. *$\mathrm{I}_{\ell-k}$ is the identity matrix of size $\ell - k$ with coefficients in $\mathbb{F}_q[G]$.*

The Quasi-Abelian Decoding Problem generalises all the instantiations of the Decoding Problem we have introduced so far, and which are used in cryptography.

- When $G = \{1\}$ is the trivial group, this corresponds exactly to the usual (average-case) Decoding Problem DP (Problem 1.10).
- When $G = \mathbb{Z}/n\mathbb{Z}$ is a cyclic group of size n, this is nothing else than the (average-case) Decoding Problem of Quasi-Cyclic codes QC-DP (Problem 1.38), as used in BIKE and HQC for instance (in the specification of those cryptosystems we have $\ell = 2$ or $\ell = 3$ and $k = \ell - 1$ or $k = \ell - 2$).

Seen as a code over $\mathbb{F}_q$, the quasi-G code has length $\ell|G|$ and dimension $k|G|$. For the applications, $|G|$ will represent the number of OLE's our PCG will be able to produce, and therefore will be very large (say $|G| \approx 2^{30}$). On the other hand, we will consider a constant rate regime of the form $1 - 1/\ell$, with ℓ rather small (say, less than 4). In particular, when k is not specified, it will be implicitly understood to be $\ell - 1$. Allowing to have $\ell > 2$ enables a slight efficiency improvement in the construction of our PCG. In this situation, $\mathbf{H}$ is of the form

$$\mathbf{H} \stackrel{\text{def}}{=} (a_1 \mid \cdots \mid a_{\ell-1} \mid 1), \quad \text{where } a_i \leftarrow \mathbb{F}_q[G],$$

and a syndrome is of the form

$$\mathbf{H}\mathbf{e}^\top \stackrel{\text{def}}{=} a_1 e_1 + \cdots + a_{\ell-1} e_{\ell-1} + e_\ell \in \mathbb{F}_q[G], \quad \text{where } e_i \leftarrow \Psi.$$

The decisional version is naturally defined as follows

Problem 7.22 (QADP(G,k,ℓ,Ψ), decisional version)

Consider the following two distributions, where $\mathbf{H}' \leftarrow \mathbb{F}_q[G]^{(\ell-k)\times k}$

- $\mathcal{D}_0 : \left(\mathbf{H} \stackrel{\text{def}}{=} (\mathbf{H}' \mid \mathrm{I}_{\ell-k}), \mathbf{s}^{\text{unif}}\right)$ and $\mathbf{s}^{\text{unif}} \leftarrow \mathbb{F}_q[G]^{\ell-k}$
- $\mathcal{D}_1 : \left(\mathbf{H} \stackrel{\text{def}}{=} (\mathbf{H}' \mid \mathrm{I}_{\ell-k}), \mathbf{e}\mathbf{H}^\top\right)$ and $\mathbf{e} \stackrel{\text{def}}{=} (e_1, \ldots, e_\ell)$ with $e_i \leftarrow \Psi$.

Given oracle access to distribution $\mathcal{D}_b$ where $b \leftarrow \{0,1\}$, the goal is to recover b.

In the applications, we only consider the case $k = \ell - 1$, for which Problem 7.22 can be simplified to

Problem 7.23 (QADP(G,ℓ,Ψ), decisional version)

Consider the following two distributions

- $\mathcal{D}_0 : \left(\mathbf{a} \stackrel{\text{def}}{=} (a_1, \ldots, a_{\ell-1}), \mathbf{s}^{\text{unif}}\right)$ where $a_i \leftarrow \mathbb{F}_q[G]$ and $\mathbf{s}^{\text{unif}} \leftarrow \mathbb{F}_q[G]$
- $\mathcal{D}_1 : \left(\mathbf{a} \stackrel{\text{def}}{=} (a_1, \ldots, a_{\ell-1}), \sum_{i=1}^{\ell-1} a_i e_i + e_\ell\right)$ where $a_i \leftarrow \mathbb{F}_q[G]$, and $e_i \leftarrow \Psi$.

Given oracle access to distribution $\mathcal{D}_b$ where $b \leftarrow \{0,1\}$, the goal is to recover b.

By a slight abuse of notation, when $t \in \{0, \ldots, |G|\}$, we will denote by $\mathrm{QADP}(G,\ell,t)$ the problem $\mathrm{QADP}(G,\ell,\Psi)$ where Ψ denotes the uniform distribution over elements of Hamming weight t.

A note about the systematic form. We can wonder what happens when we do not force the parity-check matrix to be in systematic form in the definition of QADP. The situation is very similar to that of Section 1.3.1.2. For simplicity, assume that $k = 1, \ell = 2$, *i.e.* $\mathbf{H} = (a_1 \mid a_2) \in \mathbb{F}_q[G]^{1\times 2}$ for some uniformly random $a_1, a_2 \leftarrow \mathbb{F}_q[G]$. Then, a syndrome of $\mathbf{H}$ is of the form $a_1 e_1 + a_2 e_2$, and therefore is contained in the ideal $\mathcal{I} \stackrel{\text{def}}{=} (a_1, a_2)$ of $\mathbb{F}_q[G]$ generated by a_1 and a_2.[i] In particular, when $\mathcal{I}$ is *not* the full ring, this induces an obvious bias. When working over large fields, this does not really matter since an element of $\mathbb{F}_q[G]$ will be invertible with high probability. On the other hand, this is not true anymore when working over small fields such as $\mathbb{F}_2$ or $\mathbb{F}_3$ (which is the whole point of this work). Using parity-check matrices in systematic form forces $\mathcal{I}$ to contain 1_G, which removes the bias.

[i] Note that $\mathbb{F}_q[G]$ is not necessarily a Principal Ideal Domain.

7.3.2 Instantiating the PCG with Quasi-Abelian Codes

Our new PCG uses the template of [BCGI+20b] recalled in Chapter 6, and more precisely in Section 6.4.3: it suffices to choose a group G such that

$$\mathcal{R} \stackrel{\text{def}}{=} \mathbb{F}_q[G] \simeq \underbrace{\mathbb{F}_q \times \cdots \times \mathbb{F}_q}_{N \text{ copies}}.$$

In the sequel, an element $\mathbf{a} \in \mathbb{F}_q[G]$ will be called *t-sparse* if it has Hamming weight t (with respect to the basis given by G, for any ordering).

The Construction. Assume that we have such a group G, and denote by φ the aforementioned isomorphism. Mimicking the construction of Section 6.4.3, the trusted dealer samples $\mathbf{a}$ uniformly at random in $\mathbb{F}_q[G]$, as well as some *t-sparse* elements $\mathbf{e}_u, \mathbf{e}_v, \mathbf{f}_u, \mathbf{f}_v \in \mathbb{F}_q[G]$, where the sparsity is defined with respect to the canonical basis given by G.

Remark 7.24. *Note that working with a group algebra allows to define the* sparsity *of elements for free. This fact may seem meaningless at first glance, but as we will see in Section 7.4, this is not always obvious for general $\mathbb{F}_q$-algebras, for which no basis seems to be better than another. We already mentioned this issue in Chapter 4, and more particularly in Section 4.4.3.2, where we introduced the Normal Ring–LPN distribution (Definition 4.57).*

Now, the dealer sets

$$\mathbf{U} \stackrel{\text{def}}{=} \mathbf{a} \cdot \mathbf{e}_u + \mathbf{f}_u \in \mathbb{F}_q[G] \quad \text{and} \quad \mathbf{V} \stackrel{\text{def}}{=} \mathbf{a} \cdot \mathbf{e}_v + \mathbf{f}_v \in \mathbb{F}_q[G],$$

so that the product

$$\mathbf{U} \cdot \mathbf{V} = \mathbf{a}^2(\mathbf{e}_u \mathbf{e}_v) + \mathbf{a}(\mathbf{e}_u \mathbf{f}_v + \mathbf{e}_v \mathbf{f}_u) + \mathbf{f}_u \mathbf{f}_v \in \mathbb{F}_q[G]$$

is again a linear combination of t^2-sparse elements of $\mathbb{F}_q[G]$.

Remark 7.25. *Another advantage of using group algebras is that the product of t-sparse elements directly yields a t^2-sparse element* in $\mathbb{F}_q[G]$*, while in the construction of [BCGI+20b], the authors were forced to share the products seen as degree $2N-2$ polynomials over $\mathbb{F}_q[X]$, and then reduce locally modulo their defining polynomial. Indeed, for rings of the form $\mathbb{F}_q[X]/(F(X))$, it is not generally true that the product of two sparse elements remains sparse. By using group algebras, we can directly share elements of $\mathbb{F}_q[G]$, which makes the scheme slightly more efficient.*

Both $\mathbf{U}$ and $\mathbf{V}$ can be considered as syndromes of the random quasi-abelian code with parity-check matrix $(\mathbf{a} \mid 1)$, and error respectively $(\mathbf{e}_u, \mathbf{f}_u)$ and $(\mathbf{e}_v, \mathbf{f}_v)$.

Remark 7.26. *In practice, we might want to use codes with a slightly higher rate $(\ell-1)/\ell$ (with $\ell \in \{2,3,4\}$), which corresponds to using sparse polynomials of the form $(\mathbf{e}_{u_1}, \ldots, \mathbf{e}_{u_{\ell-1}}, \mathbf{f}_u)$ and $(\mathbf{e}_{v_1}, \ldots, \mathbf{e}_{v_{\ell-1}}, \mathbf{f}_v)$.*

Using function secret sharing as per Section 6.3, the trusted dealer can succinctly describe additive shares of all the cross products, in $\mathbb{F}_q[G]$. Together with the seed defining $\mathbf{a}$, as well as the description of $\mathbf{e}, \mathbf{f}$, this allows the dealer to give to each party a short description of an OLE over $\mathbb{F}_q[G]$, which can then be transformed into N OLE's over $\mathbb{F}_q$ by applying the isomorphism φ, which is nothing else than one step of the FFT algorithm in $\mathbb{F}_q[G]$.

Moreover, this construction inherits the programmability property from the original template of [BCGI+20b].

Instantiating the group G. In order to maximise the number of OLE's we can produce, we propose to use a group of the form

$$G \stackrel{\text{def}}{=} \left(\mathbb{Z}/(q-1)\mathbb{Z}\right)^r,$$

so that

$$\mathbb{F}_q[G] \simeq \mathbb{F}_q[X_1,\dots,X_r]/(X_1^{q-1}-1,\dots,X_r^{q-1}-1) \simeq \prod_{(\gamma_1,\dots,\gamma_r)\in\left(\mathbb{F}_q^\times\right)^t} \mathbb{F}_q[X_1,\dots,X_r]/(X_i-\gamma_i).$$

is a product of $N \stackrel{\text{def}}{=} (q-1)^r$ copies of $\mathbb{F}_q$.

In particular, setting $q=3$ and $r=30$, we can generate 2^{30} OLE's over $\mathbb{F}_3$.

In one wants to give concrete set of parameters which includes choosing the number of errors t, it is necessary to study in depth the best attacks on QADP.

Remark 7.27. *Note that by construction, the efficiency of our* PCG *is directly related to the number of* FSS *schemes we need to run, i.e. in the sparsity t we can afford without diminishing too much the security. In particular, this is why we are tempted to use codes of higher rate, in order to be able to slightly decrease t.*

7.3.3 On the Security of the Construction

Assuming the security of the function secret sharing scheme, which is basically based on the security of the underlying pseudorandom generator, the security of our PCG relies on the hardness of the decisional version of QADP. Moreover, by Remark 7.27 the efficiency of our PCG is directly related to the number of errors we can afford in QADP without undermining the security: the smaller the number of errors, the more efficient the scheme.

7.3.3.1 Hardness of QADP

In this section we focus on the theoretical hardness of the *Quasi-Abelian Decoding Problem*. The discussion of Section 7.3.1 already identifies some potentially weak instantiations of the decisional version. On the other hand, since QADP extends the usual DP (Problem 1.10) and QC-DP (Problem 1.38), at least *some* instantiations are believed to be hard. Therefore, it is natural to ask if all the weaknesses have been removed.

Obviously, a necessary condition for (the decisional version) of QADP to be hard, is the hardness of the search version. It turns out that all the remarks about the hardness of decoding quasi-cyclic codes extend to quasi-abelian codes. This is even listed as an open research problem in the most recent Encyclopedia of Coding Theory [Wil21, Problem 16.10.5].

A remark on Ring-LPN. In [BL12], Bernstein and Lange introduced an attack against the protocol LaPiN [HKLP+12] based on variants of Ring-LPN. It shall be noticed that none of the rings proposed to instantiate this protocol were group algebras, and this attack does not seem to apply in this context. In Section 7.3.3 we investigate concrete attacks against QADP.

A remark on Multivariate LWE. When coming from the world of lattice-based cryptography, it may be counterintuitive at the very least, if not dubious, to even consider that QADP might

be hard in general. Indeed, let us consider the simpler case where $k = 1$ and $\ell = 2$. Let

$$G \stackrel{\text{def}}{=} \prod_{i=1}^{r} \mathbb{Z}/d_i\mathbb{Z}$$

be the decomposition of G in product of cyclic groups. Assume that the characteristic of $\mathbb{F}_q$ does not divide any of the d_i, so that $\mathbb{F}_q[G]$ is a semisimple algebra. In terms of polynomial rings, we have

$$\mathcal{R} \stackrel{\text{def}}{=} \mathbb{F}_q[G] \simeq \mathbb{F}_q[X_1, \ldots, X_r]/(X_1^{d_1} - 1, \ldots, X_r^{d_r} - 1).$$

In other words, a sample from the QADP distribution will be of the form

$$(a(X_1, \ldots, X_r), a(X_1, \ldots, X_r) \cdot s(X_1, \ldots, X_r) + e(X_1, \ldots, X_r))$$

where a is uniformly distributed in the multivariate polynomial ring $\mathcal{R}$, and e is a sparse multivariate polynomial. Since we look at a syndrome in our definition of QADP, we also ask s to be a sparse multivariate polynomial, but this is not a restriction. Obviously, the resemblance with a ring-LWE sample is striking. Yet, multivariate versions of ring-LWE have already been considered in the literature such as [PTP15; PTP16], and were proven insecure in [BCV20]. Nonetheless, as we will see, the Hamming metric constraint on the error distribution is quite different than the euclidean situation.

A search-to-decision reduction. Since we are mostly interested in the decisional version of QADP, it is natural to ask if it is strictly easier than the search version. This is the first time that this question is investigated. In Chapter 4, we introduced a function field approach to derive search-to-decision reductions for structured codes. In particular, Theorem 4.44 gives the reduction in the case of quasi-cyclic codes with block length $q-1$, *i.e.* for the group algebra $\mathbb{F}_q[G]$ with $G \stackrel{\text{def}}{=} \mathbb{Z}/(q-1)\mathbb{Z}$.

The crucial part of the reduction was that

(i) the group algebra $\mathbb{F}_q[G]$ could be interpreted in terms of the Carlitz extension with respect to the T-torsion, namely $K \stackrel{\text{def}}{=} \mathbb{F}_q(T)[\Lambda_T]$:

$$\mathbb{F}_q[G] \simeq \mathcal{O}_K/T\mathcal{O}_K \simeq \prod_{\mathfrak{p}|T} \mathcal{O}_K/\mathfrak{p};$$

(ii) and the Galois group $\mathsf{Gal}(K/\mathbb{F}_q(T)) \simeq \mathbb{F}_q^\times$ acted transitively on the prime above T, permuting the factors in the product, while also keeping invariant the Hamming support of each element. On the quotient, that is on $\mathbb{F}_q[G]$, the action of $\zeta \in \mathbb{F}_q^\times$ on an element $a \stackrel{\text{def}}{=} a(X) \stackrel{\text{def}}{=} \sum_{i=0}^{n-1} a_i X^i \in \mathbb{F}_q[G]$ was simply

$$\zeta \cdot a \stackrel{\text{def}}{=} a(\zeta X) = \sum_{i=0}^{n-1} a_i \zeta^i X^i.$$

It turns out that this reduction extends easily to the multivariate setting, yielding a search-to-decision reduction for QADP, when the codes are instantiated with particular groups of the form

$$H \stackrel{\text{def}}{=} \left(\mathbb{Z}/(q-1)\mathbb{Z}\right)^r.$$

This will be the setting chosen to build our PCG. Using the heavy machinery of inverse Galois theory, it may be possible to find a Galois extension $L/\mathbb{F}_q(T)$ with Galois group H, and such that $\mathbb{F}_q[H]$ could also be interpreted as the quotient of a ring of integers. However, in this case we can directly adapt the proof, following every step, and working directly on the ground ring.

Indeed, we have

$$\mathbb{F}_q[H] = \mathbb{F}_q[X_1,\dots,X_r]\big/(X_1^{q-1}-1,\dots,X_r^{q-1}-1) \simeq \prod_{(\gamma_1,\dots,\gamma_r)\in(\mathbb{F}_q^\times)^r} \mathbb{F}_q[X_1,\dots,X_r]\big/(X_i-\gamma_i)$$

where each factor is a finite field, isomorphic to $\mathbb{F}_q$. Therefore, we only need to find a group which acts transitively on the factors, while keeping invariant the noise distribution we use to define QADP. It turns out that the group $\widehat{H} \stackrel{\text{def}}{=} (\mathbb{F}_q^\times)^r$ is suitable: it acts on $\mathbb{F}_q[H]$ via

$$(\zeta_1,\dots,\zeta_r)\cdot a(X_1,\dots,X_r) \stackrel{\text{def}}{=} a(\zeta_1 X_1,\dots,\zeta_r X_r),$$

and this action maps the ideal $(X_1-\gamma_1,\dots,X_r-\gamma_r)$ onto $(X_1-\zeta_1^{-1}\gamma_1,\dots,\zeta_r^{-1}\gamma_r)$.

Moreover, as in the univariate case, this group action keeps invariant the support of elements of $\mathbb{F}_q[G]$. In particular, it keeps invariant the uniform distribution over words of fixed Hamming weight, and the Bernoulli distribution.

Therefore, following *exactly* the path of the reduction from Chapter 4, it suffices to replace in Step 4 the action of the Galois group by that of $\widehat{H}$ to prove that the decisional version of QADP, instantiated with the group H, is at least as hard as the corresponding search version.

Resistance to Linear Attacks. Recall from Section 1.3.4.3 that most of the known attacks against the decisional version of the decoding problem fit into the linear test framework. Unconditionally resisting linear attacks is equivalent for the class of codes to have a large dual minimum distance. Since the dual of a quasi-abelian code is still a quasi-abelian code, we will focus on the minimum distance of a random quasi-abelian code. Ideally, we would like that a random quasi-abelian code reaches the Gilbert-Varshamov bound with good probability, as it is the case for random linear codes (*i.e.* quasi-group codes with the trivial group).

Extending this result to more general algebraic codes has attracted a lot of work in algebraic coding theory in the past 50 years, and is still an active field of research as many questions are still widely open. For example, it was proven in [CPJ69], later generalised in [Kas74], that under mild assumptions regarding the size of the blocks, random double-circulant codes achieve the Gilbert-Varshamov bound with high probability. Furthermore, as discussed in 1.3.4.3, it was proven in [GZ06] that certain binary quasi-cyclic codes even satisfy a logarithmic improvement over the Gilbert-Varshamov bound, in other words they reach the Gilbert-Varshamov bound *from above*.

For the case of random quasi-abelian codes, [BM06] extends the result of [Kas74] to binary quasi-abelian codes of fixed rate $1/\ell$ (or $(\ell-1)/\ell$), but only when the dimension $L_2(G)$ of the smallest irreducible $\mathbb{F}_2$ representation of G (which depends only on $|G|$ when G is abelian) grows faster than logarithmically in $|G|$ (and $|G|$ is odd). This can be extended to $\mathbb{F}_q$ using [FL22, Section 6.2], when $|G|$ is coprime to q and with an analogue restriction in the dimension $L_q(G)$, denoted by $\mu_q(|G|)$ in the latter reference. However, this condition is not satisfied by the group $G \stackrel{\text{def}}{=} \left(\mathbb{Z}\big/(q-1)\mathbb{Z}\right)^r$ which we propose to use in our PCG construction. Indeed, it can be shown (see [FL22, Section 3]) that

$$\mu_q(n) = \min\{\mathsf{ord}_p(q) \mid p \text{ is a prime divisor of } n\},$$

where $\mathsf{ord}_p(q)$ denotes the order of q in $\left(\mathbb{Z}/p\mathbb{Z}\right)^\times$. In odd characteristics, $(q-1)^r$ is even, and $\mathsf{ord}_2(q)=1$. Therefore, $\mu_q((q-1)^r)=1$, is independent of r.

However, all those works were mostly concerned in finding explicit codes with optimal parameters, and/or infinite sequences of codes with good parameters within those families. Moreover, they did not restrict themselves to codes in systematic form, which we know can induce a bias. This condition on $L_q(G)$ (or equivalently $\mu_q(|G|)$) was enough for their goal.

On the other hand, when one considers additional constraints, such as demanding that the codes be self dual, some previous works such as [FL22, Section 7.1] have restricted themselves to systematic quasi-abelian codes. It can be interesting in the future to explore this topic further.

Nonetheless, for quasi-abelian codes it can be interesting to look at their behaviour when the group G is *fixed*, and when this is the length ℓ (over the group algebra) that goes to infinity, while keeping k/ℓ constant. For example, this is the setting considered when dealing with random linear codes, where G is fixed to be $\{1\}$. In particular, it was proven in [FL15] without any restriction on G (this result even holds in the modular case where $\mathbb{F}_q[G]$ is not semisimple anymore) that random large quasi-abelian codes also meet the Gilbert-Varshamov bound. More precisely, they proved the following theorem:

Theorem 7.28 ([FL15, Theorem 2.1])

Let G be a finite abelian group, and let $(\mathscr{C}_\ell)_\ell$ be a sequence of random quasi-G codes of length $\ell \in \mathbb{N}$ and rate $r \in (0,1)$. Let h_q denote the q-ary entropy function, and let $\delta \in (0, 1-\frac{1}{q})$. Then,

$$\lim_{\ell\to\infty} \mathbb{P}\left(\frac{d_{min}(\mathscr{C}_\ell)}{|G|} > \delta\ell\right) = \begin{cases} 1 & \text{if } r < 1-h_q(\delta); \\ 0 & \text{if } r > 1-h_q(\delta); \end{cases}$$

and both limits converge exponentially fast. The above probability is taken over the uniform choice of a generator matrix $\mathbf{\Gamma}_\ell \in \mathbb{F}_q[G]^{k\times\ell}$ of $\mathscr{C}_\ell$.

Remark 7.29. *The proof of this theorem follows the same path as the usual result for random linear codes, using the theory of representations of G to estimate the probabilities which are involved.*

Intuitively, when the ideal of $\mathbb{F}_q[G]$ generated by the blocks $(a_1,\dots,a_\ell)$ of the parity-check matrix is not the full ring, it is possible that the minimum distance drop, but in general it should be linear in its length. On the other hand, when ℓ grows, this ideal is likely to become the full ring very quickly, removing the bias. If this intuition is correct, it may be possible to obtain a Gilbert-Varshamov like bound for random quasi-abelian codes in systematic form.

7.3.3.2 Concrete Security of the Construction

In Section 7.3.3.1, we discussed the hardness of QADP in general. We shall now focus on concrete attacks.

There are two ways of attacking this problem: one can either attack the search version, which has been more studied in the literature, or try to directly distinguish a syndrome from a uniform vector. As we already mentioned, no decoding algorithm is known for generic quasi-group codes. Moreover, our choice of the group G allows to design a search-to-decision reduction (see Section 7.3.3.1). Even though the parameters of the reduction are quite loose, solving the decisional QADP implies solving the search version (*i.e.* decoding a random quasi-G code).

