ulm university universität
uulm

Construction and Decoding of Evaluation Codes in Hamming and Rank Metric

DISSERTATION

zur Erlangung des akademischen Grades eines

DOKTOR-INGENIEURS
(DR.-ING.)

der Fakultät für Ingenieurwissenschaften,
Informatik und Psychologie der Universität Ulm

von

Sven Puchinger

geboren in Karlsruhe

Gutachter:	Prof. Dr.-Ing. Martin Bossert
	Prof. Dr. Tom Høholdt
Amtierender Dekan:	Prof. Dr. Frank Kargl

Ulm, 25. Juni 2018

Für Emma und Sandra.

Acknowledgments

This dissertation contains parts of my work as a research assistant at the Institute of Communications Engineering at Ulm University, Germany, from January 2014 to April 2018. During this time, many people inspired, motivated, and supported me, and I am profoundly thankful to them.

First of all, I would like to thank Martin Bossert for being an excellent doctoral adviser. Martin provided an extraordinarily inspiring and motivating research environment. This includes his contagious enthusiasm for new research problems and mathematical tools, his support for starting new research collaborations, and the freedom that he gives to his doctoral students. Furthermore, he is an enthusiastic teacher and encourages his students to think outside the (research) box as well. I would also like to thank Martin for supporting all my conference travels and other research stays.

I feel very grateful and honored that Tom Høholdt was the second examiner of this dissertation, especially since Tom's book (with Jørn Justesen) was—besides Martin's book—one of the first ones that I read about coding theory and which encouraged me to work in this research area. I would also like to thank Christian Waldschmidt and Maurits Ortmanns for being on my committee.

During my time as a doctoral student, I learned a lot from Johan Rosenkilde né Nielsen, Antonia Wachter-Zeh, and Vladimir (Volodya) Sidorenko, for which I am very thankful. Johan, with his always enthusiastic way of doing research (and everything else), has been a great co-author and mentor. His ideas and detailed comments made a significant contribution to the results presented in this thesis. Antonia already advised me during my bachelor's thesis, and, with her encouraging manner, she motivated me to work in coding theory. She has a great intuition for interesting research topics and is always full of ideas, some of which have found their way into this dissertation. I would also like to thank Volodya for the inspiring discussions about open research problems that we have had, and the numerous chocolate bars that helped solving some of them.

Several colleagues have made my time at Ulm University fun and enjoyable. In particular, I would like to thank Sven Müelich for being a great co-author, travel mate, and moral support, Mostafa Hosni Mohamed for being an awesome office mate and for the fantastic discussions, and George Yammine for the entertaining early-morning coffee breaks. Moreover, thanks to all the other colleagues at the institute for the great time.

During my time as a doctoral student, I (co-)advised several talented bachelor's and master's thesis student: Frederik Walter, David Mödinger, Karim Ishak, Yonatan Marin, Michael Zurell, Ranjith Ponnusamy, Sven Kahle, Veniamin Stukalov, Liming Fan, and Carmen Sippel. I enjoyed working together with them.

I would like to thank my co-authors that have not been mentioned above: Peter Beelen, Irene Bouw, Yuval Cassuto, Michael Cyran, Robert F. H. Fischer, Evyatar Hemo, Matthias Hiller, Johannes B. Huber, Ludwig Kürzinger, Wenhui Li, Julian Renner, John Sheekey, Georg Sigl, Ulrich Speidel, and Sebastian Stern.

Furthermore, I am grateful to Irene Bouw, Sven Müelich, Cornelia Ott, Vladimir Sidorenko, and Antonia Wachter-Zeh for proofreading parts of this dissertation.

Finally, I would like to thank my family, in particular my parents Susanne and Jürgen and my sister Anouk, who have always supported, encouraged, and motivated me through my entire life. I am also infinitely grateful to my wife Sandra and my daughter Emma for always loving and supporting me: you make me happy every day.

Ulm, June 2018 Sven Puchinger

Abstract

This dissertation considers constructions and decoders of several evaluation codes in Hamming and rank metric. Codes in Hamming metric have been studied since the 1950s and the known codes considered in this thesis have found many applications, e.g., satellite communication, CDs, DVDs, BluRays, RAID systems, QR codes, code-based cryptography, and modern storage systems. Rank-metric codes were first introduced in 1978 and have recently become an active research area due to their possible applications in network coding, cryptography, distributed storage systems, low-rank matrix recovery, and space-time coding.

In Hamming metric, we propose new partial unique decoding algorithms for decoding interleaved Reed–Solomon and interleaved one-point Hermitian codes beyond half the minimum distance. The algorithms combine ideas of collaborative decoding of interleaved codes and power decoding. For interleaved Reed–Solomon codes, we achieve the same maximal decoding radius as the previous best decoder, but at a better complexity. Our decoder for interleaved one-point Hermitian codes improves upon the previous best maximal decoding radius at all rates. Simulation results for various parameters indicate that both algorithms achieve their maximal decoding radii with high probability for random errors.

Inspired by a recent rank-metric code construction by Sheekey, called twisted Gabidulin codes, we present a new code class in Hamming metric: Twisted Reed–Solomon codes. The class contains many maximum distance separable (MDS) codes that are inequivalent to Reed–Solomon codes. We study the duals and Schur squares of the new codes and propose a list decoder that is efficient for many parameters. Furthermore, we single out two subclasses of long non-Reed–Solomon MDS codes and show that there is a subclass resisting some known structural attacks on the McEliece code-based cryptosystem.

In rank metric, we present the first sub-quadratic runtime half-the-minimum-distance decoding algorithm for Gabidulin codes, a well-known class of maximum rank distance (MRD) codes that can be seen as the rank-metric analog of Reed–Solomon codes. The result is achieved by accelerating many operations over skew polynomial rings that occur in a known decoding algorithm.

Further, we show that the core steps of several known decoding algorithms for interleaved Gabidulin codes, both key-equation- and interpolation-based, can be implemented using row reduction of skew polynomial matrices. By adapting well-known row-reduction algorithms over ordinary polynomial rings to skew polynomials, we achieve conceptually simple and in some cases faster decoding algorithms. The approach is inspired by several recent publications that found a unified description of various decoders of Reed–Solomon, one-point Hermitian, and their interleaved codes.

Finally, we propose a generalization of Sheekey's twisted Gabidulin codes, using similar methods as for our twisted Reed–Solomon codes. The new code class contains many MRD codes that are inequivalent to both Gabidulin codes and the original twisted Gabidulin codes.

Contents

1 Introduction **1**

2 Preliminaries **5**
2.1 Basic Terminology 5
2.1.1 Block Codes 5
2.1.2 Notation 6
2.2 Codes in Hamming Metric 7
2.2.1 Reed–Solomon Codes 8
2.2.2 One-Point Hermitian Codes 9
2.2.3 Interleaved Codes in Hamming Metric 11
2.3 Codes in Rank Metric 12
2.3.1 Skew Polynomial Rings 14
2.3.2 Modules over Skew Polynomial Rings 17
2.3.3 Gabidulin Codes 19
2.3.4 Interleaved Gabidulin Codes 20
2.3.5 Twisted Gabidulin Codes 21
2.3.6 Rank-Metric Codes over Fields of Characteristic Zero 22

Part I: Codes in Hamming Metric **25**

3 Improved Power Decoding of Interleaved Codes in Hamming Metric **25**
3.1 Improved Power Decoding of Interleaved Reed–Solomon Codes 28
3.1.1 Key Equations 28
3.1.2 Solving the Key Equations 30
3.1.3 Decoding Radius 33
3.1.4 Failure Behavior 36
3.1.5 Simulation Results 38
3.2 Improved Power Decoding of Interleaved One-Point Hermitian Codes 39
3.2.1 Key Equations 39
3.2.2 Solving the Key Equations 41
3.2.3 Decoding Radius and Failure Behavior 44
3.2.4 Simulation Results 47
3.3 Comparison of the New Decoders 48
3.4 Concluding Remarks 51

4 Twisted Reed–Solomon Codes **53**
4.1 Definition . . . 54
4.2 A Sufficient Condition for Twisted RS Codes to be MDS . . . 55
4.3 Decoding . . . 58
4.4 Dual Codes . . . 60
4.5 Schur Squares . . . 62
4.5.1 Schur Squares of Twisted RS Codes . . . 63
4.5.2 Codes with Maximal Schur Square Dimension . . . 65
4.6 Relation to Reed–Solomon Codes . . . 69
4.6.1 Low-Rate Non-GRS Twisted RS Codes . . . 69
4.6.2 A Combinatorial Inequivalence Argument . . . 70
4.6.3 Separation from GRS Codes Using Schur Squares . . . 74
4.7 Subclasses of Long MDS Twisted RS Codes . . . 75
4.7.1 $(*)$-Twisted Reed–Solomon Codes . . . 76
4.7.2 $(+)$-Twisted Reed–Solomon Codes . . . 79
4.7.3 Computer Searches . . . 81
4.8 Twisted RS Codes in the McEliece Cryptosystem . . . 83
4.8.1 Twisted RS Codes Resisting Some Known Structural Attacks . . . 85
4.8.2 Example Parameters Resulting in Small Key Sizes . . . 88
4.9 Concluding Remarks . . . 89

Part II: Codes in Rank Metric **93**

5 Fast Decoding of Gabidulin Codes **93**
5.1 Decoding Gabidulin Codes up to Half the Minimum Distance . . . 93
5.2 Fast Operations on Skew Polynomials . . . 96
5.2.1 Multiplication . . . 98
5.2.2 Division . . . 101
5.2.3 Minimal Subspace Polynomials and Multi-Point Evaluation . . . 102
5.2.4 Interpolation . . . 106
5.2.5 σ-Transform . . . 107
5.3 Concluding Remarks . . . 109

6 Decoding Interleaved Gabidulin Codes Using Row Reduction **111**
6.1 Implementing Known Decoding Algorithms Using Row Reduction . . . 112
6.1.1 Key-Equation-Based Decoding . . . 112
6.1.2 Interpolation-Based Decoding . . . 115
6.2 Row Reduction of Skew Polynomial Matrices . . . 117
6.2.1 The Mulders–Storjohann Algorithm over Skew Polynomials . . . 117
6.2.2 Orthogonality Defect and Cost of the Mulders–Storjohann Algorithm 119
6.2.3 A Divide-&-Conquer Variant: Alekhnovich's Algorithm . . . 123
6.3 Specialized Row Reduction Algorithms for the Decoding Problems . . . 127
6.3.1 Key-Equation Based Decoding: Rosenkilde's Demand-Driven Algorithm 127
6.3.2 Interpolation-Based Decoding: Weak Popov Walk . . . 130
6.4 Concluding Remarks . . . 132

7 Generalizations of Twisted Gabidulin Codes **135**
7.1 Definition . . . 135
7.2 A Sufficient Condition for Twisted Gabidulin Codes to be MRD . . . 136
7.3 A Suboptimal Decoder . . . 139
7.4 Inequivalence to Other MRD Codes . . . 143
7.5 Twisted Gabidulin Codes in the GPT Cryptosystem . . . 147
7.6 Concluding Remarks . . . 147

8 Conclusion **149**

A Appendices **151**
A.1 Efficiently Decoding Interleaved Reed–Solomon and One-Point Hermitian Codes 151
A.2 Proof of Theorem 4.21 (Schur Squares of Shortened Codes) . . . 154
A.3 Optimal Multiplication of Skew Polynomials of Degree m . . . 156
A.3.1 Relation of Skew Polynomial and Matrix Multiplication . . . 156
A.3.2 Faster Implementation of a Known Multiplication Algorithm . . . 157
A.3.3 Optimality of the Multiplication Algorithm . . . 158
A.4 Key Equations Solvable with the MgLSSR Problem . . . 159

B List of Symbols and Acronyms **161**

Bibliography **167**

1

Introduction

Coding Theory has its roots in the pioneering publications by Shannon [Sha48] and Hamming [Ham50]. Shannon developed a mathematical theory that formalized information transmission through communication channels and gave limits on how much redundancy must be added to information in order to transmit it error-free asymptotically. Though non-constructive, his statements lead to fundamental limits of communication systems, whose practical achievability has been studied since then. Hamming proposed one of the first constructive classes of error-correcting codes and introduced the *Hamming distance*, a metric that measures the distance of codewords by counting the number of positions in which they disagree. Within the last 70 years, his findings have resulted in a rich theory of codes and metrics tailored to different communication channels.

In this dissertation, we consider the problem of constructing and decoding codes in two metrics: Hamming and rank metric. All considered codes are algebraic block codes, which are sets of vectors of fixed length over a—typically finite—field. Furthermore, they are evaluation codes, which are defined by evaluating certain polynomials. When constructing block codes, one usually aims at finding codes of large cardinality whose codewords are pairwise far apart in a given metric, i.e., they have a large *minimum distance*. Motivated by Shannon's asymptotic statements, longer codes are often preferred over short codes and many applications require a small field size relative to the length. *Decoding* means to find close codewords to a received word with respect to a given metric, e.g., in order to correct errors introduced by the channel. Generally, the aim is to find decoding algorithms that are fast and can correct many errors, i.e., have a large *decoding radius*.

The Hamming metric is the natural one for many communication channels and the codes designed for this metric are widely used. Among these codes are *Reed–Solomon (RS)* codes [RS60], which arise from evaluating low-degree univariate polynomials and are *maximum distance separable (MDS)*, i.e., they attain Singleton's upper bound [Sin64] on the minimum distance with equality. Their applications range from satellite communication [WHPH87], error-correction in CDs/DVDs/BluRays [HTV82], RAID-6 [Anv04], QR codes [OM11], to modern storage applications (see e.g. the overview in [DDKM16, Section I.B]). RS codes can be decoded beyond half the minimum distance in polynomial time, cf. [GS98, Wu08, SSB10, Ros18]. Furthermore, the problem of fast decoding of RS codes is solved asymptotically: Using well-known methods from computational algebra, most RS decoders can be implemented in quasi-linear time in the code length, see, e.g., [Jus76, CJN+15, NRS17].

The length of RS codes, and conjecturally all non-trivial MDS codes (cf. [Seg55]), is upper-bounded by their field size plus a small constant. In the last decades, many inequivalent

constructions of MDS codes of length in the order of their field size have been found, see, e.g., [MS77, RL89a, RS85].

One-point Hermitian codes are, besides RS codes, one of the most-studied class of *algebraic geometry (AG)* codes [Gop83]. By evaluating certain bivariate polynomials at points on the Hermitian curve, they achieve larger lengths compared to RS codes at the cost of a slightly reduced minimum distance. These codes can also be decoded beyond half their minimum distance [GS98, HN99, Kam14, NB15], and the corresponding algorithms can be implemented with sub-quadratic time in the code length, cf. [NB15].

Interleaved codes are direct sums of codes of the same length and, when designed for the Hamming metric, are used to correct *burst errors*, i.e., errors that occur at the same positions in the constituent codewords. Such errors occur, e.g., in data storage applications (see, e.g., [KL97]) or when decoding concatenated codes, cf. [KL98, JTH04, SSB05, SSB09]. In the last decades, many algorithms for decoding interleaved Reed–Solomon codes have been proposed [KL97, BKY03, CS03, PV04, BMS04, Par07, SSB07, SSB09, CH13, WZB14]. The largest decoding radius is achieved by [PV04, CH13]. For interleaved one-point Hermitian codes, there are comparably fewer decoding algorithms [BMS05, Kam14].

The second metric that we consider is the rank metric, which was first considered in coding theory by Delsarte [Del78], Gabidulin [Gab85], and Roth [Rot91]. Codes in this metric can be interpreted as sets of matrices whose distance is determined by the rank of their difference. The codes have applications in network coding [KK08, SKK08], cryptography [GPT91, FL06, Ove08, Loi10, Loi16, WPR18], distributed storage systems [SRV12], construction and decoding of space-time codes [GBL00, BGL02, LFT02, LGB03, Rob15a, PSBF16], and low-rank matrix recovery [FS12, MPB17b].

The first and most prominent representatives of this code class are *Gabidulin codes* [Del78, Gab85, Rot91], which are defined by evaluating so-called *skew polynomials* of low degree and can be seen as the rank-metric analog of Reed–Solomon codes. Gabidulin codes are *maximum rank distance (MRD)* codes, i.e., they fulfill the rank-metric Singleton bound with equality. It was recently shown by Raviv and Wachter-Zeh [RW15, RW16] that certain classes of Gabidulin codes cannot be list decoded at any radius beyond half the minimum distance, which means that a general polynomial-time list decoder—analog to the Guruswami–Sudan or Wu list decoder for RS codes—does not exist for these codes. All existing algorithms for decoding up to half the minimum distance [Gab85, Rot91, Gab91, PT91, RP04a, RP04b, SKK08, SK09, GY08, SRB11, WAS13, Wac13, SB14] have at least quadratic runtime in the code length, and, despite several advances accelerating parts of the algorithms [SB14, SK09, HS10, WAS13, Wac13], it has been an open problem whether Gabidulin codes can be decoded in sub-quadratic time.

Besides Gabidulin codes, only a few classes of MRD codes, and rank-metric codes in general, are known. Recently, Sheekey [She16] presented *twisted Gabidulin codes*, a class of MRD codes that are inequivalent to Gabidulin codes, and which arise from the latter by slightly modifying the set of evaluation polynomials.

Interleaved Gabidulin codes were introduced by Loidreau and Overbeck [LO06] for cryptographic applications and can be used for overhead reduction in network coding, cf. [SKK08]. By considering synchronized rank errors, analog to burst errors in Hamming metric, one can decode beyond half the minimum distance with high probability [LO06, SB10, SJB11, WZ13, Wac13].

Outline

In **Chapter 2**, we give an introduction to the known methods and codes that we consider in this dissertation. We first recall fundamental mathematical and coding-theory-specific notions. Then, we discuss codes in Hamming metric, more specifically Reed–Solomon, one-point Hermitian codes, and interleaved codes thereof. Finally, we recall properties of the rank metric and skew polynomial rings. We give the definition of Gabidulin, interleaved Gabidulin and twisted Gabidulin codes as evaluation codes of these polynomials.

The remaining chapters contain new results. Since most of these findings are already published, we give the corresponding references at the end of each chapter introduction. The thesis is formally divided into two parts: Chapters 3 and 4 deal with codes in Hamming metric and Chapters 5, 6, and 7 contain results on rank-metric codes. Despite this separation, the parts are closely related, as mentioned in detail below.

In **Chapter 3**, we present new decoding algorithms for interleaved Reed–Solomon and interleaved one-point Hermitian codes. Simulation results indicate that the algorithms succeed for most errors up to the derived maximal decoding radius, but—as all comparable algorithms—fail for some errors beyond the unique decoding radius. The approach combines the ideas of power decoding of RS [SSB06, SSB10] and one-point Hermitian codes [Kam14, NB15], respectively, with collaborative decoding of interleaved codes [SSB09, SSB07, WZB14], and the recently developed improved power decoding principle [Ros18]. For interleaved RS codes, we obtain the same maximal decoding radius as [PV04, CH13], but at a better complexity. The maximal decoding radius of our decoder for interleaved one-point Hermitian codes improves upon the previous best one [BMS05, Kam14] at all rates.

Chapter 4 introduces a new code class, *twisted Reed–Solomon codes*, which is motivated by Sheekey's twisted Gabidulin codes [She16], utilizing the similarity of RS and Gabidulin codes. Compared to Sheekey's codes, we significantly generalize the construction based on modifying the evaluation polynomials. We also use different methods for their analysis. First, we derive a sufficient condition for the codes to be MDS. Although this yields short codes in general, we show in the subsequent sections that subclasses of longer (half the field size) MDS twisted RS codes exist. Our decoding algorithm makes use of well-known RS decoders, or in some cases even decoders for interleaved RS codes, where we can utilize the algorithm from Chapter 3. We present a subclass of twisted RS codes that are closed under duality, and show that many low-rate codes have much larger Schur square dimension than RS codes. The latter observation proves the inequivalence of these codes to RS codes, which we also study from a combinatorial perspective. Finally, we analyze the new codes in the *McEliece cryptosystem*, a candidate for post-quantum public-key cryptography based on coding theory. We show that subclasses of twisted RS codes resist some known structural attacks on variants of the system based on RS codes.

We consider fast decoding of Gabidulin codes in **Chapter 5**. First, we recall the algorithm by Wachter-Zeh et al. [WAS13] for decoding up to half the minimum distance, phrase it in skew polynomial language, and identify the skew polynomial operations that determine the algorithm's runtime. Inspired by well-known fast algorithms for ordinary polynomials, which are, for instance, used for fast decoding of RS codes, we prove that all these operations can be implemented in sub-quadratic time in the polynomials' degrees. This results in the first decoder for Gabidulin codes with sub-quadratic complexity in the code length.

Chapter 6 shows that the core steps of several known algorithms for decoding interleaved

Gabidulin codes can be solved by row reduction of skew polynomial matrices. The approach is inspired by the recent unification of various decoding algorithms for decoding RS and one-point Hermitian codes [Ale05, LO08, BB10, CH10, Nie13a, Nie13b, Nie14, NB15, Ros18] (cf. the overview in [Nie13b]) which also forms the algorithmic basis of the algorithms in Chapter 3. Then, we adapt well-known row reduction algorithms for matrices over ordinary polynomial rings to skew polynomial matrices and analyze their runtime. In this way, the core steps of many decoding algorithms for interleaved Gabidulin codes become flexible, conceptually simple, and in some cases faster.

In **Chapter 7**, we use our results on twisted RS codes to widely generalize Sheekey's construction of twisted Gabidulin codes. We use similar methods as in Chapter 4 to derive a sufficient condition for the codes to be MRD. Furthermore, we show that many of the new codes are inequivalent to both Gabidulin and Sheekey's twisted Gabidulin codes.

The thesis is concluded in **Chapter 8**.

2

Preliminaries

BLOCK CODES are subsets of finite-dimensional vector spaces whose elements' distance is measured with respect to a given metric. In this section, we give a brief overview of definitions and properties of several codes. Although they are considered in two different metrics, *Hamming* and *rank metric*, they share many fundamental properties and methods for analyzing and decoding them.

We introduce basic notation in Section 2.1, recall Hamming-metric codes and their properties in Section 2.2, and provide an overview of rank-metric codes in Section 2.3.

2.1 Basic Terminology

2.1.1 Block Codes

Coding theory commonly considers two general classes of codes: Infinite and finite length codes. The latter ones, also called *block codes*, were first studied by Golay [Gol49] and Hamming [Ham50], and all codes in this thesis belong to this family. They are defined as follows. Let $\mathbb{F}$ be a field, n a positive integer, and $\mathrm{d}(\cdot,\cdot) : \mathbb{F}^n \times \mathbb{F}^n \to \mathbb{R}_{\geq 0}$ a metric[1] on $\mathbb{F}^n$.

Definition 2.1. A *(block) code* $\mathcal{C}(n, M, d)$ over $\mathbb{F}$ of *length* n, *cardinality* M, and *minimum distance* d with respect to the metric $\mathrm{d}(\cdot,\cdot)$ is a subset $\mathcal{C} \subseteq \mathbb{F}^n$ of $M = |\mathcal{C}|$ elements that fulfill

$$d = \min_{\substack{\boldsymbol{c}_1, \boldsymbol{c}_2 \in \mathcal{C} \\ \boldsymbol{c}_1 \neq \boldsymbol{c}_2}} \mathrm{d}(\boldsymbol{c}_1, \boldsymbol{c}_2).$$

The elements of a code are called *codewords*. A code is called linear if it is a vector space over its base field $\mathbb{F}$. In this case, the *dimension* k of the code is defined by the dimension of $\mathcal{C}$ as an $\mathbb{F}$-vector space and we write $\mathcal{C}[n, k, d]$ or $\mathcal{C}[n, k]$. A linear code has *rate* $R = \frac{k}{n}$.

A *generator matrix* $\boldsymbol{G}$ of a linear code $\mathcal{C}$ is a basis of the code, written as the rows of $\boldsymbol{G}$. The *dual code* $\mathcal{C}^\perp$ of $\mathcal{C}$ is the orthogonal complement of $\mathcal{C}$, i.e., the set of vectors $\boldsymbol{y} \in \mathbb{F}^n$ such that $\boldsymbol{c} \cdot \boldsymbol{y}^\top = 0$ for all $\boldsymbol{c} \in \mathcal{C}$. A *parity-check matrix* $\boldsymbol{H}$ is a generator matrix of the dual code.

If a metric is *translation invariant*, i.e., $\mathrm{d}(\boldsymbol{x}, \boldsymbol{y}) = \mathrm{d}(\boldsymbol{x} + \boldsymbol{z}, \boldsymbol{y} + \boldsymbol{z})$ for all $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z} \in \mathbb{F}^n$, the minimum distance of a linear code is determined by the minimum distance of all non-zero codewords to the zero codeword, i.e., $d = \min_{\boldsymbol{c} \in \mathcal{C} \setminus \{\boldsymbol{0}\}} \mathrm{d}(\boldsymbol{c}, \boldsymbol{0})$.

All codes in this thesis are *evaluation codes*, which are codes of the following form.

Definition 2.2. Let R be a polynomial ring[2] in which each polynomial $f \in R$ has an

[1] i.e., $\mathrm{d}(\boldsymbol{x}, \boldsymbol{y}) \geq 0$, $\mathrm{d}(\boldsymbol{x}, \boldsymbol{y}) = 0$ iff $\boldsymbol{x} = \boldsymbol{y}$, $\mathrm{d}(\boldsymbol{x}, \boldsymbol{y}) = \mathrm{d}(\boldsymbol{y}, \boldsymbol{x})$, and $\mathrm{d}(\boldsymbol{x}, \boldsymbol{z}) \leq \mathrm{d}(\boldsymbol{x}, \boldsymbol{y}) + \mathrm{d}(\boldsymbol{y}, \boldsymbol{z})$ for all $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z} \in \mathbb{F}^n$.

[2] E.g., a univariate or multivariate ordinary polynomial ring, or a skew polynomial ring (cf. Section 2.3.1)

evaluation map $f(\cdot) : D \to K$, where D is the domain of the map and K is a field. Fix elements $\alpha_1, \ldots, \alpha_n \in D$ (*evaluation points*) and a set of polynomials $\mathcal{P} \subseteq R$ (*evaluation polynomials*). The corresponding *evaluation code* is defined by

$$\mathcal{C} = \mathrm{ev}_{[\alpha_1,\ldots,\alpha_n]}(\mathcal{P}) := \{[f(\alpha_1), \ldots, f(\alpha_n)] : f \in \mathcal{P}\} \subseteq K^n.$$

A comprehensive introduction to error-correcting codes can be found in many textbooks on coding theory, e.g., [Ber15, Gal68, PW72, MS77, CJC13, VL12, Bla83, Bos13, JH04, Rot06].

2.1.2 Notation

Vectors and Matrices

Vectors $\boldsymbol{v}$ and matrices $\boldsymbol{V}$ are denoted by bold letters, where matrices are usually capital. We use square brackets to surround elements of matrices, vectors, or tuples. Unless stated otherwise, vectors are indexed starting from 1, i.e., $\boldsymbol{v} = [v_1, \ldots, v_n]$. Also, vectors are usually considered to be row vectors. The identity matrix is denoted by $\boldsymbol{I}$.

Computational Complexity

We measure the asymptotic running time of algorithms, depending on their input size n, using the Bachmann–Landau notation, i.e., $O(f(n))$ for upper, $\Theta(f(n))$ for tight, and $\Omega(f(n))$ for lower bounds, where f is a function of the input size. If the symbol is accompanied by a tilde, then logarithmic terms are omitted, e.g., $n \log^2(n) \in O^{\sim}(n)$.

For resolving recursive complexity expressions $T(n)$ of the form $T(n) = aT(n/b) + f(n)$, where $a \geq 1$, $b > 1$, and a function f, we use the master theorem, cf. [CLR$^+$01, Theorem 4.1].

If an algorithm relies on the cost of multiplying two $n \times n$ matrices over a field, then we usually express its complexity in terms of the *matrix multiplication exponent* $\omega \in [2, 3]$, which corresponds to the asymptotic number of field operations $O(n^\omega)$ required by the used matrix multiplication algorithm.

We express the cost of algorithms corresponding to a code (e.g., a decoder) as a function of its length n. Asymptotically, we assume that the dimension k is proportional to the length, which corresponds to a fixed code rate. Hence, we have $k, n - k \in \Theta(n)$, which simplifies asymptotic complexity expressions. For codes of small ($k \in o(n)$) or large ($n - k \in o(n)$) rates, the true cost of an algorithm might be smaller than indicated by this simplification.

Algebra

We extensively deal with algebraic objects, such as *rings*, *fields*, or *vector spaces*. We give a brief notational overview here and refer to fundamental textbooks for a detailed introduction, e.g., [Art91, LN97].

Most of the fields considered in this thesis are *finite*. The cardinality of a finite field $\mathbb{F}_q$ is a prime power q and that all finite fields of the same size are isomorphic.

Let K and L be fields. If K is a *subfield* of L, denoted by $K \leq L$, we call L/K a *field extension*. It is well-known that L is a K-vector space. The *extension degree* $[L : K]$ of a field extension is defined by the dimension of L as a vector space over K. For instance, $\mathbb{F}_{q^m}/\mathbb{F}_q$ has extension degree m and $\mathbb{F}_{q^m}$ is an m-dimensional vector space over $\mathbb{F}_q$.

The *automorphism group* $\mathrm{Aut}(L/K)$ of a field extension is the set of *automorphisms*[3] $\phi : L \to L$, that fix the base field K, i.e., $\phi(\alpha) = \alpha$ for all $\alpha \in K$, equipped with composition $\circ$ as the group operation. An extension L/K of finite degree is called *Galois extension* if the extension degree equals the cardinality of the automorphism group. In this case, the automorphism group is called *Galois group* and denoted by $\mathrm{Gal}(L/K) = \mathrm{Aut}(L/K)$. For $L = \mathbb{F}_{q^m}$ and $K = \mathbb{F}_q$, the Galois group consists of all powers $\cdot^{q^i} = (\cdot^q) \circ \cdots \circ (\cdot^q)$ of the *Frobenius automorphism* $\cdot^q : \mathbb{F}_{q^m} \to \mathbb{F}_{q^m}$, $\alpha \mapsto \alpha^q$. Also, it is cyclic and the generators are the automorphisms $\cdot^{q^i}$ with $\gcd(i, m) = 1$.

For any finite-degree Galois extension L/K, there is a so-called *normal basis* of the form $\{\sigma(\beta) : \sigma \in \mathrm{Gal}(L/K)\}$, where $\beta \in L$, of L as a vector space over K. For $\mathbb{F}_{q^m}/\mathbb{F}_q$, such a basis is thus of the form $\{\beta, \beta^q, \ldots, \beta^{q^i}\} = \{\sigma^0(\beta), \sigma^1(\beta), \ldots, \sigma^{m-1}(\beta)\}$, where $\beta \in \mathbb{F}_{q^m}$ and σ is any generator of the Galois group. Elements of $\mathbb{F}_{q^m}$ can be multiplied, added, and raised to the $(q^i)^{\text{th}}$ power for any i in $O^{\sim}(m)$ operations over $\mathbb{F}_q$ using the bases described in [CL09].

By $K[x]$, we denote the univariate polynomial ring over a field K in the indeterminate x. Likewise, $K[x_1, \ldots, x_n]$ is the multivariate polynomial ring over K in n unknowns.

Decoding Principles

In this thesis, we discuss decoding algorithms for codes in different metrics $\mathrm{d}(\cdot, \cdot)$, where for a given *received word* $\boldsymbol{r} \in \mathbb{F}^n$, we aim at finding codewords $\boldsymbol{c} \in \mathcal{C}$ that have small distance $\mathrm{d}(\boldsymbol{r}, \boldsymbol{c})$ to $\boldsymbol{r}$. We distinguish the following two types of decoders of *decoding radius* $\tau \in \mathbb{N}$. By *relative decoding radius*, we denote the value $\frac{\tau}{n}$.

List decoder The decoder returns all codewords of distance at most τ to the received word.

Unique decoder The decoder returns a codeword of minimal distance to the received word among those codewords $\boldsymbol{c}$ with $\mathrm{d}(\boldsymbol{r}, \boldsymbol{c}) \leq \tau$.

If the decoding radius τ is less than half the minimum distance $\frac{d}{2}$ of a code, there is at most one codeword of distance $\mathrm{d}(\boldsymbol{r}, \boldsymbol{c}) \leq \tau$, so both decoding principles coincide. If $\tau = \lfloor \frac{d-1}{2} \rfloor$, a decoder is called *bounded minimum distance (BMD)* decoder. Furthermore, we say that a decoder is *partial*[4] if it does not work for all received words.

2.2 Codes in Hamming Metric

The Hamming metric is presumably the most common metric considered in coding theory. It counts the number of components in which two vectors differ.

Definition 2.3 ([Ham50])**.** The *Hamming distance* of two vectors in $\mathbb{F}_q^n$ is defined by

$$\begin{aligned} \mathrm{d_H} : \mathbb{F}_q^n \times \mathbb{F}_q^n &\to \mathbb{N}_0, \\ [\boldsymbol{x}, \boldsymbol{y}] &\mapsto |\{i : x_i \neq y_i\}|. \end{aligned}$$

The *Hamming weight* $\mathrm{wt_H}(\boldsymbol{x})$ of a vector $\boldsymbol{x} \in \mathbb{F}_q^n$ is the Hamming distance to the all-zero vector $\mathbf{0}$, i.e., $\mathrm{wt_H}(\boldsymbol{x}) = \mathrm{d_H}(\boldsymbol{x}, \mathbf{0})$.

[3] I.e., bijective mappings $\phi : L \to L$ with $\phi(ab + cd) = \phi(a)\phi(b) + \phi(c)\phi(d)$ for all $a, b, c, d \in L$

[4] In order to avoid confusion with randomized algorithms, the name *partial decoder* was coined in [Ros18] as a replacement for *probabilistic decoder*, which was used for the same purpose, e.g., in [SSB06, SSB10].

The Hamming distance is a metric. In the following, we consider codes whose codewords' distance is measured with respect to this metric. It is translation invariant, so the minimum distance of a linear code is determined by the minimal Hamming weight of the non-zero codewords. It can be upper-bounded by the code parameters as follows.

Theorem 2.4 (Singleton Bound [Sin64])**.** *The minimum Hamming distance of a linear code $\mathcal{C}[n,k,d]$ fulfills*

$$d \leq n - k + 1.$$

Definition 2.5 ([Sin64])**.** A linear code $\mathcal{C}[n,k,d]$ that fulfills $d = n-k+1$ is called *maximum distance separable (MDS).*

MDS codes attain the maximal possible minimum distance for a given length n and dimension k. However, there are not many known codes with this property, and all of them are relatively short compared to their field size. In fact, it is conjectured (*MDS conjecture*, cf. [Seg55]) that an MDS code over a field $\mathbb{F}_q$ of dimension $k < q$ must satisfy $n \leq q+2$ if q is even and $k \in \{3, q-1\}$, and $n \leq q+1$ otherwise.

We will define a new class of Hamming-metric codes in Chapter 4. It is important to study whether these codes are in some sense equivalent to existing codes. The semi-linear Hamming-metric isometries on $\mathbb{F}_q$ (i.e., semi-linear maps[5] that preserve the distance properties) are given by permuting positions, element-wise multiplication with non-zero field elements, and element-wise application of an automorphism in $\mathrm{Gal}(\mathbb{F}_q/\mathbb{F}_p)$, where p is the characteristic of $\mathbb{F}_q$. Based on this observation, we define equivalence w.r.t. the Hamming metric as follows.

Definition 2.6 ([Huf98])**.** Two codes $\mathcal{C}, \mathcal{C}' \subseteq \mathbb{F}_q^n$ are *(semi-linearly) equivalent (w.r.t. the Hamming metric)* if there is a permutation $\pi \in S_n$, a vector $\boldsymbol{v} \in \left(\mathbb{F}_q^*\right)^n$, and an automorphism $\sigma \in \mathrm{Gal}(\mathbb{F}_q/\mathbb{F}_p)$, where p is the characteristic of $\mathbb{F}_q$, such that $\mathcal{C}' = \varphi_{\pi,\boldsymbol{v},\sigma}(\mathcal{C})$, where $\varphi_{\pi,\boldsymbol{v},\sigma}$ is defined by

$$\begin{aligned} \varphi_{\pi,\boldsymbol{v},\sigma} &: \mathbb{F}_q^n \to \mathbb{F}_q^n, \\ [c_1, \ldots, c_n] &\mapsto \left[v_1 \sigma\left(c_{\pi(1)}\right), \ldots, v_n \sigma\left(c_{\pi(n)}\right)\right]. \end{aligned}$$

In this thesis, we discuss several codes in the Hamming metric: Besides introducing a new family of MDS codes, we consider Reed–Solomon and one-point Hermitian codes and their interleaved variants, which are defined as follows.