Moreover, random quasi-abelian codes also seem to have a large minimum distance, which means that QADP resists all the attacks from the linear test framework (see Section 1.3.4.3). Nevertheless, this observation is merely of theoretical interest: this characterisation via the minimum distance rules out *all* linear attacks, even those which are very inefficient. Indeed, let $\mathscr{C}$ be a code of parity-check matrix $\mathbf{H}$, and recall that a linear attack for distinguishing a syndrome $\mathbf{He}^\top$ from a uniform vector corresponds to finding a vector $\mathbf{v} \in \mathbb{F}_q^{n-k}$ such that $\mathbf{vH}$ has small Hamming weight. In other words this corresponds to finding a short Hamming weight codeword in the code *generated* by $\mathbf{H}$, that is in the dual code $\mathscr{C}^\perp$. The smaller this codeword, the bigger the bias. However, finding such a codeword in general is a very difficult task: it is known to be NP-complete [Var97], *i.e.* hard on the *worst-case*, and also widely believed to be hard *on average*, even for structured codes. For instance, the security of the BIKE encryption scheme, from round 4 of NIST competition, relies among other things in the hardness of finding short codewords in a random binary quasi-cyclic code. In practice, the best known algorithms for finding short codewords are basically the same as the best generic decoding algorithms. In other words, this linear test framework does not consider realistic attackers, with limited resources. Therefore, in order to assess the concrete hardness of QADP, it does make sense to focus on known attacks. First, let us see what happens when the underlying ring is *not* a group algebra. Then we will focus on generic attacks which forget the structure, before finally look with more details into it.

7.3.3.3 An Easy Bias when not Working over Group Algebras

For simplicity, consider the univariate polynomial ring with coefficients in $\mathbb{F}_q$ which is isomorphic to the direct product of the largest possible number of copies of $\mathbb{F}_q$, namely

$$\mathcal{R} \stackrel{\text{def}}{=} \mathbb{F}_q[X] \big/ (X^q - X).$$

Even if this is not the algebra of some group, it is still equipped with the Hamming metric with respect to the canonical basis $(X^i)_{0 \leqslant i \leqslant q-1}$. Fixing $t \in \{1, \dots, q\}$, similarly as before, denote by $\mathcal{R}_t$ the set of polynomials of Hamming weight t, and define the following two distributions:

- $\mathcal{D}_0 : \big(a(X), s^{\text{unif}}(X)\big)$ where $a(X) \leftarrow \mathcal{R}$ and $s^{\text{unif}}(X) \leftarrow \mathcal{R}$
- $\mathcal{D}_1 : (a(X), a(X)e_1(X) + e_2(X))$ where $a(X) \leftarrow \mathcal{R}$, and $e_i \leftarrow \mathcal{R}_t$.

In order to simplify the notations, we will write a for $a(X)$ when $a \in \mathcal{R}$.

Remark 7.30. *As for the quasi-group code setting, we may identify the output of $\mathcal{D}_1$ as a pair $(\mathbf{H}, \mathbf{s})$ where $\mathbf{H} = (\mathbf{M}_a \mid \mathbf{I}_q) \in \mathbb{F}_q^{q \times 2q}$ with $\mathbf{M}_a$ representing the multiplication by a in the monomial basis, and $\mathbf{s}$ is the syndrome of a (regular) error of weight $2t$, so that the problem of distinguishing between $\mathcal{D}_0$ and $\mathcal{D}_1$ is a well defined instance of a decisional decoding problem.*

At a high level, the issue comes from the fact that X, which is an element of the basis, is *not invertible* in $\mathcal{R}$. Therefore, the ideal generated by X is not the full ring, and reduction modulo X, that is evaluation at 0, induces a bias in the distribution.

More precisely, for $P(X) \stackrel{\text{def}}{=} \sum_{i=0}^{q-1} \alpha_i X^i \in \mathcal{R}$, denote by $P^\uparrow \in \mathbb{F}_q[X]$ the polynomial of degree less than $q-1$ which lifts P back to $\mathbb{F}_q[X]$, that is

$$P^\uparrow \equiv P \mod (X^q - X).$$

Then, since $(X) \supset (X^q - X)$, we also have

$$P^{\uparrow}(0) = P(0). \tag{7.2}$$

The idea of the bias is that when P is uniformly distributed in $\mathcal{R}$, then $P(0)$ is uniformly distributed in $\mathbb{F}_q$ and

$$\mathbb{P}(P(0)=0) = \mathbb{P}(\alpha_0 = 0) = \frac{1}{q},$$

while when P is uniformly distributed in $\mathcal{R}_t$,

$$\mathbb{P}(P(0)=0) = 1 - \mathbb{P}(\alpha_0 \in \mathrm{Supp}(P)) = 1 - \frac{(q-1)\binom{q-1}{t-1}(q-1)^{t-1}}{\binom{q}{t}(q-1)^t} = 1 - \frac{t}{q}.$$

Note that this equality also holds for a polynomial uniformly distributed amongst the t sparse polynomials of degree at most $q-1$.

Now, when $e_1, e_2 \leftarrow \mathcal{R}_t$ are independent, and $a \leftarrow \mathcal{R}$, then $e_1^{\uparrow}$ and $e_2^{\uparrow}$ are also t sparse, and therefore

$$\mathbb{P}\left((a\cdot e_1 + e_2)^{\uparrow}(0) = 0\right) \geqslant \mathbb{P}\left(e_1^{\uparrow}(0) = 0, e_2^{\uparrow}(0) = 0\right) = \left(1 - \frac{t}{q}\right)^2 \gg \frac{1}{q}.$$

In other words, the polynomial $a\cdot e_1 + e_2$ is way more likely to evaluate to 0 at 0 than a uniformly random polynomial.

On the other hand, when $\mathcal{R}$ is a group algebra, such a distinguisher is not possible since by definition the basis is formed by elements of the group which are therefore invertible. In particular, for a cyclic group $G \stackrel{\text{def}}{=} \mathbb{Z}/n\mathbb{Z}$, that is

$$\mathcal{R} \stackrel{\text{def}}{=} \mathbb{F}_q[X]/(X^n - 1),$$

Equation (7.2) does not hold, since the ideal generated by X is the full ring.

Remark 7.31. *One can show that this attack falls into the linear test framework. It was overlooked in previous works using such kind of constructions, and in particular in [BCGI+20b].*

7.3.3.4 Impact of Generic Decoding Algorithms

Let G be a finite abelian group, and let $t \in \{1,\dots,|G|\}$ and $\ell \in \{2,3,4\}$ be integers. Denote by $\mathbb{F}_q[G]_t$ the set of elements of $\mathbb{F}_q[G]$ of Hamming weight exactly t. Consider an instance $(\mathbf{a},\mathbf{s})$ of the $\mathrm{QADP}(G,\ell,t)$ distribution, that is

$$\mathbf{a} = (a_1,\dots,a_{\ell-1}) \leftarrow \mathbb{F}_q[G]^{\ell-1} \quad \text{and } \mathbf{s} = \sum_{i=1}^{\ell-1} a_i e_i + e_\ell, \text{ where } e_i \leftarrow \mathbb{F}_q[G]_t.$$

When unrolled to $\mathbb{F}_q$, QADP corresponds to decoding ℓt errors in a code of length $\ell|G|$ and dimension $(\ell-1)|G|$. Since $|G|$ represents the number of OLE's our PCG is able to produce, for realistic applications we may want to set $|G| \approx 2^{30}$. In other words, we consider the decoding problem in an extreme regime where the length of the code is extremely huge, while we want to set t as small as possible. Nonetheless, the rate of the code is $1-1/\ell$ which is a constant, always greater than $1/2$. In particular, generic attacks which specifically target the LPN regime where the code-rate goes to 0 do not apply here. In particular, in the sequel we will not consider

approaches such as BKW [BKW03; Lyu05], Arora-Ge [AG11], or the more recent Gröbner based attack [BØ23] which improves the complexity in the specific case of a regular error distribution.

Moreover, according to the analysis of [CS16], in the low error regime, the best generic decoding algorithm seems to be the original approach by Prange [Pra62]. The improvements made by advanced ISD tend to only be visible for a higher number of errors. The intuition behind this, is that when the number of errors is small, then an information set picked at random is likely to contain very few errors.

Moreover, in our setting of very long codes with constant code-rate we cannot consider that solving a linear system can be done for free: we need to take into account the cost of linear algebra. This is a second reason why Prange algorithm performs better than more advanced ISD algorithms in this setting.

Wrapping up, the complexity of solving the (generic) Decoding Problem of same parameters than $\mathrm{QADP}(G,\ell,t)$ is of the form (see Section 1.1.4.1 and Equation (1.8)):

$$\frac{\binom{\ell|G|}{\ell t}}{\binom{|G|}{\ell t}} T_{\mathrm{linalg}(|G|)} \approx \ell^{\ell t} \times T_{\mathrm{linalg}(|G|)} \quad \text{given that } |G| \gg \ell t. \tag{7.3}$$

where $T_{\mathrm{linalg}(|G|)}$ is the complexity of inverting a matrix of $\mathbb{F}_q^{|G|\times|G|}$, and the approximation is simply an application of Stirling's formula.

Remark 7.32. *Beware, in [BCCD23], and in particular in the parameter set given on Table 1, t corresponds to the full number of errors: with the notations used so far, this corresponds to sampling each error e_i of weight t/ℓ.*

7.3.3.5 Taking advantage of the structure.

Obviously, QADP has a strong underlying algebraic structure, which we cannot ignore when designing attacks.

Algebraic Decoding. We define the square $\mathscr{C}^2$ of a code $\mathscr{C}$ as

$$\mathscr{C}^2 \stackrel{\text{def}}{=} \mathrm{Span}\{\mathbf{c}\star\mathbf{c}' \mid \mathbf{c},\mathbf{c}' \in \mathscr{C}\}.$$

In traditional code-based cryptography, many codes which proved to offer weak instances of the Decoding Problem appeared to have a square of small dimension (or the square of their dual). In particular, for such codes it is possible to design decoding algorithms based on a framework developed by Pelikaan and Kotter [Pel92; Köt92], and now known as *Error Correcting Pairs* algorithms. The interested reader can refer to [Cou21] for further reference.

However, for the case of quasi-abelian codes, no such decoder is known, even when restricted to the class of quasi-cyclic codes. In particular, a random quasi-abelian code does not seem to have a square of small dimension. Designing an algebraic decoding algorithm for quasi-group codes has been a major open problem in algebraic coding theory ever since they were introduced, and this is one of the open research problems listed in the most recent Encyclopedia of Coding Theory from 2021 [Wil21].

Decoding-One-Out-of-Many. As seen in the preliminaries (Section 1.3.4.1), when a code has a non trivial permutation group, the decoding problem becomes easier. Indeed, in [Sen11] it was proven that when we consider a variant of the Decoding Problem, where we are given N different

noisy codewords, but at the same distance t from the code, and when we are asked to decode only one of them, we can basically speed-up the generic decoding algorithms by a factor $\sqrt{N}$.

When working with codes $\mathscr{C}$ equipped with a non trivial automorphism group G, then from one noisy codeword $y \stackrel{\text{def}}{=} c+e$ we can basically generate $|G|$ many noisy codewords for free, simply by letting G act on y: for any permutation σ we have

$$\sigma \cdot y = \sigma \cdot c + \sigma \cdot e,$$

where $\sigma \cdot e$ still has weight t. Furthermore, when $\sigma \in G$, then $\sigma \cdot c \in \mathscr{C}$ by definition. This allows to speed-up the decoding problem by a factor $\sqrt{|G|}$.

In particular, in our regime of extremely low noise rate, combining the DOOM approach with Equation (7.3), the complexity of solving $\mathrm{QADP}(G,\ell,t)$ now becomes of the form

$$\frac{\binom{\ell|G|}{\ell t}}{\sqrt{|G|}\binom{|G|}{\ell t}} T_{\mathrm{linalg}(|G|)} \approx \ell^{\ell t} \times \left(\frac{T_{\mathrm{linalg}(|G|)}}{\sqrt{|G|}}\right) \tag{7.4}$$

Folding attacks. Let us now really focus on the algebraic structure of the ring

$$\mathbb{F}_q[G] \simeq \mathbb{F}_q[X_1,\dots,X_r]\big/(X_1^{n_1}-1,\dots,X_r^{n_r}-1).$$

Remark 7.33. *The discussion that follows is an extended and more precise version of that of [BCCD23] which was a bit loose.*

Given an instance (a,b) of the decisional $\mathrm{QADP}(G,\ell,t)$ problem, an attacker may construct a new instance of the decoding problem over a code with smaller length and dimension by picking an ideal $\mathcal{I} \subset \mathbb{F}_q[X_1,\dots,X_r]$ containing $(X_1^{n_1}-1,\dots,X_r^{n_r}-1)$ and represented by a Gröbner basis, and constructing a new instance (a',b') by reducing modulo $\mathcal{I}$ with respect to the chosen Gröbner basis. However, in general, this reduction can significantly increase the noise rate. This strongly depends on how sparse are the generators of the Gröbner basis.

Heuristically, the best possible projections seem to arise from quotients of G by a subgroup H. Namely, given a subgroup H, the canonical projection $G \to G/H$ induces a morphism of algebras

$$\pi_H : \left\{ \begin{array}{ccc} \mathbb{F}_q[G] & \longrightarrow & \mathbb{F}_q[G/H] \\ \sum_{g\in G} a_g g & \longmapsto & \sum_{\bar{g}\in G/H} \left(\sum_{h\in H} a_{gh}\right) \bar{g} \end{array} \right. .$$

In algebraic coding theory, and cryptography, this operation is known as *folding* and has been introduced in [FOPP+16a] to try and attack McEliece cryptosystems based on alternant and Goppa codes equipped with a non trivial automorphism group. It has been explicitly studied in [CT19].

Example 7.34. *Let* $G \stackrel{def}{=} \left(\mathbb{Z}/n\mathbb{Z}\right)^2$, *so that*

$$\mathbb{F}_q[G] \simeq \mathbb{F}_q[X,Y]\big/(X^n-1, Y^n-1),$$

and consider the diagonal subgroup

$$H \stackrel{def}{=} \left\{(x,x) \mid x \in \mathbb{Z}/n\mathbb{Z}\right\}.$$

Then, the folding map can be made explicit as

$$\pi_H : \begin{cases} \mathbb{F}_q[X,Y]/(X^n-1, Y^n-1) & \longrightarrow & \mathbb{F}_q[Z]/(Z^n-1) \\ \displaystyle\sum_{i,j=0}^{n-1} a_{i,j} X^i Y^j & \longmapsto & \displaystyle\sum_{i=0}^{n-1} \left(\sum_{u+v \equiv i \mod n} a_{u,v} \right) Z^i. \end{cases}$$

The folding operation sends a code of length $(\ell+1)|G|$ and dimension $\ell|G|$ onto a code of length $(\ell+1)|G/H|$ and dimension $\ell|G/H|$. Moreover, a noisy codeword $y \stackrel{\text{def}}{=} c+e$ is sent onto $\pi_H(c)+\pi_H(e)$. In particular, this keeps the code-rate invariant. However, the weight of the new error $\pi_H(e)$ is bounded from above by that of the original error e, *i.e.* by ℓt, but can be slightly reduced due to the presence of collisions. In a low-noise regime, this is unlikely to happen, but we still need to consider it.

More precisely, we have the following proposition (adapted to language of group algebras):

Proposition 7.35 ([CT19, adapted from Proposition 2])

Let G be a finite abelian group, and let $\tilde{e}$ be uniformly distributed among the elements of $\mathbb{F}_q[G]$ of Hamming weight $t \leqslant |G|/2$. Let $H < G$ be a subgroup of G, and denote by $\pi_H : \mathbb{F}_q[G] \to \mathbb{F}_q[G/H]$ the folding operation with respect to H. Then,

$$\mathbb{E}(|\pi_H(\tilde{e})|) = \frac{(q-1)}{q}|G/H| \left(1 - \left(1 - \frac{qt}{|G|(q-1)}\right)^{|H|}\right)\left(1+O\left(\frac{1}{t}\right)\right).$$

For small values of $\frac{t}{|G|}$, which is the case in our application, this approximates to

$$\mathbb{E}(|\pi_H(\tilde{e})|) \approx t - \frac{(|H|-1)qt^2}{2(q-1)|G|}. \tag{7.5}$$

For $\mathrm{QADP}(G,\ell,t)$, this translates to

$$\mathbb{E}(|\pi_H(e)|) = \ell\big(\mathbb{E}(|\pi_H(\tilde{e})|)\big) \approx \ell\left(t - \frac{(|H|-1)qt^2}{2(q-1)|G|}\right). \tag{7.6}$$

In [CT19] which was concerned in actually decoding, *i.e.* solving the search version of QADP, this approach is used as a subroutine of a decoding algorithm on the whole code of dimension k, by first decoding $t' \stackrel{\text{def}}{=} \mathbb{E}(\pi_H(e))$ errors in the code folded with respect to H, and then each candidate solution e' is lifted back to $\mathbb{F}_q[G]$. The additional information on e' allows to increase the number of equations in the linear system induced by the original decoding problem. This new system may be interpreted in decoding ℓt errors for a code of dimension $\ell|G|-(\ell+1)$. Given the size of $|G|$ compared to ℓ, this does not seem to offer a huge improvement. Still, in our setting, we are only concerned by the decisional problem. Therefore, the situation is simpler, since we do not even need to lift the solutions back to $\mathbb{F}_q[G]$. Indeed, it is enough to solve a non trivial decoding problem in one quotient to get a bias on the QADP distribution.

Note that in general, this approach will only be helpful when considering subgroups H of very large order, given Equation (7.6). Nonetheless, the larger the subgroup H, the smaller the length $(\ell+1)|G/H|$ of the folded code. In particular, when H is too large, the weight of the

folded error vector will become larger than the Gilbert-Varshamov bound of the corresponding rate and length. As such, there will be exponentially many solutions to the decoding problem of the folded code, whether the original b represents a uniform element of $\mathbb{F}_q[G]$, or a syndrome of the random quasi-abelian code. Therefore, this would not allow to distinguish between those two distributions, *i.e.* to solve the decisional version of QADP.

Remark 7.36. *We may also compose the DOOM approach with folding technique to reduce the complexity of the decoding problem in the folding code by a factor* $\sqrt{|G/H|}$.

Comparison with Euclidean Lattices. In Section 7.3.3.1, we mentioned the multivariate Ring-LWE problem which has been introduced in [PTP15; PTP16]. This approach in the lattice-based setting was proven insecure in [BCV20]. It turns out that this attack can be thought of as a folding attack. In fact, this is basically the approach of Example 7.34. It turns out that this folding operation has limited impact on the noise rate in the euclidean setting, while in the Hamming setting, especially in the low noise regime, the weight is more or less preserved while decreasing the length by a factor $|H|$, and thus increasing the noise rate by the same factor $|H|$ after folding.

In other words, what enables to do homomorphic encryption with (structured) lattices, makes this folding attack devastating, while on the other hand this very same obstacle saves the day in the coding theoretic setting.

Remark 7.37. *[HPSS+14] introduced a non standard problem originally called the* Partial Fourier Recovery Problem*, and now known as the* Partial Vandermonde Knapsack Problem*, which was revisited in [LZA18] and [BSS22] as a (structured) lattice problem. However, this problem was broken in [BGP22]. It turns out that this attack can be reformulated in terms of folding.*

7.4 Towards Programmable PCG's for OT

In this section, we extend the discussion in [BCCD23, Appendix D]. More precisely, we give potential research directions in order to overcome the inherent limitations of our constructions.

7.4.1 Limitations of the Construction

In order to build our programmable PCG in Section 7.3, we considered codes over the group algebra $\mathcal{R} \stackrel{\text{def}}{=} \mathbb{F}_q[G]$ with

$$G \stackrel{\text{def}}{=} \left(\mathbb{Z}/(q-1)\mathbb{Z}\right)^r,$$

that is

$$\mathcal{R} = \mathbb{F}_q[X_1, \ldots, X_r]/(X_1^{q-1} - 1, \ldots, X_r^{q-1} - 1)$$

for some integer r. We can set q to be any prime power all the way down to $q = 3$, but obviously this construction is not relevant when $q = 2$.

The most natural approach to mimic the construction is to set

$$\mathcal{R} \stackrel{\text{def}}{=} \mathbb{B}_r \stackrel{\text{def}}{=} \mathbb{F}_2[X_1, \ldots, X_r]/(X_1^2 - X_1, \ldots, X_r^2 - X_r) \tag{7.7}$$

to be the ring of boolean functions in r variables, which is indeed isomorphic to $\underbrace{\mathbb{F}_2 \times \cdots \times \mathbb{F}_2}_{2^r \text{ times}}$. Nevertheless, the attack described in Section 7.3.3.3 extends easily to the multivariate setting, which yields in particular an easy distinguisher for $\mathbb{B}_r$ (defined in Equation 7.7).

Therefore, we seem to be forced to work with group algebras and quasi-group codes. However, a simple combinatorial argument can easily show that this quest is vain.

Theorem 7.38 (Impossibility Result)

Let G be a finite group, and let $\mathcal{R} \stackrel{\text{def}}{=} \mathbb{F}[G]$ be its group algebra with coefficients in a finite field $\mathbb{F}$. Assume that there exists a ring isomorphism $\mathcal{R} \simeq \mathbb{F}_2^N$ for some $N \geqslant 1$. Then, $\mathbb{F} = \mathbb{F}_2, N = 1$ and $G = \{1\}$.

Proof. Clearly, $\mathbb{F}$ must be of characteristics 2. Moreover, any element of G, when regarded as an element of $\mathcal{R}$, is invertible, with inverse g^{-1}. This entails that $|G| \leqslant \left|\mathcal{R}^\times\right|$. On the other hand, the isomorphism $\mathcal{R} \simeq \mathbb{F}_2^N$ induces a group isomorphism

$$\mathcal{R}^\times \simeq \mathbb{F}_2^\times \times \cdots \times \mathbb{F}_2^\times = \{(1, \ldots, 1)\}.$$

In particular, $\left|\mathcal{R}^\times\right| = 1$, and therefore $|G| \leqslant 1$. This concludes the proof. □

Theorem 7.38 shows that there is no hope to directly adapt our approach to efficiently build a PCG for the OLE correlation over $\mathbb{F}_2$, that is a PCG of the OT correlation. However, there might be a way of circumventing this limitation. Indeed, this theorem states that it is not possible to find a non trivial group algebra isomorphic to a product of copies of $\mathbb{F}_2$ *as rings* (and hence *as algebras*), but it does not say anything about an isomorphism as other algebraic structures.

From now on, and until the end of this chapter, we propose new research perspectives.

7.4.2 A Number Theoretic Intuition

Recall that in [BCGI+20b], the authors constructed their PCG for OLE's over the ring $\mathcal{R} \stackrel{\text{def}}{=} \mathbb{F}_p[X]/(P(X))$ where $P(X) \stackrel{\text{def}}{=} X^{2^\ell} + 1$ and p is a (large) prime such that $p \equiv 1 \mod 2^{\ell+1}$. Since $\mathcal{R}$ is isomorphic to a direct product of 2^ℓ copies of $\mathbb{F}_p$, this allowed them to design a PCG for the OLE correlation.

Remark 7.39. *This ring was chosen especially for the efficiency of* NTT*-based multiplication algorithms.*

In fact, this ring identifies to the quotient $\mathcal{O}_K/p\mathcal{O}_K$ of the ring of integers $\mathcal{O}_K$ of the cyclotomic number field $K \stackrel{\text{def}}{=} \mathbb{Q}(\zeta_{2^{\ell+1}})$ where $\zeta_{2^{\ell+1}}$ denotes a primitive $2^{\ell+1}$ root of unity (in $\overline{\mathbb{Q}}$). The Galois group is given by

$$G \stackrel{\text{def}}{=} \mathrm{Gal}(K/\mathbb{Q}) = \left(\mathbb{Z}/2^{\ell+1}\mathbb{Z}\right)^\times \simeq \mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}/2^{\ell-1}\mathbb{Z}.$$

and an element k coprime to $2^{\ell+1}$ acts on $\mathcal{R}$ by $k \cdot X \mapsto X^k$.