2.2.1 Reed–Solomon Codes

Reed–Solomon codes [RS60] were invented in 1960 and have become one of the most studied and used class of algebraic codes. They have found many applications, such as satellite communication [WHPH87], error-correction in CDs/DVDs/BluRays [HTV82], RAID-6 [Anv04], QR codes [OM11], and modern storage applications (see e.g. the overview in [DDKM16, Section I.B]). The codes are formally defined as follows.

[5] A map $f : \mathbb{F}_q^n \to \mathbb{F}_q^n$ is *semi-linear* if there is a $\sigma \in \mathrm{Gal}(\mathbb{F}_q/\mathbb{F}_p)$, where p is the characteristic of $\mathbb{F}_q$, such that $f(\boldsymbol{x}+\boldsymbol{y}) = f(\boldsymbol{x}) + f(\boldsymbol{y})$ and $f(\alpha\boldsymbol{x}) = \sigma(\alpha) f(\boldsymbol{x})$ for all $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{F}_q^n$ and $\alpha \in \mathbb{F}_q$, cf. [Huf98]

Definition 2.7 ([RS60])**.** Let $k \leq n$ and $\alpha_1, \ldots, \alpha_n \in \mathbb{F}_q$ be distinct. The corresponding *Reed–Solomon (RS) code* is defined by[6]

$$\mathcal{C}_{\mathrm{RS}}[n,k] = \{\boldsymbol{c} = [f(\alpha_1), \ldots, f(\alpha_n)] \,:\, f \in \mathbb{F}_q[x], \deg f < k\} \subseteq \mathbb{F}_q^n.$$

We call the α_i the *evaluation points* and f the *message polynomial* of the codeword $\boldsymbol{c}$. A code is called *generalized Reed–Solomon (GRS) code* if it is equivalent to an RS code.

Theorem 2.8 ([RS60])**.** *An RS code $\mathcal{C}_{\mathrm{RS}}[n,k]$ is MDS.*

Proof. Since $\mathcal{C}_{\mathrm{RS}}[n,k]$ is a linear code and fulfills the Singleton bound, it suffices to show that the minimum weight of a non-zero codeword $\boldsymbol{c}$ is at least $n-k+1$. Any non-zero message polynomial has degree at most $k-1$, and, as a consequence of Bézout's theorem, has at most $k-1$ zeros. Since the α_i are distinct, at least $n-(k-1)$ entries of $\boldsymbol{c}$ are non-zero, which implies the claim. □

Note that a Reed–Solomon code can have length at most $n \leq q$. It is possible to extend an RS code to length $n = q+1$ without violating the MDS property by adding an evaluation point at infinity as follows: $f(\infty) := f_{k-1}$ for all $f = \sum_{i=0}^{k-1} f_i x^i \in \mathbb{F}_q[x]$.

2.2.2 One-Point Hermitian Codes

One-point Hermitian codes belong to the class of algebraic geometry (AG) codes, which was introduced by Goppa [Gop83]. Their advantage compared to RS codes, which are also AG codes, is that they achieve greater lengths $n = q^3$ compared to the underlying field size q^2. This improvement comes at the cost of a reduced minimum distance $d \geq n-k+1-g$, where g is the genus of the Hermitian curve. In the following, we summarize properties of this curve using the notation of [HvLP98], [Bra10], and [NB15]. A comprehensive introduction to the topic is contained in the textbooks [HvLP98, Sti09].

The Hermitian Curve

The *Hermitian curve* $\mathcal{H}/\mathbb{F}_{q^2}$ is the smooth projective plane curve given by the affine equation

$$H = y^q + y - x^{q+1} = 0.$$

The curve has *genus* $g = \frac{1}{2}q(q-1)$ and q^3+1 many $\mathbb{F}_{q^2}$-rational points $\mathcal{H} = \mathcal{H}^* \cup \{P_\infty\}$. Here, P_∞ denotes the *point at infinity* and the remaining points $\mathcal{H}^* = \{P_1, \ldots, P_{q^3}\}$ are—in affine coordinates—of the form $[\alpha, \beta]$, where for each $\alpha \in \mathbb{F}_{q^2}$, there are exactly q many $\beta \in \mathbb{F}_{q^2}$ such that $[\alpha, \beta] \in \mathcal{H}$.

By $\mathbb{F}_{q^2}(\mathcal{H})$, we denote the *function field* of the curve. A *local parameter* t_P in $P \in \mathcal{H}$ is a generator of the unique maximal ideal of the *local ring* $\mathcal{O}_P$ consisting of all *rational functions* $f \in \mathbb{F}_{q^2}(\mathcal{H})$ that are regular in P. For $P_i = [\alpha, \beta] \in \mathcal{H}^*$, a local parameter is given by $t_{P_i} = x - \alpha$ and for the point at infinity, we can use $t_{P_\infty} = x/y$. For $P \in \mathcal{H}$, the mapping $v_P : \mathbb{F}_{q^2}(\mathcal{H}) \to \mathbb{Z} \cup \{\infty\}$ is defined such that $v_P(0) = \infty$ and for each non-zero rational function $f \in \mathbb{F}_{q^2}(\mathcal{H})^*$, $v_P(f)$ is the unique integer such that there is a unit u of $\mathcal{O}_P$ with $f = u t_P^{v_P(f)}$. The definition is independent of the chosen local parameter and v_P is a

[6] I.e., $= \mathrm{ev}_{[\alpha_1,\ldots,\alpha_n]}(\mathbb{F}_q[x]_{<k})$ in evaluation-code notation.

discrete valuation (cf. [HvLP98]). Furthermore, any $f \in \mathbb{F}_{q^2}(\mathcal{H})^*$ can be uniquely expanded into a Laurent series of the form $\sum_{i \geq v_P(f)} f_i t_P{}^i$, where $f_i \in \mathbb{F}_{q^2}$ and $f_{v_P(f)} \neq 0$.

A *divisor* D on $\mathcal{H}$ is a formal sum $D = \sum_{P \in \mathcal{H}} n_P P$ with $n_P \in \mathbb{Z}$. Its *degree* is given by $\deg(D) = \sum_{P \in \mathcal{H}} n_P$ and we write $D \succeq 0$ if $n_P \geq 0$ for all P. For a non-zero rational function $f \in \mathbb{F}_{q^2}(\mathcal{H})^*$, we define the divisor $\operatorname{div}(f) = \sum_{P \in \mathcal{H}} v_P(f) P$. The *Riemann–Roch space* of a divisor is the $\mathbb{F}_{q^2}$-vector space given by

$$\mathcal{L}(D) = \left\{ f \in \mathbb{F}_{q^2}(\mathcal{H})^* \,:\, \operatorname{div}(f) + D \succeq 0 \right\} \cup \{0\}.$$

If the divisor D has negative degree $\deg D < 0$, then $\mathcal{L}(D) = \{0\}$. A central result in algebraic geometry is the Riemann–Roch theorem (cf. [HvLP98, Section 2.6]), which for $\deg(D) > 2g-2$ implies that $\dim_{\mathbb{F}_{q^2}} \mathcal{L}(D) = \deg(D) - g + 1$.

As in [NB15], we use the notation $\mathcal{L}(D + \infty P_\infty) = \bigcup_{r \in \mathbb{N}_0} \mathcal{L}(D + rP_\infty)$, in particular the ring $\mathcal{R} := \mathcal{L}(\infty P_\infty)$ of functions that are regular in all points except for P_∞. Note that $\mathcal{R}$ is isomorphic to the set of bivariate polynomials in $\mathbb{F}_{q^2}[x, y]$ with y-degree $< q$ with ordinary addition and multiplication modulo the curve equation $y^q = x^{q+1} - y$ in order to guarantee the multiplicative closure. We therefore treat elements in $\mathcal{R}$ as such polynomials. A basis of $\mathcal{R}$ as an $\mathbb{F}_{q^2}$-vector space is given by $B = \{x^i y^j \,:\, 0 \leq i,\, 0 \leq j < q\}$. It was shown in [Bra10, Proposition 2.2] that for non-zero rational functions $f \in \mathbb{F}_{q^2}(\mathcal{H})$, we have

$$\mathcal{L}(-\operatorname{div}(f) + \infty P_\infty) = f\mathcal{R}. \tag{2.1}$$

We define the *degree* on $\mathcal{R}$ as

$$\begin{aligned} \deg_{\mathcal{H}} \,:\, \mathcal{R} &\to \mathbb{N}_0 \cup \{-\infty\}, \\ f &\mapsto -v_{P_\infty}(f). \end{aligned}$$

By the properties of the Hermitian curve, we have $\deg_{\mathcal{H}}(x^i y^j) = iq + j(q+1)$, which extends to polynomials in $\mathcal{R}$ by taking the maximum of this value for all non-zero monomials. Note that $\deg_{\mathcal{H}}$ is injective on the basis B since $j < q$, and fulfills $\deg_{\mathcal{H}}(f+g) \leq \max\{\deg_{\mathcal{H}}(f), \deg_{\mathcal{H}}(g)\}$ and $\deg_{\mathcal{H}}(f \cdot g) = \deg_{\mathcal{H}}(f) + \deg_{\mathcal{H}}(g)$ for all $f, g \in \mathcal{R}$. We call a polynomial in $\mathcal{R}$ *monic* if the monomial of largest degree $\deg_{\mathcal{H}}$ has coefficient 1.

Using the above notation, the Riemann–Roch space $\mathcal{L}(rP_\infty)$ contains all polynomials $f \in \mathcal{R}$ of degree $\deg_{\mathcal{H}}(f) \leq r$. A basis of $\mathcal{L}(rP_\infty)$ is therefore given by the monomials $x^i y^j$ with $j < q$ and $\deg_{\mathcal{H}}(x^i y^j) \leq r$. The Riemann–Roch theorem implies that for non-negative integers $a < b$, there are at least $b - a - g$ many monomials of the form $x^i y^j \in \mathcal{R}$ with $j < q$ of degree $\deg_{\mathcal{H}}(x^i y^j) \in [a, b)$, and this bound is attained with equality if all *Weierstraß gaps*[7] are in $[a, b)$. Due to the injectivity of $\deg_{\mathcal{H}}$ on B, there are at most $b - a$ many such monomials.

Codes over the Hermitian Curve

One-point Hermitian codes are obtained by evaluating all polynomials in the Riemann–Roch space $\mathcal{L}(m_{\mathrm{H}} P_\infty)$, for some $m_{\mathrm{H}} \in \mathbb{N}$, at distinct $\mathbb{F}_{q^2}$-rational points on $\mathcal{H}^* = \mathcal{H} \setminus \{P_\infty\}$.

[7] Here, these are the non-negative integers that cannot be expressed as $iq + j(q+1)$ with $i, j \in \mathbb{N}_0$ and $j < q$.

Definition 2.9. Let $P_1, \ldots, P_n \in \mathcal{H}^*$ be distinct, $g = \frac{1}{2}q(q-1)$ the genus of $\mathcal{H}$, and $m_{\mathrm{H}} \in \mathbb{N}$ with $2(g-1) < m_{\mathrm{H}} < n$. The *one-point Hermitian code* of length n and parameter m_{H} over $\mathbb{F}_{q^2}$ is defined by[8]

$$\mathcal{C}_{\mathrm{H}}^{(n,m_{\mathrm{H}})} = \{[f(P_1), \ldots, f(P_n)] : f \in \mathcal{L}(m_{\mathrm{H}} P_\infty)\} \subseteq \mathbb{F}_{q^2}^n.$$

The *designed minimum distance* of $\mathcal{C}_{\mathrm{H}}^{(n,m_{\mathrm{H}})}$ is $d^* := n - m_{\mathrm{H}}$.

Note that the maximal length of a one-point Hermitian code is $n = q^3$. The Riemann–Roch theorem implies the following statement.

Theorem 2.10. *The dimension of $\mathcal{C}_{\mathrm{H}}^{(n,m_{\mathrm{H}})}$ is given by $k = m_{\mathrm{H}} - g + 1$ and the minimum distance d is lower-bounded[9] by the designed minimum distance, i.e., $d \geq d^*$.*

2.2.3 Interleaved Codes in Hamming Metric

An h-interleaved code is a direct sum of h codes of the same length. Such codes were introduced to correct burst errors which occur, e.g., in data storage applications, see, e.g., [KL97]. In this thesis, we will consider interleaved codes, where the constituent codewords are from Reed–Solomon or one-point Hermitian codes, respectively. They are defined as follows.

Definition 2.11. Let $n, h \in \mathbb{N}$ and $\mathcal{C}^{(1)}[n, k_1], \ldots \mathcal{C}^{(h)}[n, k_h]$ be linear codes over $\mathbb{F}_q$.

i) The corresponding *interleaved code* is defined as the set

$$\mathcal{C}(n, k_1, \ldots, k_h; h) = \left\{ \boldsymbol{c} = \begin{bmatrix} \boldsymbol{c}_1 \\ \vdots \\ \boldsymbol{c}_h \end{bmatrix} \,:\, \boldsymbol{c}_i \in \mathcal{C}^{(i)}[n, k_i] \right\} \subseteq \mathbb{F}_q^{h \times n}.$$

An interleaved code is *homogeneous* if $k_1 = \cdots = k_h =: k$. In this case, we write $\mathcal{C}(n, k; h)$.

ii) If the constituent codes $\mathcal{C}^{(i)}[n, k_i]$ are Reed–Solomon codes with the same evaluation points $\alpha_1, \ldots, \alpha_n \in \mathbb{F}_q$, then we call the interleaved code an *(h-)interleaved Reed–Solomon (IRS) code* and denote it by $\mathcal{C}_{\mathrm{IRS}}(n, k_1, \ldots, k_h; h)$, and $\mathcal{C}_{\mathrm{IRS}}(n, k; h)$ in the homogeneous case.

iii) We call it *(h-)interleaved one-point Hermitian (IH) code*, $\mathcal{C}_{\mathrm{IH}}(n, m_{\mathrm{H}1}, \ldots, m_{\mathrm{H}h}; h)$ or $\mathcal{C}_{\mathrm{IH}}(n, m_{\mathrm{H}}; h)$, if the $\mathcal{C}^{(i)}$ are one-point Hermitian codes $\mathcal{C}_{\mathrm{H}}^{(n,m_{\mathrm{H}_i})}$ with the same evaluation points $P_1, \ldots, P_n \in \mathcal{H}^*$.

As error model, we consider column-wise burst errors, i.e., when the received word is $\boldsymbol{r} = \boldsymbol{c} + \boldsymbol{e} \in \mathbb{F}_q^{h \times n}$ then the *error positions* are given by the indices of the non-zero columns of $\boldsymbol{e}$, i.e.,

$$\mathcal{E} = \bigcup_{j=1}^{h} \{i \,:\, e_{ji} \neq 0\}.$$

We define the *number of errors* to be $|\mathcal{E}|$. The error model is illustrated in Figure 2.1.

[8] I.e., $\mathcal{C}_{\mathrm{H}}^{(n,m_{\mathrm{H}})} = \mathrm{ev}_{[P_1,\ldots,P_n]}(\mathcal{L}(m_{\mathrm{H}} P_\infty))$ in evaluation-code notation.
[9] Exact valued of d were determined in [Sti88, YK92].

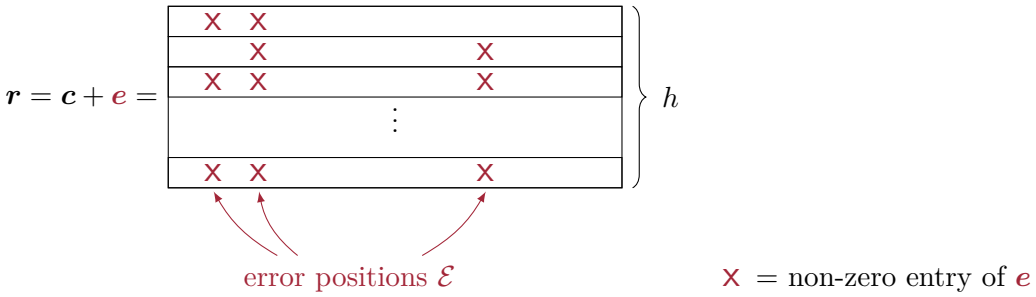


Figure 2.1: Illustration of the burst errors considered for interleaved codes in Hamming metric. In this particular example, 3 burst errors occurred.

Such burst errors occur, for instance, in data storage systems [KL97] or when decoding the outer code in concatenated codes [KL98, JTH04, SSB05, SSB09]. In general, an h-interleaved code of length n over $\mathbb{F}_q$ can be seen as a—not necessarily linear—code of length n over the field $\mathbb{F}_{q^h}$ by considering the columns of a codeword as elements in $\mathbb{F}_{q^h}$. A burst error then simply corresponds to a symbol error in $\mathbb{F}_{q^h}$. In case of a homogeneous interleaved RS code, we obtain an ($\mathbb{F}_{q^h}$-linear) RS code of the same evaluation points, cf. [SSB08].[10]

2.3 Codes in Rank Metric

Rank-metric codes can be interpreted as sets of matrices whose distance is determined by the rank of their difference. They were independently introduced by Delsarte [Del78], Gabidulin [Gab85], and Roth [Rot91], together with a constructive class of linear codes, nowadays called *Gabidulin codes*, achieving the maximal minimum distance in the rank metric for given length, dimension, and field size.

In the last years, there has been an increased interest in the codes due to their possible applications in network coding [KK08, SKK08], cryptography [GPT91, Ove08, Loi10, Loi16], distributed storage systems [SRV12], construction and decoding of space-time codes [GBL00, BGL02, LFT02, LGB03, Rob15a, PSBF16], and low-rank matrix recovery [FS12, MPB17b].

We fix a field extension $\mathbb{F}_{q^m}/\mathbb{F}_q$ of extension degree m. Since $\mathbb{F}_{q^m}$ is an m-dimensional $\mathbb{F}_q$-vector space, the *($\mathbb{F}_q$-)rank* of a vector in $\boldsymbol{x} = [x_1, \ldots, x_n] \in \mathbb{F}_{q^m}^n$ is defined by the dimension of the $\mathbb{F}_q$-span of its components, i.e.,

$$\operatorname{rank}(\boldsymbol{x}) = \dim_{\mathbb{F}_q} \langle x_1, \ldots, x_n \rangle.$$

Definition 2.12 ([Del78, Gab85, Rot91][11])**.** The *rank distance* in $\mathbb{F}_{q^m}^n$ over $\mathbb{F}_q$ is defined by

$$\begin{aligned} \mathrm{d_R} \,:\, \mathbb{F}_{q^m}^n \times \mathbb{F}_{q^m}^n &\to \mathbb{N}_0, \\ [\boldsymbol{x}, \boldsymbol{y}] &\mapsto \operatorname{rank}(\boldsymbol{x} - \boldsymbol{y}). \end{aligned}$$

The *rank weight* of a vector $\boldsymbol{x} \in \mathbb{F}_{q^m}^n$ is its distance to the zero vector, i.e., $\mathrm{wt_R}(\boldsymbol{x}) = \mathrm{d_R}(\boldsymbol{x}, \boldsymbol{0})$.

[10] Note that the evaluation points are in $\mathbb{F}_q$, so the code length is much smaller than the code's field size q^h.

[11] The rank metric was already studied under the name *arithmetic distance* by Hua in [Hua51]. Delsarte, Gabidulin, and Roth were the first to consider it in the context of coding theory.

Due to the properties of the rank, the rank distance is a metric. In the following, we consider codes over $\mathbb{F}_{q^m}$ in the rank metric with respect to the base field $\mathbb{F}_q$, also called *rank-metric codes*. In this case, the minimum distance of the code is also called *minimum rank distance*, and upper-bounded by the following rank-metric analog of the Singleton bound.

Theorem 2.13 ([Del78])**.** *The minimum rank distance of an (n, M, d) code over $\mathbb{F}_{q^m}$ fulfills*

$$d \leq \min\left\{n - \frac{\log_q M}{m} + 1, m - \frac{\log_q M}{n} + 1\right\}. \tag{2.2}$$

For linear codes of length $n \leq m$ and dimension $k \leq n$, this results in $d \leq n - k + 1$, see also [Gab85, Rot91].

Definition 2.14 ([Del78, Gab85, Rot91])**.** A rank-metric code satisfying the rank-metric Singleton bound (2.2) with equality is called *maximum rank distance (MRD) code*.

In Chapter 7, we will introduce a new class of rank-metric codes. We use the following definition for equivalence of rank-metric codes.

Definition 2.15 ([Ber03])**.** Two codes $\mathcal{C}, \mathcal{C}' \subseteq \mathbb{F}_{q^m}^n$ are *(semi-linearly) equivalent (w.r.t. to the rank metric on $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$)* if there are $\lambda \in \mathbb{F}_{q^m}^*$, $\boldsymbol{A} \in \mathbb{F}_q^{n\times n}$ invertible, and $\sigma \in \mathrm{Gal}(\mathbb{F}_{q^m}/\mathbb{F}_q)$ with

$$\mathcal{C}' = \sigma(\lambda\mathcal{C})\boldsymbol{A} := \{[\sigma(\lambda c_1), \ldots, \sigma(\lambda c_n)] \cdot \boldsymbol{A} \,:\, [c_1, \ldots, c_n] \in \mathcal{C}\}.$$

Matrix Representation of Rank-Metric Codes

Each field element $a \in \mathbb{F}_{q^m}$ can be uniquely represented as a vector $\mathrm{ext}_{\mathcal{B}}(a) \in \mathbb{F}_q^m$ with respect to an ordered basis $\mathcal{B}$ of $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$. In this way, we can represent a vector in $\boldsymbol{x} \in \mathbb{F}_{q^m}^n$ as a matrix in $\boldsymbol{X} := \mathrm{ext}_{\mathcal{B}}(\boldsymbol{x}) \in \mathbb{F}_{q^m}^{m\times n}$, where the i^{th} column of $\boldsymbol{X}$ is the vector representation $\mathrm{ext}_{\mathcal{B}}(x_i)$ of the i^{th} entry of $\boldsymbol{x}$. This representation is illustrated in Figure 2.2.

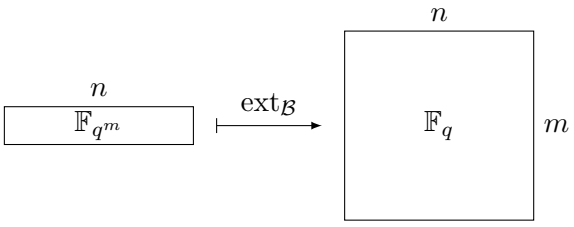


Figure 2.2: Matrix representation of a vector in $\mathbb{F}_{q^m}^n$ as a matrix in $\mathbb{F}_q^{m\times n}$.

The $\mathbb{F}_q$-rank of a vector $\boldsymbol{x}$ then equals the rank of its matrix representation $\boldsymbol{X}$, i.e.,

$$\mathrm{rank}(\boldsymbol{x}) = \mathrm{rank}(\boldsymbol{X}),$$

independent of the chosen basis $\mathcal{B}$. Hence, any rank-metric code can be interpreted as a set of $m \times n$ matrices over a finite field $\mathbb{F}_q$, where the distance of two codewords is measured by the rank of their difference.

2.3.1 Skew Polynomial Rings

All rank-metric codes considered in this thesis are evaluation codes of the non-commutative ring of skew polynomials, which was introduced by Ore in [Ore33b]. Besides their use in coding theory, these polynomials have applications in cryptography [FL06], dynamical systems [CH00] and are of theoretical interest [EGN+92, WL13]. Since skew polynomials are not widely known (e.g., they are usually not covered in lectures or textbooks), we recall their definition an properties thoroughly in the following.

Definition

Definition 2.16. Let $\mathbb{F}_{q^m}/\mathbb{F}_q$ be a field extension and $\sigma \in \mathrm{Gal}(\mathbb{F}_{q^m}/\mathbb{F}_q)$ be a generator of its Galois group.[12] The corresponding *skew polynomial ring*[13] $\mathbb{F}_{q^m}[x;\sigma]$ consists of polynomials of the form

$$f = \sum_{i=0}^{d} f_i x^i,$$

where $d \in \mathbb{N}_0$ and $f_i \in \mathbb{F}_{q^m}$. Addition is done component-wise as usual, whereas multiplication is defined by the non-commutative rule $x \cdot f_i = \sigma(f_i) \cdot x$, which—by the associative and distributive property—extends to polynomials $f, g \in \mathbb{F}_{q^m}[x;\sigma]$ as

$$f \cdot g = \sum_i \Big(\sum_{j=0}^{i} f_j \sigma^j(g_{i-j}) \Big) x^i. \tag{2.3}$$

Similar to ordinary polynomials, the *degree* of a skew polynomial $f \in \mathbb{F}_{q^m}[x;\sigma]$ is given by

$$\deg(f) = \begin{cases} \max\{i : f_i \neq 0\}, & \text{if } f \neq 0 \\ -\infty, & \text{if } f = 0. \end{cases}$$

We use the abbreviations $\mathbb{F}_{q^m}[x;\sigma]_{\leq s}$ for the set of skew polynomials over $\mathbb{F}_{q^m}$ of degree at most s, and $\mathbb{F}_{q^m}[x;\sigma]_{<s}$ analogously. The leading coefficient of a non-zero polynomial f is $\mathrm{LC}(f) = f_{\deg f}$. Similarly, its leading term is given by $\mathrm{LT}(f) = f_{\deg f} x^{\deg f}$. We call a non-zero skew polynomial *monic* if its leading coefficient is $\mathrm{LC}(f) = 1$.

Remark 2.17. *In the literature, rank-metric evaluation codes are often defined through linearized polynomials [Ore33a]. These are ordinary polynomials of the form $f = \sum_{i=0}^{d} f_i x^{q^i}$, where $d \in \mathbb{N}_0$ and $f_i \in \mathbb{F}_{q^m}$. Using composition of polynomials as multiplication, the polynomials form a non-commutative ring denoted by $\mathcal{L}_{q,m}$. This ring is isomorphic to the skew polynomial ring over $\mathbb{F}_{q^m}$ with the Frobenius automorphism $\cdot^q$, i.e., $\mathbb{F}_{q^m}[x;\cdot^q] \cong \mathcal{L}_{q,m}$. Hence, skew polynomials can be seen as a natural generalization of linearized polynomials.*

Skew polynomials can be further generalized using so-called derivations*, cf. [Ore33b]. In this form, they can also be used to construct evaluation codes [BU14], but we do not consider this extension in this thesis.*

[12] Some authors allow σ to be any automorphism in $\mathrm{Gal}(L/K)$. Hovewer, in this case, the polynomial evaluation map does not behave as well as here: E.g., Theorems 2.22, 2.23, and 2.24 below do not hold, cf. [ALR13]. Also, all known code constructions based on $\mathbb{F}_{q^m}[x;\sigma]$ require this property.

[13] In this form, skew polynomials are sometimes called *twisted polynomials* or *σ-polynomials*.

Division

Skew polynomials admit left and right division, which is formally stated in the following theorem. In contrast to their commutative relatives, the resulting right and left quotients and remainders do not necessarily agree.

Theorem 2.18 ([Ore33b])**.** *Let $a, b \in \mathbb{F}_{q^m}[x;\sigma]$ with $b \neq 0$. Then, there are unique skew polynomials $\chi_{\mathrm{R}}, \chi_{\mathrm{L}} \in \mathbb{F}_{q^m}[x;\sigma]$ (right and left quotient) and $\varrho_{\mathrm{R}}, \varrho_{\mathrm{L}} \in \mathbb{F}_{q^m}[x;\sigma]$ (right and left remainder) such that $\deg_q \varrho_{\mathrm{R}} < \deg_q b$, $\deg_q \varrho_{\mathrm{L}} < \deg_q b$, and*

$$\begin{aligned} a &= \chi_{\mathrm{R}} \cdot b + \varrho_{\mathrm{R}} \quad \textit{(right division)}, \\ a &= b \cdot \chi_{\mathrm{L}} + \varrho_{\mathrm{L}} \quad \textit{(left division)}. \end{aligned}$$

If the right (left) remainder is zero, we say that a is divisible by b on the right (left).

Definition 2.19. Let $a, b, c \in \mathbb{F}_{q^m}[x;\sigma]$, $c \neq 0$. If $a - b$ is divisible by c on the right, we write

$$a \equiv b \mod_{\mathrm{r}} c.$$

In the left case, we have $a \equiv b \mod_{\mathrm{l}} c$. The right and left modulo operators $a \bmod_{\mathrm{r}} c$ and $a \bmod_{\mathrm{l}} c$ denote the remainders of the right and left division a by c, respectively.

The right division implies the existence of the following objects.

Definition 2.20 ([Ore33b])**.** Let $f, g \in \mathbb{F}_{q^m}[x;\sigma]$.

i) The *right union* of f and g is the monic polynomial $a \in \mathbb{F}_{q^m}[x;\sigma]$ of smallest degree that is divisible on the right by both f and g. There are unique $b, c \in \mathbb{F}_{q^m}[x;\sigma]$ with

$$a = b \cdot f = c \cdot g.$$

ii) The *greatest common right divisor* $\operatorname{gcrd}(f, g)$ of f and g is the monic polynomial $a \in \mathbb{F}_{q^m}[x;\sigma]$ of largest degree that divides both f and g on the right. There are unique $b, c \in \mathbb{F}_{q^m}[x;\sigma]$ such that

$$a = b \cdot f + c \cdot g.$$

Evaluation Map

Like ordinary polynomials, skew polynomials admit an evaluation map.[14] Since all rank-metric codes considered in this thesis are evaluation codes of skew polynomials, we study this map thoroughly. It is formally defined by

$$\begin{aligned} f(\cdot) \,:\, \mathbb{F}_{q^m} &\to \mathbb{F}_{q^m}, \\ \alpha &\mapsto \textstyle\sum_i f_i \sigma^i(\alpha). \end{aligned}$$

The map is fundamentally different from $\mathbb{F}_{q^m}[x]$-evaluation maps in several aspects, which becomes apparent by the following well-known observations.

[14] There are at least two possible evaluation maps for skew polynomials, but here we consider only the so-called *operator evaluation*, cf. [BU14].

Theorem 2.21. *Let $f, g \in \mathbb{F}_{q^m}[x;\sigma]$ and $\alpha \in \mathbb{F}_{q^m}$. Then,*

i) $(f \cdot g)(\alpha) = f(g(\alpha))$.

ii) $f(\cdot)$ *is an* $\mathbb{F}_q$*-linear map.*

Proof. Ad i): By definition, we have

$$f(g(\alpha)) = \sum_{\mu} f_\mu \sigma^\mu \left(\sum_{\nu} g_\nu \sigma^\nu(\alpha) \right) = \sum_{\mu} \sum_{\nu} f_\mu \sigma^\mu(g_\nu) \sigma^{\mu+\nu}(\alpha)$$
$$\overset{(*)}{=} \sum_{i} \Big(\sum_{j=0}^{i} f_j \sigma^j(g_{i-j}) \Big) \sigma^i(\alpha) = (f \cdot g)(\alpha),$$

where $(*)$ is obtained by reordering the sum, combining all terms with $\mu + \nu = i$.

Ad ii): Since $\sigma^i(\cdot)$ is an $\mathbb{F}_q$-linear map for all $i \in \mathbb{N}_0$, this property extends to the evaluation map $f(\cdot)$, which is an $\mathbb{F}_{q^m}$-linear combination of finitely many $\sigma^i(\cdot)$. □

Using the evaluation map, we define the root space of a polynomial $f \in \mathbb{F}_{q^m}[x;\sigma]$ by

$$\operatorname{roots}(f) = \{\alpha \in \mathbb{F}_{q^m} \,:\, f(\alpha) = 0\}.$$

Due to Theorem 2.21, the root space is an $\mathbb{F}_q$-subspace of $\mathbb{F}_{q^m}$. The following property can be seen as the $\mathbb{F}_{q^m}[x;\sigma]$-analog of the fact that a non-zero ordinary polynomial of degree s has at most s roots.

Theorem 2.22 ([ALR13])**.** *Let $f \in \mathbb{F}_{q^m}[x;\sigma] \setminus \{0\}$. Then,*

$$\dim(\operatorname{roots}(f)) \leq \deg f.$$

We define some special polynomials which are defined through their evaluation map.

Theorem 2.23 ([LN97] ($\sigma = \cdot^q$) or [ALR13])**.** *Let $\mathcal{U}$ be a subspace of $\mathbb{F}_{q^m}$. There is a unique nonzero monic polynomial $\mathcal{M}_\mathcal{U} \in \mathbb{F}_{q^m}[x;\sigma]$ of minimal degree such that* $\operatorname{roots}(\mathcal{M}_\mathcal{U}) = \mathcal{U}$*. Its degree is* $\deg \mathcal{M}_\mathcal{U} = \dim \mathcal{U}$.

The polynomial $\mathcal{M}_\mathcal{U}$ in Theorem 2.23 is called *minimal subspace polynomial* (MSP) of $\mathcal{U}$.[15]

Evaluating one skew polynomial $f \in \mathbb{F}_{q^m}[x;\sigma]$ at multiple points $x_1, \ldots, x_s \in \mathbb{F}_{q^m}$ is called *multi-point evaluation (MPE)*. The dual problem to this is to find the polynomial of smallest degree that evaluates to some given values at fixed evaluation points, and is called *interpolation*. It is based on the following lemma.

Lemma 2.24 ([SK07, Sec. III-A] ($\sigma = \cdot^q$) or [Rob15b, Théorème 11])**.** *Let $[x_i, y_i] \in \mathbb{F}_{q^m}^2$ for $i = 1, \ldots, s$, where $x_1, \ldots, x_s$ are linearly independent over $\mathbb{F}_q$. There is a unique interpolation polynomial $\mathcal{I}_{\{[x_i,y_i]\}_{i=1}^s} \in \mathbb{F}_{q^m}[x;\sigma]_{<s}$ such that*

$$\mathcal{I}_{\{[x_i,y_i]\}_{i=1}^s}(x_i) = y_i, \quad \forall\, i = 1, \ldots, s.$$

[15] In the literature, the polynomial is sometimes called *annihilator polynomial*, e.g., in [ALR13].

In the case $s \mid m$ and the $x_i = \sigma^i(\beta)$ being an ordered normal basis of $\mathbb{F}_{q^s} \leq \mathbb{F}_{q^m}$, multi-point evaluation and interpolation is called the (inverse) σ-transform[16], which is formally defined as follows. Note that the inverse σ-transform exists due to Theorem 2.24.

Definition 2.25. Let $\mathcal{B} = [\sigma^0(\beta), \dots, \sigma^{s-1}(\beta)]$ be an ordered normal basis of $\mathbb{F}_{q^s} \leq \mathbb{F}_{q^m}$. The *$\sigma$-transform (w.r.t. $\mathcal{B}$)* is the mapping $\hat{\cdot} : \mathbb{F}_{q^m}[x;\sigma]_{<s} \to \mathbb{F}_{q^m}[x;\sigma]_{<s}$, $a \mapsto \hat{a}$, where

$$\hat{a}_j = a(\sigma^j(\beta)) = \sum_{i=0}^{s-1} a_i \sigma^{i+j}(\beta), \quad \forall\, j = 0, \dots, s-1. \tag{2.4}$$

Its inverse is called *inverse σ-transform (w.r.t. $\mathcal{B}$)*.

The σ-transform can be seen as the $\mathbb{F}_{q^m}[x;\sigma]$-analog of the discrete Fourier transform over ordinary polynomial rings. Its motivation in coding theory is a transform-based definition of rank-metric codes, similar to Blahut's [Bla79] description of several codes in Hamming metric (see also [Bos13]). This observation was utilized in [SK09] and [WAS13, Wac13] to obtain fast decoding methods for these codes.

2.3.2 Modules over Skew Polynomial Rings

A *module* is a generalization of a vector space, where the underlying scalar space is a (unitary) ring instead of a field. If the ring is non-commutative, we distinguish between left and right modules, which indicate the side from which a scalar is multiplied to a vector in the module.

In this thesis, we only consider left modules and often omit the "left" in front of attributes. Considering the definition of *scalar multiplication*, *linear combination*, *linear independence*, *generating set*, *basis*, *submodule*, and other notions immediately generalize from vector spaces. A module is called *free* if it has a basis.

In general, modules do not behave as nicely as vector spaces. However, in this thesis, we only consider left modules over $\mathbb{F}_{q^m}[x;\sigma]$, which share many important properties with their well-studied commutative analogs over $\mathbb{F}_{q^m}[x]$. For instance, the following properties follow analog to [Lan02, Theorem 7.1] over $\mathbb{F}_{q^m}[x]$, and by the statements [Pet12, Theorem 18] and [Pet12, Proposition 16], using that $\mathbb{F}_{q^m}[x;\sigma]$ is left Euclidean, which implies left Noetherian[17].

Theorem 2.26. *Let $\mathcal{V}$ be a (left) submodule of $\mathbb{F}_{q^m}[x;\sigma]^r$. Then,*

- *$\mathcal{V}$ is free.*
- *$\mathcal{V}$ has a basis with at most r elements.*
- *Any two bases of $\mathcal{V}$ have the same number of elements.*

Using these properties, the *dimension* of $\mathcal{V}$ is well-defined as the number of its basis elements.

Skew Polynomial Matrices

In this thesis, we often consider vectors and matrices over $\mathbb{F}_{q^m}[x;\sigma]$. Similar to [Nie13b] over $\mathbb{F}_{q^m}[x]$, we define the following notation for vectors $\boldsymbol{v} \in \mathbb{F}_{q^m}[x;\sigma]^r$, where $r \in \mathbb{N}$, and matrices $\boldsymbol{V} \in \mathbb{F}_{q^m}[x;\sigma]^{r\times r}$. By $\boldsymbol{v}_i$, we denote the i^{th} row of the matrix $\boldsymbol{V}$.

[16] The σ-transform is called *q-transform* in case of linearized polynomials ($\sigma = \cdot^q$), cf. [SK09, Wac13].
[17] I.e., every left ideal is finitely generated.

Definition 2.27. Let $\boldsymbol{v} \in \mathbb{F}_{q^m}[x;\sigma]^r$ and $\boldsymbol{V} \in \mathbb{F}_{q^m}[x;\sigma]^{r\times r}$.

- The *degree of* $\boldsymbol{v}$ is defined by $\deg \boldsymbol{v} = \max_i\{\deg v_i\}$.
- The *degree of* $\boldsymbol{V}$ is $\deg \boldsymbol{V} = \sum_i \deg \boldsymbol{v}_i$, where $\boldsymbol{v}_i$ is the i^{th} row of $\boldsymbol{V}$.
- The *maximal degree of* $\boldsymbol{V}$ is $\operatorname{maxdeg} \boldsymbol{V} = \max_i\{\deg \boldsymbol{v}_i\}$.
- The *leading position* of $\boldsymbol{v} \neq \boldsymbol{0}$ is the right-most position of maximal degree, i.e.,

$$\operatorname{LP}(\boldsymbol{v}) = \max\{i \,:\, \deg v_i = \deg \boldsymbol{v}\}.$$

  Its *leading term* is given by the polynomial $\operatorname{LT}(\boldsymbol{v}) = v_{\operatorname{LP}(\boldsymbol{v})}$.

The *row space* of a matrix $\boldsymbol{V} \in \mathbb{F}_{q^m}[x;\sigma]^{r\times r}$ is the set of all linear combinations of its rows. The *rank* of a matrix $\boldsymbol{V} \in \mathbb{F}_{q^m}[x;\sigma]^{r\times r}$ is the dimension of its row space. We denote the set of matrices in $\mathbb{F}_{q^m}[x;\sigma]^{r\times r}$ of (full) rank r by $\operatorname{GL}_r(\mathbb{F}_{q^m}[x;\sigma])$. Two matrices $\boldsymbol{V}, \boldsymbol{V'} \in \mathbb{F}_{q^m}[x;\sigma]^{r\times r}$ have the same row space if and only if there is a matrix $\boldsymbol{U} \in \operatorname{GL}_r(\mathbb{F}_{q^m}[x;\sigma])$ such that $\boldsymbol{V} = \boldsymbol{U}\cdot\boldsymbol{V'}$. Elementary row operations, which consist of adding a scalar multiple of one row to another, neither change the row space nor the rank of a matrix.

The Weak Popov Form

In Chapter 6, we will extensively deal with algorithms for transforming skew polynomial matrices into the following matrix normal form, the weak Popov form. We use a similar notation as in the $\mathbb{F}_{q^m}[x]$-case, cf. [Nie13b].

Definition 2.28. A matrix $\boldsymbol{V} \in \mathbb{F}_{q^m}[x;\sigma]^{r\times r}$ is in *weak Popov form (wPf)* if the leading positions of its non-zero rows are different.

We call a matrix *row reduced* if it is in weak Popov form and the process of obtaining such a matrix *row reduction*.[18] Row reduction is also sometimes called *module minimization* since it finds a module basis, which is minimal in the following way.

Lemma 2.29. *Let* $\boldsymbol{V} \in \mathbb{F}_{q^m}[x;\sigma]^{r\times r}$ *be in weak Popov form and* $\mathcal{V} \subseteq \mathbb{F}_{q^m}[x;\sigma]^r$ *be its row space. Then, the rows of* $\boldsymbol{V}$ *are linearly independent and every* $\boldsymbol{u} \in \mathcal{V}\setminus\{\boldsymbol{0}\}$ *satisfies* $\deg \boldsymbol{u} \geq \deg \boldsymbol{v}$*, where* $\boldsymbol{v}$ *is the row of* $\boldsymbol{V}$ *with* $\operatorname{LP}(\boldsymbol{u}) = \operatorname{LP}(\boldsymbol{v})$.

Proof. The proof works as its $\mathbb{F}_{q^m}[x]$-analog (cf. [Nie13a]), but we include it for completeness. Since $\boldsymbol{u}$ is in the row space of $\boldsymbol{V}$, we must have $\boldsymbol{u} = \sum_i a_i\boldsymbol{v}_i$, where $\boldsymbol{v}_i$ are the rows of $\boldsymbol{V}$ and $a_i \in \mathbb{F}_{q^m}[x;\sigma]$. The $\boldsymbol{v}_i$ have different leading positions and if $a_i \neq 0$, then $\operatorname{LP}(\boldsymbol{v}_i) = \operatorname{LP}(a_i\boldsymbol{v}_i)$. If two vectors have different LP, then their sum has both the LP and degree of one of the two (the "dominant" one is the vector of larger degree, or, in case of the same degree, the one with the larger LP). By an inductive argument, there is a unique row $\boldsymbol{v}_i$ with $\operatorname{LP}(\boldsymbol{u}) = \operatorname{LP}(a_i\boldsymbol{v}_i) = \operatorname{LP}(\boldsymbol{v}_i)$ and $\deg(\boldsymbol{u}) = \deg(a_i\boldsymbol{v}_i) \geq \deg(\boldsymbol{v}_i)$, which proves the claim. □

In some cases, we need to "shift" a vector in order to change the degrees of its entries. Given a "shift vector" $\boldsymbol{w} = [w_1,\ldots,w_r] \in \mathbb{N}_0^r$, we define the mapping

$$\begin{aligned}\Phi_{\boldsymbol{w}} \,:\, &\mathbb{F}_{q^m}[x;\sigma]^r \to \mathbb{F}_{q^m}[x;\sigma]^r\\ &\boldsymbol{u} = [u_1,\ldots,u_r] \mapsto [u_1x^{w_1},\ldots,u_rx^{w_r}].\end{aligned}$$

[18] Over $\mathbb{F}_{q^m}[x]$, a "row reduced" basis of a module $\mathcal{V}$ is a basis $\boldsymbol{V}$ of minimal degree among all bases of $\mathcal{V}$, cf. [Kai80][p. 384]. Being in weak Popov form implies row reduced in this sense.

Due to non-commutativity, the shifts x^{w_i} need to be multiplied from the right for two reasons. First, the coefficients of the skew polynomials are merely shifted to higher degrees and not changed, and second, the mapping $\Phi_{\boldsymbol{w}}$ is an injective (left) module isomorphism. In particular, it is possible to compute the inverse $\Phi_{\boldsymbol{w}}^{-1}$ of any vector in the module $\Phi_{\boldsymbol{w}}(\mathbb{F}_{q^m}[x;\sigma]^r)$. For notational convenience, we introduce $\mathrm{LP}_{\boldsymbol{w}}(\boldsymbol{v}) := \mathrm{LP}(\Phi_{\boldsymbol{w}}(\boldsymbol{v}))$ and $\deg_{\boldsymbol{w}}(\boldsymbol{v}) := \deg \Phi_{\boldsymbol{w}}(\boldsymbol{v})$.

The mapping $\Phi_{\boldsymbol{w}}$ can be extended to $\mathbb{F}_{q^m}[x;\sigma]$-matrices by applying it row-wise. Since the row space of $\Phi_{\boldsymbol{w}}(\boldsymbol{V})$ for any matrix $\boldsymbol{V} \in \mathbb{F}_{q^m}[x;\sigma]^{r\times r}$ is a subset of $\Phi_{\boldsymbol{w}}(\mathbb{F}_{q^m}[x;\sigma]^r)$, we can easily compute the inverse $\Phi_{\boldsymbol{w}}^{-1}$ of any matrix with the same row space. Using the shift mapping, we define a shifted weak Popov form as follows.

Definition 2.30. Let $\boldsymbol{w} = [w_1, \dots, w_r] \in \mathbb{N}_0^r$. A matrix $\boldsymbol{V}$ is in *$\boldsymbol{w}$-shifted weak Popov form* if $\Phi_{\boldsymbol{w}}(\boldsymbol{V})$ is in weak Popov form.

2.3.3 Gabidulin Codes

Gabidulin codes were independently introduced by Delsarte [Del78], Gabidulin [Gab85], and Roth [Rot91]. They are evaluation codes of skew polynomials and considered to be the rank-metric analog of Reed–Solomon codes due to their resembling definition. Indeed, many properties and decoding algorithms of the codes carry over from or are inspired by their Hamming-metric analog.

Definition 2.31 ([Del78, Gab85, Rot91])**.** Let $\alpha_1, \dots, \alpha_n \in \mathbb{F}_{q^m}$ be linearly independent over $\mathbb{F}_q$, and $k \le n$. The corresponding *Gabidulin code* of length n and dimension k is defined by[19]

$$\mathcal{C}_{\mathrm{G}}[n,k] = \left\{ \left[f(\alpha_1), \dots, f(\alpha_n) \right] \,:\, f \in \mathbb{F}_{q^m}[x;\sigma],\, \deg f < k \right\} \subseteq \mathbb{F}_{q^m}^n.$$

We call the α_i *evaluation points* and f the *message polynomial* of the codeword $\boldsymbol{c}$.

Similar to RS codes in Hamming metric, Gabidulin codes attain the rank-metric Singleton bound with equality.

Theorem 2.32 ([Del78, Gab85, Rot91])**.** *Any Gabidulin code $\mathcal{C}_{\mathrm{G}}[n,k]$ is MRD.*

Proof. Let $f \in \mathbb{F}_{q^m}[x;\sigma]$ be a non-zero evaluation polynomial. Due to $\deg f \le k-1$, by Theorem 2.22 we have

$$\dim \ker(f) \le k-1.$$

Since the $\alpha_1, \dots, \alpha_n$ are linearly independent, the rank nullity theorem gives

$$\mathrm{wt_R}(\boldsymbol{c}) = \operatorname{rank}\left[f(\alpha_1), \dots, f(\alpha_n) \right] = \dim\langle \alpha_1, \dots, \alpha_n \rangle - \underbrace{\dim\left(\ker(f) \cap \langle \alpha_1, \dots, \alpha_n \rangle \right)}_{\le k-1} \ge n-k+1,$$

which implies the claim as $\mathcal{C}_{\mathrm{G}}[n,k]$ is a linear code. □

Note that a Gabidulin code can have length at most $n \le m$ since the α_i are linearly independent over $\mathbb{F}_q$. In contrast to MDS codes in the Hamming metric, MRD codes exist for any field size and length: If $n > m$, then we simply transpose the codeword matrix representation of a $\mathcal{C}_{\mathrm{G}}(m,k)$ Gabidulin code over $\mathbb{F}_{q^n}$ and consider the resulting codewords as vectors in $\mathbb{F}_{q^m}^n$.

[19] I.e., $\mathcal{C}_{\mathrm{G}}[n,k] = \mathrm{ev}_{[\alpha_1,\dots,\alpha_n]}(\mathbb{F}_{q^m}[x;\sigma]_{<k})$ in evaluation-code notation.

2.3.4 Interleaved Gabidulin Codes

Similar to the Hamming-metric case, an h-interleaved Gabidulin code is a direct sum of h Gabidulin codes. The codes were first introduced by Loidreau and Overbeck [LO06] for cryptographic applications and independently proposed by Silva, Kschischang, and Kötter [SKK08] to increase the rates of codes for random linear network coding.

In the literature, there exist two variants: *vertically* [LO06, Ove07, SKK08, Sil09, WZ13, Wac13] and *horizontally* [SB10, SJB11] interleaved codes, which solely differ in the assumed error model. The distinction is due to the invariance of a rank-metric code under the transposition of its codewords in matrix representation, and both variants, i.e., error models, can be motivated by it. The codes are defined as follows.

Definition 2.33. Let $\mathcal{C}_{\mathrm{G}}^{(1)}[n, k_1], \ldots, \mathcal{C}_{\mathrm{G}}^{(h)}[n, k_h]$ be Gabidulin codes over $\mathbb{F}_{q^m}$ with the same evaluation points $\alpha_1, \ldots, \alpha_n$. The corresponding *(h-)interleaved Gabidulin code* is defined by

$$\mathcal{C}_{\mathrm{IG}}(n, k_1, \ldots, k_h; h) = \left\{ \boldsymbol{c} = \begin{bmatrix} \boldsymbol{c}_1 \\ \vdots \\ \boldsymbol{c}_h \end{bmatrix} : \boldsymbol{c}_i \in \mathcal{C}_{\mathrm{G}}^{(i)}[n, k_i] \right\} \subseteq \mathbb{F}_{q^m}^{h \times n}.$$

We consider the following two error models, which are illustrated in Figure 2.3. Let the received word be of the form $\boldsymbol{r} = \boldsymbol{c} + \boldsymbol{e} \in \mathbb{F}_{q^m}^{h \times n}$, where $\boldsymbol{c} \in \mathcal{C}_{\mathrm{IG}}(n, k_1, \ldots, k_h; h)$ is a codeword and $\boldsymbol{e}$ is an error with components $\boldsymbol{e}_i \in \mathbb{F}_{q^m}^n$ for all $i = 1 \ldots, h$. Furthermore, let $\boldsymbol{E}_i = \mathrm{ext}_{\mathcal{B}}(\boldsymbol{e}_i)$ be the matrix representation of $\boldsymbol{e}_i$ for a fixed ordered $\mathbb{F}_q$-basis $\mathcal{B}$ of $\mathbb{F}_{q^m}$.

Error Model in Vertically Interleaved Gabidulin Codes

In *vertically* interleaved Gabidulin codes, the number of errors $\tau = \mathrm{rank}(\boldsymbol{E}_{\mathrm{v}})$ is given by the rank of the matrix

$$\boldsymbol{E}_{\mathrm{v}} = \begin{bmatrix} \boldsymbol{E}_1 \\ \boldsymbol{E}_2 \\ \vdots \\ \boldsymbol{E}_h \end{bmatrix} \in \mathbb{F}_q^{hm \times n}. \tag{2.5}$$

This means that the row spaces of the $\boldsymbol{E}_i$ are contained in a τ-dimensional subspace of $\mathbb{F}_q^n$. The error model is motivated by interpreting the codewords $\boldsymbol{c} \in \mathbb{F}_{q^m}^{h \times n}$ as matrices $\boldsymbol{C} \in \mathbb{F}_q^{hm \times n}$ by vertically concatenating the matrix representations $\boldsymbol{C}_i = \mathrm{ext}_{\mathcal{B}}(\boldsymbol{c}_i)$ of the constituent codewords $\boldsymbol{c}_i$ for $i = 1, \ldots, h$.

Error Model in Horizontally Interleaved Gabidulin Codes

In *horizontally* interleaved Gabidulin codes, the number of errors $\tau = \mathrm{rank}(\boldsymbol{E}_{\mathrm{h}})$ is given by the rank of the matrix

$$\boldsymbol{E}_{\mathrm{h}} = [\boldsymbol{E}_1, \boldsymbol{E}_2, \ldots, \boldsymbol{E}_h] \in \mathbb{F}_q^{m \times hn}. \tag{2.6}$$

This means that the column spaces of the $\boldsymbol{E}_i$ are contained in a τ-dimensional subspace of $\mathbb{F}_q^m$. Here, a codeword's matrix representation $\boldsymbol{C} = [\boldsymbol{C}_1, \ldots, \boldsymbol{C}_h] \in \mathbb{F}_q^{m \times hn}$ is obtained by horizontally concatenating the matrices $\boldsymbol{C}_i$.

In the case of horizontally interleaved Gabidulin codes, the number of errors can be directly determined from the rows $\boldsymbol{e}_i$ of the error $\boldsymbol{e}$ without considering their matrix representation: We define the *error space* $\mathcal{E} \subseteq \mathbb{F}_{q^m}$ to be smallest $\mathbb{F}_q$-subspace of $\mathbb{F}_{q^m}$ that contains the spans of the error word components, i.e.,

$$\langle e_{i,1}, \dots, e_{i,n} \rangle \subseteq \mathcal{E} \quad \forall\, i = 1, \dots, h.$$

The number of errors τ is then given by the dimension of $\mathcal{E}$.

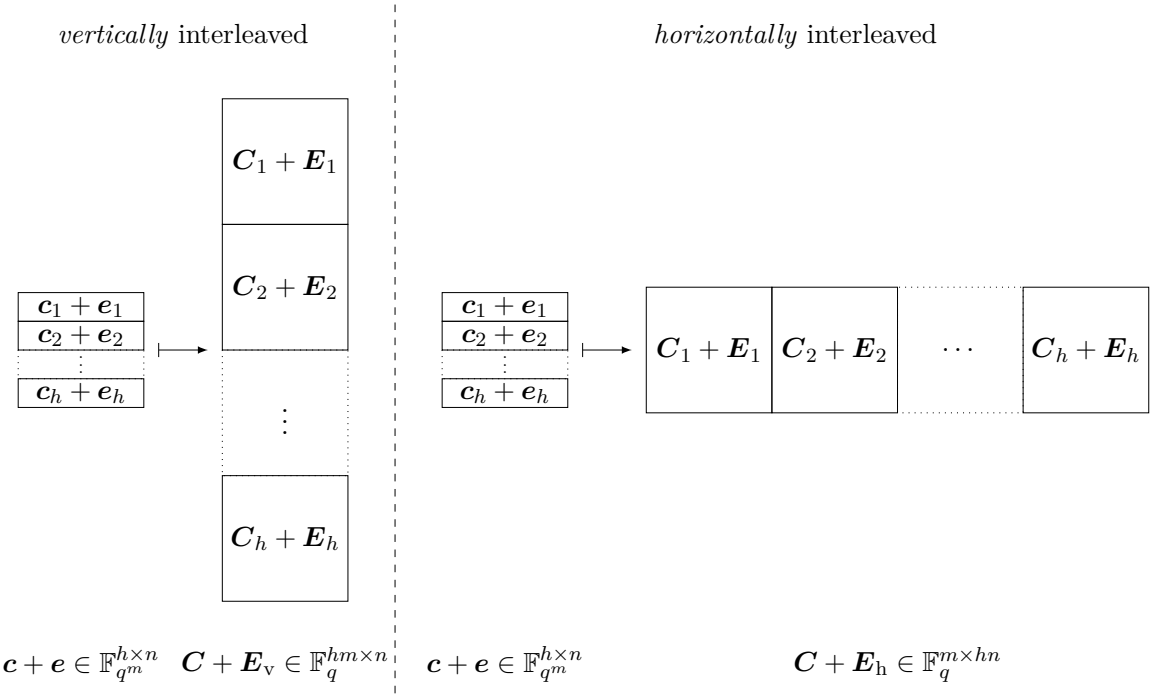


Figure 2.3: Matrix representations of vertically and horizontally interleaved Gabidulin codes, which motivate the respective number of errors: $\tau = \operatorname{rank}(\boldsymbol{E}_\mathrm{v})$ and $\tau = \operatorname{rank}(\boldsymbol{E}_\mathrm{h})$.

2.3.5 Twisted Gabidulin Codes

Quite recently, Sheekey [She16] introduced a new class of MRD codes that are inequivalent to Gabidulin codes: Twisted Gabidulin codes. Together with a special case of theirs, which was independently found by Otal and Özbudak [OÖ16], the codes form—to the best of our knowledge—the only known general family of linear MRD codes besides Gabidulin codes that exist for a large variety of code parameters.

The idea of the code construction is to evaluate skew polynomials of degree k, where the k^{th} coefficient depends on the 0^{th} one. By choosing this dependency in a suitable way (see Theorem 2.35 below), the resulting codes are MRD.

Definition 2.34 ([She16])**.** Let $\alpha_1, \dots, \alpha_n \in \mathbb{F}_{q^m}$ be linearly independent over $\mathbb{F}_q$, $\eta \in \mathbb{F}_{q^m}$, and $k < n$. The corresponding *twisted Gabidulin code* is defined by[20]

$$\mathcal{C}_{\mathrm{TG}}[n,k] = \left\{ \boldsymbol{c} = [f(\alpha_1), \dots, f(\alpha_n)] \,:\, f = \sum_{i=0}^{k-1} f_i x^i + \eta f_0 x^k \in \mathbb{F}_{q^m}[x;\sigma] \right\} \subseteq \mathbb{F}_{q^m}^n.$$

Theorem 2.35 ([She16])**.** *If $\eta^{\frac{q^m-1}{q-1}} \neq (-1)^{mk}$, then the code $\mathcal{C}_{\mathrm{TG}}[n,k]$ is MRD.*

Remark 2.36. *The construction in [She16] is in fact slightly more general: The k^{th} coefficient of f can be chosen as $f_k = \eta f_0^{q^i}$ for some $i \in \{0, \dots, m-1\}$ instead of $f_k = \eta f_0$. For $i \neq 0$, the resulting codes are not $\mathbb{F}_{q^m}$-linear, which is why we do not consider them here.*

2.3.6 Rank-Metric Codes over Fields of Characteristic Zero

Recently, there has been a raising interest in Gabidulin codes that are not defined over a finite field, but over fields of characteristic zero, in particular number fields [Rot96, ALR13, Rob15b, ALR17]. Such codes can be applied for low-rank matrix recovery [FS12, MPB17b] or constructing space-time codes [Rob15a].

They are defined exactly as Gabidulin codes over finite fields, where instead of a field extension $\mathbb{F}_{q^m}/\mathbb{F}_q$, we take a Galois extension L/K of two—possibly infinite—fields K and L whose Galois group $\mathrm{Gal}(L/K)$ is cyclic (e.g. cyclotomic extensions of $\mathbb{Q}$ or Kummer extensions thereof, cf. [ALR13, Rob15b]). Since L is a finite-dimensional vector space over K, the rank metric is well-defined over L^n.

If the automorphism $\sigma \in \mathrm{Gal}(L/K)$ is a generator of the Galois group and we define $L[x;\sigma]$ to be the set of skew polynomials with coefficients in L and automorphism σ, then the definitions and statements in Section 2.3.1 (skew polynomials), Section 2.3.2 (modules over skew polynomial rings), Section 2.3.3 (Gabidulin codes), and Section 2.3.4 (interleaved Gabidulin codes) immediately carry over by replacing $\mathbb{F}_{q^m} \leftarrow L$ and $\mathbb{F}_q \leftarrow K$, cf. [ALR13, Rob15b].

In this thesis, we concentrate on codes over finite fields. However, since most results in Chapter 5 and Chapter 6 directly apply to codes over fields of characteristic zero, we will briefly discuss this connection in the respective conclusion sections.

[20] I.e., $\mathcal{C}_{\mathrm{TG}} = \mathrm{ev}_{[\alpha_1,\dots,\alpha_n]}(\{f = \sum_{i=0}^{k-1} f_i x^i + \eta f_0 x^k \in \mathbb{F}_{q^m}[x;\sigma]\})$ in evaluation-code notation.

Part I

Codes in Hamming Metric

3

Improved Power Decoding of Interleaved Codes in Hamming Metric

Decoding h-interleaved Reed–Solomon codes has received much attention in the last two decades. By assuming that errors occur at the same positions in the constituent codewords, i.e., burst errors, one can decode beyond half the minimum distance of the code, see e.g.,() [KL97, BKY03, CS03, PV04, BMS04, Par07, SSB07, SSB09, CH13, WZB14]. All of these decoders are *partial*, i.e., they fail for some error patterns beyond half the minimum distance. For notational convenience, we consider homogeneous IRS codes, i.e., where the constituent RS codes all have the same code rate R.[1]

The first such algorithm was presented by Krachkovsky and Lee in [KL97], which can correct a fraction of $\frac{h}{h+1}(1-R)$ errors with high probability by collaboratively retrieving a joint error locator polynomial from the constituent received words. The same decoding radius was also achieved by Bleichenbacher, Kiayias, and Yung in [BKY03], Brown, Minder, and Shokrollahi [BMS04, BMS05] and Schmidt, Sidorenko, and Bossert in [SSB09].

Coppersmith and Sudan [CS03] presented a partial interpolation-based algorithm with relative decoding radius at least $1-R-R^{h/h+1}$. The method can be seen as a generalization of the Sudan [Sud97] and Guruswami–Sudan [GS98] list decoder, and improves upon $\frac{h}{h+1}(1-R)$ for low rates. Parvaresh and Vardy [PV04, Par07] and Cohn and Heninger [CH13] introduced another generalization based on $(h+1)$-variate interpolation that can correct up to a fraction of $1-R^{h/h+1}$ errors under certain assumptions.[2] The cost of the algorithm is dominated by the expensive root-finding step, which relies on resultant or Gröbner basis computation.

Power decoding, which was introduced by Schmidt, Sidorenko, and Bossert [SSB06, SSB10], is a partial decoding algorithm for Reed–Solomon codes that can decode up to roughly the same radius as Sudan's algorithm. The idea is to generate several linearly independent key equations from element-wise powers of the received word, which is a non-linear operation in general. In [SSB07], the same authors proposed to combine collaborative decoding of IRS codes with the power decoding method in order to obtain a larger system of key equations, resulting in a decoding radius similar to the Coppersmith–Sudan algorithm and improving upon it at some rates $R \leq 1/3$. The method was further refined by Wachter-Zeh, Zeh, and Bossert in [WZB14] by mixing (i.e., element-wise multiplying) powers of received words.

Recently, Rosenkilde [Nie14, Ros18] introduced an improved power decoding algorithm for

[1] The generalization of the new algorithms to the inhomogeneous case is technical, but straightforward.

[2] In [PV04, Par07], the root-finding step was described for $h = 2$ although the interpolation step works for arbitrary h. The root-finding step was later generalized in [CH13].

RS codes that—with high probability—achieves roughly the same decoding radius as the Guruswami–Sudan list decoder, asymptotically attaining the Johnson radius. The improvement is due to a larger system of key equations, which again depends on element-wise powers of the received word. In contrast to the Sudan and Guruswami–Sudan decoders, both power decoding and its improved version do not require a root-finding step.

In Section 3.1, we join the ideas of the improved power decoder [Ros18], combining collaborative with power decoding [SSB07], and the mixing technique [WZB14]. We argue that—under certain assumptions—the resulting algorithm is able to decode up to a relative radius of $1 - R^{h/h+1}$, which equals the radii of [PV04, Par07] and [CH13]. Simulation results for various code and decoder parameters indicate that the new algorithm indeed achieves this maximal decoding radius with high probability. Compared to [PV04, Par07] and [CH13], our algorithm is faster due to the lack of a root-finding step. For $\varepsilon > 0$, we can achieve a relative decoding radius $1 - R^{h/h+1} - \varepsilon$ at the cost of $O\tilde{}(n(1/\varepsilon)^{h\omega+1})$ field operations.

All mentioned decoding radii are illustrated in Figure 3.1.

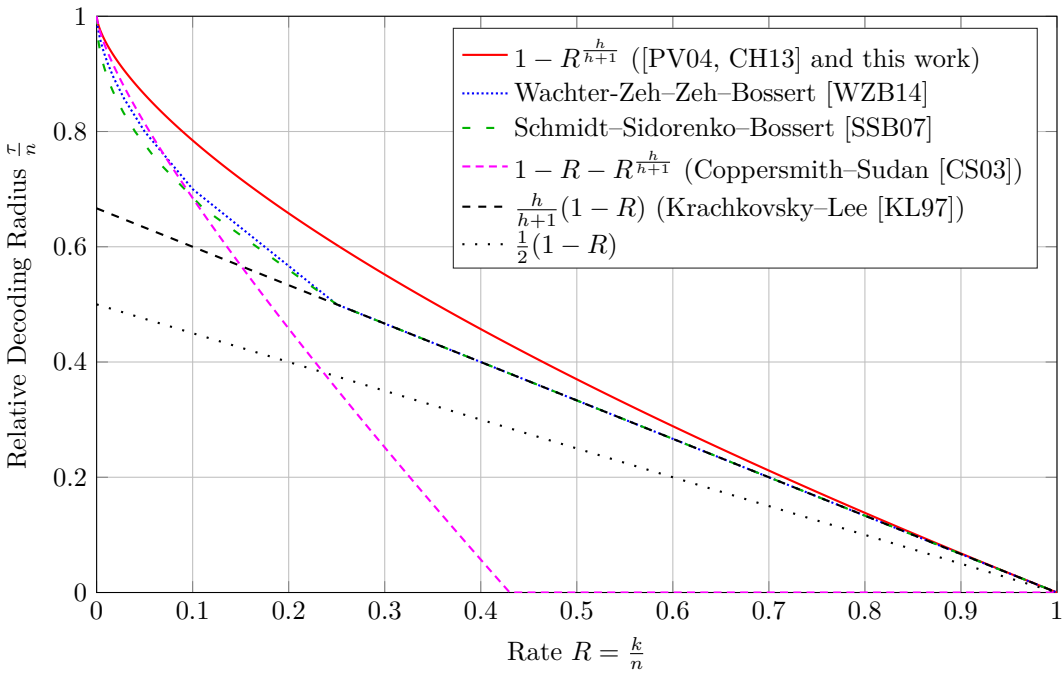


Figure 3.1: Comparison of Relative Decoding Radii of IRS Decoders for $h = 2$.

In Section 3.2, we extend our results to homogeneous interleaved one-point Hermitian codes of maximal length $n = q^3$. So far, the original power decoder was generalized to one-point Hermitian codes by Kampf [Kam14] and Rosenkilde and Beelen [NB15], achieving the same decoding radius as the adaption of Sudan's algorithm by Høholdt and Nielsen [HN99]. Furthermore, Brown et al. [BMS05] and Kampf [Kam14] adapted the collaborative decoder of IRS codes to IH codes, reaching a relative decoding radius of about $\frac{h}{h+1}(1 - \frac{k+g-1}{n})$, where k is the dimension, n the length, and g the genus of the code. To the best of our knowledge, other decoding methods for IRS codes have not been extended to IH codes.

We show that our arguments for combining improved power with collaborative decoding

and mixing can be directly adapted. For efficiently solving the system of key equations, we rely on methods similar to the fast power decoder for one-point Hermitian codes in [NB15]. The resulting relative decoding radius asymptotically achieves $1 - (\frac{k+g-1}{n})^{h/h+1}$, improving upon the previous best algorithms by Brown et al. [BMS05] and Kampf [Kam14]. As a by-product (i.e., for $h = 1$), we obtain an adaption of Rosenkilde's improved power decoder to one-point Hermitian codes that attains the same radius as the Guruswami–Sudan decoder [GS98]. Figure 3.2 provides an overview of existing and new decoding algorithms.

In Section 3.3, we compare the decoding radii of the new decoding algorithms for the same overall field size and length.

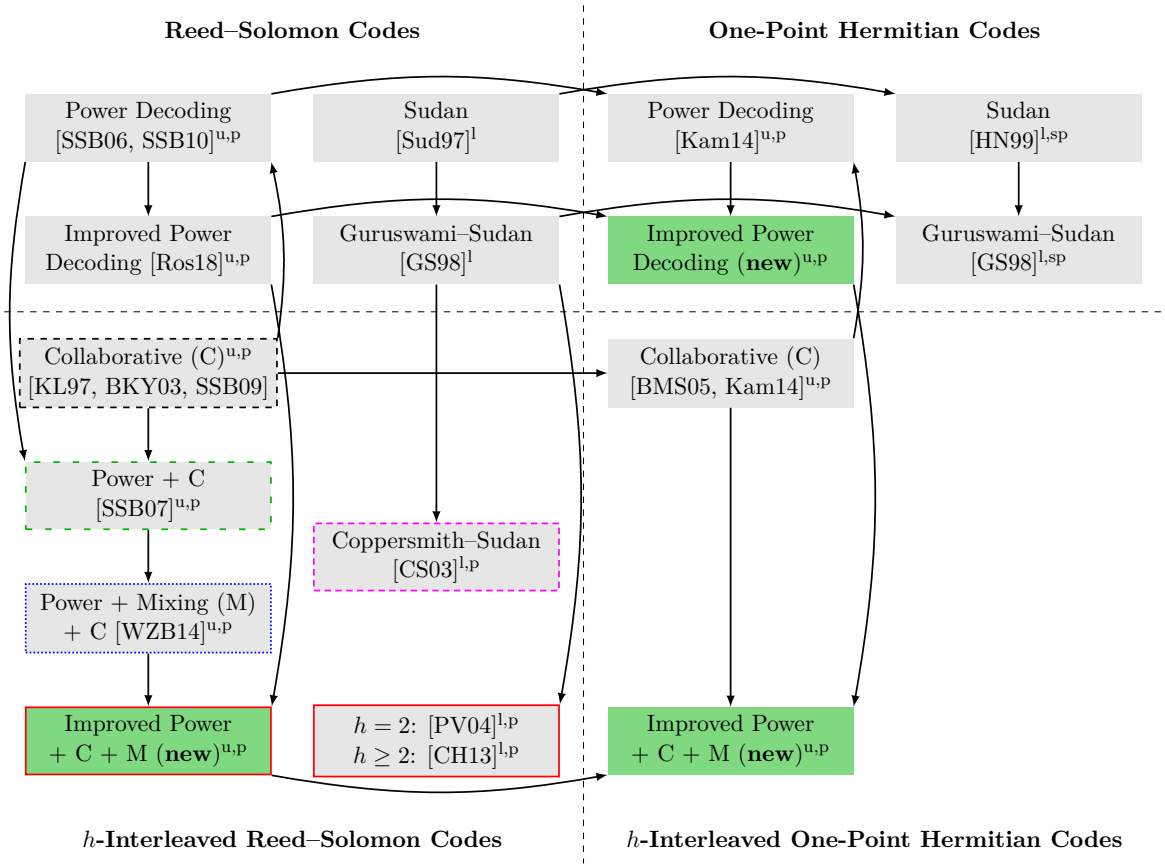


Figure 3.2: Algorithms for decoding h-interleaved Reed–Solomon and one-point Hermitian codes beyond the unique decoding radius, roughly sorted by their decoding radius (increasing from the top to the bottom). New algorithms: . Arrows indicate that an algorithm is based on the idea of the arrow's origin. Line types of frames are chosen as in Figure 3.1.
Legend: $^{\text{l}}$list decoder, $^{\text{l,p}}$partial list decoder, $^{\text{u,p}}$partial unique decoder, $^{\text{l,sp}}$list decoder that is partial above its guaranteed radius.

The results of this chapter were partly published in [PR17], [PBR17], and [PRB17] (preprint).

3.1 Improved Power Decoding of Interleaved Reed–Solomon Codes

3.1.1 Key Equations

In the following, we derive a system of key equations that combines the ideas of enhancing the error correction capability of interleaved RS codes using power decoding [SSB07], its improvement using "mixing" of codewords [WZB14], and the recent result about improved power decoding of (non-interleaved) RS codes [Ros18]. We use a similar notation for power decoding as in [Ros18].

Channel Model

Let $\boldsymbol{r} = \boldsymbol{c} + \boldsymbol{e} \in \mathbb{F}_q^{h\times n}$ be the received word, where $\boldsymbol{c} \in \mathcal{C}_{\mathrm{IRS}}(n, k; h)$ is a codeword of a homogeneous interleaved Reed–Solomon code with corresponding message polynomials, written as a vector, $\boldsymbol{f} = [f_1, \dots, f_h] \in \mathbb{F}_q[x]^h_{<k}$. Recall from Section 2.2.3 that we consider burst errors, i.e., the error positions $\mathcal{E}$ are the non-zero columns of the error word $\boldsymbol{e}$.

Polynomials Occurring in the Key Equations

The receiver can compute the following $h+1$ polynomials in $O^{\sim}(hn)$ field operations (cf. [GG99]).