Remark 7.40. *This Galois group does not act on the monomial basis (for example* $(2^\ell+1)\cdot X = -X$ *is not an element of the monomial basis), however it still maps the uniform distribution over*

words of weight t to itself. In particular, since it acts transitively over the prime ideals of $\mathcal{O}_K$ above p, this allows to adapt the search-to-decision reduction of Chapter 4 (or that of [LPR10] since we make use of number fields) to this instantiation of Ring-LPN. In fact, the roots of $X^{2^\ell}+1$ form a coset of the group of 2^ℓ-th roots of unity.

This instantiation seems to resist known attacks, although this is not an instantiation of QADP. However, similarly to the discussion of Section 4.4.3.2 since p does not ramify in $\mathcal{O}_K$, there exists a *local normal integral basis*, that is an element $\varepsilon \in \mathcal{R}$ such that $(\sigma \cdot \varepsilon)_{\sigma \in G}$ forms a basis of $\mathcal{R}$. Such a basis can easily be found via Lagrange Interpolation.

Example 7.41. *Let $\ell \stackrel{def}{=} 3$ and $p = 17$. That is, we consider the ring $\mathbb{F}_{17}[X]/(X^8+1)$. We have the following factorisation*

$$X^8+1 = (X+3)(X+5)(X+6)(X+7)(X+10)(X+11)(X+12)(X+14)$$

and a generator of a normal basis is given for instance by the polynomial which is non zero at -3 and vanishes at $\{-5,-6,-7,-10,-11,-12,-14\}$. In this example, one can easily check that

$$\varepsilon(X) \stackrel{def}{=} 6+15X+12X^2+13X^3+7X^4+9X^5+14X^6+X^7$$

generates a normal basis.

Given such a generator ε, we can write any element $a \in \mathcal{R}$ as

$$\sum_{\sigma \in G} a_\sigma (\sigma \cdot \varepsilon) = \left(\sum_{\sigma \in G} a_\sigma \sigma \right) \cdot \varepsilon, \qquad \text{where } a_\sigma \in \mathbb{F}_p.$$

In other words, we can *uniquely* represent any element of $\mathcal{R}$ as the action of any element of the group algebra $\mathbb{F}_p[G]$ on ε. This exactly means that $\mathcal{R}$ is a *free* $\mathbb{F}_p[G]$-module *of rank one*, a basis of which is given by ε. In our setting, this suggests that we could work in a rank-one free $\mathbb{F}_q[G]$-module, instead of directly in $\mathbb{F}_q[G]$.

7.4.3 An Approach Based on the Carlitz Module

In order to circumvent the result of Theorem 7.38, we propose to look for some $\mathbb{F}_2$-algebra $\mathcal{R}$ which is

- isomorphic, as an algebra, to $(\mathbb{F}_2)^N$ for a large N;
- a free $\mathbb{F}_2[G]$-module of rank one for some group G.

In general, $\mathcal{R}$ will not have a distinguished basis and therefore the Hamming metric on $\mathcal{R}$ makes little sense, or at least is not canonical. Moreover, it does not seem obvious to succinctly distribute pseudorandom elements of $\mathcal{R}$, never mind their products. On the other hand, working over group algebras allows to bring all the power of the construction presented in Section 7.3. The isomorphism of modules would preserve (pseudo)randomness.

7.4.3.1 Construction of $\mathcal{R}$

Faithful to the Number Field - Function Field analogy recalled in Chapter 4, we propose an analogous construction to that of 7.4.2, but with function fields, and especially *Carlitz* extensions (see Section 4.2.3).

We will go even further in the analogy. In the number theoretic interpretation of the ring $\mathbb{F}_p[X]/(X^{2^\ell}+1)$, the number field involved was the cyclotomic extension with respect to a primitive $2^{\ell+1}$ root of unity. Similarly, we propose to look at Carlitz extensions with respect to the $T^{\ell+1}$ torsion for some integer ℓ, and set $\mathcal{R}$ to be a quotient of its ring of integers. Let

$$K_\ell \stackrel{\text{def}}{=} \mathbb{F}_2(T)[\Lambda_{T^{\ell+1}}], \quad \text{and } \mathcal{O}_{K_\ell} \stackrel{\text{def}}{=} \mathbb{F}_2[T][\Lambda_{T^{\ell+1}}]$$

where

$$\Lambda_{T^{\ell+1}} \stackrel{\text{def}}{=} \left\{ \lambda \in \overline{\mathbb{F}_2(T)} \mid [T^{\ell+1}](\lambda) = 0 \right\}.$$

Recall that $K/\mathbb{F}_2(T)$ is a Galois extension of degree $N \stackrel{\text{def}}{=} 2^\ell$, whose field of constants is $\mathbb{F}_2$ (*i.e.* the extension is geometric), and whose Galois group is

$$G \stackrel{\text{def}}{=} \mathrm{Gal}(K/\mathbb{F}_2(T)) \simeq \left(\mathbb{F}_2[T]/(T^{\ell+1})\right)^\times.$$

Its structure can be computed through the following proposition:

Proposition 7.42 ([CL17, Proposition 2.4])

The group of units of $\mathbb{F}_2[T]/(T^n)$ is isomorphic to

$$\bigoplus_{1 \leqslant k < \lceil \log(n) \rceil} \left(\mathbb{Z}/2^k\mathbb{Z}\right)^{\left\lceil \frac{n}{2^{k-1}} \right\rceil - 2\left\lceil \frac{n}{2^k} \right\rceil + \left\lceil \frac{n}{2^{k+1}} \right\rceil}.$$

In order to maximise the number of factors, we need to look for an irreducible polynomial $Q \in \mathbb{F}_2[T]$ which splits completely in $\mathcal{O}_K$. In other words, we would get

$$\mathcal{O}_K/Q\mathcal{O}_K = \mathbb{F}_{2^{\deg(Q)}} \times \cdots \times \mathbb{F}_{2^{\deg(Q)}}. \tag{7.8}$$

By Theorem 4.19, $Q\mathcal{O}_K$ splits completely if and only if $Q \equiv 1 \mod T^{\ell+1}$. In particular, $\deg(Q)$ should be large enough. On the other hand, since we want to get a product of copies of $\mathbb{F}_2$, by Equation (7.8) $\deg(Q)$ should be equal to 1, and both conditions are incompatible.

Remark 7.43. *This is the analogue of having $p \equiv 1 \mod 2^{\ell+1}$ in the number field situation, and in particular p large enough.*

Therefore, one needs to make concessions and accept to loosen one of the conditions. Clearly, the second condition ($\deg(Q) = 1$) cannot be relaxed, because all factors of $\mathcal{O}_K/Q\mathcal{O}_K$ are extensions of $\mathbb{F}_{2^{\deg(Q)}}$. One can only hope to relax the first condition, *i.e.* one needs to accept some inertia.

Remark 7.44. *Note that the ramification needs to be trivial, otherwise we would not get a direct product of* fields.

From now on, let us fix some irreducible polynomial $Q \in \mathbb{F}_2[T]$ of degree 1. There are only two possibilities, namely $Q(T) = T$ or $Q(T) = T + 1$. Since T divides $T^{\ell+1}$, it ramifies in $\mathcal{O}_K$. Therefore, the only possibility is $Q(T) = T + 1$. Let f be its inertia degree. By Theorem 4.19,

this is the multiplicative order of $T+1 \mod T^{\ell+1}$. For any integer k it holds that

$$(T+1)^{2^k} = T^{2^k} + 1.$$

When $2^k \geqslant \ell+1$, this becomes

$$(T+1)^{2^k} \equiv 1 \mod (T^{\ell+1}). \tag{7.9}$$

In particular, f must be a power-of-two. Moreover, f is the least power-of-two such that 7.9 holds. In other words, we have

$$f \stackrel{\text{def}}{=} 2^{\lceil \log(\ell+1) \rceil}. \tag{7.10}$$

and

$$\mathcal{O}_K \big/ (T+1)\mathcal{O}_K \simeq \underbrace{\mathbb{F}_{2^f} \times \cdots \times \mathbb{F}_{2^f}}_{N/f \text{ times}}. \tag{7.11}$$

Using only Carlitz extensions, this is the best we can produce using only finite places (We discuss the impact of the place at infinity at the end of Section 7.4.3.2). Nonetheless, it is not required to work with the *full* Carlitz extension, and considering a subextension would clearly decrease the extension degree, but would also allow us to get rid of the inertia.

Let $\mathfrak{P}$ be a prime of $\mathcal{O}_K$ lying above $(T+1)$, and recall (see 4.2) that the decomposition group of $\mathfrak{P}$ is the stabiliser of $\mathfrak{P}$:

$$D_{\mathfrak{P}} \big/ (T+1) \stackrel{\text{def}}{=} \left\{ \sigma \in \mathsf{Gal}(K/\mathbb{F}_2(T)) \;\middle|\; \sigma(\mathfrak{P}) = \mathfrak{P} \right\}.$$

Since G is abelian, this group does not depend on the choice of the prime $\mathfrak{P}$, and will simply be denoted by D_{T+1}. Denote by

$$L \stackrel{\text{def}}{=} K^{D_{T+1}} = \left\{ x \in K \;\middle|\; \sigma(x) = x \quad \forall \sigma \in D_{T+1} \right\}$$

the fixed field of D_{T+1}, called the *decomposition field* of $T+1$. It is readily seen that its ring of integers $\mathcal{O}_L$ is nothing else than $\mathcal{O}_K^{D_{T+1}}$ the subring of $\mathcal{O}_K$ pointwise fixed by D_{T+1}. The particularity of L is that this is the largest subextension in which $(T+1)$ totally splits, hence the name *decomposition field*. In other words, $T+1$ splits completely in L, and then all the places of L above $(T+1)$ are totally inert in K/L. Moreover, since G is abelian, $L/\mathbb{F}_2(T)$ is a Galois extension, of Galois group $H \stackrel{\text{def}}{=} G \big/ D_{T+1}$. In particular, $[L : \mathbb{F}_2(T)] = N/f$ and

$$\mathcal{R} \stackrel{\text{def}}{=} \mathcal{O}_L \big/ (T+1)\mathcal{O}_L \simeq \underbrace{\mathbb{F}_2 \times \cdots \times \mathbb{F}_2}_{N/f \text{ times}}.$$

Similarly to the number field situation, $\mathcal{R}$ admits a local normal integral basis: there exists $\varepsilon \in \mathcal{R}$ such that $(\sigma \cdot \varepsilon)_{\sigma \in H}$ forms an $\mathbb{F}_2$-basis of $\mathcal{R}$. In other words, $\mathcal{R}$ is a free $\mathbb{F}_2[H]$ module of rank 1, which concludes the construction.

On the effectiveness of the construction. As the ring of integers of a Carlitz extension, $\mathcal{O}_K$ is a ring of the form $\mathbb{F}_2[T][X] \big/ (\Phi_{\ell+1}(T, X))$, where $\Phi_{\ell+1}$ can be explicitly computed.

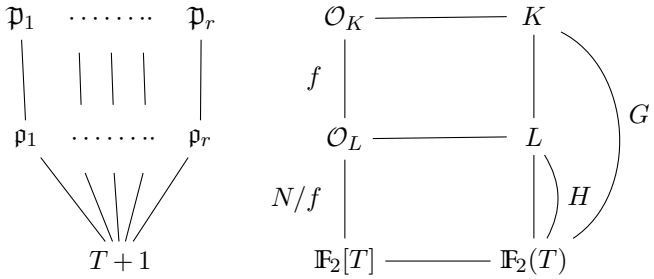


Figure 7.3: Representation of the considered algebraic extensions, with their ring of integers and the splitting behaviour of their primes lying above $T+1$

Remark 7.45. *There is a little difficulty here, though. Computer algebra softwares such as SageMath [Ste+23] or Magma [BCP97] have tremendous difficulties to handle Carlitz extensions, at the time of writing this manuscript. In Sage, no work has been done to implement those extensions (contrary to cyclotomic number fields for example). Although an implementation of Drinfeld modules has recently been made available, starting with SageMath version 10.0 (Released May 20, 2023) [LC23; ACLM23]. On the other hand, in Magma there is a construction of Carlitz modules and Carlitz extensions, however there seems to be some bugs in the implementation, and it randomly crashes with segmentation faults for medium values of ℓ (say above 20). I submitted a bug report, but without any success so far.*

I believe this is mostly due to a lack of consideration of those extensions in the past, especially in characteristic 2, but this should not be a real issue to program efficient implementations. Note that it is not really necessary to actually compute those extensions of function fields for the applications. It is more than enough to work in the finite quotient.

More precisely, $\Phi_{\ell+1}$ is a polynomial of degree 2^ℓ (in X) ; this is the irreducible factor of largest degree of the polynomial $[T^{\ell+1}](X)$ where $[\cdot]$ denotes the Carlitz action (as defined in Section 4.2.3.2). Therefore, we have

$$\mathcal{O}_K \big/ (T+1)\mathcal{O}_K \simeq \mathbb{F}_2[X] \big/ (\Phi_{\ell+1}(1,X)),$$

where $\Phi_{\ell+1}(1,X)$ is now a polynomial of degree 2^ℓ. It turns out that this polynomial is actually of the form $1+P_{\ell+1}(X)$ where $P_{\ell+1}$ is a (linear) 2-polynomial of 2-degree ℓ. In particular, $\Phi_{\ell+1}(1,X)$ is very *sparse*.

Example 7.46. *For $\ell \overset{def}{=} 25$, the Carlitz extension with respect to the $T^{\ell+1}$-torsion yields an extension $K/\mathbb{F}_2(T)$ of degree $N = 2^{25}$. Moreover, $(T+1)$ has inertia $f = 2^{\lceil \log(\ell+1) \rceil} = 2^5 = 32$.*

Furthermore, we can compute (with Magma for instance) that

$$\Phi_{26}(1,X) = 1 + X + X^2 + X^{256} + X^{512} + X^{2^{16}} + X^{2^{17}} + X^{2^{24}} + X^{2^{25}}$$

and the theory of Carlitz extensions allows to prove that

$$\mathcal{O}_K \big/ (T+1)\mathcal{O}_K = \mathbb{F}_2[X] \big/ \Phi_{26}(1,X) = \underbrace{\mathbb{F}_{2^{32}} \times \cdots \times \mathbb{F}_{2^{32}}}_{\frac{2^{25}}{32} = 2^{20} \text{ copies}}.$$

On the other hand, the ring $\mathcal{O}_L$, and its quotient $\mathcal{R} \stackrel{\text{def}}{=} \mathcal{O}_L \big/ (T+1)\mathcal{O}_L$, do not seem to inherit from a "nice" form. In fact, $\mathcal{R}$ cannot be described as the quotient of a univariate polynomial ring with coefficients in $\mathbb{F}_2$ since it is isomorphic to a direct product of a large number N/f of copies of $\mathbb{F}_2$. Nonetheless, it can be described as a *subring* of a polynomial ring. Indeed, recall that by definition D_{T+1} acts as the cyclic group generated by the Frobenius $x \mapsto x^2$ on each factor of $\mathcal{O}_K \big/ (T+1)\mathcal{O}_K$. In other words, $\mathcal{R}$ is the subring of $\mathbb{F}_2[X] \big/ (1+P_{\ell+1}(X))$ fixed by the Frobenius on each factor (after applying the Chinese Remainder Theorem).

In reality, this action can be directly understood before application of the CRT: it is isomorphic to the cyclic group (of order $2^{\lceil \log(\ell+1) \rceil}$) generated by $(T+1)$ in $\left(\mathbb{F}_2[T] \big/ (T^{\ell+1})\right)^\times$, where

$$(T+1) \cdot F(T,X) \stackrel{\text{def}}{=} F(T,[T+1](X)) \quad \text{for } F(T,X) \in \mathcal{O}_K,$$

and

$$[T+1](X) = [T](X) + [1](X) = X^2 + (T+1)X$$

denotes the Carlitz action. Modulo $(T+1)$ we recover that

$$\mathcal{R} \stackrel{\text{def}}{=} \mathcal{O}_L \big/ (T+1)\mathcal{O}_L = \left\{ F(X) \in \mathbb{F}_2[X] \big/ (1+P_{\ell+1}(X)) \;\middle|\; F(X^2) = F(X) \right\}$$

is isomorphic to a direct product of $2^{\ell - \lceil \log(\ell+1) \rceil}$ copies of $\mathbb{F}_2$.

7.4.3.2 Generating many OT's?

If one wants to mimic the construction of [BCGI+20b], it suffices to generate two pseudorandom elements $U, V \in \mathcal{R}$ admitting a *sparse* description, and to distribute to the parties a succinct additive sharing of the product $U \cdot V$. However, there are two issues here:

1. As already mentioned earlier, $\mathcal{R}$ does not appear to have a distinguished basis, and therefore sparsity in $\mathcal{R}$ is not well defined.
2. It does not seem clear how to generate a pseudorandom element in $\mathcal{R}$.

Nevertheless, by construction, $\mathcal{R}$ is a free $\mathbb{F}_2[H]$-module of rank 1, where

$$H \stackrel{\text{def}}{=} \mathrm{Gal}\big(L/\mathbb{F}_2(T)\big) = \mathrm{Gal}\big(K/\mathbb{F}_2(T)\big) \big/ (D_{T+1}) = \left(\mathbb{F}_2[T] \big/ (T^{\ell+1})\right)^\times \Big/ (T+1)\cdot$$

Therefore, we can generate pseudorandom elements in $\mathbb{F}_2[H]$ using the hardness of QADP in a similar fashion as what was done in Section 7.3, and then apply this module isomorphism to get pseudorandom elements in $\mathcal{R}$. This would solve both the problems.

There is a caveat, though. Indeed, the fact that $\mathcal{R}$ and $\mathbb{F}_2[H]$ are isomorphic *as* $\mathbb{F}_2[H]$*-modules* but not *as* $\mathbb{F}_2$*-algebras* essentially means that their multiplicative structures are completely different. In other words, even if we are able to generate an OLE

$$UV = X + Y \in \mathbb{F}_2[H]$$

over $\mathbb{F}_2[H]$, it is not clear how to send it back to $\mathcal{R}$. In short, given ε a generator of $\mathcal{R}$ as an $\mathbb{F}_2[H]$-module, we can compute $U \cdot \varepsilon$ and $V \cdot \varepsilon$, as well as

$$X \cdot \varepsilon + Y \cdot \varepsilon = (X+Y) \cdot \varepsilon = (UV) \cdot \varepsilon$$

but

$$(U \cdot \varepsilon)(V \cdot \varepsilon) \neq (UV) \cdot \varepsilon.$$

Wrapping up, the only problem that needs to be solved to design a programmable PCG for OT's following this Carlitz approach is to be able to succinctly share elements of $\mathcal{R}$ of the form

$$(UV) \cdot \varepsilon$$

given only $U \cdot \varepsilon$ and $V\varepsilon$. In order to solve this problem, it seems to be interesting to start looking into so-called (Reverse) Multiplication Friendly Embeddings.

Multiplication Friendly Embeddings. Cascudo, Chen, Cramer and Xing introduced in [CCCX09] the notion of *Multiplication Friendly Embedding* (MFE), which provides a way to embed some field $\mathbb{F}_{q^m}$ into a ring of the form $\mathbb{F}_q^r$.

Definition 7.47 ([CCCX09, Definition 11] Multiplication Friendly Embedding)

An MFE of $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$ is a triple (r, σ, ψ) where r is a positive integer called the *embedding expansion* and where $\sigma \colon \mathbb{F}_{q^m} \to \mathbb{F}_q^r$ and $\psi \colon \mathbb{F}_q^r \to \mathbb{F}_{q^m}$ are $\mathbb{F}_q$-linear maps such that

$$xy = \psi(\sigma(x) \star \sigma(y)), \quad \forall x, y \in \mathbb{F}_{q^m}.$$

Remark 7.48. *Note that for any such triple (r, σ, ψ), the map σ is necessarily* injective*, and therefore this really defines an* embedding *of $\mathbb{F}_{q^m}$ into $\mathbb{F}_q^r$.*

Remark 7.49. *Such maps are also at the core of Chudnovsky-Chudnosky type algorithms which often provide the best algorithms for multiplying elements in a finite field (with respect to the bilinear complexity), see [BPRR+21; BP23].*

This notion was also introduced in the other way around by Cascudo, Cramer, Xing and Yuan in [CCXY18] under the name *Reverse Multiplication Friendly Embeddings* (RMFE), in order to embed some ring $\mathbb{F}_q^k$ into an extension field $\mathbb{F}_{q^m}$, somehow preserving the multiplication.

Definition 7.50 ([CCXY18, Definition 1] Reverse Multiplication Friendly Embedding)

Let $k, n \geqslant 1$ be integers. A pair (φ, ψ) is called a (k, m)-*Reverse Multiplication Friendly Embedding* if $\varphi : \mathbb{F}_q^k \to \mathbb{F}_{q^m}$ and $\psi : \mathbb{F}_{q^m} \to \mathbb{F}_q^k$ are two $\mathbb{F}_q$-linear maps satisfying

$$\mathbf{x} \star \mathbf{y} = \psi(\varphi(\mathbf{x}) \cdot \varphi(\mathbf{y})), \quad \forall \mathbf{x}, \mathbf{y} \in \mathbb{F}_q^k.$$

Remark 7.51. *The aforementioned references, as well as subsequent works, constructed such* RMFE*′s and* RMFE*′s using tools coming from algebraic geometry, and more precisely recursive towers of function fields.*

Coming back to our problem, composing the module isomorphism $\mathbb{F}_2[H] \simeq \mathcal{R}$ with the algebra isomorphism $\mathcal{R} \simeq \mathbb{F}_2^N$ yields a linear isomorphism $\theta \colon \mathbb{F}_2[H] \simeq \mathbb{F}_2^N$. The Holy Grail would be to find a map $\sigma : \mathbb{F}_2^N \simeq \mathbb{F}_2[H]$ such that

$$\mathbf{x} \star \mathbf{y} = \theta(\sigma(\mathbf{x}) \cdot \sigma(\mathbf{y})).$$

Indeed, given our pseudorandom OLE

$$U \cdot V = X + Y \in \mathbb{F}_2[H]$$

then we would have

$$\sigma^{-1}(U) \star \sigma^{-1}(V) = \theta(U \cdot V) = \theta(X + Y) = \theta(X) + \theta(Y)$$

which would directly induce N OLE over $\mathbb{F}_2$ if σ and θ behave nicely with respect to the uniform distribution. Note that σ^{-1} and θ could be computed locally by each party. This would therefore conclude the PCG construction.

Under this generality, this goal is maybe unrealistic, but some further relaxations might allow to finish the construction.