Definition 3.1. Let $\boldsymbol{r} \in \mathbb{F}_q^{h\times n}$ be as above and $\alpha_1, \dots, \alpha_n$ be the distinct evaluation points of the corresponding code $\mathcal{C}_{\mathrm{IRS}}(n, k; h)$. For $t = 1, \dots, h$, let $R_t \in \mathbb{F}_q[x]$ be the unique interpolation polynomial of degree $\deg R_t < n$ such that

$$R_t(\alpha_i) = r_{t,i} \quad \forall i = 1, \dots, n.$$

We write $\boldsymbol{R} = [R_1, \dots, R_h]$. Furthermore, we define $G := \prod_{i=1}^{n}(x - \alpha_i) \in \mathbb{F}_q[x]$.

Our system of key equations (see Theorem 3.6 below) contains the unknown message polynomial vector $\boldsymbol{f}$ and the *error locator polynomial* Λ, whose roots are exactly the evaluation points of the error positions.

Definition 3.2. The *error locator polynomial* is defined by

$$\Lambda = \textstyle\prod_{i\in\mathcal{E}}(x - \alpha_i).$$

As a last component of the system of key equations, we define the unknown vector

$$\boldsymbol{\Omega} = \Lambda(\boldsymbol{f} - \boldsymbol{R})/G \in \mathbb{F}_q[x]^h,$$

which is well-defined by the following lemma.[3]

Lemma 3.3. *Let G and $\boldsymbol{R}$ be as in Definition 3.1 and Λ as in Definition 3.2. Then, G divides each component of $\Lambda(\boldsymbol{f} - \boldsymbol{R})$.*

Proof. For each $i = 1, \dots, h$ and $j = 1, \dots, n$, we have $\big(\Lambda(f_i - R_i)\big)(\alpha_j) = 0$ since

$$\big(\Lambda(f_i - R_i)\big)(\alpha_j) = \Lambda(\alpha_j) \cdot (-e_{i,j}) = \begin{cases} 0 \cdot (-e_{i,j}), & j \in \mathcal{E} \\ \Lambda(\alpha_j) \cdot 0, & \text{else.} \end{cases}$$

Thus, $(x - \alpha_j) \mid \Lambda(f_i - R_i)$ and the claim follows by the definition of $G = \prod_{j=1}^{n}(x - \alpha_j)$. □

[3] The components Ω_i of $\boldsymbol{\Omega}$ also occur in other key equations and are usually called *error evaluator polynomials*.

The Mixing Technique

In our system of key equations, we "mix" powers of the entries of the vectors $\boldsymbol{R}$, $\boldsymbol{f}$, and $\boldsymbol{\Omega}$, similar to [WZB14], as follows.

Definition 3.4 (Mixing Notation)**.** Let $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{F}_q[x]^h$ and $\boldsymbol{i}, \boldsymbol{j} \in \mathbb{N}_0^h$.

i) The *size* of $\boldsymbol{i}$ is defined as $|\boldsymbol{i}| := \sum_\mu i_\mu$.

ii) By $\preceq$, we denote the product partial order on $\mathbb{N}_0^h$, i.e., $\boldsymbol{i} \preceq \boldsymbol{j}$ if $i_\mu \leq j_\mu \; \forall \, \mu = 1, \ldots, h$.

iii) Furthermore, we define

$$\boldsymbol{a}^{\boldsymbol{i}} := \prod_{t=1}^{m} a_t^{i_t}, \quad \binom{\boldsymbol{j}}{\boldsymbol{i}} := \prod_{t=1}^{m} \binom{j_t}{i_t}.$$

Note that the number of vectors $\boldsymbol{i} \in \mathbb{N}_0^m$ of size $|\boldsymbol{i}| = \mu$ is given by $\binom{m+\mu-1}{\mu}$. Using the above notation, we can formulate a vectorized binomial theorem.

Lemma 3.5. *Let $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{F}_q[x]^h$, and $\boldsymbol{j} \in \mathbb{N}_0^h$. Then,*

$$(\boldsymbol{a} + \boldsymbol{b})^{\boldsymbol{j}} = \sum_{\boldsymbol{i} \preceq \boldsymbol{j}} \binom{\boldsymbol{j}}{\boldsymbol{i}} \boldsymbol{a}^{\boldsymbol{i}} \boldsymbol{b}^{\boldsymbol{j}-\boldsymbol{i}}.$$

Proof. We re-write

$$(\boldsymbol{a} + \boldsymbol{b})^{\boldsymbol{j}} = \prod_{\mu=1}^{r} (a_\mu + b_\mu)^{j_\mu} = \prod_{\mu=1}^{r} \sum_{i_\mu=0}^{j_\mu} \tbinom{j_\mu}{i_\mu} a_\mu^{i_\mu} b_\mu^{j_\mu - i_\mu} = \sum_{\boldsymbol{i} \preceq \boldsymbol{j}} \prod_{\mu=1}^{r} \tbinom{j_\mu}{i_\mu} a_\mu^{i_\mu} b_\mu^{j_\mu - i_\mu} = \sum_{\boldsymbol{i} \preceq \boldsymbol{j}} \tbinom{\boldsymbol{j}}{\boldsymbol{i}} \boldsymbol{a}^{\boldsymbol{i}} \boldsymbol{b}^{\boldsymbol{j}-\boldsymbol{i}}.$$

□

Key Equations

The system of key equations is stated in the following theorem. It consists of $\binom{h+\ell}{h} - 1$ many relations, indexed by the parameter $\boldsymbol{j} \in \mathbb{N}_0^h$, between the known polynomials in $\boldsymbol{R}$ and G and the unknown ones in $\boldsymbol{f}$, $\boldsymbol{\Omega}$, and Λ. The system depends on two parameters, ℓ and s, which are also parameters of the resulting decoding algorithm. As in the key equations of the original power decoder [SSB10], ℓ is the maximal power in which the components of the message polynomials f_i occur. The key equations also contain the s^{th} power of the error locator, which has the evaluation points of the error positions as zeros of multiplicity s.[4]

Theorem 3.6 (System of Key Equations)**.** *Let $s, \ell \in \mathbb{N}$ be such that $s \leq \ell$, and Λ, G, $\boldsymbol{R}$, $\boldsymbol{f}$, and $\boldsymbol{\Omega}$ be as above. Then, for all $\boldsymbol{j} \in \mathbb{N}_0^h$ with $1 \leq |\boldsymbol{j}| \leq \ell$, we have*

$$\Lambda^s \boldsymbol{f}^{\boldsymbol{j}} = \sum_{\boldsymbol{i} \preceq \boldsymbol{j}} \left[\Lambda^{s-|\boldsymbol{i}|} \boldsymbol{\Omega}^{\boldsymbol{i}}\right] \left[\binom{\boldsymbol{j}}{\boldsymbol{i}} \boldsymbol{R}^{\boldsymbol{j}-\boldsymbol{i}} G^{|\boldsymbol{i}|}\right], \qquad 1 \leq |\boldsymbol{j}| < s \tag{3.1}$$

$$\Lambda^s \boldsymbol{f}^{\boldsymbol{j}} \equiv \sum_{\substack{\boldsymbol{i} \preceq \boldsymbol{j} \\ |\boldsymbol{i}| < s}} \left[\Lambda^{s-|\boldsymbol{i}|} \boldsymbol{\Omega}^{\boldsymbol{i}}\right] \left[\binom{\boldsymbol{j}}{\boldsymbol{i}} \boldsymbol{R}^{\boldsymbol{j}-\boldsymbol{i}} G^{|\boldsymbol{i}|}\right] \mod G^s, \qquad s \leq |\boldsymbol{j}| \leq \ell. \tag{3.2}$$

[4] s was called *multiplicity parameter* in [Ros18] (case $h = 1$), in analogy to the parameters of the GS decoder [GS98]. Using the same parameters, these two algorithms yield almost the same decoding radius.

Proof. Using Lemma 3.5, we can write

$$\Lambda^s \boldsymbol{f}^{\boldsymbol{j}} = \Lambda^s((\boldsymbol{f}-\boldsymbol{R})+\boldsymbol{R})^{\boldsymbol{j}} = \sum_{\boldsymbol{i} \preceq \boldsymbol{j}} \binom{\boldsymbol{j}}{\boldsymbol{i}} \Lambda^s (\boldsymbol{f}-\boldsymbol{R})^{\boldsymbol{i}} \boldsymbol{R}^{\boldsymbol{j}-\boldsymbol{i}}.$$

For $|\boldsymbol{i}| < s$, the summand $\Lambda^s(\boldsymbol{f}-\boldsymbol{R})^{\boldsymbol{i}}$ can be re-written as

$$\Lambda^s(\boldsymbol{f}-\boldsymbol{R})^{\boldsymbol{i}} = \Lambda^{s-|\boldsymbol{i}|}[\Lambda(\boldsymbol{f}-\boldsymbol{R})]^{\boldsymbol{i}} = \Lambda^{s-|\boldsymbol{i}|}\boldsymbol{\Omega}^{\boldsymbol{i}} G^{|\boldsymbol{i}|}.$$

If $|\boldsymbol{i}| \geq s$, we can decompose $\boldsymbol{i} = \boldsymbol{i}_s + \boldsymbol{i}'$, where $\boldsymbol{i}_s, \boldsymbol{i}' \in \mathbb{Z}^h_{\geq 0}$ with $|\boldsymbol{i}_s| = s$ and obtain

$$\Lambda^s(\boldsymbol{f}-\boldsymbol{R})^{\boldsymbol{i}} = [\Lambda(\boldsymbol{f}-\boldsymbol{R})]^{\boldsymbol{i}_s}(\boldsymbol{f}-\boldsymbol{R})^{\boldsymbol{i}'} = G^s \boldsymbol{\Omega}^{\boldsymbol{i}_s}(\boldsymbol{f}-\boldsymbol{R})^{\boldsymbol{i}'},$$

so all summands $\boldsymbol{i}$ with $|\boldsymbol{i}| \geq s$ are divisible by G^s, which implies the claim. □

Remark 3.7. *For $h = 1$, the system of key equations in Theorem 3.6 coincides with the one in [Ros18]. Similarly, we obtain the relations in [WZB14] in the special case $s = 1$. The notation $\boldsymbol{f}^{\boldsymbol{j}} = \prod_{\mu=1}^h f_\mu^{j_\mu}$ and $\boldsymbol{R}^{\boldsymbol{j}-\boldsymbol{i}} = \prod_{\mu=1}^h R_\mu^{j_\mu - i_\mu}$ corresponds to mixing the codewords and received words. This mixing technique was used in [WZB14] in order to improve the decoding radius of [SSB07]. The original power decoding key equations [SSB06] (written as in [Ros18]), $\Lambda f^j \equiv \Lambda R^j \mod G$ for $j = 1, \dots, \ell$, where $f = f_1$ and $R = R_1$, are obtained for $h = s = 1$. The name "power decoding" is due to the powers of the polynomials in $\boldsymbol{f}^{\boldsymbol{j}}$ and $\boldsymbol{R}^{\boldsymbol{j}-\boldsymbol{i}}$.*

In the following, we show how to solve the key equations and use the following abbreviations:

$$\begin{aligned} \Psi_{\boldsymbol{j}} &:= \Lambda^s \boldsymbol{f}^{\boldsymbol{j}} && \text{(unknown at the receiver)}, \\ \Lambda_{\boldsymbol{i}} &:= \Lambda^{s-|\boldsymbol{i}|}\boldsymbol{\Omega}^{\boldsymbol{i}} && \text{(unknown at the receiver), and} \\ A_{\boldsymbol{i},\boldsymbol{j}} &:= \tbinom{\boldsymbol{j}}{\boldsymbol{i}} \boldsymbol{R}^{\boldsymbol{j}-\boldsymbol{i}} G^{|\boldsymbol{i}|} && \text{(known at the receiver)}. \end{aligned} \tag{3.3}$$

By *known at the receiver*, we mean that the polynomial can be directly computed from the received word. Similarly, *unknown* means that our algorithm aims at finding this polynomial.

Remark 3.8. *The polynomials $\Psi_{\boldsymbol{j}}$, $\Lambda_{\boldsymbol{i}}$, and $A_{\boldsymbol{i},\boldsymbol{j}}$ as in* (3.3) *have degree bounds*

$$\begin{aligned} \deg \Psi_{\boldsymbol{j}} &= \deg(\Lambda^s) + \deg(\boldsymbol{f}^{\boldsymbol{j}}) \leq \deg \Lambda_{\boldsymbol{0}} + |\boldsymbol{j}|(k-1) \\ \deg \Lambda_{\boldsymbol{i}} &= \deg(\Lambda^{s-|\boldsymbol{i}|}) + \deg(\Lambda^{|\boldsymbol{i}|}) + \deg((\boldsymbol{f}-\boldsymbol{R})^{\boldsymbol{i}}) - \deg(G^{|\boldsymbol{i}|}) \leq \deg \Lambda_{\boldsymbol{0}} - |\boldsymbol{i}| \\ \deg A_{\boldsymbol{i},\boldsymbol{j}} &\leq (n-1)(|\boldsymbol{j}| - |\boldsymbol{i}|) + |\boldsymbol{i}|n = (n-1)|\boldsymbol{j}| + |\boldsymbol{i}| \quad \text{for } \boldsymbol{i} \preceq \boldsymbol{j}. \end{aligned}$$

The bounds are tight in general: The first one is fulfilled with equality for all message polynomial vectors with $\deg f_\mu = k-1$ for all μ with $j_\mu > 0$. The second and third bound are tight in case of $\deg R_\mu = n-1$ for all μ with $i_\mu > 0$, or $j_\mu - i_\mu > 0$, respectively. For a fixed μ, a fraction of $1 - 1/q$ codewords, or received words, results in these maximal $\deg f_\mu$, or $\deg R_\mu$, by definition and properties of the Lagrange interpolation. Note that $A_{\boldsymbol{i},\boldsymbol{j}} = 0$ if not $\boldsymbol{i} \preceq \boldsymbol{j}$.

3.1.2 Solving the Key Equations

As in any other decoding-related key equation, the relations given in Theorem 3.6 are non-linear in the unknown polynomials. Therefore, we relax the problem into a linear one, i.e., where we do not assume that $\lambda_{\boldsymbol{i}}$ and $\psi_{\boldsymbol{j}}$, are of the form as in (3.3), and hope that its solution agrees with the sought one, i.e., $\lambda_{\boldsymbol{i}} = \Lambda_{\boldsymbol{i}}$ and $\psi_{\boldsymbol{j}} = \Psi_{\boldsymbol{j}}$. The linearized version is a wide generalization of a multi-sequence shift register synthesis problem [FT89, SS11, Nie13a][5] of

[5] Such problems are closely related to simultaneous Hermite Padé approximation problems [BL92, BL94].

the following form, whose degree restrictions are motivated by Remark 3.8.

Problem 3.9. *Given positive integers $s \leq \ell$, $A_{\boldsymbol{i},\boldsymbol{j}} = \binom{\boldsymbol{j}}{\boldsymbol{i}} \boldsymbol{R}^{\boldsymbol{j}-\boldsymbol{i}} G^{|\boldsymbol{i}|} \in \mathbb{F}_q[x]$ as in* (3.3) *for all $\boldsymbol{i} \in \mathcal{I} := \{\boldsymbol{i} \in \mathbb{N}_0^h : 0 \leq |\boldsymbol{i}| < s\}$ and $\boldsymbol{j} \in \mathcal{J} := \{\boldsymbol{j} \in \mathbb{N}_0^h : 1 \leq |\boldsymbol{j}| \leq \ell\}$, and $G \in \mathbb{F}_q[x]$ as in Definition 3.1. Find $\lambda_{\boldsymbol{i}}, \psi_{\boldsymbol{j}} \in \mathbb{F}_q[x]$ for $\boldsymbol{i} \in \mathcal{I}$ and $\boldsymbol{j} \in \mathcal{J}$ with monic $\lambda_{\boldsymbol{0}}$, such that*

$$\psi_{\boldsymbol{j}} = \sum_{\boldsymbol{i} \in \mathcal{I}} \lambda_{\boldsymbol{i}} A_{\boldsymbol{i},\boldsymbol{j}} \qquad 1 \leq |\boldsymbol{j}| < s, \tag{3.4}$$

$$\psi_{\boldsymbol{j}} \equiv \sum_{\boldsymbol{i} \in \mathcal{I}} \lambda_{\boldsymbol{i}} A_{\boldsymbol{i},\boldsymbol{j}} \mod G^s \qquad s \leq |\boldsymbol{j}| \leq \ell, \tag{3.5}$$

$$\deg \lambda_{\boldsymbol{0}} \geq \deg \lambda_{\boldsymbol{i}} + |\boldsymbol{i}| \qquad 0 \leq |\boldsymbol{i}| < s, \tag{3.6}$$

$$\deg \lambda_{\boldsymbol{0}} \geq \deg \psi_{\boldsymbol{j}} - |\boldsymbol{j}|(k-1) \qquad 1 \leq |\boldsymbol{j}| \leq \ell. \tag{3.7}$$

Definition 3.10. Consider an instance of Problem 3.9. A solution $\lambda_{\boldsymbol{i}}, \psi_{\boldsymbol{j}}$, $i \in \mathcal{I}$, $\boldsymbol{j} \in \mathcal{J}$ has *degree* $T \in \mathbb{N}_0$ if $\deg \lambda_{\boldsymbol{0}} = T$.[6] It is *minimal* if its degree is minimal among all solutions.

Our decoding strategy is motivated by the following theorem, whose proof is straightforward using the degree bounds in Remark 3.8. Note that since these bounds are tight in general, the relative degree restrictions (3.6) and (3.7) on the degrees of $\lambda_{\boldsymbol{i}}$ and $\psi_{\boldsymbol{j}}$ are chosen minimal among all choices for which Theorem 3.11 holds.

Theorem 3.11. *Consider an instance of Problem 3.9. Then, $\lambda_{\boldsymbol{i}} = \Lambda_{\boldsymbol{i}}$ and $\psi_{\boldsymbol{j}} = \Psi_{\boldsymbol{j}}$, where $\Lambda_{\boldsymbol{i}}$ and $\Psi_{\boldsymbol{j}}$ are chosen as in* (3.3)*, are a solution of the problem of degree $T = s \cdot |\mathcal{E}|$.*

Hence, all solutions corresponding to a valid error locator Λ and codeword $\boldsymbol{c}$ are contained in the solution space of Problem 3.9. We would like to find such a solution that has minimal degree, i.e., the smallest number of errors $|\mathcal{E}|$. We thus find a minimal solution of Problem 3.9 and hope that it is of the form $\lambda_{\boldsymbol{i}} = \Lambda_{\boldsymbol{i}}$ and $\psi_{\boldsymbol{j}} = \Psi_{\boldsymbol{j}}$. If the sought solution has degree T, the strategy can fail if there is another solution of degree at most T. We will return to this issue in the following subsections. If the strategy succeeds, we can obtain the message polynomials f_i by division of $\psi_{\boldsymbol{u}_i} = \Lambda^s f_i$ by $\lambda_{\boldsymbol{0}} = \Lambda^s$, where $\boldsymbol{u}_i = [0, \ldots, 0, 1, 0, \ldots, 0]$ is the i^{th} unit vector. The procedure is outlined in Algorithm 1.

Algorithm 1: Improved Power Decoder for h-Interleaved Reed–Solomon Codes

Input: Received word $\boldsymbol{r} \in \mathbb{F}_{q^m}^{h \times n}$ and positive integers $s \leq \ell$
Output: $\boldsymbol{f}$ such that $\boldsymbol{c}_i = [f_i(\alpha_1), \ldots, f_i(\alpha_n)]$ for all $i = 1, \ldots, h$ is the codeword corresponding to the smallest number of errors $|\mathcal{E}|$; or "decoding failure".

```
Compute R and G as in Definition 3.1
A_{i,j} ← \binom{j}{i} R^{j-i} G^{|i|}
λ_i, ψ_j ← Minimal solution of Problem 3.9 with input s, ℓ, A_{i,j}, and G
if λ_0 divides all ψ_{u_i} for i = 1, ..., h then
    f ← [ψ_{u_1}/λ_0, ..., ψ_{u_h}/λ_0]
    E ← Error set corresponding to e_i = r_i − [f_i(α_1), ..., f_i(α_n)] for i = 1, ..., h
    if λ_0 = Λ^s, where Λ is the error locator of E then
        return f
return "decoding failure"
```

[6] Note that $\lambda_{\boldsymbol{0}}$ being monic ensures that $\lambda_{\boldsymbol{0}} \neq 0$, so its degree must be non-negative.

In the next subsection, we will see that we can solve Problem 3.9 using a linear system of equations. However, the problem can be solved more efficiently using known algorithms from computational algebra.

Lemma 3.12. *A minimal solution of Problem 3.9 with input s, ℓ, $A_{\boldsymbol{i},\boldsymbol{j}}$, and G can be found in $O^{\sim}(\binom{h+\ell}{h}^{\omega}\ell n)$ operations over $\mathbb{F}_q$.*

Proof. Problem 3.9 is the same computational problem as the one in [Ros18] and the claim follows by adapting the size and polynomials' degrees of the problem. For completeness, we discuss the technical details in Appendix A.1. □

Theorem 3.13. *Algorithm 1 can be implemented in $O^{\sim}(\binom{h+\ell}{h}^{\omega}\ell n)$ operations over $\mathbb{F}_q$.*

Proof. Precomputing $\boldsymbol{R}$ and G costs $O^{\sim}(hn)$ operations over $\mathbb{F}_q$, using standard methods from computation algebra, cf. [GG99]. There are $\binom{h+\ell}{h} \cdot \binom{h+s-1}{h}$ many $A_{\boldsymbol{i},\boldsymbol{j}}$, which each an be computed in $O^{\sim}(\ell n)$ operations using a divide-&-conquer strategy. Thus, Line 2 costs

$$O^{\sim}\Big(\binom{h+\ell}{h}\binom{h+s-1}{h}\ell n\Big) \subseteq O^{\sim}\Big(\binom{h+\ell}{h}^2\ell n\Big)$$

field operations. The heaviest step is Line 3, which costs $O^{\sim}(\binom{h+\ell}{h}^{\omega}\ell n)$ by Lemma 3.12. The only remaining expensive operations are the h divisions in Line 5, which cost $O^{\sim}(h\ell n)$. □

Remark 3.14. *The algorithms by Parvaresh and Vardy [PV04, Par07] and Cohn and Heninger [CH13] have the same maximal asymptotic decoding radius as our decoder (cf. Section 3.1.3 below). We compare the complexities of the algorithms.*

In [Par07, Section 2.4], the complexity of the $(h+1)$-variate interpolation step is given as $O(n^2\ell^{3h+2})$ (by choosing the parameter m of their decoder $m \in \Theta(\ell)$, which results in approximately the same decoding radius, cf. Section 3.1.3 below). This is larger than the complexity of our decoder, which has quasi-linear runtime in n.[7] It is stated that in principal, the complexity of the interpolation step can be further reduced, but—to the best of our knowledge—this has not yet been analyzed.

The second part of the algorithm is the multivariate root-finding step. It is described in [Par07, Section 2.5] only for the case $h = 2$. The idea is to find two trivariate interpolation polynomials and to eliminate one variable using resultants. Afterwards, the message polynomials can be found by the Roth–Ruckenstein algorithm [RR00]. Resultant computation can be rather heavy (cf. [GG99, Chapter 6]) and the complexity of this step is given in [Par07] as "polynomial time" without stating the exponents of ℓ or n.

The algorithm by Cohn and Heninger [CH13] is of a similar type as the Parvaresh–Vardy algorithm. The interpolation step is done via lattice basis reduction, for which many efficient algorithms exist, and h many interpolation polynomials are found. Furthermore, Cohn and Heninger suggest to solve the root-finding step for general h by iteratively eliminating h variables of the interpolation polynomials using resultants or Gröbner bases without giving the algorithmic details—in fact, the method is only described for the integer variant of the problem that they are solving. The overall complexity of the algorithm is given as "polynomial time for fixed h" and it is not apparent whether root finding can be done efficiently.

[7] Note that for a constant h, we have $O^{\sim}(\binom{h+\ell}{h}^{\omega}\ell n) \subseteq O^{\sim}(\ell^{h\omega+1}n)$, so also the exponent of ℓ is smaller here.

3.1.3 Decoding Radius

In this section, we provide an upper bound on the number of errors for which the decoding problem solution $\lambda_{\boldsymbol{i}} = \Lambda_{\boldsymbol{i}}$ and $\psi_{\boldsymbol{j}} = \Psi_{\boldsymbol{j}}$ can be a unique minimal solution of Problem 3.9. As in previous algorithms based on power decoding [SSB06, SSB07, WZB14, Ros18], we call this upper bound the *decoding radius*. Similarly, since the algorithm might fail or return a different codeword for certain error patterns of weight up to this bound, we study the failure behavior separately in Section 3.1.4. We first prove two lemmata.

Lemma 3.15. *Let $r \in \mathbb{N}$. Then,*

$$\sum_{\mu=0}^{r} \tbinom{h+\mu-1}{\mu} = \tbinom{h+r}{h}, \text{ and } \sum_{\mu=0}^{r-1} \mu \tbinom{h+\mu-1}{\mu} = h\tbinom{h+r-1}{h+1}.$$

Proof. This follows immediately from well-known properties of the binomial coefficient. □

Lemma 3.16. *Let $T, \ell, s \in \mathbb{N}$ such that $s \leq \ell$ and $T + \ell(k-1) < sn$. Then, all polynomials $\lambda_{\boldsymbol{i}}$ and $\psi_{\boldsymbol{j}}$ for $\boldsymbol{i} \in \mathcal{I}$ and $\boldsymbol{j} \in \mathcal{J}$ that fulfill* (3.4), (3.5)*, and the absolute degree restrictions*

$$\deg \lambda_{\boldsymbol{i}} \leq T - |\boldsymbol{i}| \qquad \forall\, \boldsymbol{i} \in \mathcal{I}, \tag{3.8}$$

$$\deg \psi_{\boldsymbol{j}} \leq T + |\boldsymbol{j}|(k-1) \qquad \forall\, \boldsymbol{j} \in \mathcal{J}, \tag{3.9}$$

are exactly the solutions of a homogeneous linear system of equations over $\mathbb{F}_q$ with at least

$$\delta(T) = (\tau+1)\tbinom{h+\ell}{h} - n\left[h\tbinom{h+s-1}{h+1} + s\tbinom{h+\ell}{h} - s\tbinom{h+s-1}{h}\right] + (k-1)h\tbinom{h+\ell}{h+1}$$

more variables than equations.[8] *There are received words, where the difference between the number of variables and equations is exactly $\delta(T)$.*

Proof. By Remark 3.8, the degree of $A_{i,\boldsymbol{j}}$ is at most $(n-1)|\boldsymbol{j}| + |\boldsymbol{i}|$, so any $\lambda_{\boldsymbol{i}}$ satisfying (3.8) results in

$$\deg\Big(\sum_{\boldsymbol{i}\in\mathcal{I}} \lambda_{\boldsymbol{i}} A_{\boldsymbol{i},\boldsymbol{j}}\Big) \leq T + |\boldsymbol{j}|(n-1) \quad \forall\, \boldsymbol{j} \in \mathcal{J}.$$

Consider the right-hand sides in the equations, (3.4), of Problem 3.9, i.e., $\sum_{\boldsymbol{i}\in\mathcal{I}} \lambda_{\boldsymbol{i}} A_{\boldsymbol{i},\boldsymbol{j}}$. Furthermore, view the coefficients of the $\lambda_{\boldsymbol{i}}$ as indeterminates. Then, $\sum_{\boldsymbol{i}\in\mathcal{I}} \lambda_{\boldsymbol{i}} A_{\boldsymbol{i},\boldsymbol{j}}$ is a polynomial of x-degree at most $T + |\boldsymbol{j}|(n-1)$ (in fact, for most received words, the x-degree is equal to this value), whose coefficients are linear combinations of the unknown coefficients of the $\lambda_{\boldsymbol{i}}$. The polynomial $\psi_{\boldsymbol{j}}$ on the left-hand side of (3.4) has a much lower upper bound on its degree, cf. (3.9). Thus, we obtain a homogeneous system of equations given by the coefficients of $\sum_{\boldsymbol{i}\in\mathcal{I}} \lambda_{\boldsymbol{i}} A_{\boldsymbol{i},\boldsymbol{j}}$ of degree greater than $T + |\boldsymbol{j}|(k-1)$. Note that the upper bound on the polynomial's degree implies an upper bound the number of equations. The lowest $T + |\boldsymbol{j}|(k-1)$ coefficients do not give additional equations since the coefficients of $\psi_{\boldsymbol{j}}$ are also unknown. They can be easily obtained after solving the system for the $\lambda_{\boldsymbol{i}}$.

We obtain further equations by considering the congruences, (3.5), of Problem 3.9, which are equivalent to equations in which the left- and right-hand sides are taken modulo G^s. Note that we need to assume $T + \ell(k-1) < sn$ in order for the degree bound on $\psi_{\boldsymbol{j}}$ to be smaller

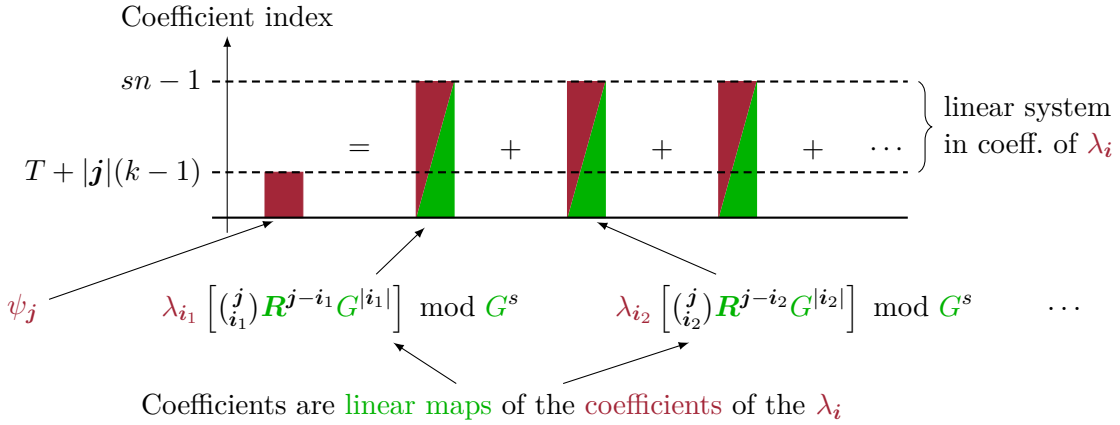


Figure 3.3: Illustration of the linear system for obtaining all polynomials $\lambda_{\boldsymbol{i}}$ and $\psi_{\boldsymbol{j}}$ satisfying (3.4), (3.5), (3.8), and (3.9), in the proof of Lemma 3.16 in the case $s \le |\boldsymbol{j}| \le \ell$ (in the other case, the upper bound on the degree is $T+|\boldsymbol{j}|(k-1)$ instead of sn). Known polynomials are displayed in green color, unknown ones are red.

than the degree bound on the right-hand-side polynomial. The system, which is illustrated in Figure 3.3, is obtained as for the equations above.

For $|\boldsymbol{j}| < s$, the number of equations is at most[9]

$$N_{\boldsymbol{j}} := (\tau + |\boldsymbol{j}|(n-1)) - (\tau + |\boldsymbol{j}|(k-1)) = |\boldsymbol{j}|(n-k)$$

and for $|\boldsymbol{j}| \ge s$ at most (note that $T+\ell(k-1) < sn$ ensures $N_{\boldsymbol{j}} \ge 0$ for all $\boldsymbol{j} \in \mathcal{J}$)

$$N_{\boldsymbol{j}} := sn - 1 - (\tau + |\boldsymbol{j}|(k-1)) = sn - \tau - |\boldsymbol{j}|(k-1) - 1.$$

Using Lemma 3.15, in total we obtain at most

$$\begin{aligned}\mathrm{NE} &:= \sum_{\boldsymbol{j}\in\mathcal{J}} N_{\boldsymbol{j}} = \sum_{1\le|\boldsymbol{j}|<s} |\boldsymbol{j}|(n-k) + \sum_{s\le|\boldsymbol{j}|\le\ell} (sn - \tau - |\boldsymbol{j}|(k-1) - 1) \\ &= (n-k)h\tbinom{h+s-1}{h+1} + (sn-\tau-1)\left(\tbinom{h+\ell}{h} - \tbinom{h+s-1}{h}\right) - (k-1)\left(h\tbinom{h+\ell}{h+1} - h\tbinom{h+s-1}{h+1}\right)\end{aligned}$$

equations. On the other hand, the number of unknowns (i.e., the coefficients of the $\lambda_{\boldsymbol{i}}$) is

$$\mathrm{NV} := \sum_{\boldsymbol{i}\in\mathcal{I}} (\tau - |\boldsymbol{i}| + 1) = (\tau+1)\tbinom{h+s-1}{h} - h\tbinom{h+s-1}{h+1}.$$

The first claim follows by subtracting NV − NE. If all R_i have maximal degree $n-1$, then the number of equations is equal to NE, which proves the second statement. □

Lemma 3.17. *Let $T, \ell, s \in \mathbb{N}$ such that $s \le \ell$ and $T + \ell(k-1) < sn$. If Problem 3.9 has a solution of degree T, it has at least $q^{\delta(T)-1}$ many such solutions.*

[8] If $\delta(T) < 0$, this statement means that #variables − #equations $\ge \delta(T)$.
[9] It might be fewer if e.g. $\deg R_i < n-1$ for some i.

Proof. Degree-T solutions of Problem 3.9 are those solutions of the system of linear equations in Lemma 3.16 where the T^{th} coefficient of λ_0 is one. In this way, we obtain an inhomogeneous system of equations, whose matrix, say $\boldsymbol{A}$, has a kernel of dimension $\dim\ker \boldsymbol{A} \geq \delta(T) - 1$. If the problem has a solution of degree T, then this system has at least $q^{\dim\ker \boldsymbol{A}}$ solutions, which proves the claim. □

Theorem 3.18. *Let $T, \ell, s \in \mathbb{N}$ such that $T = s|\mathcal{E}|$, $s \leq \ell$, $T + \ell(k-1) < sn$, and*

$$\frac{T}{s} = |\mathcal{E}| > \tau_{\text{new}} := n\left(1 - \frac{s\binom{h+s-1}{h} - h\binom{h+s-1}{h+1}}{s\binom{h+\ell}{h}}\right) - \frac{h}{h+1}\frac{\ell}{s}(k-1) - \frac{1}{s}\left(1 - \frac{2}{\binom{h+\ell}{h}}\right). \tag{3.10}$$

Then, Problem 3.9 has at least two solutions of degree T.

Proof. If (3.10) is satisfied, then we have $\delta(T) > 1$. Due to $T = s|\mathcal{E}|$, the problem has at least one degree-T solution by Theorem 3.11, i.e., $\lambda_i = \Lambda_i$ and $\psi_j = \Psi_j$. Hence, the problem has at least $q^{\delta(T)-1} > 1$ solutions by Lemma 3.17. □

We can interpret Theorem 3.18 as follows: If the number of errors $|\mathcal{E}|$ is greater than τ_{new}, then $\lambda_i = \Lambda_j$ and $\psi_j = \Psi_j$ cannot be a unique solution of Problem 3.9, and Algorithm 1 will most likely fail. On the other hand, if there are sufficiently many linearly independent equations in the system of linear equations given in Lemma 3.16, then Problem 3.9 has no other solution of degree T for $T \leq s|\mathcal{E}|$, and Algorithm 1 succeeds up to τ_{new} many errors.[10] This motivates to call τ_{new} the *decoding radius*. As we will see in Section 3.1.5, our simulation results indicate that this is indeed the radius up to which most errors can be corrected.

Remark 3.19. *In case of $T + \ell(k-1) \geq sn$, the numerical difference of variables and equations of the system in Lemma 3.16 is strictly larger than predicted by $\delta(T)$. Hence, the actual decoding radius might be smaller in this case. We will, however, see in the following that our proposed choice of the parameters ℓ and s implies that for $T \leq s\tau_{\text{new}}$, we always have $T + \ell(k-1) < sn$.*

Asymptotic Analysis

We analyze the asymptotic behavior of the decoding radius τ_{new} over the choices of the decoder parameters ℓ and s. The following lemma helps us to understand the limit of the binomial coefficient fractions occurring in the expression of τ_{new}.