To infinity, and beyond. There is an interesting property of Carlitz extensions which we have not exploited at all in this manuscript, namely the splitting behaviour of the place at infinity. This fact was already hinted in Chapter 4. Recall that $\mathbb{F}_q(T)$ contains another ring of interest without which the arithmetic of the function field is not complete, namely $\mathbb{F}_q[T]_\infty \stackrel{\text{def}}{=} \mathbb{F}_q\left[\frac{1}{T}\right]$. Given $K/\mathbb{F}_q(T)$ a Carlitz extension with respect to the torsion of *any* polynomial $M \in \mathbb{F}_q[T]$, one can define the integral closure of $\mathcal{O}_{K,\infty}$ of $\mathbb{F}_q[T]_\infty$, which is also a Dedekind domain, and look at the factorisation of $(1/T)\mathcal{O}_{K,\infty}$. It turns out that this place at infinity has *always* ramification index $e_\infty(q) \stackrel{\text{def}}{=} q-1$ (see [Vil06, Section 12.4.3]). In particular, the place at infinity does not seem attractive given our goal. Nevertheless, $e_\infty(2) = 1$ precisely means that $1/T$ splits *completely* in $\mathcal{O}_{K,\infty}$ when $q = 2$, *i.e.*

$$\mathcal{R}_\infty \simeq \mathcal{O}_{K,\infty} \big/ (1/T)\mathcal{O}_{K,\infty} \simeq \mathbb{F}_2^N, \quad \text{as algebras}$$

where $N \stackrel{\text{def}}{=} [K : \mathbb{F}_2(T)]$. Moreover, denoting by $G \stackrel{\text{def}}{=} \mathrm{Gal}(K/\mathbb{F}_2(T))$, one can prove that $\mathcal{R}_\infty$ is as well a free $\mathbb{F}_2[G]$-module of rank 1.

In other words, letting $K \stackrel{\text{def}}{=} \mathbb{F}_2(T)(\Lambda_{T^{\ell+1}})$ as before, we can easily construct two free $\mathbb{F}_2[G]$-modules of rank 1, namely

$$\mathcal{O}_K \big/ (T+1)\mathcal{O}_K \simeq \underbrace{\mathbb{F}_{2^f} \times \cdots \times \mathbb{F}_{2^f}}_{N/f \text{ times}}$$

and

$$\mathcal{O}_{K,\infty} \big/ (1/T)\mathcal{O}_{K,\infty} \simeq \underbrace{\mathbb{F}_2 \times \cdots \times \mathbb{F}_2}_{N \text{ times}},$$

which might be helpful to construct a (reverse) *multiplication-friendly* embedding

$$\mathbb{F}_2^{N/f} \hookrightarrow \mathbb{F}_2[G].$$

7.4.4 On the Efficiency of the Construction

If the previous problem was solved, this would allow to build the programmable PCG for OT, but only theoretically. Indeed, a priori many other obstacles seem to thwart efficient implementations.

For the construction to be of practical interest, we need to be able to efficiently compute the isomorphism $\mathcal{R} \stackrel{\text{def}}{=} \mathcal{O}_L \big/ (T+1)\mathcal{O}_L \simeq \mathbb{F}_2^{N/f}$. Since elements of $\mathcal{R}$ are actually seen as elements of

$\widetilde{\mathcal{R}} \stackrel{\text{def}}{=} \mathbb{F}_2[X]/(1+P_{\ell+1})$, we want to compute the evaluation at every root of $1+P_{\ell+1}$. We also need to be able to efficiently compute multiplications in the group algebra $\mathbb{F}_2[G]$ (or $\mathbb{F}_2[H]$), that is we need to be able efficiently *encode* quasi-G codes (or quasi-H) codes. In reality, those two problems are related. Note that *a priori* there is a difficulty here, since the group algebra is not semisimple and the algorithms recalled in Section 7.2.4 do not apply here.

In order to try and circumvent this state of affairs, we will here again, get inspired by the cyclotomic situation. The following presentation will seem pedantic and overly complicated in this case, but will actually be very insightful when turning to the characteristic 2 situation.

7.4.4.1 Standard NTT

Let $\zeta_{2^{\ell+1}}$ be a primitive $2^{\ell+1}$-th root of unity (in $\overline{\mathbb{Q}}$), and consider the cyclotomic number field

$$K \stackrel{\text{def}}{=} \mathbb{Q}(\zeta_{2^{\ell+1}}) = \mathbb{Q}[X]/(X^{2^\ell}+1).$$

As recalled in Section 7.4.2, if p is a prime such that $p \equiv 1 \mod 2^{\ell+1}$, then $p\mathcal{O}_K$ splits completely in $\mathcal{O}_K$ and

$$\mathcal{O}_K/p\mathcal{O}_K = \mathbb{Z}[X]/(p, X^{2^\ell}+1) = \mathbb{F}_p[X]/(X^{2^\ell}+1).$$

For $k \in \{0,\dots,\ell+1\}$, consider the group $\mathbb{U}_k \subset \mathcal{O}_K^\times$ of 2^k-th roots of unity, and denote by φ_2 the squaring operator. Note that φ_2 is a group homomorphism which maps $\mathbb{U}_{k+1}$ onto $\mathbb{U}_k$, with kernel of size 2. In particular, this defines a Jordan-Hölder composition series of $\mathbb{U}_{\ell+1}$, with all the factors equal to $\mathbb{Z}/2\mathbb{Z}$, which we write in decreasing order:

$$\mathcal{O}_K^\times \supset \mathbb{U}_{\ell+1} \xrightarrow{\varphi_2} \mathbb{U}_\ell \xrightarrow{\varphi_2} \cdots \xrightarrow{\varphi_2} \mathbb{U}_1 \xrightarrow{\varphi_2} \mathbb{U}_0 = \{1\}.$$

Denote by

$$\pi\colon \mathcal{O}_K \twoheadrightarrow \mathcal{O}_K/p\mathcal{O}_K$$

the projection modulo $p\mathcal{O}_K$. Obviously, $\pi(\mathbb{U}_k)$ is a subgroup of $\mathbb{F}_p^\times$ formed by the 2^k-th roots of unity, and the above composition series is well defined on the quotient:

$$\mathbb{F}_p^\times \supset \pi(\mathbb{U}_{\ell+1}) \xrightarrow{\varphi_2} \pi(\mathbb{U}_\ell) \xrightarrow{\varphi_2} \cdots \xrightarrow{\varphi_2} \pi(\mathbb{U}_1) \xrightarrow{\varphi_2} \pi(\mathbb{U}_0) = \{1\}.$$

Let $R_\ell \stackrel{\text{def}}{=} \{\omega_1,\dots,\omega_{2^\ell}\}$ denote the roots (in $\mathbb{F}_p$) of the polynomial $X^{2^\ell}+1$. Since $R_\ell \subset \mathbb{U}_{\ell+1}$, the multi-evaluation of a polynomial on R_ℓ is a subset of the multi-evaluation of said polynomial on the group $\mathbb{U}_{\ell+1}$. In other words, computing the Chinese isomorphism

$$\Phi_{\ell+1}\colon \left\{ \begin{array}{ccc} \mathbb{F}_p[X]/(X^{2^\ell}+1) & \longrightarrow & \mathbb{F}_p^{2^\ell} \\ P & \longmapsto & (P(\omega_1),\dots,P(\omega_{2^\ell})) \end{array} \right. ,$$

can be done by seeing P as a polynomial of degree $< 2^{\ell+1}$ and evaluating at all $\mathbb{U}_{\ell+1}$, before discarding half of the evaluations. This can be done via an FFT of size $2^{\ell+1}$: the polynomial P is first split into its even and odd parts

$$P(X) = \sum_{0\leqslant j<2^{\ell+1}} a_j X_j = \left(\sum_{0\leqslant j<2^\ell} a_{2j} X^{2j}\right) + X\cdot\left(\sum_{0\leqslant j<2^\ell} a_{2j+1} X^{2j}\right) = P_{\text{even}}(\varphi_2(X)) + X\cdot P_{\text{odd}}(\varphi_2(X)),$$

which can then be recursively evaluated onto all of $\mathbb{U}_\ell$.

Remark 7.52. *This is obviously an inefficient way of doing said task. In particular, a simple optimisation consists in remarking that $R_{\ell+1}$ is simply a coset of $\mathbb{U}_\ell$. Therefore, computing $\Phi_{\ell+1}$ is equivalent to evaluating a translated polynomial at all the 2^ℓ-th roots of unity.*

Due to its ubiquity in modern cryptography, this particular multi-evaluation has been highly optimised, but we left out those optimisations. The goal of this section is simply to hint at what is possible.

7.4.4.2 A Carlitz Module Analogy

Let us now turn to our ring $\widetilde{\mathcal{R}} \stackrel{\text{def}}{=} \mathbb{F}_2[X]/(1+P_{\ell+1}(X))$. By construction,

$$\widetilde{\mathcal{R}} = \mathcal{O}_K/(T+1)\mathcal{O}_K,$$

where $K \stackrel{\text{def}}{=} \mathbb{F}_2(T)(\Lambda_{T^{\ell+1}})$ is the Carlitz extension with respect to the $T^{\ell+1}$ torsion. For simplicity, we shall drop the T and just write $\Lambda_{\ell+1}$ for the $T^{\ell+1}$-torsion.

Recall that $\Lambda_{\ell+1}$ is a cyclic $\mathbb{F}_2[T]$-module, with respect to the Carlitz action. Under the Cyclotomic-Carlitz analogy, this will play the role of the roots of unity $\mathbb{U}_{\ell+1}$ (in $\mathcal{O}_K$). Note that the 2-to-1 map φ_2 defined earlier to be the squaring operator is nothing but the cyclotomic action of 2 on $\mathcal{O}_K^\times$. Similarly, let

$$\varphi_T\colon \left\{ \begin{array}{ccc} \mathcal{O}_K & \longrightarrow & \mathcal{O}_K \\ a & \longmapsto & [T](a) = a^2 + T\cdot a \end{array} \right.$$

denote the Carlitz action of T on $\mathcal{O}_K$. It turns out that φ_T is an $\mathbb{F}_2[T]$-module homomorphism that maps Λ_{k+1} onto Λ_k for all $k \in \{0,\dots,\ell\}$, with a kernel of size 2. In other words, we have a composition series *of $\mathbb{F}_2[T]$-modules*,

$$\mathcal{O}_K \supset \Lambda_{\ell+1} \xrightarrow{\varphi_T} \Lambda_\ell \xrightarrow{\varphi_T} \cdots \xrightarrow{\varphi_T} \Lambda_1 \xrightarrow{\varphi_T} \Lambda_0 = \{0\},$$

where the factors Λ_{k+1}/Λ_k are all $\mathbb{F}_2$-vector spaces of dimension 1. After reduction modulo $(T+1)\mathcal{O}_K$, this yields the following composition series

$$\widetilde{\mathcal{R}} \supset \pi(\Lambda_{\ell+1}) \xrightarrow{\overline{\varphi_T}} \pi(\Lambda_\ell) \xrightarrow{\overline{\varphi_T}} \cdots \xrightarrow{\overline{\varphi_T}} \Lambda_1 \xrightarrow{\overline{\varphi_T}} \pi(\Lambda_0) = \{0\},$$

where $\pi : \mathcal{O}_K \twoheadrightarrow \widetilde{\mathcal{R}}$ is the canonical projection, and

$$\overline{\varphi_T}\colon \left\{ \begin{array}{ccc} \widetilde{\mathcal{R}} & \to & \widetilde{\mathcal{R}} \\ a & \mapsto & \left\{ \begin{array}{ll} [T](a) & \mod (T+1) \\ = a^2 + Ta & \mod (T+1) \\ = a^2 + a & \end{array} \right. \end{array} \right.$$

is the reduction modulo $(T+1)\mathcal{O}_K$ of the Carlitz action of T. Using this composition series allows to design a divide-and-conquer algorithm to evaluate an element of $\widetilde{\mathcal{R}}$ at all the elements of $\pi(\Lambda_{\ell+1})$.

Remark 7.53. *Note that as $\mathbb{F}_2[T]$-modules, the Λ_k's are in particular vector spaces, whereas the roots of $1+P_{\ell+1}$ form an affine space. In fact, those roots are simply a coset of $\pi(\Lambda_\ell)$, just like*

the roots of $X^{2^\ell}+1$ *form a coset of* $\pi(\mathbb{U}_\ell)$*. After applying a translation (which does not change the degree), this allows to compute efficiently the Chinese isomorphism*

$$\widetilde{\mathcal{R}} = \mathbb{F}_2[X]\big/(1+P_{\ell+1}(X)) \to (\mathbb{F}_{2^f})^{2^\ell/f},$$

where $f = 2^{\lceil \log(\ell+1)\rceil}$.

Since we are working in characteristic 2, the operation $a \mapsto a^2 + a$ is linear, and in fact this Carlitz point of view recovers some algorithms already known in the literature as the *Additive Fast Fourier Transform*, introduced by Cantor in 1989 [Can89], and later refined in subsequent works [GG96; GM10; Cox21]. The latter reference is dedicated to optimised implementations. In those works, a basis $(\beta_1, \dots, \beta_\ell)$ of a vector space V, such that $\beta_{i+1} = \beta_i^2 + \beta_i$ is referred to as a *Cantor basis* of V. The Carlitz point of view, which to the best of my knowledge seems new, gives more insight about where such vector spaces come from.

When we restrict ourselves to the fixed ring $\mathcal{R} \stackrel{\text{def}}{=} \mathcal{O}_L\big/(T+1)\mathcal{O}_L$, since we know that the evaluation will yield elements of $\mathbb{F}_2$ (and not in general $\mathbb{F}_{2^f}$), we can win back the factor f *a priori* lost in the above Chinese isomorphism by using the same Frobenius trick as in the Frobenius Fast Fourier Transform of van der Hoeven and Larrieu [HL17]. This has been considered in [LCKC+18].

Remark 7.54. *This Carlitz approach can be generalised over larger finite fields, but seems particularly suited in small characteristics.*

7.4.4.3 Computing in $\mathbb{F}_2[G]$

Representations of a group G in $\mathbb{F}_q$ when $\gcd q, |G| > 1$ are known as *modular representations*. The literature is very wide, and the theory is more complicated than in the usual, coprime situation. In particular, the group algebra $\mathbb{F}_q[G]$ is not semisimple anymore, and some *reducible* representations (*i.e.* having non trivial subrepresentations) are actually *indecomposable* (*i.e.* they cannot be written as a direct sum of subrepresentations). As a consequence, the theory of characters, as well as the Fourier Transform, are not well defined, and effective computations in such group algebras have not been much explored.

However, in our case, G is a group of order 2^ℓ, whose complete structure is given by Proposition 7.42, and we might be able to say something.

Example 7.55. *If we continue with Example 7.46, setting* $\ell = 25$*, and considering the Carlitz extension* K *of degree* 2^{25} *given by the* T^{26}*-torsion, Proposition 7.42 entails that*

$$G \stackrel{def}{=} \mathrm{Gal}(K/\mathbb{F}_2(T)) \simeq \left(\mathbb{F}_2[T]\big/T^{26}\right)^\times \simeq \left(\mathbb{Z}\big/2\mathbb{Z}\right)^7 \times \left(\mathbb{Z}\big/4\mathbb{Z}\right)^3 \times \mathbb{Z}\big/8\mathbb{Z} \times \mathbb{Z}\big/16\mathbb{Z} \times \mathbb{Z}\big/32\mathbb{Z},$$

and since D_{T+1}*, the decomposition group relative to* $(T+1)$*, is cyclic isomorphic to* $\mathrm{Gal}\left(\mathbb{F}_{2^{32}}/\mathbb{F}_2\right)$*, we have*

$$H \stackrel{def}{=} \mathrm{Gal}(L/\mathbb{F}_2(T)) = G\big/D_{T+1} \simeq \left(\mathbb{Z}\big/2\mathbb{Z}\right)^7 \times \left(\mathbb{Z}\big/4\mathbb{Z}\right)^3 \times \mathbb{Z}\big/8\mathbb{Z} \times \mathbb{Z}\big/16\mathbb{Z}.$$

By remarking that

$$\mathbb{F}_2\left[\mathbb{Z}\big/2^k\mathbb{Z}\right] \simeq \mathbb{F}_2[X]\big/(X^{2^k}),$$

and using the tensor product representation, we note that $\mathbb{F}_2[G]$ is isomorphic to a multivariate polynomial ring of the form

$$\mathbb{F}_2[X_1, \ldots, X_r] \Big/ (X_1^{2^{n_1}}, \ldots, X_r^{2^{n_r}}),$$

with $\ell = \sum_{i=1}^{r} n_i$. As a consequence, multiplication in $\mathbb{F}_2[G]$ identifies to a multiplication in $\mathbb{F}_2[X_1, \ldots, X_r]$, where we discard the monomials of too high degrees, and it may be possible to use an additive multivariate Fast Fourier Transform algorithm to compute it more or less efficiently.

Nevertheless, this description is very naive, and further studies may actually design efficient algorithms for computing in modular group algebras over $\mathbb{F}_2$. In particular, a recent work of Hong, Viterbo and Belfiore [HVB16] introduced a *modular* FFT for specific group algebras of the form $\mathbb{F}_2\left[\left(\mathbb{Z}/2\mathbb{Z}\right)^s\right]$. Their algorithm is particularly efficient because it only involves *additions*, and could therefore benefit from hardware optimisations. Faithful to our interpretation of additive FFT in terms of Carlitz modules, a good starting point towards more general algorithms could be to think in terms of series of submodules of $\mathbb{F}_2[G]$, and not only in terms of subgroups of G.

Conclusion and Future Work

All good things must come to an end.

Old proverb

Structured error correcting codes have been used for quite a long time now to build cryptography, for they offer a very good trade-off between hardness of the Decoding Problems, and efficiency of the operations, as well as compactness of the representation. They yield interesting encryption schemes with very short key sizes, two of them (BIKE [AABB+22a] and HQC [AABB+22b]) having made their way to the fourth round of NIST competition.

Nevertheless, before the work developed in this thesis, no reduction was known between the decisional and search versions of the decoding problem, and the hardness was mainly assessed through cryptanalysis, or more precisely the lack thereof, and the test of time. However, without such theoretical guarantees, cryptosystems are not immune to be actually weaker than previously thought. In particular, the cryptosystems LIGA ([RPW21]) and RAMESSES ([LLP20]) introduced in Part I, did not benefit from any reduction to well established problems,and in Chapter 3 we managed to give a full message recovery attack against both cryptosystems.

This state of affairs suggests the necessity of having stronger guarantees of hardness on the underlying computational problems. In the realm of cryptanalysis, the *linear test framework* (Section 1.3.4.3) gives a first answer to this issue, by providing a simple condition to avoid a large class of attacks: for the decisional version of the Decoding Problem to be hard (with respect to the currently known best attacks) for a specific class of codes (*e.g.* general linear codes, quasi-cyclic codes), it is enough that those codes have a large *dual distance*. However, this does not prevent new ideas to be at the origin of new attacks which do not fit in this framework.

Another way of increasing the confidence in a cryptosystem is providing theoretical reductions, and more precisely *search-to-decision reductions* when we are interested in the decisional variant of a problem. This has been one of the strengths of lattice-based cryptography for the past decade. Indeed, many reductions have been derived using the power of algebraic number theory. Given the resemblances between the two domains, it was natural to ask if we could adapt their techniques in code-based cryptography. This was the whole idea of Part II where we introduced a new *function field interpretation* of the decoding problem of structured codes with the *Function Field Decoding Problem* (FF-DP) in Chapter 4. This can be thought as an analogue of structured variants of LWE in positive characteristics, using *Carlitz modules* instead of *cyclotomic number fields*. However, there are arithmetic limitations inherent to our approach, and we tried to lift them in Chapter 5 by adapting the most recent technique used in lattice-based cryptography to give reductions, namely the OHCP framework, and more precisely its very essence the Oracle Comparison Problem (OCP). With this technique inspired from learning theory, we were able to give a new (worst-case to average-case, search-to-decision) reduction for

the unstructured decoding problem. Unfortunately, the structured variants still remain out of reach for the moment.

Finally, besides encryption schemes, quasi-cyclic error correcting codes, and their Decoding Problems, have been considered in the context of Secure Multiparty Computation (MPC), and especially for providing more efficient ways of building (and distributing) *correlated randomness* to all the parties involved in the protocol. More precisely, they were used in [BCGI+20b] to build *Pseudorandom Correlation Generators* (see Section 6.4). However, their interpretation using univariate polynomials induced an inherent limitation in the size of the field which could be used: it should be larger than the number of correlated random elements one wants to generate. In order to compute useful functions, this means working over fields having billions of elements, or even cryptographic size (in [BCGI+20b] they proposed to instantiate the construction by working over prime fields $\mathbb{F}_p$ with a 128-bit prime p).

This suggests to try and use multivariate polynomials in order to lift the limitation. However, as we have seen with *boolean functions* in Section 7.4.1, assessing the security is not straight forward. This is where structured error correcting codes come into play. Indeed, by limiting the construction to ring of multivariate polynomials which are isomorphic to the group algebra of an abelian group, and interpreting the underlying problem as the decision version of the *Quasi-Abelian Decoding Problem* (QADP, Problem 7.22), we were able to give strong security guarantees, with a search-to-decision reduction (in the spirit of that of Chapter 4), as well as the resistance against concrete known attacks.

We believe that the context of *quasi-abelian codes* (and potentially more generally *quasi-group codes*) is the right way of understanding structured error correcting codes (endowed with the Hamming metric) in cryptography, for it offers a general framework which encompasses all the previously used versions of the Decoding Problem, namely the generic plain Decoding (Problem 1.10) and the Quasi-Cyclic Decoding Problem (QC-DP, Problem 4.1). In particular, it offers a nice intermediate between the structured and unstructured variants, in a unified way.

Future Work

We have tried in the corresponding chapters to give a detailed presentation of research directions which we would like to pursue. Instead, here we will give a more conceptual overview of future directions.

Towards more general reductions

Rényi Divergence. In Chapter 5 we tried to overcome the arithmetic limitations of the search-to-decision reduction from Chapter 4 by considering the modern approach now used in lattice-based cryptography based on the Oracle Comparison Problem (OCP), declined into OHCP in lattice-based cryptography. This allowed us to give the first direct worst-case to average-case search-to-decision reduction for the plain decoding problem, however we were unable to succeed in the case of quasi-cyclic codes. Nevertheless, even though the problem which we build does not seem to be enough for the reduction, this is mostly due to the weird shape of the noise, which does not seem to be statistically close to the ideal noise model we would like. In order to mitigate this state-of-affairs, it would be interesting to investigate other tools to measure the distance between probability distributions. For example, the so-called *Rényi Divergence* has been successfully exploited in lattice-based cryptography to improve on bounds in reductions (*e.g.* [Pre17]). Furthermore, it has even been used in a context where the *Statistical Distance* was not negligible (*e.g.* [LSS14]), which precisely seems to be our setting.

Reductions for other metrics. Our work also demonstrates that the OCP framework which had only been used in the context of lattice-based cryptography, and with the euclidean norm, could be exploited to derive reductions for the Hamming metric. This naturally raises the question of other metrics.

As we have seen in Part I, the rank metric has been used to design code-based cryptosystems, however the theoretical foundations still remain quite unclear, for the only known reduction is the randomised reduction to the Decoding Problem in the Hamming metric from [GZ16]. It would be very interesting to investigate whether the OCP framework could also be used in this context. This does not seem to be straightforward, and many tools should be introduced, such as *smoothing bounds* in the rank metric, which should involve some Fourier analysis, however this seems to be worth exploring.

As mentioned in this chapter, another metric of interest has recently been used in a cryptographic context, namely the *Lee metric*, which appears to be an intermediate between the Hamming metric in traditional code-based cryptography, and the Euclidean distance in lattice-based cryptography. However, this metric seems to be even further away than the rank metric, since the notion of support is not properly defined. As such, this metric looks closer to lattice-based cryptography and it might be necessary to adapt a *random walk technique* such as the one used inside OHCP. Nevertheless, the natural distributions capturing the Lee metric seems to behave poorly with respect to Fourier analysis. More precisely, one can show that if f is a function which only depends on the weight of its input, whether it be for the rank metric, the Hamming metric or the euclidean norm,[ii], then the Fourier transform $\hat{f}$ also only depends on the weight. It turns out that this Innocent looking fact is a crucial tool of known ways to derive smoothing bounds for a given metric. Surprisingly, for the Lee metric this is not true anymore, and new ideas might be needed.