Lemma 3.20. *Let $h \in \mathbb{N}$ and $\gamma \in (0,1)$ be fixed. Then, for $i \in \mathbb{N}$, we have*

$$\frac{\binom{h+\lfloor \gamma i \rfloor}{h}}{\binom{h+i}{h}} = \gamma^h + O\left(\tfrac{1}{i}\right).$$

Proof. Using Stirling's formula $(*)$ and

$$1 \leq \sqrt{\tfrac{(h+\gamma i)i}{(h+i)\gamma i}} = \sqrt{1 + \tfrac{(1-\gamma)m}{\gamma i}} \leq 1 + \tfrac{(1-\gamma)m}{\gamma i}, \tag{3.11}$$

$$\left|e^x - \left(1 + \tfrac{x}{i}\right)^i\right| = O\left(\tfrac{1}{i}\right) \quad \forall\, x \in \mathbb{R}, \tag{3.12}$$

$$\left(\tfrac{h+\gamma i}{h+i}\right)^h = \gamma^h + \sum_{j=1}^{h} \binom{h}{j} \gamma^{m-j} \left(\tfrac{(1-\gamma)m}{m+i}\right)^j = \gamma^h + O\left(\tfrac{1}{i}\right), \tag{3.13}$$

[10] This condition resembles, but seems weaker than, the "algebraic independence assumption" in [CH13].

we obtain

$$\frac{\binom{h+\lfloor\gamma i\rfloor}{h}}{\binom{h+i}{h}} = \frac{(h+\lfloor\gamma i\rfloor)!i!}{\lfloor\gamma i\rfloor!(h+i)!} \overset{(*)}{=} \overbrace{\sqrt{\frac{(h+\gamma i)i}{(h+i)\gamma i}}}^{\overset{(3.11)}{=}1+O(\frac{1}{i})} \cdot \frac{(h+\gamma i)^{h+\gamma i} i^i}{(\gamma i)^{\gamma i}(h+i)^{h+i}} \cdot \overbrace{e^{(h+\gamma i)+i-\gamma i-(h+i)}}^{=1} + O\left(\tfrac{1}{i}\right)$$
$$= \underbrace{\left(\frac{h+\gamma i}{h+i}\right)^h}_{\underset{(3.13)}{=}\gamma^h+O(\frac{1}{i})} \cdot \underbrace{\frac{(h+\gamma i)^{\gamma i}}{(\gamma i)^{\gamma i}}}_{\underset{(3.12)}{=}e^h+O(\frac{1}{i})} \cdot \underbrace{\frac{(i)^i}{(h+i)^i}}_{\underset{(3.12)}{=}e^{-h}+O(\frac{1}{i})} + O\left(\tfrac{1}{i}\right) = \gamma^h + O\left(\tfrac{1}{i}\right),$$

which proves the claim. □

Using Lemma 3.20, we derive the following asymptotic expression for τ_{new}. Theorem 3.21 is constructive as it provides a sequence of parameters that asymptotically achieve the limit.

Theorem 3.21. *Let $[\ell_i, s_i] = [i, \lfloor \gamma i \rfloor + 1]$ for $i \in \mathbb{Z}_{>0}$, where $\gamma = \sqrt[h+1]{\frac{k-1}{n}}$. Then,*

$$\tau_{\text{new}}(\ell_i, s_i) = n\Big(1 - \Big(\tfrac{k-1}{n}\Big)^{\frac{h}{h+1}} - O\Big(\tfrac{1}{i}\Big)\Big).$$

Proof. We choose $[\ell_i, s_i]$ as in the theorem statement. Using Lemma 3.20, we obtain

$$\tfrac{\tau_{\text{new}}}{n} = 1 - \Big[1 - \underbrace{\big(1-\tfrac{1}{s_i}\big)}_{=1-O(\frac{1}{i})}\tfrac{h}{h+1}\Big] \underbrace{\frac{\binom{h+s_i-1}{h}}{\binom{h+\ell_i}{h}}}_{=\gamma^h+O(\frac{1}{i})} - \tfrac{h}{h+1}\underbrace{\tfrac{\ell_i}{s_i}}_{\substack{=\gamma^{-1}\\+O(\frac{1}{i})}}\tfrac{k-1}{n} - \underbrace{\tfrac{1}{s_i}}_{=O(\frac{1}{i})}\underbrace{\Big[1-\tfrac{2}{\binom{h+\ell}{h}}\Big]}_{=1-O(\frac{1}{i})}$$
$$= 1 + \tfrac{m}{m+1}\underbrace{\Big(\gamma^h - \gamma^{-1}\tfrac{k-1}{n}\Big)}_{=0} - \gamma^h - O\left(\tfrac{1}{i}\right) = 1 - \Big(\tfrac{k-1}{n}\Big)^{\frac{h}{h+1}} - O\left(\tfrac{1}{i}\right),$$

which proves the claim. □

Corollary 3.22. *For a fixed code of rate $R = \frac{k}{n}$ and any constant $\varepsilon > 0$, we can choose $s, \ell \in O(1/\varepsilon)$ such that $\tau_{\text{new}} \geq n(1 - R^{\frac{h}{h+1}} - \varepsilon)$. In this case, the algorithm has complexity $O^{\sim}\Big((1/\varepsilon)^{h\omega+1}n\Big)$ in operations over $\mathbb{F}_q$, where ω is the matrix multiplication exponent.*

Remark 3.23. *The choice of $\ell \leq s(\frac{k-1}{n})^{-\frac{1}{h+1}}$ in Theorem 3.21 yields, for $T \leq s\tau_{\text{new}}$,*

$$T + \ell(k-1) < sn(1 - (\tfrac{k-1}{n})^{\frac{h}{h+1}}) + s(\tfrac{k-1}{n})^{-\frac{1}{h+1}}(k-1) = sn,$$

and therefore always fulfills the condition $T + \ell(k-1) < sn$ discussed in Remark 3.19.

3.1.4 Failure Behavior

As any other power decoding algorithm, the new decoder sometimes fails to decode below the maximal decoding radius derived in the previous subsection. By failure, we mean that for a received word $\boldsymbol{r} = \boldsymbol{c} + \boldsymbol{e}$, the codeword $\boldsymbol{c}$ does not correspond to a unique minimal solution of Problem 3.9.[11] We start with the observation that the success is independent of the codeword.

[11] Note that this kind of failure might not be detected by the decoder. E.g., it might find a solution corresponding to a codeword although there are other solutions of the same degree.

Theorem 3.24. *The success of decoding $\boldsymbol{r} = \boldsymbol{c} + \boldsymbol{e}$ depends only on the error $\boldsymbol{e}$.*

Proof. The proof is analogous to the one of [Ros18, Proposition 5.1]. We show that if decoding $\boldsymbol{r}$ fails, then for any $\hat{\boldsymbol{c}} \in \mathcal{C}_{\mathrm{IRS}}$ with corresponding message polynomial $\hat{\boldsymbol{f}}$, decoding $\boldsymbol{r} + \hat{\boldsymbol{c}}$ also fails. If decoding $\boldsymbol{r}$ fails, there is a minimal solution $\lambda_{\boldsymbol{i}}, \psi_{\boldsymbol{j}}$ of Problem 3.9 with input $\boldsymbol{r}$ such that $\lambda_{\boldsymbol{0}} \neq \Lambda^s$ and $\deg \lambda_{\boldsymbol{0}} \leq \deg \Lambda^s$. We prove that $\hat{\psi}_{\boldsymbol{j}} := \sum_{\boldsymbol{i} \preceq \boldsymbol{j}} \binom{\boldsymbol{j}}{\boldsymbol{i}} \hat{f}^{\boldsymbol{i}} \psi_{\boldsymbol{j}-\boldsymbol{i}}$ and $\hat{\lambda}_{\boldsymbol{i}} := \lambda_{\boldsymbol{i}}$ is a minimal solution Problem 3.9 with input $\boldsymbol{r} + \hat{\boldsymbol{c}}$ (using $\hat{\boldsymbol{R}} := \boldsymbol{R} + \hat{\boldsymbol{f}}$), so that decoding $\boldsymbol{r} + \hat{\boldsymbol{c}}$ also fails. The polynomials $\hat{\lambda}_{\boldsymbol{i}}$ and $\hat{\psi}_{\boldsymbol{j}}$ indeed fulfill the equalities and congruences of Problem 3.9 with input $\boldsymbol{r} + \hat{\boldsymbol{c}}$ since

$$\begin{aligned}
\sum_{\substack{\boldsymbol{i} \preceq \boldsymbol{j} \\ |\boldsymbol{i}| < s}} \hat{\lambda}_{\boldsymbol{i}} \tbinom{\boldsymbol{j}}{\boldsymbol{i}} \hat{\boldsymbol{R}}^{\boldsymbol{j}-\boldsymbol{i}} G^{|\boldsymbol{i}|} &= \sum_{\substack{\boldsymbol{i} \preceq \boldsymbol{j} \\ |\boldsymbol{i}| < s}} \sum_{\boldsymbol{h} \preceq \boldsymbol{j}-\boldsymbol{i}} \lambda_{\boldsymbol{i}} \underbrace{\tbinom{\boldsymbol{j}}{\boldsymbol{i}} \tbinom{\boldsymbol{j}-\boldsymbol{i}}{\boldsymbol{h}}}_{= \binom{\boldsymbol{j}}{\boldsymbol{h}} \binom{\boldsymbol{j}-\boldsymbol{h}}{\boldsymbol{i}}} \boldsymbol{R}^{\boldsymbol{j}-\boldsymbol{i}-\boldsymbol{h}} \hat{\boldsymbol{f}}^{\boldsymbol{h}} G^{|\boldsymbol{i}|} \\
&= \sum_{\boldsymbol{h} \preceq \boldsymbol{j}} \tbinom{\boldsymbol{j}}{\boldsymbol{h}} \hat{\boldsymbol{f}}^{\boldsymbol{h}} \sum_{\substack{\boldsymbol{i} \preceq \boldsymbol{j}-\boldsymbol{h} \\ |\boldsymbol{i}| < s}} \lambda_{\boldsymbol{i}} \tbinom{\boldsymbol{j}-\boldsymbol{h}}{\boldsymbol{i}} \boldsymbol{R}^{\boldsymbol{j}-\boldsymbol{h}-\boldsymbol{i}} G^{|\boldsymbol{i}|} \\
&= \begin{cases} \sum_{\boldsymbol{h} \preceq \boldsymbol{j}} \binom{\boldsymbol{j}}{\boldsymbol{h}} \hat{\boldsymbol{f}}^{\boldsymbol{h}} m \psi_{\boldsymbol{j}-\boldsymbol{h}} = \hat{\psi}_{\boldsymbol{j}}, & \text{if } |\boldsymbol{j}| < s \\ \sum_{\boldsymbol{h} \preceq \boldsymbol{j}} \binom{\boldsymbol{j}}{\boldsymbol{h}} \hat{\boldsymbol{f}}^{\boldsymbol{h}} \psi_{\boldsymbol{j}-\boldsymbol{h}} + \xi G^s \equiv \hat{\psi}_{\boldsymbol{j}} \mod G^s, & \text{for some } \xi \in \mathbb{F}_q[x] \text{ if } |\boldsymbol{j}| \geq s. \end{cases}
\end{aligned}$$

Also, $\deg \hat{\psi}_{\boldsymbol{j}} \leq \min_{\boldsymbol{h}} \{\deg \psi_{\boldsymbol{j}-\boldsymbol{h}} + |\boldsymbol{h}|(k-1)\} \leq \deg \lambda_{\boldsymbol{0}} s + |\boldsymbol{j}|(k-1)$. □

In general, the success of the decoder is bound to the maximal number of linearly independent equations in the linear system described in Lemma 3.16. If the method fails, we can distinguish the following two cases:

i) Algorithm 1 formally returns "decoding failure", in which case the found minimal solution of Problem 3.9 is generic, i.e., does not correspond to a codeword.

ii) The decoder returns a different codeword $\boldsymbol{c}'$ corresponding to errors positions $\mathcal{E}'$ with $|\mathcal{E}'| \leq |\mathcal{E}|$.

A bound on the failure probability for a random error of given weight can be given as the union bound of the probabilities that the events occur. The latter probability can be estimated using classical coding theory. However, the first case is more difficult. We conjecture that its probability is closely related to the existence of a generic solution of degree $\leq s\tau_{\mathrm{new}}$ of Problem 3.9 with random input polynomials $A_{\boldsymbol{i},\boldsymbol{j}}$, but we cannot formally prove this statement. Both mentioned probabilities are exponentially decaying in $(\tau_{\mathrm{new}} - |\mathcal{E}|)$.

However, this crucial question requires further research. In fact, it is not even solved in general for the previous power-decoding-based algorithms [SSB06, SSB10, WZB14, Ros18]. The only known formal bounds are as follows:

- [SSB06, SSB10] ($h = s = 1$) gives an upper bound for the case $\ell \leq 2$.
- [WZB14] ($s = 1$) gives an upper bound for the case $\ell \leq 2$.[12]
- [Ros18] ($h = 1$) gives an upper bound for the case $s \leq 2$ and $\ell \leq 3$.

[12] The bound holds for general ℓ but yields values larger than 1 beyond the radius corresponding to $\ell = 2$.

For the interpolation-based interleaved RS decoders that potentially achieve the same decoding radius as our decoder, [PV04, Par07] ($h \leq 2$) and [CH13], the only known upper bound on the failure probability is given by Parvaresh and Vardy [PV04, Par07] for $h \leq 2$ and

$$|\mathcal{E}| \leq n\left(1 - \sqrt[3]{6(\tfrac{k-1}{n})^2}\right),$$

which is far below its asymptotic radius $n(1 - (\frac{k-1}{n})^{2/3})$.

Although we do not have a formal upper bound on the failure probability, the simulation results for various code and decoder parameters presented in the next subsection indicate that the new decoder is indeed able to decode up to τ_{new} errors with overwhelmingly large probability. Based on our observations and known formal bounds on the special cases mentioned above [SSB06, SSB10, WZB14, Ros18], we conjecture that the probability of failure for a random error $\boldsymbol{e}$ of weight $|\mathcal{E}|$ is upper-bounded by

$$\mathrm{P}_{\text{fail}} \leq q^{-b(\tau_{\text{new}} - |\mathcal{E}|)},$$

where q is the field size and $b > 1$ is a constant that depends on the code and decoder parameters n, k, h, ℓ, and s.

3.1.5 Simulation Results

In this section, we present results of Monte–Carlo simulations with the new decoder, based on an implementation of Algorithm 1 in SageMath v7.5 [S$^+$]. For a variety of code and decoder parameters, we estimate the failure probability P_{fail} for random errors of weight $|\mathcal{E}| = \tau$.

Table 3.1 shows the results for decoding at exactly the decoding radius $\tau = \lfloor \tau_{\text{new}} \rfloor$. For comparison, the table also displays the maximal decoding radii τ_{WZB} of the WZB [WZB14] and τ_{KL} of the KL [KL97] algorithm for the given code parameters.

Table 3.1: Observed Failure Rate $\hat{\mathrm{P}}_{\text{fail}}$ at $\tau = \lfloor \tau_{\text{new}} \rfloor$ errors for various code (q, n, k, m) and decoder (ℓ, s) parameters, measured by Monte–Carlo simulations of $N \in \{10^4, 10^6\}$ samples. For comparison: Decoding radii τ_{WZB} of [WZB14] and τ_{KL} of [KL97] for the chosen parameters.

q	n	k	m	$[\ell, s]$	$\lfloor \tau_{\text{new}} \rfloor$	$\hat{\mathrm{P}}_{\text{fail}}$	τ_{WZB}	τ_{KL}	N
257	257	86	2	$[3, 2]$	120	0	114	114	10^6
				$[4, 3]$	124	$1.1 \cdot 10^{-5}$	114	114	10^6
43	43	18	2	$[4, 3]$	18	$5.4 \cdot 10^{-4}$	16	16	10^6
17	17	3	2	$[3, 2]$	11	$1.9 \cdot 10^{-3}$	10	9	10^6
			4	$[4, 3]$	13	$2.8 \cdot 10^{-2}$	10	9	10^4
			5	$[5, 3]$	13	$1.9 \cdot 10^{-3}$	10	9	10^4
17	16	2	3	$[3, 2]$	12	$9.1 \cdot 10^{-5}$	12	10	10^6
				$[6, 3]$	13	$1.0 \cdot 10^{-1}$	12	10	10^4
16	16	3	3	$[2, 1]$	10	0	10	9	10^6
				$[3, 2]$	11	$2.1 \cdot 10^{-5}$	10	9	10^6

Compared to the Krachkovsky–Lee and Wachter-Zeh–Zeh–Bossert algorithm, we can often correct many more errors. For instance, we can correct 124 instead of 114 errors with the code $\mathcal{C}_{\mathrm{IRS}}(257, 86; 2)$ over $\mathbb{F}_{257}$ and decoder parameters $[\ell, s] = [4, 3]$, where the observed failure rate is $\approx 1.1 \cdot 10^{-5}$. Our decoder can also achieve a better observed failure rate at the same decoding radius. Consider the code $\mathcal{C}_{\mathrm{IRS}}(16, 2; 3)$ code over $\mathbb{F}_{17}$, which is was also numerically evaluated in [WZB14, Table 1]. The WZB algorithm corrects 12 errors with observed failure rate $6.2 \cdot 10^{-3}$. By choosing the parameters of our decoder $[3, 2]$, we observe a lower failure rate $9.1 \cdot 10^{-5}$ for the same number of errors. If we increase the decoder parameters to $[6, 3]$, we can correct one more error in $\approx 90\%$ of the samples.

In case of the $\mathcal{C}_{\mathrm{IRS}}(17, 3; 5)$ code over $\mathbb{F}_{17}$, we can almost decode up to the list decoding capacity $n - k = 14$: By choosing the decoder parameters $[\ell, s] = [5, 3]$, Algorithm 1 corrects 13 errors at an observed failure rate $\approx 1.9 \cdot 10^{-3}$.

As expected, decoding always failed in our simulations when the number of errors is chosen larger than τ_{new}. On the other hand, the estimated failure probability significantly decreases when the number of errors is below τ_{new}. For instance, the observed failure rate of the decoder with parameters $[3, 2]$ of the $\mathcal{C}_{\mathrm{IRS}}(17, 3; 2)$ code over $\mathbb{F}_{17}$ is $1.3 \cdot 10^{-5}$ at $\tau = \lfloor \tau_{\mathrm{new}} \rfloor - 1 = 10$ errors, compared to $1.9 \cdot 10^{-3}$ at 11 errors.

3.2 Improved Power Decoding of Interleaved One-Point Hermitian Codes

In this section, we adapt our results of the previous section to interleaved one-point Hermitian codes of maximal length $n = q^3$.[13] The main difference is that we are dealing with polynomials in $\mathcal{R}$. Recall from Chapter 2 that $\mathcal{R}$ is the ring of functions in $\mathbb{F}_{q^2}(\mathcal{H})$ that are regular in all points of the curve except for P_∞. The functions in $\mathcal{R}$ may be uniquely written as $\mathbb{F}_{q^2}$-linear combinations of the bivariate monomials $x^i y^j$ with y-degree $j < q$. In this case, multiplication in $\mathcal{R}$ corresponds to multiplication of the bivariate polynomials modulo the curve equation $y^q + x - x^{q+1} = 0$. Furthermore, recall that $g = \frac{1}{2}q(q-1)$ is the genus of the Hermitian curve. We use a similar notation for power decoding as in [NB15].

3.2.1 Key Equations

Let $\boldsymbol{r} = \boldsymbol{c} + \boldsymbol{e} \in \mathbb{F}_{q^2}^{h \times n}$ be the received word, where $\boldsymbol{c} \in \mathcal{C}_{\mathrm{IH}}(n, m_{\mathrm{H}}; h)$ is a codeword with corresponding message polynomial vector $\boldsymbol{f} \in \mathcal{L}(m_{\mathrm{H}} P_\infty)^h$, and $\boldsymbol{e}$ is an error word with burst error positions in $\mathcal{E}$ (cf. Section 2.2.3). Our decoder is based on a system of key equations that contains the following polynomials.

Definition 3.25. Let $s \in \mathbb{N}$. An *error locator polynomial* $\Lambda^{(s)}$ *of multiplicity* s is a non-zero monic element in $\mathcal{L}\left(-\sum_{i \in \mathcal{E}} s P_i + \infty P_\infty\right)$ of minimal degree.

Theorem 3.26. *The error locator polynomial of multiplicity s is unique and has degree*

$$s|\mathcal{E}| \leq \deg_{\mathcal{H}} \Lambda^{(s)} \leq s|\mathcal{E}| + g.$$

[13] The approach also works for any $n < q^3$, but then the evaluation points must be chosen carefully in order to retain the complexity statements.

Proof. The proof is similar to [NB15, Lemma 23]. Uniqueness is clear since the difference of two such polynomials would also be in $\mathcal{L}\left(-\sum_{i\in\mathcal{E}} sP_i + \infty P_\infty\right)$, but of smaller degree $\deg_{\mathcal{H}}$ (recall that two different monomials $x^i y^j$ with $j < q$ cannot have the same $\deg_{\mathcal{H}}$).

We prove the upper bound by showing that $\mathcal{L}\left(-\sum_{i\in\mathcal{E}} sP_i + \infty P_\infty\right)$ contains an element f of degree $\deg_{\mathcal{H}} f \leq s|\mathcal{E}| + g$. We can find such an element by a system of equations as follows: Consider the coefficients of f up to the monomial of degree $s|\mathcal{E}| + g$ to be indeterminates. Hence, there are at least $s|\mathcal{E}|+1$ many of them. The polynomial f is in $\mathcal{L}\left(-\sum_{i\in\mathcal{E}} sP_i + \infty P_\infty\right)$ if and only if it can be expanded into a power series $\sum_{j\geq s} \gamma_{i,j} {t_{P_i}}^j$ for all $i \in \mathcal{E}$, where t_{P_i} is a local parameter of P_i (e.g., take $t_{P_i} = x - \alpha_i$ if $P_i = (\alpha_i, \beta_i)$). Each $i \in \mathcal{E}$ thus puts s many homogeneous linear restrictions on the coefficients of f, given by the s lowest coefficients of the power series expansion. In total, we obtain a homogeneous linear system of $s|\mathcal{E}|$ equations and $s|\mathcal{E}| + 1$ unknowns, so there is a non-zero solution corresponding to an $f \in \mathcal{L}\left(-\sum_{i\in\mathcal{E}} sP_i + \infty P_\infty\right)$ of degree $\deg_{\mathcal{H}} f \leq s|\mathcal{E}| + g$.

There cannot be an error locator of degree smaller than $s|\mathcal{E}|$ since otherwise it would be contained in the Riemann–Roch space $\mathcal{L}(D)$ with $D = -\sum_{i\in\mathcal{E}} sP_i + (s|\mathcal{E}| - 1)P_\infty$. However, we have $\mathcal{L}(D) = \{0\}$ since $\deg D < 0$ (cf. Section 2.2.2). □

Lemma 3.27. *For each $i = 1, \ldots, h$, there is a polynomial $R_i \in \mathcal{R}$ with $\deg_{\mathcal{H}}(R_i) < n + 2g$ that satisfies $R_i(P_j) = r_{i,j}$ for all $P_j \in \mathcal{H}^*$ and can be computed in $O^{\sim}(n)$ operations over $\mathbb{F}_{q^2}$.*

Proof. This directly follows from [NB15, Lemma 6]. □

In the remainder of this section, we write $\boldsymbol{R} = [R_1, \ldots, R_h] \in \mathcal{R}^h$, where R_i is as Lemma 3.27 and choose $G \in \mathcal{R}$, with $\deg_{\mathcal{H}} G = n$, as

$$G = \textstyle\prod_{\alpha\in\mathbb{F}_{q^2}} (x - \alpha) = x^{q^2} - x.$$

Lemma 3.28. *Let $s \in \mathbb{N}$, $\boldsymbol{i} \in \mathbb{N}_0^h$ with $|\boldsymbol{i}| \leq s$, and $\Lambda^{(s)}$, $\boldsymbol{R}$, $\boldsymbol{f}$, G as above. There is a unique polynomial $\Omega_{s,\boldsymbol{i}} \in \mathcal{R}$ of degree $\deg_{\mathcal{H}} \Omega_{s,\boldsymbol{i}} \leq \deg_{\mathcal{H}} \Lambda^{(s)} + |\boldsymbol{i}|(2g - 1)$ such that*

$$\Lambda^{(s)}(\boldsymbol{f} - \boldsymbol{R})^{\boldsymbol{i}} = G^{|\boldsymbol{i}|}\Omega_{s,\boldsymbol{i}}.$$

Proof. We show that $\Lambda^{(s)}(\boldsymbol{f} - \boldsymbol{R})^{\boldsymbol{i}}) \in \mathcal{L}(-\mathrm{div}(G^{|\boldsymbol{i}|}) + \infty P_\infty)$. Existence and uniqueness of $\Omega_{s,\boldsymbol{i}}$ then follows since this Riemann–Roch space equals $G^{|\boldsymbol{i}|}\mathcal{R}$, cf. (2.1) on page 10. First note that $\mathrm{div}(G) = \sum_{j=1}^n P_j - nP_\infty$, so $-\mathrm{div}(G^{|\boldsymbol{i}|}) + \infty P_\infty = -|\boldsymbol{i}| \sum_{j=1}^n P_j + \infty P_\infty$. Hence, it suffices to show $v_{P_j}(\Lambda^{(s)}(\boldsymbol{f} - \boldsymbol{R})^{\boldsymbol{i}})) \geq |\boldsymbol{i}|$ for all $j = 1, \ldots, n$. In the case $j \in \mathcal{E}$, we have $v_{P_j}(\Lambda^{(s)}(\boldsymbol{f} - \boldsymbol{R})^{\boldsymbol{i}}) \geq v_{P_j}(\Lambda^{(s)}) \geq s \geq |\boldsymbol{i}|$. Since for each $\mu = 1, \ldots, h$, the polynomial $f_\mu - R_\mu$ has zeros at the non-error positions $j \notin \mathcal{E}$, we must have $v_{P_j}(\Lambda^{(s)}(\boldsymbol{f} - \boldsymbol{R})^{\boldsymbol{i}})) \geq v_{P_j}((\boldsymbol{f} - \boldsymbol{R})^{\boldsymbol{i}}) \geq |\boldsymbol{i}|$. The degree of $\Omega_{s,\boldsymbol{i}}$ follows from the degree bound $\deg_{\mathcal{H}}(R_\mu) < n + 2g$ and by comparing the degrees on both sides of the equation. □

The following theorem states the system of key equations that we use for decoding. Its formulation is similar to its IRS analog, Theorem 3.6 on page 29, with two major differences:

- All polynomials are in $\mathcal{R}$ instead of $\mathbb{F}_q[x]$.
- Instead of the s^{th} power of the ordinary error locator, Λ^s, we seek the error locator $\Lambda^{(s)}$ of multiplicity s, which is a natural generalization.[14]

[14] In $\mathbb{F}_q[x]$, Λ^s is the monic polynomial of smallest degree with multiplicity s at all evaluation points $\alpha_i \in \mathbb{F}_q$.

Theorem 3.29 (System of Key Equations)**.** *Let $\ell, s \in \mathbb{N}$ with $s \leq \ell$ and $\boldsymbol{f}$, Λ, G, $\boldsymbol{R}$, and $\Omega_{s,\boldsymbol{i}}$ be defined as above. Then, for all $\boldsymbol{j} \in \mathbb{N}_0^h$ with $1 \leq |\boldsymbol{j}| \leq \ell$, we have*

$$\Lambda^{(s)} \boldsymbol{f}^{\boldsymbol{j}} = \sum_{\boldsymbol{i} \preceq \boldsymbol{j}} \Omega_{s,\boldsymbol{i}} \binom{\boldsymbol{j}}{\boldsymbol{i}} \boldsymbol{R}^{\boldsymbol{j}-\boldsymbol{i}} G^{|\boldsymbol{i}|}, \qquad 1 \leq |\boldsymbol{j}| < s \tag{3.14}$$

$$\Lambda^{(s)} \boldsymbol{f}^{\boldsymbol{j}} \equiv \sum_{\substack{\boldsymbol{i} \preceq \boldsymbol{j} \\ |\boldsymbol{i}|<s}} \Omega_{s,\boldsymbol{i}} \binom{\boldsymbol{j}}{\boldsymbol{i}} \boldsymbol{R}^{\boldsymbol{j}-\boldsymbol{i}} G^{|\boldsymbol{i}|} \mod G^s, \qquad s \leq |\boldsymbol{j}| \leq \ell, \tag{3.15}$$

as a congruence over $\mathcal{R}$.

Proof. The statement follows by analogous arguments as Theorem 3.6, using Lemma 3.28. □

Analogous to (3.3) in the previous section, we define the following polynomials in $\mathcal{R}$:

$$\begin{aligned} \Psi_{\boldsymbol{j}} &:= \Lambda^{(s)} \boldsymbol{f}^{\boldsymbol{j}} && \text{(unknown at the receiver),} \\ \Lambda_{\boldsymbol{i}} &:= \Omega_{s,\boldsymbol{i}} && \text{(unknown at the receiver), and} \\ A_{\boldsymbol{i},\boldsymbol{j}} &:= \tbinom{\boldsymbol{j}}{\boldsymbol{i}} \boldsymbol{R}^{\boldsymbol{j}-\boldsymbol{i}} G^{|\boldsymbol{i}|} && \text{(known at the receiver).} \end{aligned} \tag{3.16}$$

Remark 3.30. *Theorem 3.29 is similar to the system of key equations for IRS decoding, Theorem 3.6 on page 29. However, the polynomials $\Psi_{\boldsymbol{j}}$, $\Lambda_{\boldsymbol{i}}$ and $A_{\boldsymbol{i},\boldsymbol{j}}$ as in* (3.16) *have different degree bounds (cf. Remark 3.8)*

$$\begin{aligned} \deg_{\mathcal{H}} \Psi_{\boldsymbol{j}} &= \deg_{\mathcal{H}} \Lambda^{(s)} + \deg_{\mathcal{H}}(\boldsymbol{f}^{\boldsymbol{j}}) \leq \deg_{\mathcal{H}} \Lambda_{\boldsymbol{0}} + |\boldsymbol{j}| m_{\mathrm{H}}, \\ \deg_{\mathcal{H}} \Lambda_{\boldsymbol{i}} &\leq \deg_{\mathcal{H}} \Lambda^{(s)} + |\boldsymbol{i}|(2g-1) = \deg_{\mathcal{H}} \Lambda_{\boldsymbol{0}} + |\boldsymbol{i}|(2g-1), \\ \deg_{\mathcal{H}} A_{\boldsymbol{i},\boldsymbol{j}} &\leq (n+2g-1)(|\boldsymbol{j}| - |\boldsymbol{i}|) + |\boldsymbol{i}| n = (n+2g-1)|\boldsymbol{j}| - |\boldsymbol{i}|(2g-1) \quad \text{for } \boldsymbol{i} \preceq \boldsymbol{j}. \end{aligned}$$

The bounds are tight for many received words: If for all μ with $j_\mu > 0$, we have $\deg_{\mathcal{H}} f_\mu = m_{\mathrm{H}}$, then the first bound is fulfilled with equality. The second and third bound are tight for received words with $\deg_{\mathcal{H}} R_\mu = n+2g-1$ for all μ with $i_\mu > 0$, and all μ with $j_\mu - i_\mu > 0$, respectively. Note that for a fixed μ, the polynomials f_μ and R_μ have these maximal degrees for a fraction of $1 - 1/q^2$ of codewords and received words by definition. This is a direct consequence of the choice of f_μ and the interpolation polynomial formula in [NB15, Lemma 6].

3.2.2 Solving the Key Equations

As in the IRS case in Section 3.1.2, we relax the non-linear system of key equations into a generalization of a multi-sequence linear feedback shift-register synthesis problem, whose degree restrictions are motivated by the degree bounds in Remark 3.30. By relaxed, we mean that we do not assume that the $\lambda_{\boldsymbol{i}}$ and $\psi_{\boldsymbol{j}}$, instead of $\Lambda_{\boldsymbol{i}}$ and $\Psi_{\boldsymbol{j}}$, are of the form as in (3.16).

Problem 3.31. *Given positive integers $s \leq \ell$, $A_{\boldsymbol{i},\boldsymbol{j}} = \binom{\boldsymbol{j}}{\boldsymbol{i}} \boldsymbol{R}^{\boldsymbol{j}-\boldsymbol{i}} G^{|\boldsymbol{i}|} \in \mathcal{R}$ as in* (3.16) *for all $\boldsymbol{i} \in \mathcal{I} := \{\boldsymbol{i} \in \mathbb{N}_0^h : 0 \leq |\boldsymbol{i}| < s\}$ and $\boldsymbol{j} \in \mathcal{J} := \{\boldsymbol{j} \in \mathbb{N}_0^h : 1 \leq |\boldsymbol{j}| \leq \ell\}$, and $G \in \mathcal{R}$ as in*

Section 3.2.1. Find $\lambda_{\boldsymbol{i}}, \psi_{\boldsymbol{j}} \in \mathcal{R}$ for $\boldsymbol{i} \in \mathcal{I}$ and $\boldsymbol{j} \in \mathcal{J}$ with monic $\lambda_{\boldsymbol{0}}$, such that

$$\psi_{\boldsymbol{j}} = \sum_{\boldsymbol{i} \in \mathcal{I}} \lambda_{\boldsymbol{i}} A_{\boldsymbol{i},\boldsymbol{j}} \qquad 1 \leq |\boldsymbol{j}| < s, \tag{3.17}$$

$$\psi_{\boldsymbol{j}} \equiv \sum_{\boldsymbol{i} \in \mathcal{I}} \lambda_{\boldsymbol{i}} A_{\boldsymbol{i},\boldsymbol{j}} \mod G^s \qquad s \leq |\boldsymbol{j}| \leq \ell, \tag{3.18}$$

$$\deg_{\mathcal{H}} \lambda_{\boldsymbol{0}} \geq \deg \lambda_{\boldsymbol{i}} - |\boldsymbol{i}|(2g-1) \qquad 0 \leq |\boldsymbol{i}| < s, \tag{3.19}$$

$$\deg_{\mathcal{H}} \lambda_{\boldsymbol{0}} \geq \deg \psi_{\boldsymbol{j}} - |\boldsymbol{j}| m_{\mathrm{H}} \qquad 1 \leq |\boldsymbol{j}| \leq \ell. \tag{3.20}$$

Definition 3.32. Consider an instance of Problem 3.31. A solution $\lambda_{\boldsymbol{i}}, \psi_{\boldsymbol{j}}$, $i \in \mathcal{I}$, $\boldsymbol{j} \in \mathcal{J}$ has *degree* $T \in \mathbb{N}_0$ if $\deg_{\mathcal{H}} \lambda_{\boldsymbol{0}} = T$. It is *minimal* if its degree is minimal among all solutions.

Our decoding strategy, which is outlined in Algorithm 2, is motivated by the following statement, which directly follows from the key equations and the degree bounds in Remark 3.30.

Theorem 3.33. *Consider an instance of Problem 3.31. Then, $\lambda_{\boldsymbol{i}} = \Lambda_{\boldsymbol{i}}$ and $\psi_{\boldsymbol{j}} = \Psi_{\boldsymbol{j}}$, where $\Lambda_{\boldsymbol{i}} \in \mathcal{R}$ and $\Psi_{\boldsymbol{j}} \in \mathcal{R}$ are chosen as in* (3.16)*, are a solution of the problem of degree $T = \deg_{\mathcal{H}} \Lambda^{(s)}$, where $s \cdot |\mathcal{E}| \leq T \leq s \cdot |\mathcal{E}| + g$.*

In the following, we show how to obtain a minimal solution of Problem 3.31 (Line 3 of Algorithm 2). If the found solution is $\lambda_{\boldsymbol{i}} = \Lambda_{\boldsymbol{i}}$ and $\psi_{\boldsymbol{j}} = \Psi_{\boldsymbol{j}}$, then we can proceed analogously to the IRS case in Section 3.1.2: Let $\boldsymbol{u}_i = [0, \ldots, 0, 1, 0, \ldots, 0]$ be the i^{th} unit vector for $i = 1, \ldots, h$. Then, $\lambda_{\boldsymbol{0}} = \Lambda^{(s)}$ and $\psi_{\boldsymbol{u}_i} = \Lambda^{(s)} f_i$, so we can obtain the message polynomials f_i by a division of $\psi_{\boldsymbol{u}_i}$ by $\lambda_{\boldsymbol{0}}$. Division in $\mathcal{R}$ can be performed efficiently as described in [NB15, Section V.C], using truncated power series.

Algorithm 2: Improved Power Decoder for h-Interleaved One-Point Hermitian Codes

Input: Received words $\boldsymbol{r}_i = \boldsymbol{c}_i + \boldsymbol{e}_i \in \mathbb{F}_{q^m}^{h \times n}$ for $i = 1, \ldots, h$ and positive integers $s \leq \ell$

Output: $\boldsymbol{f} \in \mathcal{L}(m_{\mathrm{H}} P_\infty)^h$ such that $\boldsymbol{c}_i = [f_i(P_1), \ldots, f_i(P_n)]$ for all $i = 1, \ldots, h$ is a codeword corresponding to a minimal $\deg \Lambda^{(s)}$; or "decoding failure".

1 Compute $\boldsymbol{R}$ and G as in Section 3.2.1 using the interpolation algorithm in [NB15]
2 $A_{\boldsymbol{i},\boldsymbol{j}} \leftarrow \binom{\boldsymbol{j}}{\boldsymbol{i}} \boldsymbol{R}^{\boldsymbol{j}-\boldsymbol{i}} G^{|\boldsymbol{i}|}$ for all $\boldsymbol{i} \preceq \boldsymbol{j}$
3 $\lambda_{\boldsymbol{i}}, \psi_{\boldsymbol{j}} \leftarrow$ Minimal solution of Problem 3.31 with input s, ℓ, $A_{\boldsymbol{i},\boldsymbol{j}}$, and G
4 **if** $\lambda_{\boldsymbol{0}}$ *divides all* $\psi_{\boldsymbol{u}_i}$ *(in $\mathcal{R}$) for $i = 1, \ldots, h$, where $\boldsymbol{u}_i$ is the i^{th} unit vector* **then**
5 $\boldsymbol{f} \leftarrow [\psi_{\boldsymbol{u}_1}/\lambda_{\boldsymbol{0}}, \ldots, \psi_{\boldsymbol{u}_h}/\lambda_{\boldsymbol{0}}]$ using the division algorithm in [NB15]
6 $\mathcal{E} \leftarrow$ Error set corresponding to $\boldsymbol{e}_i = \boldsymbol{r}_i - [f_i(\alpha_1), \ldots, f_i(\alpha_n)]$ for $i = 1, \ldots, h$
7 **if** $\lambda_{\boldsymbol{0}}$ *is the error locator of multiplicity s corresponding to $\mathcal{E}$* **then**
8 **return** $\boldsymbol{f}$
9 **return** *"decoding failure"*

Complexity

We show how to implement Algorithm 2 efficiently. Since Problem 3.31 is over $\mathcal{R}$, we cannot immediately apply the same methods as for the IRS case: We first need to reduce it to an equivalent problem over $\mathbb{F}_{q^2}[x]$, similar to [NB15]. For this purpose, we use the $\mathbb{F}_{q^2}[x]$-vector representation of an element of $\mathcal{R}$. Any polynomial $a \in \mathcal{R}$ can be uniquely written

as $a = \sum_{i=0}^{q-1} a_i y^i \in \mathcal{R}$ with $a_i \in \mathbb{F}_{q^2}[x]$. Then, the *vector representation* of a is defined by $\boldsymbol{\nu}(a) = [a_0, \dots, a_{q-1}] \in \mathbb{F}_{q^2}[x]^q$, cf. [NB15]. We can determine the degree of a over $\mathcal{R}$ from its vector representation by

$$\deg_{\mathcal{H}}(a) = \max_i \{q \deg(a_i) + i(q+1)\}, \tag{3.21}$$

Furthermore, the vector representations of $a, b \in \mathcal{R}$ fulfill

$$\boldsymbol{\nu}(a+b) = \boldsymbol{\nu}(a) + \boldsymbol{\nu}(b), \qquad \boldsymbol{\nu}(ab) = \boldsymbol{\nu}(a)\boldsymbol{\mu}(b)\boldsymbol{\Xi},$$

where $\boldsymbol{\mu}(b) \in \mathbb{F}_{q^2}[x]^{q\times(2q-1)}$ and $\boldsymbol{\Xi} \in \mathbb{F}_{q^2}[x]^{(2q-1)\times q}$ are defined by

$$\boldsymbol{\mu}(b) := \begin{bmatrix} b_0 & b_1 & b_2 & \dots & b_{q-1} & & & \\ & b_0 & b_1 & \dots & b_{q-2} & b_{q-1} & & \\ & & \ddots & \ddots & \dots & \ddots & \ddots & \\ & & & b_0 & b_1 & \dots & b_{q-2} & b_{q-1} \end{bmatrix}, \quad \boldsymbol{\Xi} := \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \\ x^{q+1} & -1 & & \\ & x^{q+1} & -1 & \\ & & \ddots & \ddots \\ & & & x^{q+1} & -1 \end{bmatrix},$$

cf. [NB15]. For any polynomial $c \in \mathbb{F}_{q^2}[x] \subseteq \mathcal{R}$, we have $\boldsymbol{\mu}(ac) = \boldsymbol{\mu}(a)c$. In the following, let $[q]$ denote $\{0, \dots, q-1\}$. We reformulate Problem 3.31 into the following equivalent problem over $\mathbb{F}_{q^2}[x]$ using the notation above.

Problem 3.34. *Let $\ell, s \in \mathbb{N}$ with $s \le \ell$, $\boldsymbol{R}, G \in \mathcal{R}$ as in Section 3.2.1, and*

$$\boldsymbol{A}^{(\boldsymbol{i},\boldsymbol{j})} := \boldsymbol{\mu}(A_{\boldsymbol{i},\boldsymbol{j}})\boldsymbol{\Xi} = \boldsymbol{\mu}\left(\tbinom{\boldsymbol{j}}{\boldsymbol{i}} \boldsymbol{R}^{\boldsymbol{j}-\boldsymbol{i}} G^{|\boldsymbol{i}|}\right)\boldsymbol{\Xi} \in \mathbb{F}_{q^2}[x]^{q\times q}$$

for all $\boldsymbol{i} \in \mathcal{I} := \{\boldsymbol{i} \in \mathbb{N}_0^h : 0 \le |\boldsymbol{i}| < s\}$ and $\boldsymbol{j} \in \mathcal{J} := \{\boldsymbol{j} \in \mathbb{N}_0^h : 1 \le |\boldsymbol{j}| \le \ell\}$. Find $\lambda_{\boldsymbol{i},\iota}, \psi_{\boldsymbol{j},\kappa} \in \mathbb{F}_{q^2}[x]$ for $\boldsymbol{i} \in \mathcal{I}$, $\boldsymbol{j} \in \mathcal{J}$, $\iota, \kappa \in [q)$, which are not all zero, such that

$$\psi_{\boldsymbol{j},\kappa} = \sum_{\boldsymbol{i}\in\mathcal{I}} \sum_{\iota=0}^{q-1} \lambda_{\boldsymbol{i},\iota} A_{\iota,\kappa}^{(\boldsymbol{i},\boldsymbol{j})}, \qquad 1 \le |\boldsymbol{j}| < s, \tag{3.22}$$

$$\psi_{\boldsymbol{j},\kappa} \equiv \sum_{\boldsymbol{i}\in\mathcal{I}} \sum_{\iota=0}^{q-1} \lambda_{\boldsymbol{i},\iota} A_{\iota,\kappa}^{(\boldsymbol{i},\boldsymbol{j})} \mod G^s, \qquad s \le |\boldsymbol{j}| \le \ell, \tag{3.23}$$

$$\max_{\iota\in[q]} \{q \deg \lambda_{\boldsymbol{0},\iota} + \iota(q+1)\} \ge q \deg \lambda_{\boldsymbol{i},\iota} + \iota(q+1) - |\boldsymbol{i}|(2g-1), \quad 0 \le |\boldsymbol{i}| < s,\ \iota \in [q), \tag{3.24}$$

$$\max_{\iota\in[q]} \{q \deg \lambda_{\boldsymbol{0},\iota} + \iota(q+1)\} \ge q \deg \psi_{\boldsymbol{j},\kappa} + \kappa(q+1) - |\boldsymbol{j}| m_{\mathrm{H}}, \qquad 1 \le |\boldsymbol{j}| \le \ell,\ \kappa \in [q). \tag{3.25}$$

Motivated by (3.21), we call $T = \max_{\iota\in[q)} \{q \deg \lambda_{\boldsymbol{0},\iota} + \iota(q+1)\}$ the *degree* of a solution $\lambda_{\boldsymbol{i},\iota}, \psi_{\boldsymbol{j},\kappa}$ of Problem 3.34. Also, a solution is called *monic* if the leading coefficient of the (unique) polynomial $\lambda_{\boldsymbol{0},\iota}$ that maximizes $\max_{\iota\in[q)} \{q \deg \lambda_{\boldsymbol{0},\iota} + \iota(q+1)\}$ is one. We establish the connection between Problem 3.31 and Problem 3.34 in the following.

Theorem 3.35. *Let $T \in \mathbb{N}_0$. The polynomials $\lambda_{\boldsymbol{i}}, \psi_{\boldsymbol{j}} \in \mathcal{R}$ for $\boldsymbol{i} \in \mathcal{I}$ and $\boldsymbol{j} \in \mathcal{J}$ are a degree-T solution of Problem 3.31 if and only if*

$$[\lambda_{\boldsymbol{i},0}, \dots, \lambda_{\boldsymbol{i},q-1}] := \boldsymbol{\nu}(\lambda_{\boldsymbol{i}}) \quad \text{and} \quad [\psi_{\boldsymbol{j},0}, \dots, \psi_{\boldsymbol{j},q-1}] := \boldsymbol{\nu}(\psi_{\boldsymbol{j}})$$

are a monic solution of degree T of Problem 3.34.

Proof. The equivalence of (3.17) and (3.22) is clear by the properties of the vector representation. Also the degree restrictions are equivalent due to (3.21). It remains to show that the congruences mean the same. If $\lambda_{\boldsymbol{i}}$ and $\psi_{\boldsymbol{j}}$ are a solution to Problem 3.31, then for $|\boldsymbol{j}| \geq s$, there is some $u_{\boldsymbol{j}} \in \mathcal{R}$ such that:

$$\psi_{\boldsymbol{j}} = \sum_{\substack{\boldsymbol{i} \preceq \boldsymbol{j} \\ |\boldsymbol{i}|<s}} \lambda_{\boldsymbol{i}} A_{\boldsymbol{i},\boldsymbol{j}} + u_{\boldsymbol{j}} G^s \quad \iff \quad \boldsymbol{\nu}(\psi_{\boldsymbol{j}}) = \sum_{\substack{\boldsymbol{i} \preceq \boldsymbol{j} \\ |\boldsymbol{i}|<s}} \boldsymbol{\nu}(\lambda_{\boldsymbol{i}}) \boldsymbol{\mu}(A_{\boldsymbol{i},\boldsymbol{j}}) \boldsymbol{\Xi} + \boldsymbol{\mu}(u_{\boldsymbol{j}}) G^s,$$

where $\boldsymbol{\mu}(G^s u_{\boldsymbol{j}}) = \boldsymbol{\mu}(u_{\boldsymbol{j}}) G^s$ due to $G^s \in \mathbb{F}_{q^2}[x]$, which implies the claim. □

Problem 3.34 is of a similar form as the problem discussed in [NB15, Section V.B], which can be solved with the following cost.

Lemma 3.36. *A minimal solution of Problem 3.31 with input $\ell, s, A_{\boldsymbol{i},\boldsymbol{j}}, G$ can be found in*

$$O^{\sim}\Big(\tbinom{h+\ell}{h}^{\omega} \ell n^{\frac{\omega+2}{3}}\Big) \subseteq O^{\sim}\Big(\ell^{h\omega} \ell n^{\frac{\omega+2}{3}}\Big)$$

operations over $\mathbb{F}_{q^2}$.

Proof. Problem 3.34 is of a similar kind as the one considered in [NB15]. For completeness, the technical details are contained in Appendix A.1. □

Theorem 3.37. *Algorithm 2 can be implemented in $O^{\sim}(\binom{h+\ell}{h}^{\omega} \ell n^{\frac{\omega+2}{3}})$ operations over $\mathbb{F}_{q^2}$.*

Proof. The pre- and post-computations in Algorithm 2 are negligible compared to Line 3 by similar arguments as in Theorem 3.13 and the fast interpolation and division algorithm over $\mathcal{R}$ in [NB15]. The claim follows by Lemma 3.36. □

3.2.3 Decoding Radius and Failure Behavior

Similar to the IRS case, we derive an upper bound on the degree of $\Lambda^{(s)}$ for which it can be a unique solution of Problem 3.31.

Lemma 3.38. *Let $T, \ell, s \in \mathbb{N}$ such that $s \leq \ell$ and $T + \ell m_{\mathrm{H}} < sn$. Then, all polynomials $\lambda_{\boldsymbol{i}}, \psi_{\boldsymbol{j}} \in \mathcal{R}$ for $\boldsymbol{i} \in \mathcal{I}$ and $\boldsymbol{j} \in \mathcal{J}$ that fulfill (3.17), (3.18), and the absolute degree restrictions*

$$\deg_{\mathcal{H}} \lambda_{\boldsymbol{i}} - |\boldsymbol{i}|(2g-1) \leq T, \tag{3.26}$$

$$\deg_{\mathcal{H}} \psi_{\boldsymbol{j}} - |\boldsymbol{j}| m_{\mathrm{H}} \leq T, \tag{3.27}$$

are exactly the solutions of a homogeneous linear system of equations over $\mathbb{F}_{q^2}$ with at least

$$\delta(T) = (T+1)\tbinom{h+\ell}{h} - n\Big[h\tbinom{h+s-1}{h+1} + s\tbinom{h+\ell}{h} - s\tbinom{h+s-1}{h}\Big] + m_{\mathrm{H}} h \tbinom{h+\ell}{h+1} - g\tbinom{h+\ell}{h}$$

more variables than equations.[15] *If $T \geq 2g-1$, there are received words for which this difference is exactly $\delta(T)$.*

[15] If $\delta(T) < 0$, this statement means that #variables − #equations $\geq \delta(T)$.

Proof. The proof works in a similar way as the one of Lemma 3.16 on page 33, but we need to take care that all polynomials are in $\mathcal{R}$. Due to $\deg_{\mathcal{H}} A_{\boldsymbol{i},\boldsymbol{j}} \le (n+2g-1)|\boldsymbol{j}| - (2g-1)|\boldsymbol{i}|$, we get

$$\deg_{\mathcal{H}} \Big(\sum_{\boldsymbol{i}\in\mathcal{I}} \lambda_{\boldsymbol{i}} A_{\boldsymbol{i},\boldsymbol{j}} \Big) \le T + |\boldsymbol{j}|(n+2g-1) \quad \forall\, \boldsymbol{j} \in \mathcal{J}.$$

Hence, for $|\boldsymbol{j}| < s$, the polynomial $\psi_{\boldsymbol{j}}$ has lower degree than the terms in $\sum_{\boldsymbol{i}\in\mathcal{I}} \lambda_{\boldsymbol{i}} A_{\boldsymbol{i},\boldsymbol{j}}$, where the coefficients of $\lambda_{\boldsymbol{i}}$ are considered to be indeterminates. For $|\boldsymbol{j}| \ge s$, $\psi_{\boldsymbol{j}}$ has also lower degree than the modulus G^s due to the condition $T + \ell m_{\mathrm{H}} < sn$. Thus, the coefficients of the polynomials $\sum_{\boldsymbol{i}\in\mathcal{I}} \lambda_{\boldsymbol{i}} A_{\boldsymbol{i},\boldsymbol{j}}$ (case $|\boldsymbol{j}| < s$) and $(\sum_{\boldsymbol{i}\in\mathcal{I}} \lambda_{\boldsymbol{i}} A_{\boldsymbol{i},\boldsymbol{j}} \bmod G^s)$ (case $|\boldsymbol{j}| \ge s$) of index greater than $T + |\boldsymbol{j}|m_{\mathrm{H}}$ must be zero. Since the coefficients are known linear combinations of the unknown coefficients of the $\lambda_{\boldsymbol{i}}$, this gives us a homogeneous linear system of equations (cf. Figure 3.3 on page 34).

We count the number of equations and variables. Recall that for $a, b \in \mathbb{N}_0$, $a < b$, there are at least $b - a - g$ and at most $b - a$ many monomials $x^i y^j \in \mathcal{R}$ with $j < q$ of degree $\deg_{\mathcal{H}}(x^i y^j) \in [a, b)$, cf. Section 2.2.2.

Due to the degrees of the polynomials, the number of equations for each $|\boldsymbol{j}| < s$ is at most

$$N_{\boldsymbol{j}} = (T + |\boldsymbol{j}|(n+2g-1)) - (T + |\boldsymbol{j}|m_{\mathrm{H}}) = |\boldsymbol{j}|(n+2g-1-m_{\mathrm{H}}).$$

The analysis is more complicated for $|\boldsymbol{j}| \ge s$: Since G^s is a polynomial only in x with x-degree $\deg_x(G^s) = sq^2$, the modulo-G^s operation forces the x-degree of all monomials in $(\sum_{\boldsymbol{i}\in\mathcal{I}} \lambda_{\boldsymbol{i}} A_{\boldsymbol{i},\boldsymbol{j}} \bmod G^s)$ to be at most $sq-1$. Thus, it can be written as

$$\left(\sum_{\boldsymbol{i}\in\mathcal{I}} \lambda_{\boldsymbol{i}} A_{\boldsymbol{i},\boldsymbol{j}} \bmod G^s \right) = \sum_{j=0}^{q-1} \sum_{i=0}^{sq^2-1} a_{ij} x^i y^j,$$

where $a_{ij} \in \mathbb{F}_{q^2}$ are again known linear combinations of the unknown coefficients of the $\lambda_{\boldsymbol{i}}$. Due to $\deg \psi_{\boldsymbol{j}} \le T + |\boldsymbol{j}|m_{\mathrm{H}}$, we get at most

$$N_{\boldsymbol{j}} := \Big(\sum_{j=0}^{q-1} sq^2 \Big) - \underbrace{|\{[i,j] \,:\, qi + (q+1)j \le T + |\boldsymbol{j}|m_{\mathrm{H}}\}|}_{= T + |\boldsymbol{j}|m_{\mathrm{H}} - g + 1} = sn - T - |\boldsymbol{j}|m_{\mathrm{H}} + g - 1$$

homogeneous linear equations. Using Lemma 3.15 repeatedly, we obtain the following upper bound on the number of equations:

$$\begin{aligned} \mathrm{NE} &:= \sum_{\boldsymbol{j}\in\mathcal{J}} N_{\boldsymbol{j}} = \sum_{1\le|\boldsymbol{j}|<s} |\boldsymbol{j}|(n+2g-1-m_{\mathrm{H}}) + \sum_{s\le|\boldsymbol{j}|\le\ell} (sn - T - |\boldsymbol{j}|m_{\mathrm{H}} + g - 1) \\ &= (n+2g-1)h\tbinom{h+s-1}{h+1} + (sn - T - 1 + g)\left(\tbinom{h+\ell}{h} - \tbinom{h+s-1}{h} \right) - m_{\mathrm{H}} h \tbinom{h+\ell}{h+1}. \end{aligned}$$

The number of variables (i.e., the number of $\mathbb{F}_{q^2}$-coefficients of the $\lambda_{\boldsymbol{i}}$) is at least

$$\mathrm{NV} := \Big(\sum_{\boldsymbol{i}\in\mathcal{I}} (T + |\boldsymbol{i}|(2g-1) + 1 - g) \Big) = (T+1-g)\tbinom{h+s-1}{h} + (2g-1)h\tbinom{h+s-1}{h+1},$$

which implies the claim by subtracting NV − NE.

If $T \ge 2g-1$, then all Weierstraß gaps are below the degree bounds of the $\lambda_{\boldsymbol{i}}$ and $\psi_{\boldsymbol{j}}$, so the number of variables is exactly equal to NV. Since there are received words with $\deg_{\mathcal{H}} R_i = n + 2g - 1$, we can have $\deg_{\mathcal{H}} \left(\sum_{\boldsymbol{i}\in\mathcal{I}} \lambda_{\boldsymbol{i}} A_{\boldsymbol{i},\boldsymbol{j}} \right) = T + |\boldsymbol{j}|(n+2g-1)$, so the derived NE also equals the number of equations in this case. □

From the above lemma, we get the following statement, whose proof works exactly as the one of Lemma 3.17.

Lemma 3.39. *Let $T, \ell, s \in \mathbb{N}$ such that $s \leq \ell$ and $T + \ell m_{\mathrm{H}} < sn$. If Problem 3.31 has a solution of degree T, then it has at least $(q^2)^{\delta(T)-1}$ many such solutions.*

Lemma 3.39 implies the following statement on the uniqueness of a minimal solution corresponding to the error locator polynomial.

Theorem 3.40. *Let $T, s, \ell \in \mathbb{N}$ such that $s \leq \ell$, $T = \deg_{\mathcal{H}} \Lambda^{(s)}$, $T + \ell m_{\mathrm{H}} < sn$, and*

$$T > T_{\max} := sn \left(1 - \frac{s\binom{h+s-1}{h} - h\binom{h+s-1}{h+1}}{s\binom{h+\ell}{h}}\right) - \frac{h}{h+1}\ell m_{\mathrm{H}} + \left(\frac{1}{\binom{h+\ell}{h}} - 1\right) + g. \tag{3.28}$$

Then, Problem 3.31 has at least two solutions of degree T.

Proof. Condition (3.28) holds if and only if $\delta(T) > 1$. Due to $T = \deg_{\mathcal{H}} \Lambda^{(s)}$, Problem 3.31 has a solution of degree T, which implies the claim using Lemma 3.39. □

We can interpret Theorem 3.40 as follows: If $\deg_{\mathcal{H}} \Lambda^{(s)} > T_{\max}$, then even if $\Lambda^{(s)}$ corresponds to a minimal solution of Problem 3.31, there are also other solutions of the same degree. Since Algorithm 2 only specifies that *one* minimal solution of the problem should be found, there is no reason to assume that the decoder succeeds in this case. In particular, if we choose a random minimal solution, then the decoder will fail with high probability.

Recall that $|\mathcal{E}| \leq \deg_{\mathcal{H}} \Lambda^{(s)} \leq s|\mathcal{E}|+g$, and—for a random error—often $\deg_{\mathcal{H}} \Lambda^{(s)} = s|\mathcal{E}|+g$. Thus, $\Lambda^{(s)}$ can usually not be a unique solution for $s|\mathcal{E}| + g > T_{\max}$, i.e., $|\mathcal{E}| > \frac{T_{\max}-g}{s}$, and for sure if $|\mathcal{E}| > \frac{T_{\max}}{s}$. This motivates to call

$$\tau_{\mathrm{H,new}} = \frac{T_{\max}-g}{s} = n\left(1 - \frac{s\binom{h+s-1}{h} - h\binom{h+s-1}{h+1}}{s\binom{h+\ell}{h}}\right) - \frac{h}{h+1}\frac{\ell}{s}m_{\mathrm{H}} + \frac{1}{s}\left(\frac{1}{\binom{h+\ell}{h}} - 1\right) \tag{3.29}$$

the *decoding radius* of Algorithm 2.

Remark 3.41. *As stated in Lemma 3.38, for $T \geq 2g-1$ and $T + \ell m_{\mathrm{H}} < sn$, there are received words such that the inhomogeneous system for computing monic degree-T solutions of Problem 3.31 has exactly $\delta(T) - 1$ more variables than equations. In fact, most received words fulfill this condition: For a fixed $\boldsymbol{j}$, we only require one $A_{\boldsymbol{i},\boldsymbol{j}}$ to fulfill the upper bound on its degree with equality, i.e., $\deg_{\mathcal{H}} A_{\boldsymbol{i},\boldsymbol{j}} = (n + 2g - 1)|\boldsymbol{j}| - (2g - 1)|\boldsymbol{i}|$ (cf. Remark 3.30). If the system contains enough linearly independent equations for T with $\delta(T) < 0$, then $\Lambda^{(s)}$ corresponds to a unique minimal solution.*

In the case $T < 2g-1$, the degree bounds of both $\lambda_{\mathbf{0}}$ and $\psi_{\mathbf{0}}$ are smaller than $2g-1$ (due to $m_{\mathrm{H}} \geq 2g-1$; those of all other $\lambda_{\boldsymbol{i}}$ and $\psi_{\boldsymbol{j}}$ are still at least $2g-1$). Thus, the maximal number of equations can be up to g smaller and there can be up to g more variables than predicted in Lemma 3.38. By the arguments above, the decoding radius is reduced by at most $2g/[s\binom{h+\ell}{h}]$.

For $T + \ell m_{\mathrm{H}} \geq sn$, the number of equations is also smaller than predicted by Lemma 3.38. As in the IRS case, we will see that our proposed choice of ℓ and s (see below) circumvents this case for $T \leq s\tau_{\mathrm{H,new}}$.

Similar to our decoder for interleaved RS codes in the previous section, the decoder fails to return the sent codeword $\boldsymbol{c}$ for some errors of weight less than the decoding radius. If this is the case, then Problem 3.31 either has a generic solution of degree $\leq \deg_{\mathcal{H}} \Lambda^{(s)}$ or there is another codeword $\boldsymbol{c}' \neq \boldsymbol{c}$ with corresponding error positions $\mathcal{E}'$ and $\Lambda^{(s)'}$ of degree $\deg_{\mathcal{H}} \Lambda^{(s)'} \leq \deg_{\mathcal{H}} \Lambda^{(s)}$. It is important to note that—in contrast to IRS codes—the latter case does not imply $|\mathcal{E}'| \leq |\mathcal{E}|$. A formal upper bound on the failure probability of the decoder remains an open question. However, the simulation results in the following subsection indicate that the new decoder can indeed correct up to $\tau_{\mathrm{H,new}}$ errors. Sometimes, it can even decode beyond this radius, in which case we usually have $\deg_{\mathcal{H}} \Lambda^{(s)} < s|\mathcal{E}| + g$.

Asymptotic Analysis and Parameter Choice

Using the same arguments as in the IRS case (cf. Theorem 3.21 on page 36), we obtain the following asymptotic decoding radius. The statement also implies a practical choice of the parameters ℓ and s.

Theorem 3.42. *Let $[\ell_i, s_i] = [i, \lfloor \gamma i \rfloor + 1]$ for $i \in \mathbb{N}$, where $\gamma = \sqrt[h+1]{\frac{m_{\mathrm{H}}}{n}}$. Then,*

$$\tau_{\mathrm{H,new}}(\ell_i, s_i) = n\Big(1 - \big(\tfrac{m_{\mathrm{H}}}{n}\big)^{\frac{h}{h+1}} - O(\tfrac{1}{i})\Big) \quad \text{for } (i \to \infty).$$

Corollary 3.43. *Let $\varepsilon > 0$ and $R = \frac{k}{n}$. Then, there are $\ell, s \in O(1/\varepsilon)$ such that $\tau_{\mathrm{H,new}} \geq n(1 - (R + \frac{g-1}{n})^{\frac{h}{h+1}} - \varepsilon)$. In this case, Algorithm 2 costs $O^{\sim}((1/\varepsilon)^{h\omega+1} n^{\frac{\omega+2}{3}})$.*

Remark 3.44. *The choice of ℓ and s in Theorem 3.42 ensures that the condition $T + \ell m_{\mathrm{H}} < sn$ is fulfilled for all $T \leq s \cdot \tau_{\mathrm{H,new}}(\ell, s)$.*

3.2.4 Simulation Results

This section presents Monte-Carlo simulations with $N \in \{10^2, 10^3, 10^4, 10^5\}$ samples for estimating the failure probability of the new decoder in a channel that randomly adds $\tau \in \mathbb{N}$ burst errors. Our implementation of Algorithm 2 in SageMath v7.5 [S$^+$] is based on the available implementations of the algorithms in [NB15] and [Ros18].

All simulated examples satisfy $\deg_{\mathcal{H}} \Lambda^{(s)} \geq s\tau \geq 2g - 1$. If this condition was not fulfilled, the decoder might not be able to decode up to $\tau_{\mathrm{H,new}}$ errors (cf. Remark 3.41).

Case $h = 1$

Since Rosenkilde's improved power decoder has so far not been adapted to one-point Hermitian codes, we first treat the special case $h = 1$ (i.e., no interleaving) and compare its failure probability ($\hat{\mathrm{P}}_{\mathrm{fail,IPD}}$) to the Guruswami–Sudan (GS) decoder ($\hat{\mathrm{P}}_{\mathrm{fail,GS}}$) using the implementation from [NB15]. The GS algorithm is guaranteed to return all codewords of distance

$$n[1 - \tfrac{s+1}{2(\ell+1)}] - \tfrac{\ell}{2s} m_{\mathrm{H}} - \tfrac{g}{s} = \tau_{\mathrm{H,new}} - \tfrac{g}{s} + \tfrac{\ell}{s(\ell+1)}$$

to the received word, but it is well-known that it often succeeds beyond this radius, cf. [NB15]. In our simulations, we consider values of τ greater than this guaranteed radius.

Table 3.2: Observed failure rate of the improved power ($\hat{\mathrm{P}}_{\mathrm{fail,IPD}}$) and Guruswami–Sudan ($\hat{\mathrm{P}}_{\mathrm{fail,GS}}$) decoder for one-point Hermitian codes ($h = 1$). Code parameters q, m, n, k, d^*. Decoder parameters ℓ, s. Number of errors τ ($^+\tau = \tau_{\mathrm{H,new}}$ decoding radius as in (3.29)). Number of experiments N.

q	m	n	k	d^*	ℓ	s	τ	$\hat{\mathrm{P}}_{\mathrm{fail,IPD}}$	$\hat{\mathrm{P}}_{\mathrm{fail,GS}}$	N
4	15	64	10	49	4	2	28	0	0	10^4
							29^+	0	$3.30 \cdot 10^{-3}$	10^4
							30	$9.93 \cdot 10^{-1}$	$9.39 \cdot 10^{-1}$	10^4
5	55	125	46	70	3	2	35	0	0	10^4
							36^+	0	$4.00 \cdot 10^{-4}$	10^4
							37	$9.57 \cdot 10^{-1}$	$9.60 \cdot 10^{-1}$	10^4
5	20	125	11	105	5	2	67	0	0	10^3
							68^+	0	$7.00 \cdot 10^{-3}$	10^3
							69	$9.91 \cdot 10^{-1}$	$9.60 \cdot 10^{-1}$	10^3
7	70	343	50	273	3	2	160	0	0	10^3
							161^+	0	0	10^3
							162	$9.78 \cdot 10^{-1}$	$9.86 \cdot 10^{-1}$	10^3
7	70	343	50	273	4	2	168	0	0	10^3
							169^+	0	0	10^3
							170	$9.79 \cdot 10^{-1}$	$2.2 \cdot 10^{-2}$	10^3
7	55	343	35	288	4	2	183	0	0	10^3
							184^+	0	0	10^3
							185	$9.82 \cdot 10^{-1}$	$1.9 \cdot 10^{-2}$	10^3

The results for various code (q, m, n, k, d^*), decoder (ℓ, s), and channel (τ) parameters are shown in Table 3.2. Both algorithms can correct most error patterns of weight up to $\tau_{\mathrm{H,new}}$, improving upon classical power decoding. Also, neither of the two algorithms is generally superior in terms of observed failure rate.

General Case

For $h \geq 1$, the previous best relative decoding radius is $\tau_{\mathrm{H,old}} = \frac{h}{h+1}(n - m_{\mathrm{H}})$ [BMS05, Kam14]. We present several simulation results in Table 3.3. Algorithm 2 corrected all error patterns up to weight $\tau_{\mathrm{H,new}}$ in the simulated samples. Furthermore, it failed with large observed probability when only one more error was added.

3.3 Comparison of the New Decoders

In this section, we compare the two new decoding algorithms for codes of the same length and overall field size, both to each other and to other codes and their decoders.

Table 3.3: Observed failure rate $\hat{\mathrm{P}}_{\mathrm{fail,IPD}}$ of Algorithm 2 for $h > 1$. Code parameters q, m_{H}, n, k, d^*, h. Decoder parameters ℓ, s. Number of errors τ ($^+\tau = \tau_{\mathrm{H,new}}$ as in (3.29)). Number of experiments N. Previous best decoding radius $\tau_{\mathrm{H,old}}$ [BMS05, Kam14].

q	m_{H}	n	k	d^*	h	ℓ	s	τ	$\hat{\mathrm{P}}_{\mathrm{fail,IPD}}$	N	$\tau_{\mathrm{H,old}}$
4	15	64	10	49	2	3	2	35^+	0	10^5	32
								36	$9.18 \cdot 10^{-1}$	10^3	32
4	15	64	10	49	2	5	3	36^+	0	10^3	32
								37	$9.31 \cdot 10^{-1}$	10^3	32
4	15	64	10	49	3	3	2	38^+	0	10^5	36
								39	$9.42 \cdot 10^{-1}$	10^3	36
4	15	64	10	49	3	4	3	39^+	0	10^2	36
								40	1	10^2	36
4	22	64	17	42	2	4	3	29^+	0	10^3	28
								30	$9.44 \cdot 10^{-1}$	10^3	28
5	20	125	11	105	2	3	2	79^+	0	10^5	70
								80	$9.37 \cdot 10^{-1}$	10^3	70
5	20	125	11	105	2	4	2	81^+	0	10^3	70
								82	$9.93 \cdot 10^{-1}$	10^3	70
5	20	125	11	105	3	3	2	86^+	0	10^3	78
								87	$9.94 \cdot 10^{-1}$	10^3	78
5	55	125	46	70	2	4	3	48^+	0	10^3	46
								49	$9.86 \cdot 10^{-1}$	10^3	46
7	90	343	70	253	2	3	2	183^+	0	10^3	168
								184	$9.72 \cdot 10^{-1}$	10^3	168
8	128	512	101	384	2	3	2	281^+	0	10^2	256
								282	1	10^2	256

Interleaved Reed–Solomon and One-Point Hermitian codes

We first compare RS, IRS, and IH codes of the same length and overall field size.[16] Using an isomorphism $\mathbb{F}_Q^h \simeq \mathbb{F}_{Q^h}$, an h-interleaved code over a field $\mathbb{F}_Q$ can be considered as a, not necessarily $\mathbb{F}_{Q^h}$-linear, code over $\mathbb{F}_{Q^h}$. A burst error then simply corresponds to a Q^h-ary symbol error. We compare the following decoding radii for a prime power q, $n = q^3$, $k < n$, and $h \in \mathbb{N}$:

i) $\tau_{\mathrm{RS}} = n(1 - (\frac{k-1}{n})^{\frac{1}{2}})$: Johnson radius for decoding an $[n, k]$ RS code over $\mathbb{F}_{q^{6h}}$, using one of the algorithms in [GS98, Wu08, Ros18].

ii) $\tau_{\mathrm{IRS}} = n(1 - (\frac{k-1}{n})^{\frac{2h}{2h+1}})$: Decoding radius of [CH13] and the new decoder in Section 3.1 for a $2h$-interleaved RS code over $\mathbb{F}_{q^3}$.

iii) $\tau_{\mathrm{IH}} = n(1 - (\frac{k+g-1}{n})^{\frac{3h}{3h+1}})$: Decoding radius of the new decoder in Section 3.2 for a $3h$-interleaved RS code over $\mathbb{F}_{q^2}$.

[16] A similar comparison was drawn in [Arm08] between the decoders in [BKY03] and [BMS05].

All codes have length $n = q^3$ and overall field size q^{6h}. The special case $h = 1$ is illustrated in Figure 3.4.

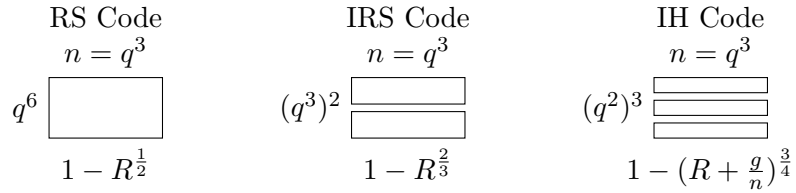


Figure 3.4: Decoding radii of RS, IRS, and IH codes of length $n = q^3$ and overall field size q^6.

From the expressions of τ_{RS}, τ_{IRS}, and τ_{IH}, it becomes clear that asymptotically, we have

$$\tau_{\text{RS}} < \tau_{\text{IRS}} < \tau_{\text{IH}}$$

for any $h \in \mathbb{N}$ since $\frac{g}{n} \to \infty$ for $n \to \infty$. We provide some examples for finite n below.

The prime power $q = 13$ is the smallest field size for which $\tau_{\text{IH}} > \tau_{\text{IRS}}$ for codes of rate $\approx 1/2$. In this case, the interleaved codes are $[2197, 1098]$ codes over $\mathbb{F}_{13^6}$, and the decoding radii are $t_{\text{RS}} = 644$, $\tau_{\text{IRS}} = 814$ and $\tau_{\text{IH}} = 823$. Table 3.4 gives decoding radii of rate $1/2$ codes for several values of even q.