New tools for reductions ; using representation theory ? In this work, we were focused in adapting the techniques used in lattice-based cryptography to derive reductions. As mentioned in the end of Chapter 5, a new technique for deriving a random self-reducibility of structured problems in lattice-based cryptography has recently been investigated, namely random walks on some algebraic structures such as the *Arakelov class group*. This research direction might be worth exploring in the context of structured error correcting codes, especially in light of the quasi-abelian decoding framework.

However, it may also be interesting to try to go beyond adapting techniques used in lattice-based cryptography, and to develop new techniques especially targeting error correcting codes. In particular, there is a notion which does not seem to have really been explored so far in code-based cryptography, namely *representation theory*. Quasi-group codes are precisely modules over a group algebra $\mathbb{F}_q[G]$, and as such are strongly connected to the representations of G with coefficients in $\mathbb{F}_q$. Moreover, Fourier analysis is typically a tool from representation theory. It could be interesting to see if techniques arising from the theory of representations of finite groups could be used to design more useful reductions for structured codes.

Number theory in function fields and applications.

Pseudorandom Correlation Generators. As already mentioned in the end of Chapter 7, the function field framework which we have developed in Chapter 4, and especially the introduction of *Carlitz modules* in a cryptographic context, seems to offer an interesting starting point in order to overcome the limitations of the construction of our *programmable Pseudorandom Correlation*

[ii] such a function is called *radial*

Generator (PCG). If this could be pushed further, this would potentially have important applications in secure multiparty computation. The story would be similar to the development of tools from algebraic number theory which are used to build efficient *Fully Homomorphic Encryption*.

Towards efficient computations in modular group algebras ? Even if we manage to overcome all the theoretical limitations, there is still much work to do in order to make it practical. More precisely, we would need to develop fast algorithms for multiplying two elements of $\mathbb{F}_2[G]$ when $|G|$ is a power-of-two. This is a particular case of a *modular* group algebra, and since it is not semisimple, and the usual approaches based on the Discrete Fourier Transform do not really make sense anymore. However, the story is not necessarily dead in its tracks. In particular, some algorithms have been considered for developing fast arithmetic in the context of *Additive Fourier Transform*. They have even already been used in a cryptographic context in [BBHR18a; BBHR18b] to build efficient *Interactive Oracle Proofs* in characteristics 2, to be used inside Zero-Knowledge protocols.

The very essence of our idea presented in Section 7.4.4.3 is simple: yes, we cannot rely on classical Fourier theory, and especially characters. However, we might still be able to do something by replacing subgroups by submodules, in a similar fashion that building Carlitz extensions corresponds to adjoining an $\mathbb{F}_2[T]$-module of torsion elements, instead of the abelian group of the roots of unity.

Bibliography

[AAAC+20] Gorjan Alagic, Jacob Alperin-Sheriff, Daniel Apon, David Cooper, Quynh Dang, John Kelsey, Yi-Kai Liu, Carl Miller, Dustin Moody, Rene Peralta, Ray Perlner, Angela Robinson, and Smith-Tone Daniel. *Status report on the second round of the NIST post-quantum cryptography standardization process*. https://nvlpubs.nist.gov/nistpubs/ir/2020/NIST.IR.8309.pdf. 2020 (cit. on pp. x, 96).

[AABB+20] Carlos Aguilar Melchor, Nicolas Aragon, Slim Bettaieb, Loïc Bidoux, Olivier Blazy, Maxime Bros, Alain Couvreur, Jean-Christophe Deneuville, Philippe Gaborit, Gilles Zémor, and Adrien Hauteville. *Rank Quasi Cyclic (RQC)*. Second Round submission to NIST Post-Quantum Cryptography call. Apr. 2020 (cit. on p. 58).

[AABB+22a] Carlos Aguilar Melchor, Nicolas Aragon, Paulo Barreto, Slim Bettaieb, Loïc Bidoux, Olivier Blazy, Jean-Christophe Deneuville, Philippe Gaborit, Santosh Ghosh, Shay Gueron, Tim Güneysu, Rafael Misoczki, Edoardo Persichetti, Jan Richter-Brockmann, Nicolas Sendrier, Jean-Pierre Tillich, Valentin Vasseur, and Gilles Zémor. *BIKE*. Round 4 Submission to the NIST Post-Quantum Cryptography Call, v. 5.1. Version 5.1. Oct. 2022 (cit. on pp. viii, 2, 4, 22, 42, 43, 50, 54, 58, 95, 227).

[AABB+22b] Carlos Aguilar Melchor, Nicolas Aragon, Slim Bettaieb, Loïc Bidoux, Olivier Blazy, Jurjen Bos, Jean-Christophe Deneuville, Arnaud Dion, Philippe Gaborit, Jérôme Lacan, Edoardo Persichetti, Jean-Marc Robert, Pascal Véron, Gilles Zémor, and Jurjen Bos. *HQC*. Round 4 Submission to the NIST Post-Quantum Cryptography Call. https://pqc-hqc.org/. Oct. 2022 (cit. on pp. viii, 2, 5, 22, 44, 46, 50, 54, 58, 95, 169, 227).

[AACD+22] Gorjan Alagic, Daniel Apon, David Cooper, Quynh Dang, Thinh Dang, John Kelsey, Jacob Lichtinger, Carl Miller, Dustin Moody, Rene Peralta, et al. *Status report on the third round of the NIST post-quantum cryptography standardization process*. https://nvlpubs.nist.gov/nistpubs/ir/2022/NIST.IR.8413.pdf. 2022 (cit. on p. x).

[ABCC+22] Martin Albrecht, Daniel J. Bernstein, Tung Chou, Carlos Cid, Jan Gilcher, Tanja Lange, Varun Maram, Ingo von Maurich, Rafael Mizoczki, Ruben Niederhagen, Edoardo Persichetti, Kenneth Paterson, Christiane Peters, Peter Schwabe, Nicolas Sendrier, Jakub Szefer, Cen Jung Tjhai, Martin Tomlinson, and Wang Wen. *Classic McEliece (merger of Classic McEliece and NTS-KEM)*. https://classic.mceliece.org. Fourth round finalist of the NIST post-quantum cryptography call. Nov. 2022 (cit. on pp. viii, 2, 3, 24, 26, 41).

[ABDG+16] Carlos Aguilar Melchor, Olivier Blazy, Jean-Christophe Deneuville, Philippe Gaborit, and Gilles Zémor. "Efficient Encryption from Random Quasi-Cyclic Codes". In: *CoRR* abs/1612.05572 (2016) (cit. on p. 44).

[ABDG+19] Nicolas Aragon, Olivier Blazy, Jean-Christophe Deneuville, Philippe Gaborit, Adrien Hauteville, Olivier Ruatta, Jean-Pierre Tillich, Gilles Zémor, Carlos Aguilar Melchor, Slim Bettaieb, Loïc Bidoux, Magali Bardet, and Ayoub Otmani. *ROLLO (merger of Rank-Ouroboros, LAKE and LOCKER)*. Second round submission to the NIST post-quantum cryptography call. NIST Round 2 submission for Post-Quantum Cryptography. Mar. 2019 (cit. on pp. 58, 59).

[ABDK+21] Roberto Avanzi, Joppe Bos, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, John M Schanck, Peter Schwabe, Gregor Seiler, and Damien Stehlé. *CRYSTALS-Kyber: algorithm specifications and supporting documentation*. Version 3.02. https://pq-crystals.org/kyber/data/kyber-specification-round3-20210804.pdf. Aug. 2021 (cit. on pp. viii, 2, 44).

[ACLM23] David Ayotte, Xavier Caruso, Antoine Leudière, and Joseph Musleh. "Drinfeld Modules in SageMath". In: *ACM Communications in Computer Algebra* (2023). https://arxiv.org/abs/2305.00422. arXiv: 2305.00422 (cit. on p. 219).

[AF03] Daniel Augot and Matthieu Finiasz. "A Public Key Encryption Scheme Based on the Polynomial Reconstruction Problem". In: *Advances in Cryptology - EUROCRYPT 2003*. Vol. 2656. LNCS. Springer, 2003, pp. 229–240 (cit. on p. 65).

[AFL03] Daniel Augot, Matthieu Finiasz, and Pierre Loidreau. "Using the Trace Operator to repair the Polynomial Reconstruction based Cryptosystem presented at Eurocrypt 2003". In: *IACR Cryptology ePrint Archive* 2003 (2003), p. 209 (cit. on p. 66).

[AG11] Sanjeev Arora and Rong Ge. "New Algorithms for Learning in Presence of Errors". English. In: *Automata, Languages and Programming*. Ed. by Luca Aceto, Monika Henzinger, and Jiří Sgall. Vol. 6755. LNCS. Springer Berlin Heidelberg, 2011, pp. 403–415. ISBN: 978-3-642-22005-0 (cit. on p. 211).

[AGHR+18] Nicolas Aragon, Philippe Gaborit, Adrien Hauteville, Olivier Ruatta, and Gilles Zémor. "Low Rank Parity Check Codes: New Decoding Algorithms and Application to Cryptography". Available on http://arxiv.org/abs/1111.4301. 2018 (cit. on p. 59).

[AGHT18] Nicolas Aragon, Philippe Gaborit, Adrien Hauteville, and Jean-Pierre Tillich. "A new algorithm for solving the rank syndrome decoding problem". In: *2018 IEEE International Symposium on Information Theory, ISIT 2018, Vail, CO, USA, June 17-22, 2018*. IEEE. 2018, pp. 2421–2425 (cit. on pp. 56, 57).

[AJAC+20] Gorjan Alagic, Alperin-Sheriff Jacob, Daniel Apon, David Cooper, Quynh Dang, John Kelsey, Yi-Kai Liu, Carl Miller, Dustin Moody, Rene Peralta, Ray Perlner, Angela Robinson, and Daniel Smith-Tone. *Status Report on the Second Round of the NIST Post-Quantum Cryptography Standardization Process*. Tech. rep. NISTIR 8309. NIST, July 2020 (cit. on p. 58).

[AJLT+12] Gilad Asharov, Abhishek Jain, Adriana López-Alt, Eran Tromer, Vinod Vaikuntanathan, and Daniel Wichs. "Multiparty Computation with Low Communication, Computation and Interaction via Threshold FHE". In: *Advances in Cryptology - EUROCRYPT 2012 - 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cambridge, UK, April 15-19, 2012. Proceedings*. Ed. by David Pointcheval and Thomas Johansson. Vol. 7237. Lecture Notes in Computer Science. Springer, 2012, pp. 483–501 (cit. on p. 174).

[AL87] Dana Angluin and Philip D. Laird. "Learning From Noisy Examples". In: *Mach. Learn.* 2.4 (1987), pp. 343–370 (cit. on p. 22).

[Ale03] Alekhnovich, Michael. "More on Average Case vs Approximation Complexity". In: *44th Symposium on Foundations of Computer Science (FOCS 2003), 11-14 October 2003, Cambridge, MA, USA, Proceedings*. IEEE Computer Society, 2003, pp. 298–307 (cit. on pp. ix, 4, 27, 32).

[Bar18] Élise Barelli. "On the security of short McEliece keys from algebraic and algebraic geometry codes with automorphisms". PhD thesis. École Polytechnique X ; Université Paris Saclay, 2018 (cit. on pp. 41, 47).

[BBBC+17] Gustavo Banegas, Paulo S.L.M Barreto, Brice Odilon Boidje, Pierre-Louis Cayrel, Gilbert Ndollane Dione, Kris Gaj, Cheikh Thiécoumba Gueye, Richard Haeussler, Jean Belo Klamti, Ousmane N'diaye, Duc Tri Nguyen, Edoardo Persichetti, and Jefferson E. Ricardini. *DAGS : Key Encapsulation for Dyadic GS Codes*. https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/round-1/submissions/DAGS.zip. First round submission to the NIST post-quantum cryptography call. Nov. 2017 (cit. on p. 41).

[BBBG+20] Magali Bardet, Pierre Briaud, Maxime Bros, Philippe Gaborit, Vincent Neiger, Olivier Ruatta, and Jean-Pierre Tillich. "An Algebraic Attack on Rank Metric Code-Based Cryptosystems". In: *Advances in Cryptology – EUROCRYPT 2020*. Ed. by Anne Canteaut and Yuval Ishai. Cham: Springer International Publishing, 2020, pp. 64–93 (cit. on p. 58).

[BBBG+22] Magali Bardet, Pierre Briaud, Maxime Bros, Philippe Gaborit, and Jean-Pierre Tillich. *Revisiting Algebraic Attacks on MinRank and on the Rank Decoding Problem*. ArXiv:2208.05471. 2022 (cit. on p. 58).

[BBCG+18] Slim Bettaieb, Loïc Bidoux, Yann Connan, Philippe Gaborit, and Adrien Hauteville. "The Learning with Rank Errors problem and an application to symmetric authentication". In: *2018 IEEE International Symposium on Information Theory, ISIT 2018, Vail, CO, USA, June 17-22, 2018*. 2018, pp. 2629–2633 (cit. on p. 153).

[BBCG+20] Magali Bardet, Maxime Bros, Daniel Cabarcas, Philippe Gaborit, Ray Perlner, Daniel Smith-Tone, Jean-Pierre Tillich, and Javier Verbel. "Improvements of Algebraic Attacks for solving the Rank Decoding and MinRank problems". In: *Advances in Cryptology - ASIACRYPT 2020, International Conference on the Theory and Application of Cryptology and Information Security, 2020. Proceedings*. 2020, pp. 507–536 (cit. on p. 58).

[BBCO19] Magali Bardet, Manon Bertin, Alain Couvreur, and Ayoub Otmani. "Practical Algebraic Attack on DAGS". In: *Code-Based Cryptography - 7th International Workshop, CBC 2019, Darmstadt, Germany, May 18-19, 2019, Revised Selected Papers*. Ed. by Marco Baldi, Edoardo Persichetti, and Paolo Santini. Vol. 11666. LNCS. Springer, 2019, pp. 86–101 (cit. on p. 41).

[BBHR18a] Eli Ben-Sasson, Iddo Bentov, Yinon Horesh, and Michael Riabzev. "Fast Reed-Solomon Interactive Oracle Proofs of Proximity". In: *45th International Colloquium on Automata, Languages, and Programming, ICALP 2018, July 9-13, 2018, Prague, Czech Republic*. Ed. by Ioannis Chatzigiannakis, Christos Kaklamanis, Dániel Marx, and Donald Sannella. Vol. 107. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018, 14:1–14:17 (cit. on p. 230).

[BBHR18b] Eli Ben-Sasson, Iddo Bentov, Yinon Horesh, and Michael Riabzev. "Scalable, transparent, and post-quantum secure computational integrity". In: *IACR Cryptol. ePrint Arch.* (2018), p. 46 (cit. on p. 230).

[BC18] Élise Barelli and Alain Couvreur. "An Efficient Structural Attack on NIST Submission DAGS". In: *Advances in Cryptology - ASIACRYPT'18*. Ed. by Thomas Peyrin and Steven Galbraith. Vol. 11272. LNCS. Springer, Dec. 2018, pp. 93–118 (cit. on pp. 41, 47).

[BC21] Maxime Bombar and Alain Couvreur. "Decoding Supercodes of Gabidulin Codes and Applications to Cryptanalysis". In: *Post-Quantum Cryptography - 12th International Conference*. Ed. by Jung Hee Cheon and Jean-Pierre Tillich. Vol. 12841. LNCS. Daejeon, South Korea: Springer, July 2021, pp. 3–22 (cit. on pp. xi, 7, 53, 65, 76, 78).

[BC22] Maxime Bombar and Alain Couvreur. "Right–hand side decoding of Gabidulin codes and applications". In: *WCC 2022 - Workshop on Coding Theory and Cryptography*. Rostock, Germany, Mar. 2022 (cit. on pp. xi, 7, 53, 65, 70).

[BC60] Raj Chandra Bose and Dwijendra K Chaudhuri. "On a class of error correcting binary group codes". In: *Information and control* 3.1 (1960), pp. 68–79 (cit. on p. 34).

[BCCD23] Maxime Bombar, Geoffroy Couteau, Alain Couvreur, and Clément Ducros. "Correlated Pseudorandomness from the Hardness of Quasi-Abelian Decoding". In: *Advances in Cryptology - CRYPTO 2023 - 43nd International Cryptology Conference*. Ed. by Helena Handschuh and Anna Lysyanskaya. Santa Barbara, CA, USA: Springer, Aug. 2023 (cit. on pp. xiv, 7, 8, 174, 188, 191, 192, 211, 212, 214).

[BCD22] Maxime Bombar, Alain Couvreur, and Thomas Debris-Alazard. "On Codes and Learning With Errors over Function Fields". In: *Advances in Cryptology - CRYPTO 2022 - 42nd International Cryptology Conference*. Ed. by Yevgeniy Dodis and Thomas Shrimpton. Vol. 13508. LNCS. Santa Barbara, CA, USA: Springer, Aug. 2022 (cit. on pp. xiii, 7, 93).

[BCD23] Maxime Bombar, Alain Couvreur, and Thomas Debris-Alazard. "Pseudorandomness of Decoding, Revisited: Adapting OHCP to Code-Based Cryptography". In: *Advances in Cryptology - ASIACRYPT 2023 29th International Conference on the Theory and Application of Cryptology and Information Security*. Ed. by Jian Guo and Ron Steinfeld. LNCS. Guangzhou, China: Springer, Dec. 2023 (cit. on pp. xiii, 6, 7, 133, 151).

[BCGI+19] Elette Boyle, Geoffroy Couteau, Niv Gilboa, Yuval Ishai, Lisa Kohl, and Peter Scholl. "Efficient Pseudorandom Correlation Generators: Silent OT Extension and More". In: *Advances in Cryptology - CRYPTO 2019 - 39th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2019, Proceedings, Part III*. Ed. by Alexandra Boldyreva and Daniele Micciancio. Vol. 11694. Lecture Notes in Computer Science. Springer, 2019, pp. 489–518 (cit. on pp. xi, 6, 179, 185–187, 192, 193).

[BCGI+20a] Elette Boyle, Geoffroy Couteau, Niv Gilboa, Yuval Ishai, Lisa Kohl, and Peter Scholl. "Correlated Pseudorandom Functions from Variable-Density LPN". In: *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020, Durham, NC, USA, November 16-19, 2020*. Ed. by Sandy Irani. IEEE, 2020, pp. 1069–1080 (cit. on pp. 48, 49).

[BCGI+20b] Elette Boyle, Geoffroy Couteau, Niv Gilboa, Yuval Ishai, Lisa Kohl, and Peter Scholl. "Efficient Pseudorandom Correlation Generators from Ring-LPN". In: *Advances in Cryptology - CRYPTO*. Ed. by Daniele Micciancio and Thomas Ristenpart. Cham: Springer International Publishing, 2020, pp. 387–416 (cit. on pp. xi, xiii, 6, 8, 40, 124, 180, 187–193, 201, 204, 210, 215, 220, 228).

[BCGI+22] Elette Boyle, Geoffroy Couteau, Niv Gilboa, Yuval Ishai, Lisa Kohl, Nicolas Resch, and Peter Scholl. "Correlated Pseudorandomness from Expand-Accumulate Codes". In: *Advances in Cryptology - CRYPTO 2022 - 42nd Annual International Cryptology Conference, CRYPTO 2022, Santa Barbara, CA, USA, August 15-18, 2022, Proceedings, Part II*. Ed. by Yevgeniy Dodis and Thomas Shrimpton. Vol. 13508. Lecture Notes in Computer Science. Springer, 2022, pp. 603–633 (cit. on pp. 192, 193).

[BCGI18] Elette Boyle, Geoffroy Couteau, Niv Gilboa, and Yuval Ishai. "Compressing Vector OLE". In: *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, CCS 2018, Toronto, ON, Canada, October 15-19, 2018*. Ed. by David Lie, Mohammad Mannan, Michael Backes, and XiaoFeng Wang. ACM, 2018, pp. 896–912 (cit. on pp. xi, 6, 179, 185).

[BCGI22] Elette Boyle, Geoffroy Couteau, Niv Gilboa, and Yuval Ishai. "Function Secret Sharing and Homomorphic Secret Sharing". https://geoffroycouteau.github.io/assets/pdf/HSS_FSS.pdf. 2022 (cit. on pp. 180, 184).

[BCP97] Wieb Bosma, John Cannon, and Catherine Playoust. "The Magma algebra system. I. The user language". In: *J. Symbolic Comput.* 24.3-4 (1997). Computational algebra and number theory (London, 1993), pp. 235–265. ISSN: 0747-7171 (cit. on p. 219).

[BCV20] Carl Bootland, Wouter Castryck, and Frederik Vercauteren. "On the Security of the Multivariate Ring Learning With Errors Problem". eng. In: *ANTS-XIV, Fourteenth Algorithmic Number Theory Symposium, Proceedings*. Vol. 4. Open Book Series 1. Auckland, New Zealand: Mathematical Sciences Publishers, 2020, pp. 57–71 (cit. on pp. 206, 214).

[BDEF+22] Daniel J Bernstein, Christoph Dobraunig, Maria Eichlseder, Scott Fluhrer, Stefan-Lukas Gazdag, Andreas Hülsing, Panos Kampanakis, Stefan Kölbl, Tanja Lange, Martin M Lauridsen, et al. *SPHINCS+*. Version 3.1. https://sphincs.org/data/sphincs+-r3.1-specification.pdf. June 2022 (cit. on p. 2).

[BDKP+11] Boaz Barak, Yevgeniy Dodis, Hugo Krawczyk, Olivier Pereira, Krzysztof Pietrzak, François-Xavier Standaert, and Yu Yu. "Leftover Hash Lemma, Revisited". In: *Advances in Cryptology - CRYPTO 2011 - 31st Annual Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2011. Proceedings*. 2011, pp. 1–20 (cit. on p. 154).

[BDPW20] Koen de Boer, Léo Ducas, Alice Pellet-Mary, and Benjamin Wesolowski. "Random Self-reducibility of Ideal-SVP via Arakelov Random Walks". In: *Advances in Cryptology - CRYPTO 2020 - 40th Annual International Cryptology Conference*. Ed. by Daniele Micciancio and Thomas Ristenpart. Vol. 12171. Lecture Notes in Computer Science. Springer, 2020, pp. 243–273 (cit. on p. 170).

[Bea91] Donald Beaver. "Efficient Multiparty Protocols Using Circuit Randomization". In: *Advances in Cryptology - CRYPTO '91, 11th Annual International Cryptology Conference, Santa Barbara, California, USA, August 11-15, 1991, Proceedings*. Ed. by Joan Feigenbaum. Vol. 576. Lecture Notes in Computer Science. Springer, 1991, pp. 420–432 (cit. on pp. x, 174, 176).

[Bea95] Donald Beaver. "Precomputing Oblivious Transfer". In: *Advances in Cryptology - CRYPTO '95, 15th Annual International Cryptology Conference, Santa Barbara, California, USA, August 27-31, 1995, Proceedings*. Ed. by Don Coppersmith. Vol. 963. Lecture Notes in Computer Science. https://link.springer.com/content/pdf/10.1007/3-540-44750-4_8.pdf. Springer, 1995, pp. 97–109 (cit. on p. x).

[Ber03] Thierry P. Berger. "Isometries for rank distance and permutation group of Gabidulin codes". In: *IEEE Trans. Inform. Theory* 49.11 (2003), pp. 3016–3019 (cit. on p. 70).

[Ber10] Daniel J. Bernstein. "Grover vs. McEliece". In: *Post-Quantum Cryptography 2010*. Ed. by Nicolas Sendrier. Vol. 6061. LNCS. Springer, 2010, pp. 73–80 (cit. on p. 21).