Table 3.4: Examples of τ_{RS}, τ_{IRS}, and τ_{IH} for rate $R = 1/2$ codes of several lengths for $h = 1$.

q	$n = q^3$	$k = \frac{n}{2}$	τ_{RS}	τ_{IRS}	τ_{IH}	$\tau_{\text{IH}}/\tau_{\text{RS}} \approx$	$\tau_{\text{IH}}/\tau_{\text{IRS}} \approx$
2^3	512	256	150	190	183	1.22	0.96
2^4	4096	2048	1200	1516	1555	1.30	1.03
2^5	32768	16384	9598	12126	12844	1.34	1.06
2^6	262144	131072	76780	97004	104478	1.36	1.08
2^7	2097152	1048576	614242	776029	842936	1.37	1.09

Comparison to Folded and Univariate Multiplicity Codes

We compare our IRS decoder to *h-folded Reed–Solomon codes*, which are codes with an overall field size q^h and can be list-decoded beyond the Johnson radius [GR08, Theorem 4.4]. More precisely, for any $\varepsilon > 0$, there are h-folded RS codes of rate R with $h \in O(1/\varepsilon^2)$ such that the decoder returns all codewords of relative distance $1 - R - \varepsilon$ to the received word. Another family of codes that achieve $1-R-\varepsilon$ for an overall field size $q^{O(1/2\varepsilon^2)}$ are *univariate multiplicity codes* [Kop12], sometimes called *Derivative codes*.

In comparison, for $\varepsilon > 0$, an h-interleaved RS code with $h = \frac{\log(1/(R+\varepsilon))}{\log(1+\varepsilon/R)} \leq \frac{\log(1/R)}{\varepsilon/R} \in O(1/\varepsilon)$ can presumably be decoded by [CH13] and our decoding algorithm in Section 3.1 up to a fraction of

$$1 - R^{\frac{h}{h+1}} = 1 - R^{\frac{\log(R+\varepsilon)}{\log(R)}} = 1 - R - \varepsilon$$

errors. Compared to folded or univariate multiplicity codes, the IRS decoder's radius converges to the list decoding capacity $1 - R$ much faster in the overall field size q^h. The

comparison is not entirely fair, though: The decoders of the folded and univariate multiplicity codes codes are list decoders, whereas neither [CH13] nor we have a guarantee that the IRS decoder succeeds.

3.4 Concluding Remarks

In this chapter, we have presented new partial decoding algorithms for interleaved Reed–Solomon and interleaved one-point Hermitian codes, which join ideas of power decoding [SSB06, SSB10, Kam14, NB15] and improved power decoding [Ros18] for RS and one-point Hermitian codes, as well as collaborative decoding [SSB09, SSB07, WZB14, Kam14] of their interleaved variants.

The new decoder for IRS codes achieves, under certain assumptions, the decoding radius

$$n(1 - (\tfrac{k-1}{n})^{\frac{h}{h+1}}),$$

which is on par with the interpolation-based algorithms by [PV04, Par07] ($h = 2$) and [CH13], but at a better complexity. Simulation results indicate that for random errors, the expected decoding radius is achieved with high probability.

Our decoder for IH codes can correct, with presumably high probability, up to

$$n(1 - (\tfrac{k+g-1}{n})^{\frac{h}{h+1}}),$$

errors, improving upon the previous best radius by [BMS05, Kam14] at all rates.

Proving an analytic upper bound on the failure probability for random errors of a given weight remains an open question. The results on special cases of previous power decoding algorithms, cf. Section 3.1.4, can yield a partial answer to this problem. Furthermore, it appears possible to adapt improved power decoding to general interleaved AG codes, analog to the adaptions of the Sudan [SW99] or Guruswami–Sudan [GS98] algorithm.

Since all existing decoding algorithms for interleaved Reed–Solomon and one-point Hermitian codes are partial, it is an open problem to find a polynomial-time list decoding algorithm that guarantees to return all codewords within a distance greater than the Johnson radius. The decoding radius of such an algorithm cannot be greater than the one of a list decoder for a single, non-interleaved, constituent code: If one row of the interleaved received word contains a word with exponential list size in the constituent code, then also the list size of the interleaved decoder is exponential. Finding a polynomial-time list decoder for decoding Reed–Solomon codes beyond the Johnson radius has been, and still is, an open problem for many years.

4

Twisted Reed–Solomon Codes

Maximum distance separable (MDS) codes attain the maximal achievable minimum distance $d = n-k+1$ for a fixed code length n and dimension k. The codes have been an interesting object to study since Singleton [Sin64] presented his famous upper bound on the minimum distance of a code. In general, it is conjectured that, except for a few trivial examples, a linear MDS code over a field $\mathbb{F}_q$ must have length $n \leq q+2$. Presumably the most prominent MDS codes are generalized Reed–Solomon (GRS) codes [RS60], which—when extended—achieve the conjectured upper bound on the code length. There are many other constructions, e.g., based on finding n-arcs in projective geometry (see, e.g., [MS77]), Hankel matrices [RS85], or k-sum generators [RL89a].

Inspired by the construction of Sheekey's twisted Gabidulin codes [She16] in rank metric, we introduce a new class of linear evaluation codes in Hamming metric: Twisted Reed–Solomon codes. Compared to RS codes, where polynomials of degree smaller than the code dimension k are evaluated, we add further monomials, so-called "twists", of degree at least k to the evaluation polynomials. The twists' coefficients depend linearly on the k lowest coefficients. We give the formal definition in Section 4.1.

In Section 4.2, we show that by a suitable choice of the evaluation points and twist coefficients, the resulting twisted RS code is MDS. The length of this constructive subclass of MDS twisted RS codes is comparably small to the field size ($n \leq \sqrt{q}$), but the subsequent sections show that longer codes ($n \approx \frac{q}{2}$) are possible. A decoding algorithm is given in Section 4.3, whose complexity is polynomial in the code length for a fixed number of twists.

We study the duals of the new codes in Section 4.4. In Section 4.5, we investigate the Schur squares of low-rate twisted RS codes and show that many of them have much larger Schur square dimension than GRS codes of the same dimension. We use these findings to prove inequivalence of subfamilies of the new codes to GRS codes in Section 4.6. In addition, we provide a combinatorial argument showing that many more twisted RS codes are non-GRS.

Finally, we emphasize two notable properties of the new codes based on the results of the preceding sections: First, Section 4.7 presents two subclasses of MDS twisted RS codes that achieve lengths up to $\approx \frac{q}{2}$ for a given field size q, as well as computer searches for other long MDS twisted RS codes that are not equivalent to any Reed–Solomon code. Second, we analyze twisted RS codes for their suitability in the McEliece cryptosystem, cf. Section 4.8. An explicit subfamily resists some known structural attacks on the system based on RS-like codes, and other attacks at least do not apply in an obvious way.

The results of this chapter were partly published in [BPR17] and [BBPR18].

4.1 Definition

Idea and Goal

In this section, we define the new codes. We first provide an intuition of the construction and then present the formal definition. Reed–Solomon codes of dimension k are MDS since a polynomial of degree less than k,

$$f = \sum_{i=0}^{k-1} f_i x^i, \ f_i \in \mathbb{F}_q,$$

is evaluated at distinct evaluation points $\alpha_1, \ldots, \alpha_n \in \mathbb{F}_q$, and, as a non-zero f cannot have more than $k-1$ zeros, its corresponding codeword has Hamming weight at least $n-k+1$.

The idea of twisted Reed–Solomon codes is to add further monomials of degree at least k and to choose the coefficients of these new monomials as a (linear) function of the lowest k coefficients. Figure 4.1 shows several possible evaluation polynomial constructions (a)-(d).[1] We use such illustrations of the evaluation polynomials throughout this chapter.

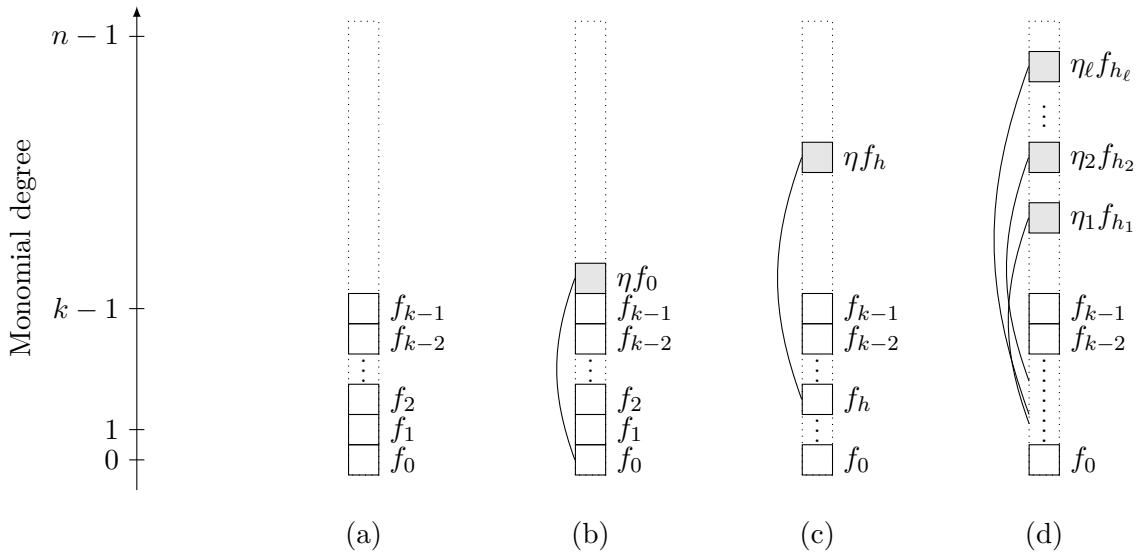


Figure 4.1: Illustration of evaluation polynomials of twisted RS codes. Boxes □ correspond to possibly non-zero coefficients. Arcs connect the corresponding hook and twist (grey background) coefficients (cf. Definition 4.1 below). Interpretation: See text.

Figure 4.1 (a) corresponds to evaluation polynomials of a Reed–Solomon code. The codes resulting from the polynomials in Figure 4.1 (b) can be seen as the $\mathbb{F}_q[x]$-analogs of Sheekey's [She16] twisted Gabidulin codes, where the additional twist has degree k and depends on the 0^{th} coefficient, i.e.,

$$f = \sum_{i=0}^{k-1} f_i x^i + \eta f_0 x^k,$$

[1] Note the analogy to the transform-domain illustration of RS and BCH codes in [Bla83] and [Bos13].

with $\eta \in \mathbb{F}_q$. We consider this case in Section 4.7.1. In Figure 4.1 (c), the construction is generalized such that the twist can be at any monomial degree $k-1+t$ for $1 \leq t \leq n-k$ and depend on any of the lowest k coefficients f_h for $0 \leq h < k$, i.e.,

$$f = \sum_{i=0}^{k-1} f_i x^i + \eta f_h x^{k-1+t}.$$

Our twisted RS codes arise from a further generalization of this construction (cf. Figure 4.1 (d)), which allows multiple ($\ell \leq n-k$ many) such twists, i.e.,

$$f = \sum_{i=0}^{k-1} f_i x^i + \sum_{j=1}^{\ell} \eta_j f_{h_j} x^{k-1+t_j}.$$

Our goal is to choose the coefficients $\eta_i \in \mathbb{F}_q$ and evaluation points $\alpha_i \in \mathbb{F}_q$, such that again any non-zero evaluation polynomial has at most $k-1$ roots among the evaluation points, resulting in MDS codes. We return to this problem in Section 4.2 and Section 4.7.

Formal Definition

The new codes are formally defined as follows.

Definition 4.1. Let $n, k, \ell \in \mathbb{N}$ be positive integers with $k < n$ and $\ell \leq n-k$. Choose a *hook vector* $\boldsymbol{h} \in \{0, \ldots, k-1\}^\ell$ and a *twist vector* $\boldsymbol{t} \in \{1, \ldots, n-k\}^\ell$ such that the t_i are distinct, and let $\boldsymbol{\eta} \in \mathbb{F}_q^\ell$. The set of $[k, \boldsymbol{t}, \boldsymbol{h}, \boldsymbol{\eta}]$-*twisted polynomials* over $\mathbb{F}_q$ is defined by

$$\mathcal{P}^{n,k}_{\boldsymbol{t},\boldsymbol{h},\boldsymbol{\eta}} = \left\{ f = \sum_{i=0}^{k-1} f_i x^i + \sum_{j=1}^{\ell} \eta_j f_{h_j} x^{k-1+t_j} : f_i \in \mathbb{F}_q \right\}.$$

Let $\alpha_1, \ldots, \alpha_n \in \mathbb{F}_q$ be distinct and write $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_n]$. The $[\boldsymbol{\alpha}, \boldsymbol{t}, \boldsymbol{h}, \boldsymbol{\eta}]$-*twisted Reed–Solomon code* of length n and dimension k is given by

$$\mathcal{C}_{\boldsymbol{\alpha},\boldsymbol{t},\boldsymbol{h},\boldsymbol{\eta}}[n,k] = \mathrm{ev}_{\boldsymbol{\alpha}}(\mathcal{P}^{n,k}_{\boldsymbol{t},\boldsymbol{h},\boldsymbol{\eta}}) \subseteq \mathbb{F}_q^n.$$

For brevity, we often say *twisted polynomials* and *twisted RS codes*[2], respectively.

The set of twisted polynomials $\mathcal{P}^{n,k}_{\boldsymbol{t},\boldsymbol{h},\boldsymbol{\eta}} \subseteq \mathbb{F}_q[x]$ forms a k-dimensional $\mathbb{F}_q$-linear subspace, so a twisted RS code is linear. The code $\mathcal{C}_{\boldsymbol{\alpha},\boldsymbol{t},\boldsymbol{h},\boldsymbol{\eta}}[n,k]$ indeed has dimension k since the evaluation map is injective: Any evaluation polynomial f satisfies $\deg f \leq k-1+\max_i\{t_i\} < n$.

4.2 A Sufficient Condition for Twisted RS Codes to be MDS

Not all twisted RS codes as in Definition 4.1 are MDS. In this section, we show that by choosing the evaluation points $\boldsymbol{\alpha}$ and the twist coefficients $\boldsymbol{\eta}$ in a suitable way, we can ensure that any non-zero evaluation polynomial has at most $k-1$ roots among the α_i, resulting in an MDS code due to its linearity. We start with a lemma that relates this property to a set of homogeneous systems of linear equations, whose coefficients depend on $\boldsymbol{\alpha}$ and $\boldsymbol{\eta}$, having only the zero solution.

[2] Twisted RS codes are not related to twisted BCH codes [EB97]. The name is inspired by Sheekey's twisted Gabidulin codes [She16], which are related to so-called *generalized twisted fields*.

Lemma 4.2. *Let $n, k, \boldsymbol{t}, \boldsymbol{h}, \boldsymbol{\eta}$ be chosen as in Definition 4.1 with $t_1 < t_2 < \cdots < t_\ell$ and $\eta_i \neq 0$ for all i. For any $\mathcal{I} \subseteq \{1, \ldots, n\}$ with $|\mathcal{I}| = k$, consider the homogeneous, linear system of equations in $g_0, \ldots, g_{t_\ell - 1} \in \mathbb{F}_q$:*

$$\sum_{j=0}^{t_\ell - 1} g_j T_{i,j}^{(\mathcal{I})} = 0 \quad \text{for } i = k, \ldots, k - 1 + t_\ell \ , \tag{4.1}$$

where

$$T_{i,j}^{(\mathcal{I})} = \begin{cases} \eta_\kappa^{-1} a_{i-j} - a_{h_\kappa - j}, & \text{if } i = k - 1 + t_\kappa \text{ for } \kappa \in \{1, \ldots, \ell\}, \\ a_{i-j}, & \text{otherwise,} \end{cases}$$

and $\sum_{i=0}^{k} a_i x^i := \prod_{i \in \mathcal{I}} (x - \alpha_i)$ and $a_i := 0$ for $i < 0$ or $i \geq k$. The twisted RS code $\mathcal{C}_{\boldsymbol{\alpha},\boldsymbol{t},\boldsymbol{h},\boldsymbol{\eta}}[n, k]$ is MDS if and only if there is no choice of $\mathcal{I}$ admitting a non-zero solution of the linear system (4.1).

Proof. Let $f \in \mathcal{P}_{\boldsymbol{t},\boldsymbol{h},\boldsymbol{\eta}}^{n,k}$ be a polynomial with at least k roots among the $\alpha_1, \ldots, \alpha_n$. This means that there is a set $\mathcal{I} \subseteq \{1, \ldots, n\}$ with $|\mathcal{I}| = k$ such that $a := \prod_{i \in \mathcal{I}} (x - \alpha_i)$ divides f, i.e., $f = g \cdot a$ for some $g = \sum_{i=0}^{t_\ell - 1} g_i x^i \in \mathbb{F}_q[x]$. Then the i^{th} coefficient of f is given by $f_i = \sum_{j=0}^{t_\ell - 1} g_j a_{i-j}$. On the other hand, for $i \geq k$, then f_i also satisfies

$$f_i = \begin{cases} \eta_\kappa f_{h_\kappa} & \text{if } i = k - 1 + t_\kappa \text{ for } \kappa \in \{1, \ldots, \ell\}, \\ 0 & \text{otherwise.} \end{cases}$$

For the case $i = k - 1 + t_\kappa$, combining and rewriting yields $0 = \sum_{j=0}^{t_\ell - 1} g_j \cdot (\eta_\kappa^{-1} a_{i-j} - a_{h_\kappa - j})$. Thus, for a given $\mathcal{I}$, a non-zero solution of the linear system given in (4.1) corresponds to a non-zero polynomial $f = g \cdot a \in \mathcal{P}_{\boldsymbol{t},\boldsymbol{h},\boldsymbol{\eta}}^{n,k}$ with $\mathrm{wt_H}(\mathrm{ev}_f(\boldsymbol{\alpha})) \leq n - k$. The code $\mathcal{C}_{\boldsymbol{\alpha},\boldsymbol{t},\boldsymbol{h},\boldsymbol{\eta}}[n, k]$ is MDS if and only if all non-zero $f \in \mathcal{P}_{\boldsymbol{t},\boldsymbol{h},\boldsymbol{\eta}}^{n,k}$ result in codewords of weight $\geq n - k + 1$, which is true if and only if System (4.1) has no non-zero solution for any set $\mathcal{I}$. □

For a given index set $\mathcal{I}$, System (4.1) is of the form as in Figure 4.2.

$$\underbrace{\begin{bmatrix} \eta_{t_\ell}^{-1} + \square & \square & \cdots & \square & \square & \square & \cdots & \square & \square & \cdots \\ \square & 1 & & & & & & & & \\ \vdots & \vdots & \ddots & & & & & & & \\ \square & \square & \cdots & 1 & & & & & & \\ \eta_{t_{\ell-1}}^{-1} \square + \square & \eta_{t_{\ell-1}}^{-1} \square + \square & \cdots & \eta_{t_{\ell-1}}^{-1} \square + \square & \eta_{t_{\ell-1}}^{-1} + \square & \square & \cdots & \square & \square & \cdots \\ \square & \square & \cdots & \square & \square & 1 & & & & \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & & & \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & & & \\ \square & \square & \cdots & \square & \square & \square & \cdots & 1 & & \\ \eta_{t_1}^{-1} \square + \square & \eta_{t_1}^{-1} \square + \square & \cdots & \eta_{t_1}^{-1} \square + \square & \eta_{t_1}^{-1} \square + \square & \eta_{t_1}^{-1} \square + \square & \cdots & \eta_{t_1}^{-1} \square + \square & \eta_{t_1}^{-1} + \square & \cdots \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots \end{bmatrix}}_{=: \boldsymbol{B}_{\mathcal{I}}} \begin{bmatrix} g_{t_\ell} \\ g_{t_\ell - 1} \\ \vdots \\ g_1 \\ g_0 \end{bmatrix} = \mathbf{0},$$

Figure 4.2: Illustration of the linear system in (4.1). Boxes □ represent elements obtained by multiplications and $\mathbb{F}_q$-linear combinations of $\alpha_1, \ldots, \alpha_n$. Diagonal elements are 1 (if the row corresponds to an index i with $i \neq k - 1 + t_\kappa$ for all κ) or $\eta_{t_\kappa}^{-1} + \square$ (if $i \neq k - 1 + t_\kappa$) due to $a_{i-(i-k)} = a_k = 1$ for all $i = k, k + 1, \ldots, k - 1 + t_\ell$ (note that the diagonal elements are the $T_{i,j}^{(\mathcal{I})}$ of (4.1) with $j = i - k$). Also, all elements above the diagonal do not depend on the $\eta_{t_\kappa}^{-1}$ since $a_{i-j} = 0$ for all $i > j + k$.

By choosing the evaluation points $\boldsymbol{\alpha}$ and twist coefficients $\boldsymbol{\eta}$ as in the following theorem, the system's matrix $\boldsymbol{B}_{\mathcal{I}}$ is always regular, independent of the choice of $\mathcal{I}$. Using Lemma 4.2, this proves that the resulting code is MDS. Theorem 4.3 is illustrated in Figure 4.3.

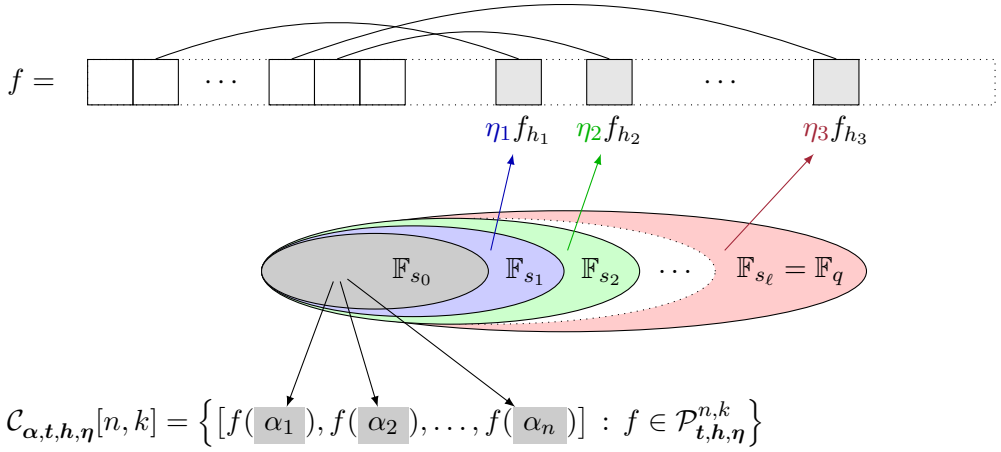


Figure 4.3: Illustration of the choices of $\boldsymbol{\alpha}$ and $\boldsymbol{\eta}$ in Theorem 4.3, resulting in MDS codes.

Theorem 4.3. *Let $s_0, \ldots, s_\ell \in \mathbb{N}$ such that $\mathbb{F}_{s_0} \subsetneq \mathbb{F}_{s_1} \subsetneq \cdots \subsetneq \mathbb{F}_{s_\ell} = \mathbb{F}_q$ is a chain of subfields. Let $k < n \leq s_0$ and $\alpha_1, \ldots, \alpha_n \in \mathbb{F}_{s_0}$ be distinct, and let $\boldsymbol{t}$, $\boldsymbol{h}$, and $\boldsymbol{\eta}$ be chosen as in Definition 4.1 with the additional requirements*

$$\eta_i \in \mathbb{F}_{s_i} \setminus \mathbb{F}_{s_{i-1}} \quad \forall\, i = 1, \ldots, \ell$$

and $t_1 < t_2 < \cdots < t_\ell$.[3] *Then, the twisted RS code $\mathcal{C}_{\boldsymbol{\alpha},\boldsymbol{t},\boldsymbol{h},\boldsymbol{\eta}}[n,k]$ is MDS.*

Proof. We prove the claim using Lemma 4.2. Let $\mathcal{I} \subseteq \{1, \ldots, n\}$ be a set of k elements. We show that the system (4.1) only has the zero solution. Since $a_i \in \mathbb{F}_{s_0}$, the boxes $\square$ of the system's matrix $\boldsymbol{B}_{\mathcal{I}}$ as in Figure 4.2 represent elements from $\mathbb{F}_{s_0}$.

We consider the $\eta_{t_\kappa}^{-1}$'s to be indeterminates. This means that $\det(\boldsymbol{B}_{\mathcal{I}}) \in \mathbb{F}_{s_0}[\eta_{t_1}^{-1}, \ldots, \eta_{t_\ell}^{-1}]$ is a multivariate polynomial, where each indeterminate $\eta_{t_\kappa}^{-1}$ appears at most of degree 1 in a monomial. Let $\boldsymbol{B}_{\mathcal{I}}^{(\mu)}$ be the $(\mu \times \mu)$-bottom-right submatrix of $\boldsymbol{B}_{\mathcal{I}}$. We distinguish two cases:

(i) If $\mu \neq t_\kappa$ for all κ, then the first row of $\boldsymbol{B}_{\mathcal{I}}^{(\mu)}$ is of the form $[1, 0, \ldots, 0]$ and by Laplace's rule we get

$$\det\left(\boldsymbol{B}_{\mathcal{I}}^{(\mu)}\right) = \det\left(\boldsymbol{B}_{\mathcal{I}}^{(\mu-1)}\right).$$

(ii) If $\mu = t_\kappa$ for some κ, then $\boldsymbol{B}_{\mathcal{I}}^{(\mu)}$ contains only $\eta_{t_1}^{-1}, \ldots, \eta_{t_{\kappa-1}}^{-1}$ in its rows 2 to μ and since the first row is of the form $[\eta_{t_\kappa}^{-1} + \square, \square, \ldots, \square]$, the determinant fulfills

$$\det\left(\boldsymbol{B}_{\mathcal{I}}^{(\mu)}\right) = (\eta_{t_\kappa}^{-1} + T_\mu) \cdot \det\left(\boldsymbol{B}_{\mathcal{I}}^{(\mu-1)}\right) + U_\mu \in \mathbb{F}_{s_0}[\eta_{t_1}^{-1}, \ldots, \eta_{t_\kappa}^{-1}],$$

where $T_\mu \in \mathbb{F}_{s_0}$ and $U_\mu \in \mathbb{F}_{s_0}[\eta_{t_1}^{-1}, \ldots, \eta_{t_{\kappa-1}}^{-1}]$.

[3] The entries of $\boldsymbol{t}$ and $\boldsymbol{h}$ can be permuted such that any twisted RS code fulfills the latter condition.

Combined, we get $\det(\boldsymbol{B}_{\mathcal{I}}^{(t_\kappa)}) = (\eta_{t_\kappa}^{-1} + T_{t_\kappa}) \det(\boldsymbol{B}_{\mathcal{I}}^{(t_{\kappa-1})}) + U_{t_\kappa} \in \mathbb{F}_{s_0}[\eta_{t_1}^{-1}, \dots, \eta_{t_\kappa}^{-1}]$, where $\det(\boldsymbol{B}_{\mathcal{I}}^{(t_1)}) = \eta_{t_1}^{-1}$, and by recursively substituting $\eta_\kappa \in \mathbb{F}_{s_\kappa} \setminus \mathbb{F}_{s_{\kappa-1}}$ for $\kappa = 1, \dots, \ell$, we obtain

$$\det\left(\boldsymbol{B}_{\mathcal{I}}^{(t_\kappa)}\right) \in \mathbb{F}_{s_\kappa} \setminus \{0\},$$

since $\eta_{t_\kappa}^{-1} \in \mathbb{F}_{s_\kappa} \setminus \mathbb{F}_{s_{\kappa-i}}$, $\det(\boldsymbol{B}_{\mathcal{I}}^{(t_{\kappa-1})}) \in \mathbb{F}_{s_{\kappa-1}} \setminus \{0\}$, and $U_{t_\kappa} \in \mathbb{F}_{s_{\kappa-1}}$. Hence, also $\det(\boldsymbol{B}_{\mathcal{I}}) = \det(\boldsymbol{B}_{\mathcal{I}}^{(t_\ell)}) \neq 0$ and System (4.1) has only the zero solution. □

Theorem 4.3 implies that if the code length n is a prime power, there is a twisted RS code with ℓ twists over the field of size $q = n^{2^\ell}$ (choose $s_i = n^{2^i}$). For a large number of twists, we therefore obtain codes whose length is much smaller than the field size.

We will see in Section 4.7 that for $\ell = 1$, we can improve this lower bound on the maximal length from $n \approx \sqrt{q}$ to $n \approx \frac{q}{2}$ in some cases, thereby obtaining codes close to the conjectured maximal length. The "short" codes given by Theorem 4.3 are nevertheless interesting, e.g., for cryptographic applications as discussed in Section 4.8.

4.3 Decoding

An efficient decoding algorithm is a key ingredient in the study of a code. If the coefficients $f_{h_i}\eta_i$ of the twist monomials x^{k-1+t_i} are known, we can subtract the twists from the evaluation polynomial and obtain a polynomial of degree less than k. Thus, if $\boldsymbol{c} = \mathrm{ev}_{\boldsymbol{\alpha}}(f)$, then $\boldsymbol{c}' = \boldsymbol{c} - \mathrm{ev}_{\boldsymbol{\alpha}}(\sum_{i=1}^{\ell} f_{h_i}\eta_i x^{k-1+t_i})$ is a codeword of the Reed–Solomon code $\mathcal{C}_{\mathrm{RS}}[n,k]$ with evaluation points $\boldsymbol{\alpha}$. Twisted RS codes can therefore be decoded based on a given RS decoder by brute-forcing the twist coefficients. The procedure is formally stated in Algorithm 3.

Algorithm 3: Twisted RS list decoder by brute-forcing the twists.

Input: Received word $\boldsymbol{r} \in \mathbb{F}_q^n$, RS (list) decoder RSdecoder$(\cdot)$ with decoding radius τ
Output: List $\mathcal{L}$ of codewords $\boldsymbol{c} \in \mathcal{C}_{\boldsymbol{\alpha},\boldsymbol{t},\boldsymbol{h},\boldsymbol{\eta}}[n,k]$ such that $\mathrm{d_H}(\boldsymbol{r},\boldsymbol{c}) \leq \tau$.

```
L ← ∅
for [g_1, ..., g_ℓ] ∈ F_q^ℓ do
    r' ← r − ev_α(Σ_{i=1}^{ℓ} g_i η_i x^{k−1+t_i})
    L' ← RSdecoder(r')
    for c' ∈ L' do
        c ← c' + ev_α(Σ_{i=1}^{ℓ} g_i η_i x^{k−1+t_i})
        if c ∈ C_{α,t,h,η}[n,k] and d_H(r,c) ≤ τ then
            L ← L ∪ {c}
return L
```

Theorem 4.4. *Let k, n, $\boldsymbol{\alpha}$, $\boldsymbol{t}$, $\boldsymbol{h}$, and $\boldsymbol{\eta}$ be chosen as in Definition 4.1. Furthermore, let* RSdecoder$(\cdot)$ *be a list decoder with decoding radius $\tau \in \mathbb{N}$, maximal list size $L(\tau)$, and complexity $\mathcal{T}(n,\tau)$ over $\mathbb{F}_q$ of the RS code $\mathcal{C}_{\mathrm{RS}}[n,k]$ with evaluation points $\boldsymbol{\alpha}$.*

Then, Algorithm 3 is correct, returns a list of size at most $q^\ell L(\tau)$, and can be implemented in $O(q^\ell \mathcal{T}(n,\tau))$ operations over $\mathbb{F}_q$.

Proof. Note that $\boldsymbol{c} \in \mathcal{C}_{\boldsymbol{\alpha},\boldsymbol{t},\boldsymbol{h},\boldsymbol{\eta}}[n,k]$ such that $\mathrm{d_H}(\boldsymbol{r},\boldsymbol{c}) \leq \tau$ if

$$\boldsymbol{c}' = \boldsymbol{c} - \mathrm{ev}_{\boldsymbol{\alpha}}\Big(\sum_{i=1}^{\ell} f_{h_i}\eta_i x^{k-1+t_i}\Big) \in \mathcal{C}_{\mathrm{RS}}$$

with $\mathrm{d_H}(\boldsymbol{r}',\boldsymbol{c}') \leq \tau$, where $\boldsymbol{r}' = \boldsymbol{r} - \mathrm{ev}_{\boldsymbol{\alpha}}(\sum_{i=1}^{\ell} f_{h_i}\eta_i x^{k-1+t_i})$. Thus, the algorithm finds all sought codewords in the iteration corresponding to $g_i = f_{h_i}$ for all $i = 1,\ldots,\ell$ and the overall list size is the same as the one of the RS decoder in this iteration. The complexity of the algorithm is determined by Line 4, which is executed q^ℓ times. The list size statement follows from $|\mathcal{L}'| \leq L(\tau)$. □

By evaluating Theorem 4.4 for different decoders, e.g., a bounded minimum distance (BMD) decoder or the Guruswami–Sudan list decoder [GS98], we obtain the decoding radii, list sizes and complexities in Table 4.1. The last row of the table gives the resulting values in case of the improved power decoder for IRS codes from Chapter 3. We can use this decoder if the evaluation points α_i are in a proper subfield $\mathbb{F}_s \lneq \mathbb{F}_q$. In this case, the RS code $\mathcal{C}_{\mathrm{RS}}[n,k]$ over $\mathbb{F}_q$ is a homogeneous interleaved RS code of interleaving degree $h = [\mathbb{F}_q : \mathbb{F}_s]$. For the constructive class of MDS twisted RS codes in Section 4.2 with minimal field size $q = n^{2^\ell}$ for a given length, we thus have $h = 2^\ell$.

Table 4.1: Decoding radii, list sizes, and complexities for different RS decoders, both for a general field size q and for the minimal field size of the MDS twisted RS codes constructed in Section 4.2, i.e., $q = n^{2^\ell}$.

RS decoder	τ	List size	Complexity	Complexity ($q = n^{2^\ell}$)
BMD	$\lfloor \frac{n-k}{2} \rfloor$	q^ℓ MDS: 1	$O^{\sim}\big(q^\ell n\big)$ [Jus76][4]	$O^{\sim}\big(n^{\ell 2^\ell+1}\big)$
GS [GS98]	$n\left(1-\sqrt{\frac{k-1}{n}}-\varepsilon\right)$	$q^\ell \frac{1}{\varepsilon}$	$O^{\sim}\big(q^\ell n 4^\ell \varepsilon^{-4}\big)$ [CJN+15, NRS17][5]	$O^{\sim}\big(n^{\ell 2^\ell+1} 4^\ell \varepsilon^{-4}\big)$
IRS (Sec. 3.1)	$n\left(1-\left(\frac{k-1}{n}\right)^{\frac{h}{h+1}}-\varepsilon\right)$ $h = [\mathbb{F}_q : \mathbb{F}_s]$, $\alpha_i \in \mathbb{F}_s$ e.g.: $h = 2^\ell$	q^ℓ (partial)	$O^{\sim}\big(q^\ell n \varepsilon^{-(h\omega+1)}\big)$ (cf. Corollary 3.22)	$O^{\sim}\big(n^{\ell 2^\ell+1} \varepsilon^{-(2^\ell\omega+1)}\big)$

In case of the MDS codes in Section 4.2 with $q \approx n^{2^\ell}$, all listed decoders have complexities in the order of at least $n^{\ell 2^\ell+1}$. For $\ell = 1$, this gives n^3, for $\ell = 2$, it is n^9, but already for $\ell = 3$, we get n^{25}. Hence, the decoder is feasible for very small values of ℓ only (e.g., $\ell = 1, 2$), which is due to the relatively large field size compared to the code length.

[4] Exact expression: $O\big(q^\ell n \log^2(n)\log(\log(n))\big)$

[5] Exact expression: $O\big(q^\ell n \varepsilon^{-4}(\log^2(n/\varepsilon)+\log(q))\log(\log(n/\varepsilon))\big)$

Remark 4.5. *Note that Theorem 4.4, as well as the decoding radii, maximal list sizes, and complexities in Table 4.1, also hold for non-MDS twisted RS codes since our algorithm consists of a brute-force step and RS decoders. However, the average list size might be larger in this case. Furthermore, the maximal list size is bounded by well-known bounds on the list size, such as the q-ary generalization of the Johnson bound [Joh62] or the Cassuto–Bruck bound [CB04]. In particular, the list size is polynomial in n for decoding radii up to* $\tau < n - \sqrt{n(n-d)}$*, independent of* ℓ*.*

4.4 Dual Codes

We study the dual codes of twisted RS codes. The following analysis shows that some subfamilies of the new codes with conditions on the evaluation points are—up to equivalence—closed under duality. In contrast to GRS codes, the statement does not generalize to arbitrary evaluation points: In our computer searches (cf. Section 4.7.3), we found twisted RS codes with one twist whose dual code is not equivalent to any twisted RS code with one twist.