[Beu22] Ward Beullens. "Breaking Rainbow Takes a Weekend on a Laptop". In: *Advances in Cryptology - CRYPTO 2022 - 42nd Annual International Cryptology Conference, CRYPTO 2022, Santa Barbara, CA, USA, August 15-18, 2022, Proceedings, Part II*. Ed. by Yevgeniy Dodis and Thomas Shrimpton. Vol. 13508. Lecture Notes in Computer Science. Springer, 2022, pp. 464–479 (cit. on p. 2).

[BF02] Alexander Barg and G. David Forney. "Random codes: Minimum distances and error exponents". In: *IEEE Trans. Inf. Theory* 48.9 (2002), pp. 2568–2573 (cit. on pp. 18, 165).

[BFS99] Jonathan F. Buss, Gudmund S. Frandsen, and Jeffrey O. Shallit. "The Computational Complexity of Some Problems of Linear Algebra". In: *J. Comput. System Sci.* 58.3 (June 1999), pp. 572–596 (cit. on p. 57).

[BGI16] Elette Boyle, Niv Gilboa, and Yuval Ishai. "Function Secret Sharing: Improvements and Extensions". In: *ACM CCS 2016: 23rd Conference on Computer and Communications Security*. Ed. by Edgar R. Weippl, Stefan Katzenbeisser, Christopher Kruegel, Andrew C. Myers, and Shai Halevi. Vienna, Austria: ACM Press, Oct. 2016, pp. 1292–1303 (cit. on pp. 180, 181, 184).

[BGP22] Katharina Boudgoust, Erell Gachon, and Alice Pellet-Mary. "Some Easy Instances of Ideal-SVP and Implications on the Partial Vandermonde Knapsack Problem". In: *Advances in Cryptology - CRYPTO 2022 - 42nd Annual International Cryptology Conference, CRYPTO 2022, Santa Barbara, CA, USA, August 15-18, 2022, Proceedings, Part II*. Ed. by Yevgeniy Dodis and Thomas Shrimpton. Vol. 13508. Lecture Notes in Computer Science. Springer, 2022, pp. 480–509 (cit. on p. 214).

[BJMM12] Anja Becker, Antoine Joux, Alexander May, and Alexander Meurer. "Decoding Random Binary Linear Codes in $2^{n/20}$: How $1+1=0$ Improves Information Set Decoding". In: *Advances in Cryptology - EUROCRYPT 2012*. LNCS. Springer, 2012 (cit. on pp. 20, 49).

[BJRW20a] Katharina Boudgoust, Corentin Jeudy, Adeline Roux-Langlois, and Weiqiang Wen. "Towards Classical Hardness of Module-LWE: The Linear Rank Case". In: *Advances in Cryptology - ASIACRYPT 2020*. Ed. by Shiho Moriai and Huaxiong Wang. Springer, Dec. 2020 (cit. on p. xii).

[BJRW20b] Katharina Boudgoust, Corentin Jeudy, Adeline Roux-Langlois, and Weiqiang Wen. "Towards Classical Hardness of Module-LWE: The Linear Rank Case". In: *Advances in Cryptology - ASIACRYPT 2020 - 26th International Conference on the Theory and Application of Cryptology and Information Security, Daejeon, South Korea, December 7-11, 2020, Proceedings, Part II*. Ed. by Shiho Moriai and Huaxiong Wang. Vol. 12492. Lecture Notes in Computer Science. Springer, 2020, pp. 289–317 (cit. on p. 134).

[BKW03] Avrim Blum, Adam Kalai, and Hal Wasserman. "Noise-tolerant learning, the parity problem, and the statistical query model". In: *Journal of the ACM (JACM)* 50.4 (2003), pp. 506–519 (cit. on p. 211).

[BL12] Daniel J. Bernstein and Tanja Lange. "Never Trust a Bunny". In: *Radio Frequency Identification. Security and Privacy Issues - 8th International Workshop, RFIDSec 2012, Nijmegen, The Netherlands, July 2-3, 2012, Revised Selected Papers*. Ed. by Jaap-Henk Hoepman and Ingrid Verbauwhede. Vol. 7739. Lecture Notes in Computer Science. Springer, 2012, pp. 137–148 (cit. on pp. 124, 205).

[BLVW19] Zvika Brakerski, Vadim Lyubashevsky, Vinod Vaikuntanathan, and Daniel Wichs. "Worst-Case Hardness for LPN and Cryptographic Hashing via Code Smoothing". In: *Advances in Cryptology - EUROCRYPT 2019 - 38th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Darmstadt, Germany, May 19-23, 2019, Proceedings, Part III*. Ed. by Yuval Ishai and Vincent Rijmen. Vol. 11478. LNCS. Springer, 2019, pp. 619–635 (cit. on pp. 139, 154, 163, 165).

[BM06] L. M. J. Bazzi and Sanjoy K. Mitter. "Some Randomized Code Constructions From Group Actions". In: *IEEE Trans. Inf. Theory* 52.7 (2006), pp. 3210–3219 (cit. on p. 207).

[BM17] Leif Both and Alexander May. "Optimizing BJMM with Nearest Neighbors: Full Decoding in $2^{2/21n}$ and McEliece Security". In: *WCC Workshop on Coding and Cryptography*. Sept. 2017 (cit. on pp. 20, 49).

[BMT23] Magali Bardet, Rocco Mora, and Jean-Pierre Tillich. "Polynomial time key-recovery attack on high rate random alternant codes". In: *CoRR* abs/2304.14757 (2023). arXiv: 2304.14757 (cit. on pp. 4, 41).

[BMT78] Elwyn Berlekamp, Robert McEliece, and Henk van Tilborg. "On the inherent intractability of certain coding problems". In: *IEEE Trans. Inform. Theory* 24.3 (May 1978), pp. 384–386 (cit. on pp. ix, 3, 14).

[BØ23] Pierre Briaud and Morten Øygarden. *A New Algebraic Approach to the Regular Syndrome Decoding Problem and Implications for PCG Constructions*. Cryptology ePrint Archive, Paper 2023/176. https://eprint.iacr.org/2023/176. 2023 (cit. on p. 211).

[BP23] Stéphane Ballet and Bastien Pacifico. “Chudnovsky-Type Algorithms over the Projective Line Using Generalized Evaluation Maps”. In: *Codes, Cryptology and Information Security - 4th International Conference, C2SI 2023, Rabat, Morocco, May 29-31, 2023, Proceedings*. Ed. by Said El Hajji, Sihem Mesnager, and El Mamoun Souidi. Vol. 13874. Lecture Notes in Computer Science. Springer, 2023, pp. 360–375 (cit. on p. 221).

[BPRR+21] Stéphane Ballet, Julia Pieltant, Matthieu Rambaud, Hugues Randriambololona, Robert Rolland, and Jean Chaumine. “On the Tensor Rank of Multiplication in Finite Extensions of Finite Fields and Related Issues in Algebraic Geometry”. In: *Russian Mathematical Surveys* 76.1 (2021), p. 29 (cit. on p. 221).

[BSS22] Katharina Boudgoust, Amin Sakzad, and Ron Steinfeld. “Vandermonde meets Regev: public key encryption schemes based on partial Vandermonde problems”. In: *Des. Codes Cryptogr.* 90.8 (2022), pp. 1899–1936 (cit. on p. 214).

[BT06] Andrej Bogdanov and Luca Trevisan. “On Worst-Case to Average-Case Reductions for NP Problems”. In: *SIAM J. Comput.* 36.4 (2006), pp. 1119–1159 (cit. on p. 15).

[Can89] David G. Cantor. “On Arithmetical Algorithms over Finite Fields”. In: *J. Comb. Theory, Ser. A* 50.2 (1989), pp. 285–300 (cit. on p. 225).

[CBBB+17] Alain Couvreur, Magali Bardet, Élise Barelli, Olivier Blazy, Rodolfo Canto Torres, Phillipe Gaborit, Ayoub Otmani, Nicolas Sendrier, and Jean-Pierre Tillich. *BIG QUAKE*. https://bigquake.inria.fr. NIST Round 1 submission for Post-Quantum Cryptography. Nov. 2017 (cit. on p. 41).

[CC19] Daniel Coggia and Alain Couvreur. “On the security of a Loidreau’s rank metric code based encryption scheme”. In: *WCC 2019 - Workshop on Coding Theory and Cryptography*. Saint Jacut de la mer, France, 2019 (cit. on p. 85).

[CC20] Daniel Coggia and Alain Couvreur. “On the security of a Loidreau’s rank metric code based encryption scheme”. In: *Des. Codes Cryptogr.* 88 (2020), pp. 1941–1957 (cit. on p. 63).

[CCCX09] Ignacio Cascudo, Hao Chen, Ronald Cramer, and Chaoping Xing. “Asymptotically Good Ideal Linear Secret Sharing with Strong Multiplication over *Any* Fixed Finite Field”. In: *Advances in Cryptology - CRYPTO 2009, 29th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2009. Proceedings*. Ed. by Shai Halevi. Vol. 5677. Lecture Notes in Computer Science. https://www.iacr.org/archive/crypto2009/56770460/56770460.pdf. Springer, 2009, pp. 466–486 (cit. on p. 221).

[CCXY18] Ignacio Cascudo, Ronald Cramer, Chaoping Xing, and Chen Yuan. “Amortized Complexity of Information-Theoretically Secure MPC Revisited”. In: *Advances in Cryptology - CRYPTO 2018 - 38th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2018, Proceedings, Part III*. Ed. by Hovav Shacham and Alexandra Boldyreva. Vol. 10993. Lecture Notes in Computer Science. https://eprint.iacr.org/2018/429.pdf. Springer, 2018, pp. 395–426 (cit. on p. 221).

[CD23] Wouter Castryck and Thomas Decru. "An Efficient Key Recovery Attack on SIDH". In: *Advances in Cryptology - EUROCRYPT 2023 - 42nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Lyon, France, April 23-27, 2023, Proceedings, Part V*. Ed. by Carmit Hazay and Martijn Stam. Vol. 14008. Lecture Notes in Computer Science. Springer, 2023, pp. 423–447 (cit. on p. 2).

[CDGK+20] Melissa Chase, David Derler, Steven Goldfeder, Daniel Kales, Jonathan Katz, Vladimir Koleshnikov, Claudio Orlandi, Sebastian Ramacher, Christian Rechberger, Daniel Slamanig, Xiao Wang, and Zaverucha Greg. *Picnic: A Family of Post-Quantum Secure Digital Signature Algorithms*. Third round submission to the NIST post-quantum cryptography call. https://microsoft.github.io/Picnic/. Apr. 2020 (cit. on p. 174).

[CDMT22] Kevin Carrier, Thomas Debris-Alazard, Charles Meyer-Hilfiger, and Jean-Pierre Tillich. "Statistical Decoding 2.0: Reducing Decoding to LPN". In: *Advances in Cryptology - ASIACRYPT 2022*. LNCS. Springer, 2022 (cit. on pp. 20, 49).

[CDN01] Ronald Cramer, Ivan Damgård, and Jesper Buus Nielsen. "Multiparty Computation from Threshold Homomorphic Encryption". In: *Advances in Cryptology - EUROCRYPT 2001, International Conference on the Theory and Application of Cryptographic Techniques, Innsbruck, Austria, May 6-10, 2001, Proceeding*. Ed. by Birgit Pfitzmann. Vol. 2045. Lecture Notes in Computer Science. Springer, 2001, pp. 280–299 (cit. on p. 174).

[CDN15] Ronald Cramer, Ivan Damgård, and Jesper Buus Nielsen. *Secure Multiparty Computation and Secret Sharing*. Cambridge University Press, 2015. ISBN: 9781107043053 (cit. on pp. 174, 176).

[CGGO+13] Alain Couvreur, Philippe Gaborit, Valérie Gautier, Ayoub Otmani, and Jean-Pierre Tillich. "Distinguisher-Based Attacks on Public-Key Cryptosystems Using Reed-Solomon Codes". In: *International Workshop on Coding and Cryptography - WCC 2013*. Bergen, Norway, Apr. 2013, pp. 181–193 (cit. on p. 4).

[Cha17] Julia Chaulet. "Étude de cryptosystèmes à clé publique basés sur les codes MDPC quasi-cycliques". PhD thesis. University Pierre et Marie Curie, Mar. 2017 (cit. on p. 21).

[Cha96] Robin J. Chapman. "A simple proof of Noether's Theorem". In: *Glasgow Math. J.* 38 (1996), pp. 49–51 (cit. on p. 127).

[CIV16] Wouter Castryck, Ilia Iliashenko, and Frederik Vercauteren. "Provably Weak Instances of Ring-LWE Revisited". In: *Advances in Cryptology - EUROCRYPT 2016 - 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Vienna, Austria, May 8-12, 2016, Proceedings, Part I*. Ed. by Marc Fischlin and Jean-Sébastien Coron. Vol. 9665. Lecture Notes in Computer Science. Springer, 2016, pp. 147–167 (cit. on p. 190).

[CL17] Sunil K Chebolu and Keir Lockridge. "Fuchs' Problem for Dihedral Groups". In: *Journal of Pure and Applied Algebra* 221.4 (2017), pp. 971–982 (cit. on p. 217).

[CMT23] Alain Couvreur, Rocco Mora, and Jean-Pierre Tillich. "A new approach based on quadratic forms to attack the McEliece cryptosystem". In: *arXiv preprint arXiv:2306.10294* (2023) (cit. on pp. 4, 41).

[Con] Keith Conrad. "Carlitz extensions". https://kconrad.math.uconn.edu/blurbs/gradnumthy/carlitz.pdf (cit. on p. 105).

[Cor04] Jean-Sébastien Coron. "Cryptanalysis of a Public-Key Encryption Scheme Based on the Polynomial Reconstruction Problem". In: *Public Key Cryptography - PKC 2004, 7th International Workshop on Theory and Practice in Public Key Cryptography, Singapore, March 1-4, 2004*. 2004, pp. 14–27 (cit. on p. 66).

[Cou01] Nicolas Courtois. "Efficient zero-knowledge authentication based on a linear algebra problem MinRank". In: *Advances in Cryptology - ASIACRYPT 2001*. Vol. 2248. LNCS. Gold Coast, Australia: Springer, 2001, pp. 402–421 (cit. on p. 57).

[Cou19] Alain Couvreur. "Codes algébriques et géométriques, applications à la cryptographie et à l'information quantique". Accreditation to supervise research. Université Paris Diderot, Dec. 2019 (cit. on p. 24).

[Cou21] Alain Couvreur. "How arithmetic and geometry make error correcting codes better". In: *Société Mathématique de France, Panoramas & Synthèses (to appear)* (2021). https://arxiv.org/abs/2110.11282 (cit. on p. 211).

[Cox21] Nicholas Coxon. "Fast transforms over finite fields of characteristic two". In: *J. Symb. Comput.* 104 (2021), pp. 824–854 (cit. on p. 225).

[CPJ69] C. L. Chen, W. W. Peterson, and E. J. Weldon Jr. "Some Results on Quasi-Cyclic Codes". In: *Information and Control* 15.5 (1969), pp. 407–423 (cit. on pp. 50, 207).

[CRR21] Geoffroy Couteau, Peter Rindal, and Srinivasan Raghuraman. "Silver: Silent VOLE and Oblivious Transfer from Hardness of Decoding Structured LDPC Codes". In: *Advances in Cryptology - CRYPTO 2021 - 41st Annual International Cryptology Conference, CRYPTO 2021, Virtual Event, August 16-20, 2021, Proceedings, Part III*. Ed. by Tal Malkin and Chris Peikert. Vol. 12827. Lecture Notes in Computer Science. Springer, 2021, pp. 502–534 (cit. on pp. 48–50, 192, 193).

[CS16] Rodolfo Canto-Torres and Nicolas Sendrier. "Analysis of Information Set Decoding for a Sub-linear Error Weight". In: *Post-Quantum Cryptography 2016*. LNCS. Fukuoka, Japan, Feb. 2016, pp. 144–161 (cit. on pp. 21, 161, 211).

[CT19] Rodolfo Canto-Torres and Jean-Pierre Tillich. "Speeding up decoding a code with a non-trivial automorphism group up to an exponential factor". In: *Proc. IEEE Int. Symposium Inf. Theory - ISIT 2019*. 2019, pp. 1927–1931 (cit. on pp. 47, 48, 212, 213).

[CT65] James W. Cooley and John W. Tukey. "An Algorithm for the Machine Calculation of Complex Fourier Series". In: *Mathematics of Computation* 19 (1965). https://web.stanford.edu/class/cme324/classics/cooley-tukey.pdf, pp. 297–301. ISSN: 0025–5718 (cit. on p. 200).

[CZ23] Alain Couvreur and Ilaria Zappatore. "An extension of Overbeck's attack with an application to cryptanalysis of Twisted Gabidulin-based schemes". In: *Post-Quantum Cryptography - 14th International Workshop, PQCrypto 2023, College Park, MD, USA, August 16-18, 2023, Proceedings*. Ed. by Thomas Johansson and Daniel Smith-Tone. Lecture Notes in Computer Science. Springer, 2023 (cit. on pp. 62, 63).

[DDRT23] Thomas Debris-Alazard, Léo Ducas, Nicolas Resch, and Jean-Pierre Tillich. "Smoothing Codes and Lattices: Systematic Study and New Bounds". In: *IEEE Trans. Inform. Theory* 69.9 (2023), pp. 6006–6027 (cit. on pp. 50, 139, 154).

[Deb23] Thomas Debris-Alazard. *Code-based Cryptography: Lecture Notes*. https://arxiv.org/abs/2304.03541. 2023. arXiv: 2304.03541 (cit. on pp. 17–21, 31, 50, 136).

[Del78] Philippe Delsarte. "Bilinear Forms over a Finite Field, with Applications to Coding Theory". In: *J. Comb. Theory, Ser. A* 25.3 (1978), pp. 226–241 (cit. on p. 59).

[DH76] Whitfield Diffie and Martin Hellman. "New directions in cryptography". In: *IEEE transactions on Information Theory* 22.6 (1976), pp. 644–654 (cit. on pp. vii, 1, 23).

[DK12] Yurj A Drozd and Vladimir V Kirichenko. *Finite dimensional algebras*. Springer Science & Business Media, 2012 (cit. on pp. 121, 193).

[DK22] Samed Düzlü and Juliane Krämer. "Application of Automorphic Forms to Lattice Problems". In: *J. Math. Cryptol.* 16.1 (2022), pp. 156–197 (cit. on p. 170).

[Dor71] Larry L Dornhoff. *Group representation theory: Part A - Ordinary representation theory*. Vol. 7. M. Dekker, 1971 (cit. on p. 193).

[DP12] Ivan Damgård and Sunoo Park. "Is Public-Key Encryption Based on LPN Practical?" In: *IACR Cryptol. ePrint Arch.* (2012), p. 699 (cit. on pp. 38, 124, 129).

[DPSZ12] Ivan Damgård, Valerio Pastro, Nigel P. Smart, and Sarah Zakarias. "Multiparty Computation from Somewhat Homomorphic Encryption". In: *Advances in Cryptology - CRYPTO 2012 - 32nd Annual Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2012. Proceedings*. Ed. by Reihaneh Safavi-Naini and Ran Canetti. Vol. 7417. Lecture Notes in Computer Science. Springer, 2012, pp. 643–662 (cit. on pp. 174, 177, 179, 185).

[DR22] Thomas Debris-Alazard and Nicolas Resch. *Worst and Average case hardness of Decoding via Smoothing Bounds*. preprint. eprint. Dec. 2022 (cit. on pp. 23, 139, 154, 158, 162, 163).

[DST19a] Thomas Debris-Alazard, Nicolas Sendrier, and Jean-Pierre Tillich. *Wave: A New Family of Trapdoor One-Way Preimage Sampleable Functions Based on Codes*. Cryptology ePrint Archive, Report 2018/996. https://eprint.iacr.org/2018/996. May 2019 (cit. on p. 22).

[DST19b] Thomas Debris-Alazard, Nicolas Sendrier, and Jean-Pierre Tillich. "Wave: A New Family of Trapdoor One-Way Preimage Sampleable Functions Based on Codes". In: *Advances in Cryptology - ASIACRYPT 2019, Part I*. Ed. by Steven D. Galbraith and Shiho Moriai. Vol. 11921. LNCS. Kobe, Japan: Springer, Dec. 2019, pp. 21–51 (cit. on pp. 139, 154).

[DT17] Thomas Debris-Alazard and Jean-Pierre Tillich. "Statistical decoding". In: *Proc. IEEE Int. Symposium Inf. Theory - ISIT 2017*. Aachen, Germany, June 2017, pp. 1798–1802 (cit. on pp. 20, 49).

[Dum91] Ilya Dumer. "On minimum distance decoding of linear codes". In: *Proc. 5th Joint Soviet-Swedish Int. Workshop Inform. Theory*. Moscow, 1991, pp. 50–52 (cit. on pp. 20, 49).

[EKR18] David Evans, Vladimir Kolesnikov, and Mike Rosulek. "A Pragmatic Introduction to Secure Multi-Party Computation". In: *Found. Trends Priv. Secur.* 2.2-3 (2018), pp. 70–246 (cit. on pp. 174, 176, 177).

[Fau09] Cédric Faure. "Etudes de systèmes cryptographiques construits à l'aide de codes correcteurs, en métrique de Hamming et en métrique rang." PhD thesis. Ecole Polytechnique X, 2009 (cit. on p. 67).

[FHKL+20] Pierre-Alain Fouque, Jeffrey Hoffstein, Paul Kirchner, Vadim Lyubashevsky, Thomas Pornin, Thomas Prest, Thomas Ricosset, Gregor Seiler, William Whyte, and Zhenfei Zhang. *Falcon: Fast-Fourier lattice-based compact signatures over NTRU*. Version 1.2. https://falcon-sign.info/falcon.pdf. Oct. 2020 (cit. on pp. viii, 2).

[FJR22] Thibauld Feneuil, Antoine Joux, and Matthieu Rivain. "Syndrome Decoding in the Head: Shorter Signatures from Zero-Knowledge Proofs". In: *IACR Cryptol. ePrint Arch.* (2022), p. 188 (cit. on p. 174).

[FL15] Yun Fan and Liren Lin. "Thresholds of Random Quasi-Abelian Codes". In: *IEEE Transactions on Information Theory* 61.1 (2015), pp. 82–90 (cit. on p. 208).

[FL22] Yun Fan and Liren Lin. "Asymptotic Properties of Quasi-Group Codes". In: *CoRR* abs/2203.00958 (2022). https://arxiv.org/abs/2203.00958. arXiv: 2203.00958 (cit. on pp. 207, 208).

[FLP08] Jean-Charles Faugère, Françoise Levy-dit-Vehel, and Ludovic Perret. "Cryptanalysis of Minrank". In: *Advances in Cryptology - CRYPTO 2008*. Ed. by David Wagner. Vol. 5157. LNCS. 2008, pp. 280–296 (cit. on p. 58).

[FOPP+16a] Jean-Charles Faugère, Ayoub Otmani, Ludovic Perret, Frédéric de Portzamparc, and Jean-Pierre Tillich. "Folding Alternant and Goppa Codes With Non-Trivial Automorphism Groups". In: *IEEE Trans. Inform. Theory* 62.1 (2016), pp. 184–198 (cit. on pp. 47, 212).

[FOPP+16b] Jean-Charles Faugère, Ayoub Otmani, Ludovic Perret, Frédéric de Portzamparc, and Jean-Pierre Tillich. "Structural Cryptanalysis of McEliece Schemes with Compact Keys". In: *Des. Codes Cryptogr.* 79.1 (2016), pp. 87–112 (cit. on p. 47).

[FPS22] Joël Felderhoff, Alice Pellet-Mary, and Damien Stehlé. "On Module Unique-SVP and NTRU". In: *Advances in Cryptology - ASIACRYPT 2022 - 28th International Conference on the Theory and Application of Cryptology and Information Security, Taipei, Taiwan, December 5-9, 2022, Proceedings, Part III*. Ed. by Shweta Agrawal and Dongdai Lin. Vol. 13793. Lecture Notes in Computer Science. Springer, 2022, pp. 709–740 (cit. on p. 43).

[FS96] Jean-Bernard Fischer and Jacques Stern. "An efficient pseudo-random generator provably as secure as syndrome decoding". In: *Advances in Cryptology - EUROCRYPT'96*. Ed. by Ueli Maurer. Vol. 1070. LNCS. Springer, 1996, pp. 245–255. ISBN: ISBN 978-3-540-61186-8 (cit. on pp. 5, 25, 27, 30, 31, 44, 48, 135, 154).

[Gab05] Philippe Gaborit. "Shorter keys for code based cryptography". In: *Proceedings of the 2005 International Workshop on Coding and Cryptography (WCC 2005)*. Bergen, Norway, Mar. 2005, pp. 81–91 (cit. on pp. 4, 26, 34, 40).

[Gab85] Ernst M. Gabidulin. "Theory of codes with maximum rank distance". In: *Problemy Peredachi Informatsii* 21.1 (1985), pp. 3–16 (cit. on p. 59).

[Gal63] Robert G. Gallager. *Low Density Parity Check Codes*. Cambridge, Massachusetts: M.I.T. Press, 1963 (cit. on pp. ix, 3, 13, 26).

[Gen09] Craig Gentry. "Fully Homomorphic Encryption using ideal lattices". In: *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009, Bethesda, MD, USA, May 31 - June 2, 2009*. Ed. by Michael Mitzenmacher. https://www.cs.cmu.edu/~odonnell/hits09/gentry-homomorphic-encryption.pdf. ACM, 2009, pp. 169–178 (cit. on pp. 5, 174).

[GG96] Joachim von zur Gathen and Jürgen Gerhard. "Arithmetic and Factorization of Polynomial Over F_2 (extended abstract)". In: *Proceedings of the 1996 International Symposium on Symbolic and Algebraic Computation, ISSAC '96, Zurich, Switzerland, July 24-26, 1996*. Ed. by Erwin Engeler, B. F. Caviness, and Yagati N. Lakshman. ACM, 1996, pp. 1–9 (cit. on p. 225).

[GGM84] Oded Goldreich, Shafi Goldwasser, and Silvio Micali. "How to Construct Random Functions (Extended Abstract)". In: *25th Annual Symposium on Foundations of Computer Science, West Palm Beach, Florida, USA, 24-26 October 1984*. IEEE Computer Society, 1984, pp. 464–479 (cit. on p. 181).

[GHPT16] Philippe Gaborit, Adrien Hauteville, Duong Hieu Phan, and Jean-Pierre Tillich. *Identity-based Encryption from Rank Metric*. IACR Cryptology ePrint Archive, Report2017/623. http://eprint.iacr.org/. May 2016 (cit. on pp. 58, 59).

[GI14] Niv Gilboa and Yuval Ishai. "Distributed Point Functions and Their Applications". In: *Advances in Cryptology - EUROCRYPT 2014 - 33rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Copenhagen, Denmark, May 11-15, 2014. Proceedings*. Ed. by Phong Q. Nguyen and Elisabeth Oswald. Vol. 8441. Lecture Notes in Computer Science. Springer, 2014, pp. 640–658 (cit. on p. 180).

[GL05] Ernst M. Gabidulin and Pierre Loidreau. "On subcodes of codes in rank metric". In: *Proc. IEEE Int. Symposium Inf. Theory - ISIT*. IEEE. 2005, pp. 121–123 (cit. on p. 63).

[GL08] Ernst M. Gabidulin and Pierre Loidreau. "Properties of subspace subcodes of Gabidulin codes". In: *Adv. Math. Commun.* 2.2 (2008), pp. 147–157 (cit. on p. 63).

[GL89] Oded Goldreich and Leonid A. Levin. "A Hard-Core Predicate for all One-Way Functions". In: *Proceedings of the 21st Annual ACM Symposium on Theory of Computing, May 14-17, 1989, Seattle, Washington, USA*. Ed. by David S. Johnson. ACM, 1989, pp. 25–32 (cit. on p. 31).

[GM10] Shuhong Gao and Todd D. Mateer. "Additive Fast Fourier Transforms Over Finite Fields". In: *IEEE Trans. Inf. Theory* 56.12 (2010), pp. 6265–6272 (cit. on p. 225).

[GMRZ13] Philippe Gaborit, Gaétan Murat, Olivier Ruatta, and Gilles Zémor. "Low Rank Parity Check codes and their application to cryptography". In: *Proceedings of the Workshop on Coding and Cryptography WCC'2013*. Bergen, Norway, 2013 (cit. on p. 59).

[GMW87] Oded Goldreich, Silvio Micali, and Avi Wigderson. "How to Play any Mental Game or A Completeness Theorem for Protocols with Honest Majority". In: *Proceedings of the 19th Annual ACM Symposium on Theory of Computing, 1987, New York, New York, USA*. Ed. by Alfred V. Aho. ACM, 1987, pp. 218–229 (cit. on p. 174).

[Gop70] Valerii D. Goppa. “A new class of linear error-correcting codes”. In: *Problemy Peredachi Informatsii* 6.3 (1970). In Russian, pp. 24–30 (cit. on pp. ix, 3).

[Gop81] Valerii D. Goppa. “Codes on algebraic curves”. In: *Dokl. Akad. Nauk SSSR* 259.6 (1981). In Russian, pp. 1289–1290 (cit. on pp. 13, 26).

[GOT18] Philippe Gaborit, Ayoub Otmani, and Hervé Talé-Kalachi. “Polynomial-time key recovery attack on the Faure-Loidreau scheme based on Gabidulin codes”. In: *Des. Codes Cryptogr.* 86.7 (2018), pp. 1391–1403 (cit. on pp. 67, 69, 75, 76, 79).

[GPT91] Ernst M. Gabidulin, A. V. Paramonov, and O. V. Tretjakov. “Ideals over a non-commutative ring and their applications to cryptography”. In: *Advances in Cryptology - EUROCRYPT’91*. LNCS 547. Brighton, Apr. 1991, pp. 482–489 (cit. on pp. 59, 62).

[GRS00] Oded Goldreich, Ronitt Rubinfeld, and Madhu Sudan. “Learning Polynomials with Queries: The Highly Noisy Case”. In: *SIAM J. Discret. Math.* 13.4 (2000). (Long version of a paper appeared in *36th Annual Symposium on Foundations of Computer Science*, pages 294-303, Milwaukee, Wisconsin, 23-25 October 1995. IEEE.) - https://www.wisdom.weizmann.ac.il/~oded/PSX/grs.pdf, pp. 535–570 (cit. on p. 31).

[GRS13] Philippe Gaborit, Olivier Ruatta, and Julien Schrek. “On the complexity of the Rank Syndrome Decoding problem”. In: *CoRR* abs/1301.1026 (2013) (cit. on pp. 56, 58).

[GS98] Venkatesan Guruswami and Madhu Sudan. “Improved decoding of Reed–Solomon and algebraic-geometric codes”. In: *Proceedings 39th Annual Symposium on Foundations of Computer Science (Cat. No. 98CB36280)*. IEEE. 1998, pp. 28–37 (cit. on p. 61).

[GZ06] Philippe Gaborit and Gilles Zémor. “Asymptotic improvement of the Gilbert-Varshamov bound for linear codes”. In: *Proc. IEEE Int. Symposium Inf. Theory - ISIT 2006*. Seattle, USA, June 2006, pp. 287–291 (cit. on pp. 50, 207).

[GZ16] Philippe Gaborit and Gilles Zémor. “On the hardness of the decoding and the minimum distance problems for rank codes”. In: *IEEE Trans. Inform. Theory* 62(12) (2016), pp. 7245–7252 (cit. on pp. 57, 229).

[GZ61] Daniel Gorenstein and Neal Zierler. “A class of error-correcting codes in pˆm symbols”. In: *Journal of the Society for Industrial and Applied Mathematics* 9.2 (1961), pp. 207–214 (cit. on p. 34).

[Hay74] David R Hayes. “Explicit class field theory for rational function fields”. In: *Transactions of the American Mathematical Society* 189 (1974), pp. 77–91 (cit. on pp. xiii, 105).

[Hes02] Florian Hess. “Computing Riemann–Roch Spaces in Algebraic Function Fields and Related Topics”. In: *Journal of Symbolic Computation* 33.4 (2002). https://www.staff.uni-oldenburg.de/florian.hess/publications/rr.pdf, pp. 425–445. ISSN: 0747-7171 (cit. on p. 102).

[HHK17] Dennis Hofheinz, Kathrin Hövelmanns, and Eike Kiltz. “A modular analysis of the Fujisaki-Okamoto transformation”. In: *Theory of Cryptography Conference*. Springer. 2017, pp. 341–371 (cit. on p. 26).

[HKLP+12] Stephan Heyse, Eike Kiltz, Vadim Lyubashevsky, Christof Paar, and Krzysztof Pietrzak. "Lapin: An efficient authentication protocol based on Ring-LPN". In: *Fast Software Encryption - 19th International Workshop, FSE 2012, Washington, DC, USA, March 19-21, 2012*. Ed. by Anne Canteaut. Vol. 7549. LNCS. Washington DC, United States: Springer, 2012, pp. 346–365 (cit. on pp. 38, 124, 126, 205).

[HL17] Joris van der Hoeven and Robin Larrieu. "The Frobenius FFT". In: *Proceedings of the 2017 ACM on International Symposium on Symbolic and Algebraic Computation*. https://hal.science/hal-01453269/. 2017, pp. 437–444 (cit. on pp. 201, 225).

[HLS13] Joris van der Hoeven, Romain Lebreton, and Éric Schost. "Structured FFT and TFT: Symmetric and Lattice Polynomials". In: *International Symposium on Symbolic and Algebraic Computation, ISSAC'13, Boston, MA, USA, June 26-29, 2013*. Ed. by Manuel Kauers. https://www.texmacs.org/joris/symtft/symtft.pdf. ACM, 2013, pp. 355–362 (cit. on p. 201).

[Hoc59] Alexis Hocquenghem. "Codes correcteurs d'erreurs". In: *Chiffres* 2 (1959). (In French), pp. 147–156 (cit. on p. 34).

[HPS98] Jeffrey Hoffstein, Jill Pipher, and Joseph H Silverman. "NTRU: A ring-based public key cryptosystem". In: *International algorithmic number theory symposium*. Springer. 1998, pp. 267–288 (cit. on p. 43).

[HPSS+14] Jeffrey Hoffstein, Jill Pipher, John M. Schanck, Joseph H. Silverman, and William Whyte. "Practical Signatures from the Partial Fourier Recovery Problem". In: *Applied Cryptography and Network Security - 12th International Conference, ACNS 2014, Lausanne, Switzerland, June 10-13, 2014. Proceedings*. Ed. by Ioana Boureanu, Philippe Owesarski, and Serge Vaudenay. Vol. 8479. Lecture Notes in Computer Science. Springer, 2014, pp. 476–493 (cit. on p. 214).

[HVB16] Yi Hong, Emanuele Viterbo, and Jean-Claude Belfiore. "The Two-Modular Fourier Transform of Binary Functions". In: *IEEE Trans. Inf. Theory* 62.5 (2016), pp. 2813–2826 (cit. on p. 226).

[IKNP03] Yuval Ishai, Joe Kilian, Kobbi Nissim, and Erez Petrank. "Extending Oblivious Transfers Efficiently". In: *Advances in Cryptology - CRYPTO 2003, 23rd Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 2003, Proceedings*. Ed. by Dan Boneh. Vol. 2729. Lecture Notes in Computer Science. Springer, 2003, pp. 145–161 (cit. on pp. 179, 185).

[IKOS07] Yuval Ishai, Eyal Kushilevitz, Rafail Ostrovsky, and Amit Sahai. "Zero-knowledge from secure multiparty computation". In: *Proceedings of the 39th Annual ACM Symposium on Theory of Computing, San Diego, California, USA, June 11-13, 2007*. Ed. by David S. Johnson and Uriel Feige. ACM, 2007, pp. 21–30 (cit. on p. 174).

[IPS08] Yuval Ishai, Manoj Prabhakaran, and Amit Sahai. "Founding Cryptography on Oblivious Transfer - Efficiently". In: *Advances in Cryptology - CRYPTO 2008, 28th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2008. Proceedings*. Ed. by David A. Wagner. Vol. 5157. Lecture Notes in Computer Science. Springer, 2008, pp. 572–591 (cit. on p. 174).

[Jab01] Abdulrahman Al Jabri. "A statistical decoding algorithm for general linear block codes". In: *Cryptography and coding. Proceedings of the 8th IMA International Conference*. Ed. by Bahram Honary. Vol. 2260. LNCS. Cirencester, UK: Springer, Dec. 2001, pp. 1–8 (cit. on pp. 20, 49).

[JMV09] David Jao, Stephen D Miller, and Ramarathnam Venkatesan. "Expander Graphs Based on GRH with an application to Elliptic Curve Cryptography". In: *Journal of Number Theory* 129.6 (2009), pp. 1491–1504 (cit. on p. 170).

[JN03] Antoine Joux and Kim Nguyen. "Separating Decision Diffie-Hellman from Computational Diffie-Hellman in Cryptographic Groups". In: *J. Cryptol.* 16.4 (2003), pp. 239–247 (cit. on pp. 4, 27).

[Joz01] Richard Jozsa. "Quantum factoring, discrete logarithms, and the hidden subgroup problem". In: *Comput. Sci. Eng.* 3.2 (2001), pp. 34–43 (cit. on p. 2).

[Kas74] Tadao Kasami. "A Gilbert-Varshamov bound for quasi-cycle codes of rate 1/2 (Corresp.)" In: *IEEE Transactions on Information Theory* 20.5 (1974), pp. 679–679 (cit. on p. 207).

[Kon12] Aryeh Kontorovich. "Obtaining measure concentration from Markov contraction". In: *Markov Processes and Related Fields* 18.4 (2012), pp. 613–638 (cit. on p. 16).

[Köt92] Ralf Kötter. "A unified description of an error locating procedure for linear codes". In: *Proc. Algebraic and Combinatorial Coding Theory*. Voneshta Voda, 1992, pp. 113–117 (cit. on p. 211).

[KPR18] Marcel Keller, Valerio Pastro, and Dragos Rotaru. "Overdrive: Making SPDZ Great Again". In: *Advances in Cryptology - EUROCRYPT 2018 - 37th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tel Aviv, Israel, April 29 - May 3, 2018 Proceedings, Part III*. Ed. by Jesper Buus Nielsen and Vincent Rijmen. Vol. 10822. Lecture Notes in Computer Science. Springer, 2018, pp. 158–189 (cit. on pp. 179, 185).

[KT17] Ghazal Kachigar and Jean-Pierre Tillich. "Quantum Information Set Decoding Algorithms". In: *Post-Quantum Cryptography 2017*. Vol. 10346. LNCS. Utrecht, The Netherlands: Springer, June 2017, pp. 69–89 (cit. on p. 21).

[Lan12] Serge Lang. *Algebra*. Vol. 211. Springer Science & Business Media, 2012 (cit. on p. 200).

[LC23] Antoine Leudière and Xavier Caruso. *Drinfeld modules in Sage*. https://doc.sagemath.org/html/en/reference/drinfeld_modules/sage/rings/function_field/drinfeld_modules/drinfeld_module.html. 2023 (cit. on p. 219).

[LCKC+18] Wen-Ding Li, Ming-Shing Chen, Po-Chun Kuo, Chen-Mou Cheng, and Bo-Yin Yang. "Frobenius Additive Fast Fourier Transform". In: *Proceedings of the 2018 ACM on International Symposium on Symbolic and Algebraic Computation, ISSAC 2018, New York, NY, USA, July 16-19, 2018*. Ed. by Manuel Kauers, Alexey Ovchinnikov, and Éric Schost. ACM, 2018, pp. 263–270 (cit. on p. 225).

[LDKL+20] Vadim Lyubashevsky, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Peter Schwabe, Gregor Seiler, Damien Stehlé, and Shi Bai. *CRYSTALS-Dilithium: algorithm specifications and supporting documentation*. Version 3.1. https://pq-crystals.org/dilithium/data/dilithium-specification-round3-20210208.pdf. Feb. 2020 (cit. on pp. viii, 2).

[LDW94] Yuan Xing Li, Robert H. Deng, and Xin Mei Wang. "On the equivalence of McEliece's and Niederreiter's public-key cryptosystems". In: *IEEE Trans. Inform. Theory* 40.1 (1994), pp. 271–273 (cit. on p. 25).

[Lee58] C. Y. Lee. "Some properties of nonbinary error-correcting codes". In: *IRE Trans. Inf. Theory* 4.2 (1958), pp. 77–82 (cit. on p. 153).

[Lev93] Leonid A. Levin. "Randomness and Non-determinism". In: *Journal of Symbolic Logic* 58(3):1102-1103 (1993). http://arxiv.org/abs/1211.0071. Also available in *International Congress of Mathematicians: Abstracts of Invited Lectures.* p.155 Zuruch, August 4, 1994 (cit. on p. 31).

[LLP20] Julien Lavauzelle, Pierre Loidreau, and Ba-Duc Pham. "RAMESSES, a Rank Metric Encryption Scheme with Short Keys". working paper or preprint. Jan. 2020 (cit. on pp. xi, 7, 62, 65, 67, 76, 78, 81, 227).

[LN97] Rudolf Lidl and Harald Niederreiter. *Finite fields*. Second. Vol. 20. Encyclopedia of Mathematics and its Applications. With a foreword by P. M. Cohn. Cambridge University Press, Cambridge, 1997, pp. xiv+755. ISBN: 0-521-39231-4 (cit. on p. 127).

[LO06] Pierre Loidreau and Raphael Overbeck. "Decoding rank errors beyond the error-correction capability". In: *Proceedings of the Tenth International Workshop on Algebraic and Combinatorial Coding Theory, ACCT-10*. 2006, pp. 168–190 (cit. on p. 70).

[Loi06a] Pierre Loidreau. "A Welch–Berlekamp Like Algorithm for Decoding Gabidulin Codes". In: *Coding and Cryptography*. Ed. by Øyvind Ytrehus. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 36–45 (cit. on pp. 61, 62).

[Loi06b] Pierre Loidreau. *Properties of codes in rank metric*. 2006 (cit. on p. 57).

[Lor21] Dino Lorenzini. *An invitation to arithmetic geometry*. Vol. 9. American Mathematical Society, 2021 (cit. on pp. 97–99, 115).

[LP06a] Françoise Levy-dit-Vehel and Ludovic Perret. "Algebraic decoding of rank metric codes". In: *Talk at the Special Semester on Gröbner Bases - Workshop D1* (2006), pp. 1–19 (cit. on p. 58).

[LP06b] Françoise Lévy-dit-Vehel and Ludovic Perret. "Algebraic decoding of codes in rank metric". communication at YACC06, Porquerolles, France. June 2006 (cit. on p. 58).

[LPR10] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. "On Ideal Lattices and Learning with Errors over Rings". In: *Advances in Cryptology - EUROCRYPT2010*. Vol. 6110. LNCS. Springer, 2010, pp. 1–23 (cit. on pp. x, xii, 5, 48, 93, 95, 105, 111, 114–117, 216).

[LS15] Adeline Langlois and Damien Stehlé. "Worst-case to average-case reductions for module lattices". In: *Des. Codes Cryptogr.* 75 (2015), pp. 565–599 (cit. on pp. xii, 5, 40, 48, 93, 113, 115).

[LSS14] Adeline Langlois, Damien Stehlé, and Ron Steinfeld. "GGHLite: More Efficient Multilinear Maps from Ideal Lattices". In: *Advances in Cryptology - EUROCRYPT 2014 - 33rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Copenhagen, Denmark, May 11-15, 2014. Proceedings*. Ed. by Phong Q. Nguyen and Elisabeth Oswald. Vol. 8441. Lecture Notes in Computer Science. Springer, 2014, pp. 239–256 (cit. on p. 228).

[Lyu05] Vadim Lyubashevsky. "On Random High Density Subset Sums". In: *Electronic Colloquium on Computational Complexity (ECCC)* 1.007 (2005) (cit. on p. 211).

[Lyu11] Vadim Lyubashevsky. "Search to decision reduction for the learning with errors over rings problem". In: *ITW*. IEEE, 2011, pp. 410–414 (cit. on pp. 112, 114).

[LZA18] Xingye Lu, Zhenfei Zhang, and Man Ho Au. "Practical Signatures from the Partial Fourier Recovery Problem Revisited: A Provably-Secure and Gaussian-Distributed Construction". In: *Information Security and Privacy - 23rd Australasian Conference, ACISP 2018, Wollongong, NSW, Australia, July 11-13, 2018, Proceedings*. Ed. by Willy Susilo and Guomin Yang. Vol. 10946. Lecture Notes in Computer Science. Springer, 2018, pp. 813–820 (cit. on p. 214).

[McE78] Robert J. McEliece. "A Public-Key System Based on Algebraic Coding Theory". In: DSN Progress Report 44. Jet Propulsion Lab, 1978, pp. 114–116 (cit. on pp. 3, 22, 23, 26).

[Mer78] Ralph C. Merkle. "Secure Communications Over Insecure Channels". In: *Commun. ACM* 21.4 (1978). *Although it was published in 1978, it was already submitted and known by Diffie and Hellman when they published their famous papers in 1976.*, pp. 294–299 (cit. on p. 2).

[MG02] Daniele Micciancio and Shafi Goldwasser. *Complexity of lattice problems: a cryptographic perspective*. Vol. 671. Springer Science & Business Media, 2002 (cit. on p. 15).

[MMPP+23] Luciano Maino, Chloe Martindale, Lorenz Panny, Giacomo Pope, and Benjamin Wesolowski. "A Direct Key Recovery Attack on SIDH". In: *Advances in Cryptology - EUROCRYPT 2023 - 42nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Lyon, France, April 23-27, 2023, Proceedings, Part V*. Ed. by Carmit Hazay and Martijn Stam. Vol. 14008. Lecture Notes in Computer Science. Springer, 2023, pp. 448–471 (cit. on p. 2).

[MO15] Alexander May and Ilya Ozerov. "On Computing Nearest Neighbors with Applications to Decoding of Binary Linear Codes". In: *Advances in Cryptology - EUROCRYPT 2015*. Ed. by E. Oswald and M. Fischlin. Vol. 9056. LNCS. Springer, 2015, pp. 203–228 (cit. on pp. 20, 49).

[MR04] D. Micciancio and O. Regev. "Worst-case to average-case reductions based on Gaussian measures". In: *45th Annual IEEE Symposium on Foundations of Computer Science*. 2004, pp. 372–381 (cit. on pp. 50, 115).

[MS86] Florence J. MacWilliams and Neil J. A. Sloane. *The Theory of Error-Correcting Codes*. Fifth. Amsterdam: North–Holland, 1986 (cit. on pp. 13, 26).

[MT22] Rocco Mora and Jean-Pierre Tillich. "On the dimension and structure of the square of the dual of a Goppa code". In: *WCC 2022 - Workshop on Coding Theory and Cryptography*. 2022 (cit. on pp. 4, 41).

[MTSB13] Rafael Misoczki, Jean-Pierre Tillich, Nicolas Sendrier, and Paulo S. L. M. Barreto. "MDPC-McEliece: New McEliece variants from Moderate Density Parity-Check codes". In: *Proc. IEEE Int. Symposium Inf. Theory - ISIT*. 2013, pp. 2069–2073 (cit. on pp. ix, 3, 13, 26, 27).

[Mul54] David E. Muller. "Application of Boolean algebra to switching circuit design and to error detection". In: *Transactions of the IRE professional group on electronic computers* 3.3 (1954), pp. 6–12 (cit. on p. 26).

[MW16] Pratyay Mukherjee and Daniel Wichs. "Two Round Multiparty Computation via Multi-key FHE". In: *Advances in Cryptology - EUROCRYPT 2016 - 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Vienna, Austria, May 8-12, 2016, Proceedings, Part II*. Ed. by Marc Fischlin and Jean-Sébastien Coron. Vol. 9666. Lecture Notes in Computer Science. Springer, 2016, pp. 735–763 (cit. on p. 174).

[Nan06] Mridul Nandi. "A Simple and Unified Method of Proving Indistinguishability". In: *Progress in Cryptology - INDOCRYPT 2006, 7th International Conference on Cryptology in India, Kolkata, India, December 11-13, 2006, Proceedings*. Ed. by Rana Barua and Tanja Lange. Vol. 4329. Lecture Notes in Computer Science. Springer, 2006, pp. 317–334 (cit. on p. 28).

[Nie86] Harald Niederreiter. "Knapsack-type cryptosystems and algebraic coding theory". In: *Problems of Control and Information Theory* 15.2 (1986), pp. 159–166 (cit. on p. 25).

[Noe32] Emmy Noether. "Normalbasis bei Körpern ohne Höhere Verzweigung". In: *J. Reine Angew. Math.* 167 (1932). (in German), pp. 147–152 (cit. on p. 127).

[Obe07] Ulrich Oberst. "The Fast Fourier Transform". In: *SIAM Journal on Control and Optimization* 46.2 (2007). https://www.uibk.ac.at/mathematik/personal/oberst/fftsicon065824.pdf, pp. 496–540 (cit. on pp. 200, 201).

[Ore33] Øystein Ore. "On a special class of polynomials". In: *Trans. Amer. Math. Soc.* 35.3 (1933), pp. 559–584 (cit. on p. 60).

[OS09] Raphael Overbeck and Nicolas Sendrier. "Code-based cryptography". In: *Post-quantum cryptography*. Ed. by Daniel J. Bernstein, Johannes Buchmann, and Erik Dahmen. Springer, 2009, pp. 95–145. ISBN: 978-3-540-88701-0 (cit. on p. 21).

[OTD10] Ayoub Otmani, Jean-Pierre Tillich, and Léonard Dallot. "Cryptanalysis of Two McEliece Cryptosystems Based on Quasi-Cyclic Codes". In: *Special Issues of Mathematics in Computer Science* 3.2 (Jan. 2010), pp. 129–140 (cit. on p. 40).

[Ove05] Raphael Overbeck. "A New Structural Attack for GPT and Variants". In: *Mycrypt*. Vol. 3715. LNCS. 2005, pp. 50–63 (cit. on pp. 59, 62).

[Ove06] Raphael Overbeck. "Statistical decoding revisited". In: *Information security and privacy : 11th Australasian conference, ACISP 2006*. Ed. by Reihaneh Safavi-Naini Lynn Batten. Vol. 4058. LNCS. Springer, 2006, pp. 283–294 (cit. on pp. 20, 49).

[Pel19] Alice Pellet–Mary. "On ideal lattices and the GGH13 multilinear map". https://theses.hal.science/tel-02337930. PhD thesis. École Normale Supérieure de Lyon, Oct. 2019 (cit. on p. 196).

[Pel92] Ruud Pellikaan. "On decoding by error location and dependent sets of error positions". In: *Discrete Math.* 106–107 (1992), pp. 368–381 (cit. on p. 211).

[Pie67] John N. Pierce. "Limit distribution of the minimum distance of random linear codes". In: *IEEE Trans. Inform. Theory* 13.1 (1967), pp. 595–599 (cit. on p. 18).

[Pra57] Eugene Prange. *Cyclic Error Correcting Codes in Two Symbols*. Version TN-57-103. Air Force Cambridge Research Labs, Bedford, Massachussetts. Sept. 1957 (cit. on p. 34).

[Pra58] Eugene Prange. *Some Cyclic Error-Correcting Codes with Simple Decoding Algorithms*. Version TN-58-156. Air Force Cambridge Research Labs, Bedford, Massachussetts. Apr. 1958 (cit. on p. 34).

[Pra62] Eugene Prange. "The use of information sets in decoding cyclic codes". In: *IRE Transactions on Information Theory* 8.5 (1962), pp. 5–9 (cit. on pp. 20, 49, 160, 161, 211).

[Pre17] Thomas Prest. "Sharper Bounds in Lattice-Based Cryptography Using the Rényi Divergence". In: *ASIACRYPT 2017*. Ed. by Tsuyoshi Takagi and Thomas Peyrin. Vol. 10624. LNCS. Springer, 2017, pp. 347–374 (cit. on p. 228).

[PRS17] Chris Peikert, Oded Regev, and Noah Stephens-Davidowitz. "Pseudorandomness of ring-LWE for any ring and modulus". In: *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*. 2017, pp. 461–473 (cit. on pp. xii, xiii, 5, 8, 48, 134, 135, 139, 140, 144).

[PS21a] Alice Pellet-Mary and Damien Stehlé. "On the Hardness of the NTRU Problem". In: *Advances in Cryptology - ASIACRYPT 2021 - 27th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, December 6-10, 2021, Proceedings, Part I*. Ed. by Mehdi Tibouchi and Huaxiong Wang. Vol. 13090. Lecture Notes in Computer Science. Springer, 2021, pp. 3–35 (cit. on p. 43).

[PS21b] Alice Pellet-Mary and Damien Stehlé. "On the hardness of the NTRU problem". In: *Asiacrypt 2021 - 27th Annual International Conference on the Theory and Applications of Cryptology and Information Security*. Advances in Cryptology – ASIACRYPT 2021. Lecture Notes in Computer Science, vol 13090. Singapore, Singapore, Dec. 2021 (cit. on pp. xii, 134).

[PTP15] Alberto Pedrouzo-Ulloa, Juan Ramón Troncoso-Pastoriza, and Fernando Pérez-González. "Multivariate lattices for encrypted image processing". In: *2015 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2015, South Brisbane, Queensland, Australia, April 19-24, 2015*. IEEE, 2015, pp. 1707–1711 (cit. on pp. 206, 214).

[PTP16] Alberto Pedrouzo-Ulloa, Juan Ramón Troncoso-Pastoriza, and Fernando Pérez-González. "On Ideal Lattices over the Tensor Product of Number Fields and Ring Learning with Errors over Multivariate Rings". In: abs/1607.05244 (2016). arXiv: 1607.05244 (cit. on pp. 206, 214).

[RAD+78] Ronald L Rivest, Len Adleman, Michael L Dertouzos, et al. "On data banks and privacy homomorphisms". In: *Foundations of secure computation* 4.11 (1978), pp. 169–180 (cit. on p. 174).

[RBKM+23] Stefan Ritterhoff, Sebastian Bitzer, Patrick Karl, Georg Maringer, Thomas Schamberger, Jonas Schupp, Georg Sigl, Antonia Wachter-Zeh, and Violetta Weger. *FuLeeca: A Lee-based Signature Scheme*. Round 1 Submission to the NIST second Call for Post-Quantum signatures. Version 1. https://www.ce.cit.tum.de/lnt/forschung/professur-fuer-coding-and-cryptography/fuleeca/. June 2023 (cit. on p. 153).

[Ree54] Irving S. Reed. "A class of multiple-error-correcting codes and the decoding scheme". In: *Trans. IRE Prof. Group Inf. Theory* 4 (1954), pp. 38–49 (cit. on p. 26).

[Reg05] Oded Regev. "On lattices, learning with errors, random linear codes, and cryptography". In: *Proceedings of the 37th Annual ACM Symposium on Theory of Computing, Baltimore, MD, USA, May 22-24, 2005*. 2005, pp. 84–93 (cit. on p. 4).

[Rob23] Damien Robert. "Breaking SIDH in Polynomial Time". In: *Advances in Cryptology - EUROCRYPT 2023 - 42nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Lyon, France, April 23-27, 2023, Proceedings, Part V*. Ed. by Carmit Hazay and Martijn Stam. Vol. 14008. Lecture Notes in Computer Science. Springer, 2023, pp. 472–503 (cit. on p. 2).

[Ros02] Michael Rosen. *Number Theory in Function Fields*. Graduate Texts in Mathematics. Springer, 2002. ISBN: 978-1-4757-6046-0 (cit. on pp. 105, 109, 110).

[RPW21] Julian Renner, Sven Puchinger, and Antonia Wachter–Zeh. "LIGA: A Cryptosystem Based on the Hardness of Rank–Metric List and Interleaved Decoding". In: *Des. Codes Cryptogr.* 89 (2021), pp. 1279–1319 (cit. on pp. xi, 7, 61, 65, 67, 68, 76, 85, 227).

[RS60] Irving S. Reed and Gustave Solomon. "Polynomial codes over certain finite fields". In: *Journal of the society for industrial and applied mathematics* 8.2 (1960), pp. 300–304 (cit. on pp. ix, 3, 13, 26, 61).

[RSA78] Ronald L. Rivest, Adi Shamir, and Leonard M. Adleman. "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems". In: *Commun. ACM* 21.2 (1978), pp. 120–126 (cit. on pp. vii, 2, 23).

[RSW18] Miruna Rosca, Damien Stehlé, and Alexandre Wallet. "On the ring-LWE and polynomial-LWE problems". In: *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer. 2018, pp. 146–173 (cit. on pp. xii, 48, 134, 140, 169).

[RW15] Netanel Raviv and Antonia Wachter-Zeh. "Some Gabidulin codes cannot be list decoded efficiently at any radius". In: *Proc. IEEE Int. Symposium Inf. Theory - ISIT*. 2015, pp. 6–10 (cit. on p. 61).

[Sen11] Nicolas Sendrier. "Decoding One Out of Many". In: *Post-Quantum Cryptography 2011*. Vol. 7071. LNCS. 2011, pp. 51–67 (cit. on pp. 38, 46, 211).

[Sha48] Claude E. Shannon. "A Mathematical Theory of Communication". In: *Bell System Technical Journal* 27.3 (1948), pp. 379–423. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/j.1538-7305.1948.tb01338.x (cit. on pp. ix, 1, 3, 13).

[Sha49] Claude E. Shannon. "Communication Theory of Secrecy Systems". In: *Bell Syst. Tech. J.* 28.4 (1949). https://www.cs.virginia.edu/~evans/greatworks/shannon1949.pdf, pp. 656–715 (cit. on pp. ix, 1, 3).

[Sha79] Adi Shamir. "How to Share a Secret". In: *Commun. ACM* 22.11 (1979), pp. 612–613 (cit. on pp. 175, 176).

[Sho09] Victor Shoup. *A computational introduction to number theory and algebra*. Cambridge university press, 2009 (cit. on p. 15).

[Sho94] Peter W. Shor. "Algorithms for Quantum Computation: Discrete Logarithms and Factoring". In: *35th Annual Symposium on Foundations of Computer Science, Santa Fe, New Mexico, USA, 20-22 November 1994*. IEEE Computer Society, 1994, pp. 124–134 (cit. on pp. vii, 2).

[SJB11] Vladimir Sidorenko, Lan Jiang, and Martin Bossert. "Skew-Feedback Shift-Register Synthesis and Decoding Interleaved Gabidulin Codes". In: *IEEE Trans. Inf. Theory* 57.2 (2011), pp. 621–632 (cit. on p. 76).

[SK11] Danilo Silva and Frank R. Kschischang. "Universal Secure Network Coding via Rank-Metric Codes". In: *IEEE Trans. Inform. Theory* 57.2 (Feb. 2011), pp. 1124–1135. ISSN: 1557-9654 (cit. on pp. 13, 59).

[SS11] Damien Stehlé and Ron Steinfeld. "Making NTRU as Secure as Worst-Case Problems over Ideal Lattices". In: *Advances in Cryptology - EUROCRYPT 2011*. Vol. 6632. LNCS. 2011, pp. 27–47 (cit. on p. 43).

[SS92] Vladimir Michilovich Sidelnikov and S.O. Shestakov. "On the insecurity of cryptosystems based on generalized Reed-Solomon codes". In: *Discrete Math. Appl.* 1.4 (1992), pp. 439–444 (cit. on p. 4).

[SSTX09] Damien Stehlé, Ron Steinfeld, Keisuke Tanaka, and Keita Xagawa. "Efficient Public Key Encryption Based on Ideal Lattices". In: *Advances in Cryptology - ASIACRYPT 2009, 15th International Conference on the Theory and Application of Cryptology and Information Security, Tokyo, Japan, December 6-10, 2009. Proceedings*. Ed. by Mitsuru Matsui. Vol. 5912. LNCS. Springer, 2009, pp. 617–635 (cit. on pp. xii, 5, 93, 95).

[Ste+23] W. A. Stein et al. *Sage Mathematics Software (Version 10.0.0)*. http://www.sagemath.org. The Sage Development Team. 2023 (cit. on pp. xi, 89, 219).

[Ste88] Jacques Stern. "A method for finding codewords of small weight". In: *Coding Theory and Applications*. Ed. by G. D. Cohen and J. Wolfmann. Vol. 388. LNCS. Springer, 1988, pp. 106–113 (cit. on pp. 20, 49).

[Sti09] Henning Stichtenoth. *Algebraic function fields and codes*. Second. Vol. 254. Graduate Texts in Mathematics. Springer-Verlag, Berlin, 2009, pp. xiv+355. ISBN: 978-3-540-76877-7 (cit. on p. 98).

[Til18] Jean-Pierre Tillich. "The Decoding Failure Probability of MDPC Codes". In: *2018 IEEE International Symposium on Information Theory, ISIT 2018, Vail, CO, USA, June 17-22, 2018*. 2018, pp. 941–945 (cit. on p. 33).

[Val84] Leslie G. Valiant. "A Theory of the Learnable". In: *Commun. ACM* 27.11 (1984), pp. 1134–1142 (cit. on p. 22).

[Var97] Alexander Vardy. "The Intractability of Computing the Minimum Distance of a Code". In: *IEEE Trans. Inform. Theory* 43.6 (Nov. 1997), pp. 1757–1766 (cit. on p. 209).

[Vil06] Gabriel Daniel Villa Salvador. *Topics in the Theory of Algebraic Function Fields*. Springer, 2006. ISBN: 978-0-8176-4480-2 (cit. on pp. 102, 105, 108, 109, 170, 222).

[Wac13] Antonia Wachter-Zeh. "Decoding of block and convolutional codes in rank metric". PhD thesis. Université Rennes 1, 2013 (cit. on pp. 61, 76).

[Was77] Siri Krishan Wasan. "Quasi Abelian Codes". In: *Publications de l'Institut Mathématique* 21.35 (1977). http://elib.mi.sanu.ac.rs/files/journals/publ/41/31.pdf, pp. 201–206 (cit. on p. 193).

[Wei40] André Weil. "Sur les fonctions algébriques à corps de constantes fini". In: *Comptes Rendus Hebdomadaires des Séances de l'Académie des Sciences,* **210**. In French. 1940, pp. 592–594 (cit. on p. 98).

[Wei48] André Weil. *Sur les courbes algébriques et les variétés qui s'en déduisent*. In French. Actualités Scientifiques et Industrielles, **1041**, 1948 (cit. on p. 98).

[Wil21] Wolfgang Willems. "Codes in group algebras". In: *Concise Encyclopedia of Coding Theory*. Ed. by W. Cary Huffman, Jon-Lark Kim, and Patrick Solé. Chapman and Hall/CRC, 2021. Chap. 16 (cit. on pp. 123, 205, 211).

[WPR18] Antonia Wachter-Zeh, Sven Puchinger, and Julian Renner. "Repairing the Faure-Loidreau Public-Key Cryptosystem". In: *Proc. IEEE Int. Symposium Inf. Theory - ISIT*. 2018, pp. 2426–2430 (cit. on p. 76).

[Yao82] Andrew Chi-Chih Yao. "Protocols for Secure Computations (Extended Abstract)". In: *23rd Annual Symposium on Foundations of Computer Science, Chicago, Illinois, USA, 3-5 November 1982*. IEEE Computer Society, 1982, pp. 160–164 (cit. on p. 174).

[YZ21] Yu Yu and Jiang Zhang. "Smoothing Out Binary Linear Codes and Worst-Case Sub-exponential Hardness for LPN". In: *Advances in Cryptology - CRYPTO 2021 - 41st Annual International Cryptology Conference, CRYPTO 2021, Virtual Event, August 16-20, 2021, Proceedings, Part III*. Ed. by Tal Malkin and Chris Peikert. Vol. 12827. LNCS. Springer, 2021, pp. 473–501 (cit. on pp. 154, 165).

[Zap20] Ilaria Zappatore. "Reconstruction Rationnelle Simultanée et applications à la Théorie des Codes Correcteurs d'Erreurs". https://theses.hal.science/tel-03013914v2/document. PhD thesis. Université de Montpellier, Oct. 2020 (cit. on p. 70).

[Zém16] Gilles Zémor. *Notes on Alekhnovich's cryptosystems*. https://www.math.u-bordeaux.fr/~gzemor/alekhnovich.pdf. 2016 (cit. on p. 31).

ECOLE
DOCTORALE

Titre : Codes Structurés pour la Cryptographie: des Fondations Théoriques de Sécurité aux Applications

Mots clés : Cryptographie post-quantique, Codes structurés, Cryptanalyse et réductions, Générateur d'aléas corrélé, Modules de Carlitz

Résumé : Dans cette thèse, nous nous intéressons à la cryptographie fondée sur les codes correcteurs d'erreurs, et plus particulièrement sur ceux possédant une forte structure algébrique. La cryptographie à base de codes est ancienne, puisque McEliece proposait déjà en 1978 un schéma de chiffrement reposant, entre autres, sur la difficulté du problème de décodage. Il est important de noter que le cryptosystème de McEliece est encore aujourd'hui résistant aux attaques, y compris à l'aide d'un ordinateur quantique (même si ses paramètres ont dû être mis à jour pour s'adapter aux nouvelles normes de sécurité). En particulier, cela fait de McEliece le plus ancien cryptosystème avec cette propriété. En effet, il est connu depuis les années 1990 que l'algorithme de Shor présente une menace importante pour la cryptographie actuellement utilisée en pratique. Néanmoins, le système de McEliece souffre d'un gros inconvénient : ses clés publiques sont énormes. Afin de résoudre ce problème de taille, il a été proposé d'utiliser des codes correcteurs d'erreurs avec une structure algébrique additionnelle comme les codes quasi-cycliques, offrant une représentation plus compacte et de meilleures performances. Cependant, il est fondamental de s'assurer que cette efficacité ne se fasse pas au détriment de la sécurité. Ceci est d'autant plus important que le NIST (National Institute for Standards and Technology) a annoncé que le prochain standard de chiffrement post-quantique serait choisi parmi trois candidats, dont deux (BIKE et HQC) utilisent des codes structurés. Néanmoins, HQC est fondé sur un paradigme différent de celui de McEliece : sa sécurité repose sur la difficulté d'une variante dite décisionnelle du problème de décodage. Si dans le cas des codes génériques il est connu que les variantes de recherche et de décision sont en réalité équivalentes, la situation est beaucoup moins claire dans le cas des codes structurés pour lesquels aucune réduction de recherche-à-décision n'est connue. Cependant, l'existence de réductions théoriques vers des problèmes dont la difficulté est bien établie est une caractéristique importante pour accroître la confiance en un cryptosystème. En particulier, l'une des contributions de cette thèse est une attaque sur des chiffrements à base de codes en métrique rang (une autre forme de codes structurés) qui ne possédaient pas ce type de réduction. C'est dans ce contexte que se place la deuxième partie de cette thèse : inspirés par des techniques utilisées en cryptographie à base de réseaux euclidiens, nous donnons la première réduction de recherche-à-décision pour certaines familles de codes quasi-cycliques. Elle repose sur une nouvelle interprétation de ces codes à l'aide d'outils issus de la théorie des corps de fonctions en caractéristique positive : les modules de Carlitz. Cependant, cette réduction présente certaines limitations que nous tentons ensuite de lever. Finalement, dans une dernière partie nous explorons une autre application de ces codes structurés dans le domaine du calcul multiparti sécurisé (MPC). L'objectif d'un protocole de MPC est de permettre à plusieurs joueurs d'effectuer un calcul ensemble de telle sorte qu'ils n'aient chacun qu'une connaissance partielle de l'entrée, et que personne ne puisse apprendre autre chose que la sortie. Il s'avère qu'un outil appelé PCG a été récemment introduit afin de distribuer efficacement aux joueurs de longues listes d'éléments aléatoires corrélés qui permettent ensuite d'accélérer les calculs. Notre analyse des codes quasi-cycliques permet alors de donner des fondations théoriques plus solides aux meilleurs constructions de PCG. Enfin, en utilisant pour la première fois la variante décisionnelle du problème de décodage des codes dits quasi-abéliens, dont nous analysons la difficulté à la lumière des techniques développées dans cette thèse, nous sommes capables de lever certaines limitations des constructions de l'état de l'art.

Title : Structured Codes for Cryptography: from Source of Hardness to Applications

Keywords : Post-quantum cryptography, Structured codes, Cryptanalysis and reductions, Pseudorandom correlation generators, Carlitz modules

Abstract : In this PhD thesis, we focus on cryptography based on error-correcting codes, and more specifically on those offering a strong algebraic structure. Code-based cryptography is not new, as McEliece already proposed in 1978 an encryption scheme which relies, among other things, on the hardness of the decoding problem. It is important to note that McEliece cryptosystem still appears to be resistant to known attacks (even though its parameters needed to be updated to meet new security standards), including those involving quantum computers. In particular, this makes of McEliece the oldest cryptosystem with this property. Indeed, it is known since the 1990s that Shor algorithm poses a significant threat to the cryptography currently used in practice. However, McEliece system suffers from a major drawback : its public keys are huge. In order to address this issue, it was proposed to use error-correcting codes with an additional algebraic structure, such as quasi-cyclic codes, which allow for a more compact representation and better performances. However, it is fundamental to ensure that this efficiency does not come at the expense of security. This is especially important as the National Institute for Standards and Technology (NIST) has announced that the next post-quantum encryption standard will be chosen out of three candidates, two of them (BIKE and HQC) based on structured codes. Nevertheless, HQC relies on a different paradigm than that of McEliece : its security is based on the difficulty of the so-called decisional variant of the decoding problem. In the case of generic codes, it is known that the decisional and search versions, which is the most studied one, are in fact equivalent. However, the situation is much more uncertain in the case of structured codes for which no reduction from search to decision is known. Yet, the existence of theoretical reductions to well-established hard problems is an important characteristic to increase confidence in a cryptosystem. In particular, one of the contributions of this thesis is an attack on code-based encryption schemes based on rank metric codes (another form of structured codes) that did not have this kind of reduction. This is the context of the second part of this thesis. Inspired by some techniques from lattice-based cryptography, we provide the first search-to-decision reduction for certain families of quasi-cyclic codes. It is based on a new interpretation of these codes using tools arising from the theory of function fields in positive characteristic, namely the Carlitz Modules. However, this reduction has certain limitations which we try to overcome. Finally, in the last part, we explore another application of these structured codes in the field of secure multiparty computation (MPC). The goal of an MPC protocol is to allow several players to perform a computation together so that they each only has partial knowledge of the input, and no one learns anything else than the output. It turns out that they can often benefit from long lists of correlated random elements to achieve fast computation, and a new tool called Pseudorandom Correlation Generators (PCG) was recently introduced to efficiently generate those long list. Thanks to our analysis of quasi-cyclic codes, we provide a more solid theoretical foundation for the best PCG constructions. Last but not least, we introduce for the first-time the decisional variant of the decoding problem of so-called quasi-abelian codes, and analyse its hardness in light of the techniques developed in this thesis. This allows us to overcome certain limitations of the state-of-the-art PCGs.

Institut Polytechnique de Paris
91120 Palaiseau, France