Definition 4.6 (Auxiliary Matrices)**.** Let $r, r' \in \mathbb{N}$ and $\boldsymbol{\alpha} \in \mathbb{F}_q^r$.

i) The *reversal matrix* $\boldsymbol{J}_r \in \mathbb{F}_q^{r \times r}$ is the square matrix

$$\boldsymbol{J} = \begin{bmatrix} & & 1 \\ & \iddots & \\ 1 & & \end{bmatrix}.$$

ii) The *Vandermonde matrix of* $\boldsymbol{\alpha}$ *of rank* r is denoted by

$$\boldsymbol{V}_r(\boldsymbol{\alpha}) = \begin{bmatrix} \alpha_1^0 & \alpha_2^0 & \dots & \alpha_n^0 \\ \alpha_1^1 & \alpha_2^1 & \dots & \alpha_n^1 \\ \vdots & & \ddots & \vdots \\ \alpha_1^{r-1} & \alpha_2^{r-1} & \dots & \alpha_n^{r-1} \end{bmatrix}.$$

Remark 4.7. *For a matrix* $\boldsymbol{A} \in \mathbb{F}_q^{r \times r'}$*, then* $\boldsymbol{J}_r\boldsymbol{A}$ *is* $\boldsymbol{A}$ *with the rows in reverse order. Similarly,* $\boldsymbol{A}\boldsymbol{J}_{r'}$ *is* $\boldsymbol{A}$ *with the columns in reverse order. And* $\boldsymbol{J}_r\boldsymbol{A}\boldsymbol{J}_{r'}$ *is* $\boldsymbol{A}$ *"rotated", i.e.,* $(\boldsymbol{J}_r\boldsymbol{A}\boldsymbol{J}_{r'})[i,j] = \boldsymbol{A}[n-j+1, m-j+1]$.

Lemma 4.8. *Let* $\boldsymbol{\alpha} \in \mathbb{F}_q^n$ *such that the* α_i *are distinct and form a multiplicative group. Then,*

$$\left(\boldsymbol{V}_n(\boldsymbol{\alpha})^{\mathrm{T}}\right)^{-1} = \boldsymbol{J}_n \cdot \boldsymbol{V}_n(\boldsymbol{\alpha}) \cdot \operatorname{diag}(\boldsymbol{\alpha}/n).$$

Proof. Since the entries of $\boldsymbol{\alpha}$ are a multiplicative group, we have $\prod_{i=1}^{n}(x - \alpha_i) = x^n - 1$ and

$$\left(\boldsymbol{V}_n(\boldsymbol{\alpha})^{\mathrm{T}}\right)^{-1} = \frac{1}{n}\begin{bmatrix} 1 & 1 & \dots & 1 \\ \alpha_1^{-1} & \alpha_2^{-1} & \dots & \alpha_n^{-1} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_1^{-(n-1)} & \alpha_2^{-(n-1)} & \dots & \alpha_n^{-(n-1)} \end{bmatrix} = \boldsymbol{J}_n \cdot \boldsymbol{V}_n(\boldsymbol{\alpha}) \cdot \operatorname{diag}(\boldsymbol{\alpha}/n),$$

where the first equality follows by [AL69]. □

Lemma 4.9. *Let $\mathcal{C}[n,k]$ be a linear code with a generator matrix of the form*

$$\boldsymbol{G} = [\boldsymbol{I} \mid \boldsymbol{L}] \cdot \boldsymbol{V}_n(\boldsymbol{\alpha}),$$

where $\boldsymbol{I} \in \mathbb{F}_q^{k\times k}$ is the identity matrix, $\boldsymbol{L} \in \mathbb{F}_q^{k\times n-k}$, and the entries of $\boldsymbol{\alpha} \in \mathbb{F}_q^n$ are distinct and form a multiplicative group. Then, the matrix

$$\boldsymbol{H} = [\boldsymbol{I} \mid -\boldsymbol{J}_{n-k}\boldsymbol{L}^\top \boldsymbol{J}_k] \cdot \boldsymbol{V}_n(\boldsymbol{\alpha}) \cdot \operatorname{diag}(\boldsymbol{\alpha}/n)$$

generates the dual code $\mathcal{C}^\perp$.

Proof. By construction, $\boldsymbol{H}$ has full rank $n-k$ and fulfills

$$\begin{aligned}
\boldsymbol{G}\cdot\boldsymbol{H}^\top &= [\boldsymbol{I} \mid \boldsymbol{L}]\boldsymbol{V}_n(\boldsymbol{\alpha}) \cdot \Big([\boldsymbol{I} \mid -\boldsymbol{J}_{n-k}\boldsymbol{L}^\top\boldsymbol{J}_k]\boldsymbol{V}_n(\boldsymbol{\alpha})\operatorname{diag}(\boldsymbol{\alpha}/n)\Big)^\top \\
&= [\boldsymbol{I} \mid \boldsymbol{L}]\boldsymbol{V}_n(\boldsymbol{\alpha}) \cdot \Big(\boldsymbol{J}_{n-k}[-\boldsymbol{A}^{\mathrm{T}} \mid \boldsymbol{I}] \underbrace{\boldsymbol{J}_n\boldsymbol{V}_n(\boldsymbol{\alpha})\operatorname{diag}(\boldsymbol{\alpha}/n)}_{=(\boldsymbol{V}_n(\boldsymbol{\alpha})^{-1})^\top \text{ (Lemma 4.8)}}\Big)^\top \\
&= [\boldsymbol{I} \mid \boldsymbol{L}] \begin{bmatrix} -\boldsymbol{L} \\ \boldsymbol{I} \end{bmatrix} \boldsymbol{J}^\top = \boldsymbol{0}
\end{aligned}$$

so it is a parity-check matrix of $\mathcal{C}$, and thus, a generator matrix of the dual code. □

Lemma 4.9 implies the following duality statement for twisted RS codes with evaluation points forming a multiplicative group. The statement is illustrated in Figure 4.4.

Theorem 4.10. *Let $n, k, \boldsymbol{\alpha}, \boldsymbol{t}, \boldsymbol{h}, \boldsymbol{\eta}$ be chosen as in Definition 4.1 such that the entries of $\boldsymbol{\alpha}$ form a multiplicative sub-group of $\mathbb{F}_q$ and the h_i are distinct. Then, the dual of a $[\boldsymbol{\alpha}, \boldsymbol{t}, \boldsymbol{h}, \boldsymbol{\eta}]$-twisted code is equivalent to a $[k-\boldsymbol{h}, n-k-\boldsymbol{t}, -\boldsymbol{\eta}]$-twisted code with the same evaluation points.*

Proof. By definition, a $[\boldsymbol{\alpha}, \boldsymbol{t}, \boldsymbol{h}, \boldsymbol{\eta}]$-twisted code has a generator matrix of the form

$$\boldsymbol{G} = [\boldsymbol{I} \mid \boldsymbol{L}] \cdot \boldsymbol{V}_n(\boldsymbol{\alpha}),$$

where the entries of $\boldsymbol{L} \in \mathbb{F}_q^{k\times n-k}$ are of the form

$$L_{ij} = \begin{cases} \eta_\mu, & \text{if } [i,j] = [h_\mu + 1, t_\mu], \\ 0, & \text{else.} \end{cases}$$

The dual code is generated by the matrix $\boldsymbol{H} = [\boldsymbol{I} \mid -\boldsymbol{J}_{n-k}\boldsymbol{L}^\top\boldsymbol{J}_k] \cdot \boldsymbol{V}_n(\boldsymbol{\alpha}) \cdot \operatorname{diag}(\boldsymbol{\alpha}/n)$ by Lemma 4.9, so it is equivalent to a code $\mathcal{C}'$ generated by $[\boldsymbol{I} \mid -\boldsymbol{J}_{n-k}\boldsymbol{L}^\top\boldsymbol{J}_k] \cdot \boldsymbol{V}_n(\boldsymbol{\alpha})$. Since the entries of $-\boldsymbol{J}_{n-k}\boldsymbol{L}^\top\boldsymbol{J}_k$ are of the form

$$[-\boldsymbol{J}_{n-k}\boldsymbol{L}^\top\boldsymbol{J}_k]_{ij} = \begin{cases} -\eta_\mu, & \text{if } [i,j] = [n-k-t_\mu+1, k-h_\mu] \\ 0, & \text{else,} \end{cases}$$

and the $k-h_\mu$ are distinct by assumption, the code $\mathcal{C}'$ is a $[k-\boldsymbol{h}, n-k-\boldsymbol{t}, -\boldsymbol{\eta}]$-twisted code, which proves the claim. □

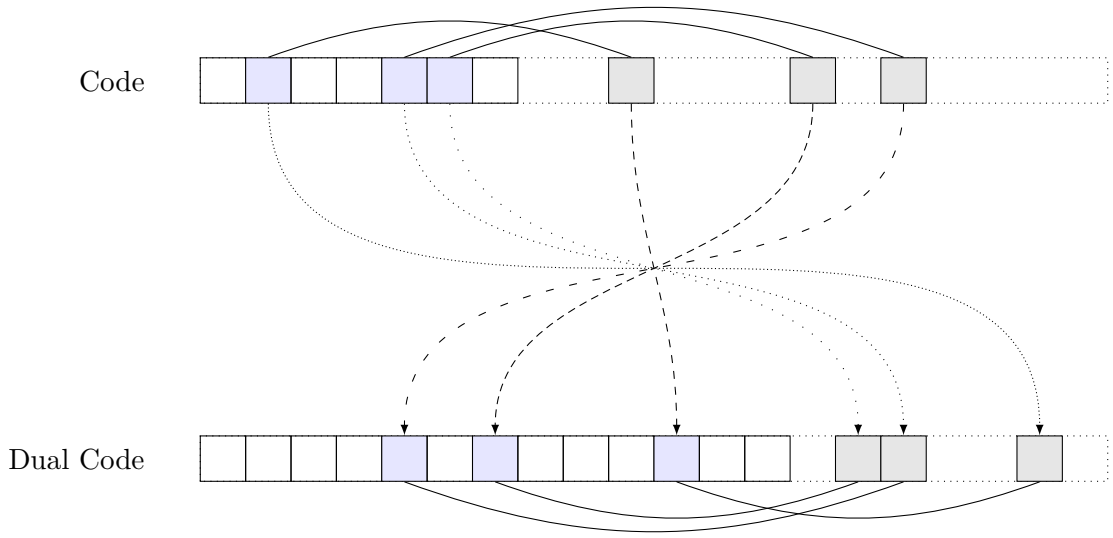


Figure 4.4: Illustration of the evaluation polynomials of the dual code of a twisted RS code as in Theorem 4.10. The twist vector of the dual code is given by $k - \boldsymbol{h}$ and the hook vector becomes $n - k - \boldsymbol{t}$. Here, we have $\boldsymbol{t} = [3, 7, 9]$ and $\boldsymbol{h} = [1, 5, 4]$.

Remark 4.11. *By analogous arguments as in the proof of Lemma 4.9, we can show the following: We join* 0 *to the evaluation points, i.e.,* $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_n, 0]$*, where the* α_i *form a multiplicative subgroup of* $\mathbb{F}_q$*. Furthermore, let a matrix* $\boldsymbol{L} \in \mathbb{F}_q^{k \times n+1-k}$ *be given whose first row is of the form* $[\boldsymbol{l}_1 \mid 0]$ *with* $\boldsymbol{l}_1 \in \mathbb{F}_q^{n-k}$*. Then,*

$$\boldsymbol{H} = \left[\boldsymbol{I} \,\middle|\, -\begin{bmatrix} 1 & \\ \boldsymbol{l}_1^\top & \boldsymbol{I} \end{bmatrix} \cdot \boldsymbol{J}_{n+1-k} \boldsymbol{L}^\top \boldsymbol{J}_k \right] \cdot \boldsymbol{V}_{n+1}(\boldsymbol{\alpha}) \cdot \operatorname{diag}(1/n, \ldots, 1/n, -1)$$

is a valid parity-check matrix for the code generated by $\boldsymbol{G} = [\boldsymbol{I} \mid \boldsymbol{L}] \cdot \boldsymbol{V}_n(\boldsymbol{\alpha}) \in \mathbb{F}_q^{k,n+1}$*. If the first row* or *the last column of* $\boldsymbol{L}$ *is zero, then we have*

$$-\begin{bmatrix} 1 & \\ \boldsymbol{l}_1^\top & \boldsymbol{I} \end{bmatrix} \cdot \boldsymbol{J}_{n+1-k} \boldsymbol{L}^\top \boldsymbol{J}_k = -\boldsymbol{J}_{n+1-k} \boldsymbol{L}^\top \boldsymbol{J}_k.$$

This implies that if all twist parameters fulfill $t_i \neq n - k$ or *all hook parameters are* $h_i \neq 0$*, then Theorem 4.10 holds as well if* 0 *is joined to the evaluation points.*

4.5 Schur Squares

In this section, we study the *Schur squares* of twisted RS codes. There has been an increased interest in these objects in the last years (see, e.g., [Ran15] and references therein) due to several applications, cf. [CGGU+14], [CDN15], and [Ran15, Section 5].

Definition 4.12. Let $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{F}_q^n$ be vectors and $\mathcal{C}, \mathcal{C}_1, \mathcal{C}_2 \subseteq \mathbb{F}_q^n$ be linear codes. The *Schur product* or *component-wise product* of $\boldsymbol{x}$ and $\boldsymbol{y}$ is defined by

$$\boldsymbol{x} \star \boldsymbol{y} = [x_1 \cdot y_1, \ldots, x_n \cdot y_n].$$

The *Schur product* of the codes $\mathcal{C}_1$ and $\mathcal{C}_2$ is similarly defined by

$$\mathcal{C}_1 \star \mathcal{C}_2 = \langle \{ \boldsymbol{c}_1 \star \boldsymbol{c}_2 \,:\, \boldsymbol{c}_1 \in \mathcal{C}_1, \boldsymbol{c}_2 \in \mathcal{C}_2 \} \rangle_{\mathbb{F}_q}$$

By $\mathcal{C}^2 = \mathcal{C} \star \mathcal{C}$, we denote the *Schur square* of $\mathcal{C}$.

It is well-known that the Schur square of a linear code $\mathcal{C}[n,k]$ has dimension

$$\dim\left(\mathcal{C}^2\right) \leq \min\{n, \tfrac{1}{2}k(k-1)\},$$

and a random linear code attains this upper bound with high probability, cf. [CCMZ15]. Furthermore, if the code is MDS, then it satisfies [Ran15, p.31]

$$\dim\left(\mathcal{C}^2\right) \geq \min\{n, 2k-1\}.$$

Generalized Reed–Solomon codes fulfill this lower bound with equality, i.e., for $k < n/2$, their Schur square is much smaller than one expects from a random code. This fact is, e.g., utilized by Couvreur et al. [CGGU$^+$14] to recover the structure of a GRS code, yielding a structural attack on the GRS-based McEliece cryptosystem, cf. Section 4.8.

In this section, we develop tools to obtain the Schur square dimension of twisted RS codes, or at least a lower bound thereof. For low-rate codes, many of the codes have much larger Schur squares than a GRS code of the same parameters. We will use this observation for two purposes in the succeeding section:

- In Section 4.6, we will separate twisted RS codes from GRS codes, implying inequivalence in many cases.
- In Section 4.8, we will show that Couvreur et al.'s attack does not work for a large subfamily of twisted RS codes.

Note that if two linear codes are equivalent, then also their Schur squares are equivalent. In particular, the Schur square dimension is invariant under equivalence.

4.5.1 Schur Squares of Twisted RS Codes

Lower Bounds on the Schur Square Dimension of Evaluation Codes

Since twisted RS codes are evaluation codes, we derive a general lower bound on the dimension of the Schur square of such codes. Due to $f(\alpha_i) \cdot g(\alpha_i) = (f \cdot g)(\alpha_i)$, for polynomials $f, g \in \mathbb{F}_q[x]$ and $\alpha_i \in \mathbb{F}_q$, the Schur square of a linear evaluation code is a linear evaluation code with the same evaluation points. Its evaluation polynomials are the span of all products of two evaluation polynomials of the original code. We will use the following objects to lower-bound the dimension of the Schur square.

Definition 4.13. Let $\boldsymbol{\alpha} \in \mathbb{F}_q^n$ with distinct entries, and $\mathcal{V} \subseteq \mathbb{F}_q[x]_{<n}$ be a k-dimensional $\mathbb{F}_q$-subspace containing polynomials of degree less than n. We define

$$\begin{aligned} \mathrm{D}(\mathcal{V}) &:= \{\deg(f \cdot g) \,:\, f, g \in \mathcal{V}\}, \quad \text{and} \\ \mathrm{D}(\mathcal{V})_{<n} &:= \{d \in \mathrm{D}(\mathcal{V}) \,:\, d < n\}. \end{aligned}$$

For $f \in \mathbb{F}_q[x]$, let $\overline{f} := (f \mod \prod_{i=1}^{n}(x-\alpha_i))$ be the remainder of the division of f by the polynomial $\prod_{i=1}^{n}(x-\alpha_i)$. Then, we define

$$\overline{\mathrm{D}}(\mathcal{V}, \boldsymbol{\alpha}) := \left\{ \deg(\overline{f \cdot g}) \,:\, f, g \in \mathcal{V} \right\}.$$

Theorem 4.14. *Let $\boldsymbol{\alpha} \in \mathbb{F}_q^n$ and $\mathcal{V} \subseteq \mathbb{F}_q[x]_{<n}$ be as in Definition 4.13, and define the evaluation code $\mathcal{C} := \mathrm{ev}_{\boldsymbol{\alpha}}(\mathcal{V})$. Then,*

$$\dim\left(\mathcal{C}^2\right) \geq \left|\overline{\mathrm{D}}(\mathcal{V}, \boldsymbol{\alpha})\right|.$$

Proof. Due to $\overline{f}(\boldsymbol{\alpha}) = f(\boldsymbol{\alpha})$ for all $f \in \mathbb{F}_q[x]$ and $\deg(\overline{f}) < n$, the code $\mathcal{C}^2$ is an evaluation code with evaluation polynomials

$$\mathcal{V}' = \left\langle \overline{f \cdot g} \,:\, f, g \in \mathcal{V} \right\rangle \subseteq \mathbb{F}_q[x]_{<n}.$$

Since the entries of $\boldsymbol{\alpha}$ are distinct, the evaluation map $\mathrm{ev}_{\boldsymbol{\alpha}}(\cdot)$ is injective on $\mathbb{F}_q[x]_{<n}$. Furthermore, $\mathrm{ev}_{\boldsymbol{\alpha}}(\cdot) : \mathbb{F}_q[x] \to \mathbb{F}_q^n$ is a linear map. Thus, the dimension of $\mathcal{C}^2$ equals the dimension of $\mathcal{V}'$ as an $\mathbb{F}_q$-vector space. Since $\{1, x, \ldots, x^{n-1}\}$ is a basis of $\mathbb{F}_q[x]_{<n}$, we have

$$\dim\left(\mathcal{C}^2\right) = \dim \mathcal{V}' \geq \left|\overline{\mathrm{D}}(\mathcal{V}, \boldsymbol{\alpha})\right|. \qquad \square$$

For some evaluation points, the set $\overline{\mathrm{D}}(\mathcal{V}, \boldsymbol{\alpha})$ is easy to determine: For instance, when the α_i form a multiplicative group, we have $\prod_{i=1}^{n}(x-\alpha_i) = x^n - 1$ and the polynomials $\overline{f \cdot g}$ are easy to compute. In other cases, we can utilize $\mathrm{D}(\mathcal{V})_{<n} \subseteq \overline{\mathrm{D}}(\mathcal{V}, \boldsymbol{\alpha})$, which implies the following slightly weaker bound.

Corollary 4.15. *Let $\boldsymbol{\alpha} \in \mathbb{F}_q^n$, $\mathcal{V} \subseteq \mathbb{F}_q[x]_{<n}$, and $\mathcal{C} = \mathrm{ev}_{\boldsymbol{\alpha}}(\mathcal{V})$ be as in Theorem 4.14. Then,*

$$\dim\left(\mathcal{C}^2\right) \geq \left|\mathrm{D}(\mathcal{V})_{<n}\right|.$$

Schur Squares of Twisted RS Codes

For twisted RS codes, the set $\mathrm{D}(\mathcal{V})$, and thus $\mathrm{D}(\mathcal{V})_{<n}$, is easy to compute. We will use the following observation in the subsequent sections.

Theorem 4.16. *Let $\boldsymbol{\alpha}, \boldsymbol{t}, \boldsymbol{h}$, and $\boldsymbol{\eta}$ be as in Definition 4.1 with $\eta_i \neq 0$ for all i, and*

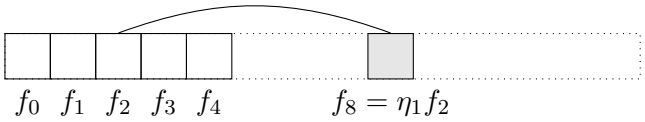
$$S_{k,\boldsymbol{h},\boldsymbol{t}} = \Big(\{0, \ldots, k-1\} \setminus \{h_1, \ldots, h_\ell\}\Big) \cup \{k-1+t_1, \ldots, k-1+t_\ell\}.$$

Then, $\mathrm{D}(\mathcal{P}^{n,k}_{\boldsymbol{t},\boldsymbol{h},\boldsymbol{\eta}}) = \{d_1 + d_2 \,:\, d_1, d_2 \in S_{k,\boldsymbol{h},\boldsymbol{t}}\}$ and

$$\dim\Big(\mathcal{C}_{\boldsymbol{\alpha},\boldsymbol{t},\boldsymbol{h},\boldsymbol{\eta}}[n,k]^2\Big) \geq \left|\{d_1 + d_2 \,:\, d_1, d_2 \in S_{k,\boldsymbol{h},\boldsymbol{t}}\}\right|. \tag{4.2}$$

Proof. Since the t_i are distinct, the elements of $S_{k,\boldsymbol{h},\boldsymbol{t}}$ are exactly the degrees of the polynomials in $\mathcal{P}^{n,k}_{\boldsymbol{t},\boldsymbol{h},\boldsymbol{\eta}}$, which implies the claim. $\square$

Example 4.17. *Let $\ell = 1$, $n = 14$, $k = 5$, $t_1 = 4$, $h_1 = 2$, and $\eta_1 \neq 0$. The corresponding evaluation polynomials have the following form:*



In this example, we get

$$S_{k,\boldsymbol{h},\boldsymbol{t}} = (\{0,\dots,4\} \setminus \{2\}) \cup \{8\} = \{0,1,3,4,8\}.$$

The degrees of the polynomials $f \cdot g$, where $f, g \in \mathcal{P}^{n,k}_{\boldsymbol{t},\boldsymbol{h},\boldsymbol{\eta}}$, can be written as sums of elements in $S_{k,\boldsymbol{h},\boldsymbol{t}}$ as follows:

$$\begin{aligned}
0 &= 0+0, & 4 &= 4+0, & 8 &= 8+0,\\
1 &= 1+0, & 5 &= 4+1, & 9 &= 8+1,\\
2 &= 1+1, & 6 &= 3+3, & 11 &= 8+3,\\
3 &= 3+0, & 7 &= 4+3, & 12 &= 8+4.
\end{aligned}$$

Thus, we have $\mathrm{D}(\mathcal{P}^{n,k}_{\boldsymbol{t},\boldsymbol{h},\boldsymbol{\eta}}) = \mathrm{D}(\mathcal{P}^{n,k}_{\boldsymbol{t},\boldsymbol{h},\boldsymbol{\eta}})_{<n} = \{0,1,2,3,4,5,6,7,8,9,11,12\}$, so only the degrees (less than n) 10 and 13 do not occur as $\deg(f \cdot g)$. Hence, the Schur square code of $\mathcal{C}_{\boldsymbol{\alpha},\boldsymbol{t},\boldsymbol{h},\boldsymbol{\eta}}[n,k]$ has dimension

$$\dim\Big(\mathcal{C}_{\boldsymbol{\alpha},\boldsymbol{t},\boldsymbol{h},\boldsymbol{\eta}}[n,k]^2\Big) \geq \Big|\mathrm{D}\Big(\mathcal{P}^{n,k}_{\boldsymbol{t},\boldsymbol{h},\boldsymbol{\eta}}\Big)_{<n}\Big| = 12.$$

Note that any GRS code with $[n,k] = [14,5]$ has Schur square dimension $2k-1 = 9$, so the code $\mathcal{C}_{\boldsymbol{\alpha},\boldsymbol{t},\boldsymbol{h},\boldsymbol{\eta}}[n,k]$ as above is not a GRS code. We will generalize this approach in Section 4.6.

4.5.2 Codes with Maximal Schur Square Dimension

In this subsection, we present two constructive classes of twisted RS codes whose Schur square has dimension n. This is of theoretical interest as it shows that twisted RS codes can have much larger Schur squares than RS codes. The Schur square of the latter class stays also maximal if the code is shortened at several positions. We will use this fact in Section 4.8 to obtain codes that are resistant to known structural attacks on the McEliece cryptosystem based on RS-like codes.

Twisted RS Codes with Maximal Schur Square Dimension

Theorem 4.18. *Let $k, n, \ell \in \mathbb{N}$ be chosen such that*

$$2\sqrt{n} + 4 < k \leq \tfrac{n}{2} \quad \text{and} \tag{4.3}$$

$$\tfrac{n+1}{k-\sqrt{n}} - 2 < \ell < \tfrac{2n}{k} - 2. \tag{4.4}$$

We define $r := \left\lceil \frac{n+1}{\ell+2} \right\rceil$ and, for $i = 1,\dots,\ell$, choose the hook and twist parameters as

$$\begin{aligned}
h_i &= r - 1 + i,\\
t_i &= (i+1)r - k.
\end{aligned}$$

Then, for any $\boldsymbol{\eta} \in (\mathbb{F}_q \setminus \{0\})^\ell$, the corresponding twisted RS code has Schur square dimension

$$\dim\Big(\mathcal{C}_{\boldsymbol{\alpha},\boldsymbol{t},\boldsymbol{h},\boldsymbol{\eta}}[n,k]^2\Big) = n.$$

Proof. We first show that the hook and twist parameters are well-defined. First note that $\ell < \frac{2n}{k} - 2 < \frac{2n}{2\sqrt{n}+4} - 2 < \sqrt{n} - 2$.

- Due to $\ell < \frac{2n}{k} - 2$, we have $r > \frac{k}{2} > 0$. Hence, $h_i > 0$, $t_i \geq 2r - k > 0$, and the t_i are distinct.
- Due to $\frac{n+1}{k-\sqrt{n}} - 2 < \ell$ and $\ell < \sqrt{n} - 2$, we have $\frac{n+1}{k-(\ell+2)} < \frac{n+1}{k-\sqrt{n}} < \ell + 2$, which implies

$$h_i \leq r - 1 + \ell \leq \tfrac{n+1}{\ell+2} + \ell + 2 < k.$$

- Due to $\ell < \sqrt{n} - 2$, we have $(\ell+3)(\ell+1) \leq (\ell+2)^2 < n$, so $\frac{n+1}{\ell+2} + 1 \leq \frac{n}{\ell+1}$, and

$$t_i \leq (\ell+1)r - k \leq (\ell+1)(\tfrac{n+1}{\ell+2} + 1) - k \leq (\ell+1)\tfrac{n}{\ell+1} - k = n - k.$$

Hence, the h_i and t_i are valid parameters by Definition 4.1. We now prove that the Schur square has full dimension using Theorem 4.16. The argument is illustrated in Figure 4.5. By construction, the following degrees are in the degree set of the evaluation polynomials:

$$\{0, \ldots, r-1\} \cup \{(i+1)r - 1 \,:\, i = 1, \ldots, \ell\} \subseteq S_{k,\boldsymbol{h},\boldsymbol{t}}.$$

Hence, the degrees $0, r-1, 2r-1, \ldots, (\ell+1)r-1$ divide the interval $[0, (\ell+2)r-2]$ into sub-intervals of length r, so any integer

$$(ir - 1) + j \quad \text{with} \quad i = 0, \ldots, \ell+1 \quad \text{and} \quad j = 0, \ldots, r-1$$

can be expressed as the sum of two elements in $S_{k,\boldsymbol{h},\boldsymbol{t}}$. Due to

$$(\ell+2)r - 2 \geq n - 1,$$

this means that $\{0, \ldots, n-1\} \subseteq \mathrm{D}(\mathcal{P}^{n,k}_{\boldsymbol{t},\boldsymbol{h},\boldsymbol{\eta}})_{<n}$, which proves the claim. □

Remark 4.19. *We analyze for which code lengths n, there are dimensions k and numbers of twists ℓ satisfying the conditions in Theorem 4.18. For this, we need to show that there are integers between the lower and upper bounds in* (4.3) *and* (4.4)*, which translates to*

$$2\sqrt{n} + 5 < \tfrac{n}{2} \quad \textit{and} \tag{4.5}$$
$$\tfrac{n+1}{k-\sqrt{n}} - 1 < \tfrac{2n}{k} - 2. \tag{4.6}$$

Condition (4.5) *is fulfilled for $n > 32$. We can rewrite* (4.6) *as*

$$0 < -k^2 + k(n - 1 + \sqrt{n}) - 2n^{3/2}.$$

The right-hand-side is monotonically increasing in k for $2\sqrt{n} + 4 < k \leq \frac{n}{2}$, so it suffices to show the inequality for $k = 2\sqrt{n} + 4$. In this case, it simplifies to an inequality in n,

$$0 < 2n - 14\sqrt{n} - 20,$$

which is fulfilled for $n > 67$. Hence, twisted RS codes of maximal Schur square dimension as in Theorem 4.18 exist for any $n > 67$.

Furthermore, for a fixed rate $R = \frac{k}{n} \leq \frac{n}{2}$, there is a twisted RS code of length $n \in \Omega(1/R^2)$ (due to $2\sqrt{n} + 4 < k = Rn$) and maximal Schur square dimension.

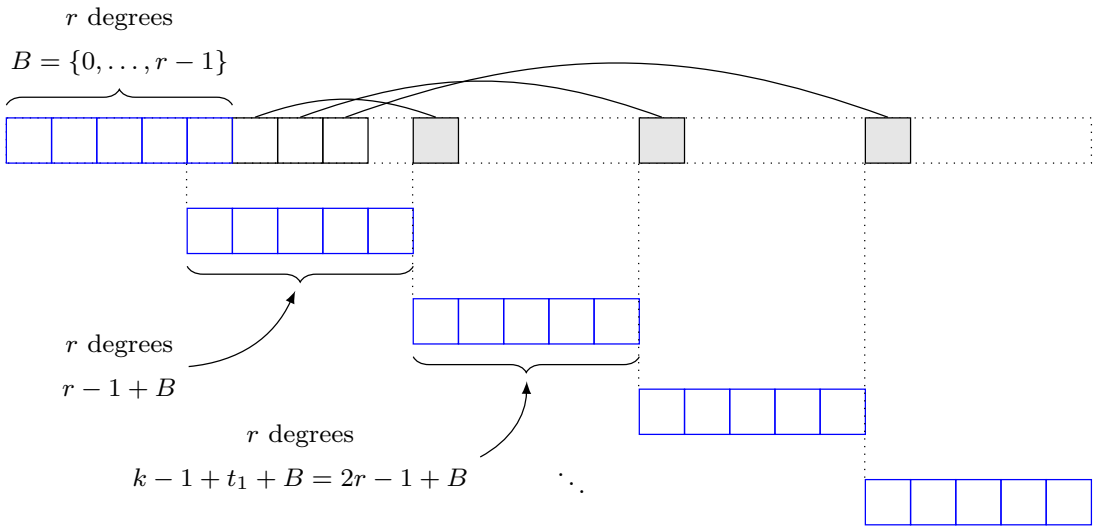


Figure 4.5: Illustration of the proof of Theorem 4.18 for the example[6] $n = 24$, $k = 8$, $\ell = 3$, $r = 5$. All evaluation polynomial degrees $0, \ldots, n-1$ can be expressed as sums of an element in $B = \{0, \ldots, r-1\}$ and one in $\{0, r-1, k-1+t_1, \ldots, k-1+t_\ell\}$.

Remark 4.20. *For a fixed rate $R = \frac{k}{n}$, the lower bound on ℓ in Theorem 4.18 converges to*

$$\tfrac{n+1}{k-\sqrt{n}} - 2 \to \tfrac{1}{R} - 2, \quad (n \to \infty)$$

so it suffices to choose $\ell = \lfloor \frac{1}{R} \rfloor - 1$ twists for large code lengths n. Note that the number of required twists ℓ does not grow in n.

Shortened Twisted RS Codes with Maximal Schur Square Dimension

In [CGGU+14], the low-dimensional Schur square of shortened Reed–Solomon codes was used to attack the RS-based McEliece cryptosystem. In the following, we show that twisted RS codes can have large Schur square also when they are shortened at several positions. The *shortening* of a linear code $\mathcal{C}[n, k, d]$ at position i is the code (see e.g. [Rot06, Problem 2.14])

$$\mathcal{C}' = \{[c_1, \ldots, c_{i-1}, c_{i+1}, \ldots, c_n] \,:\, [c_1, \ldots, c_{i-1}, 0, c_{i+1}, \ldots, c_n] \in \mathcal{C}\} \subseteq \mathbb{F}_q^{n-1}.$$

The resulting code has parameters $\mathcal{C}'[n', k', d']$ with $n' = n - 1$, $k' \geq k - 1$, and $d' \geq d$. This also implies that a shortened MDS code is again MDS with $k' = k-1$ and $d' = d$. A code can be shortened at several positions by applying the procedure above iteratively. The following family of codes has maximal Schur square also when shortened at $\varrho \geq 0$ positions. It can be seen as a natural generalization of the family in Theorem 4.18.

[6] These values are chosen for didactic reasons. They do not fulfill (4.3) and (4.4), but yield a maximal Schur square code anyways. It also shows that these conditions are sufficient, but not necessary for this purpose.

Theorem 4.21. *Let $\varrho, k, n, \ell \in \mathbb{N}$ be such that $\varrho \leq k$, $\nu(\varrho) \in \mathbb{N}_0$ (cf. Remark 4.22 below),*

$$2\sqrt{n} + \nu(\varrho) < k \leq \tfrac{n}{2}, \quad \text{and} \tag{4.7}$$

$$\tfrac{n+1}{k-\sqrt{n}} - 2 < \ell < \min\left\{\tfrac{2n}{k} - 2, \sqrt{n} - 2 - \varrho\right\}. \tag{4.8}$$

We define $r := \left\lceil \frac{n+1}{\ell+2} \right\rceil + \varrho$ and, for $i = 1, \ldots, \ell$, choose the hook and twist parameters as

$$\begin{aligned} h_i &= r - 1 + i, \\ t_i &= (i+1)(r-\varrho) - k + \varrho. \end{aligned}$$

Furthermore, let the entries of $\boldsymbol{\alpha} \in (\mathbb{F}_q \setminus \{0\})^n$ be distinct and non-zero, and $\boldsymbol{\eta} \in (\mathbb{F}_q \setminus \{0\})^\ell$. Then, for any $\varrho' \leq \varrho$, the code $\mathcal{C}[n-\varrho', k-\varrho']$ obtained from shortening $\mathcal{C}_{\boldsymbol{\alpha},\boldsymbol{t},\boldsymbol{h},\boldsymbol{\eta}}[n,k]$ at arbitrary ϱ' positions has maximal Schur square dimension

$$\dim\left(\mathcal{C}^2\right) = n - \varrho'.$$

Proof. The proof is similar to the one of Theorem 4.18. It can be found in Appendix A.2. □

Remark 4.22. *The role of the parameter $\nu(\varrho)$ becomes apparent when considering the existence of k and ℓ satisfying (4.7) and (4.8). For the analysis, we assume that $\nu : \mathbb{N}_0 \to \mathbb{N}_0$ is a function in ϱ, which also motivates the notation. As in Remark 4.19, we must have*

$$2\sqrt{n} + \nu(\varrho) - 1 < \tfrac{n}{2}, \tag{4.9}$$

$$\tfrac{n+1}{k-\sqrt{n}} - 1 < \tfrac{2n}{k} - 2, \quad \text{and} \tag{4.10}$$

$$\tfrac{n+1}{k-\sqrt{n}} - 1 < \sqrt{n} - 2 - \varrho. \tag{4.11}$$

The first inequality gives a restriction on n as $n > (\sqrt{2(\nu(\varrho)+3)}+2)^2 \in \Theta(\nu(\varrho))$. For $\nu(\varrho) > 3$ and $n > 67$, Inequality (4.10) is satisfied by the arguments in Remark 4.19. The left-hand side of (4.11) is monotonically decreasing in k for $2\sqrt{n} + \nu(\varrho) < k$, so is suffices to consider $\frac{n+1}{\sqrt{n}+\nu(\varrho)} < \sqrt{n} - 1 - \varrho$, which for $\nu(\varrho) > 1 + \varrho$ is fulfilled if and only if $n > \left(\frac{\nu(\varrho)(1+\varrho)}{\nu(\varrho)-(1+\varrho)}\right)^2$. In total, we must choose

$$\nu(\varrho) > \max\{3, 1+\varrho\}, \quad \text{and} \tag{4.12}$$

$$n > \max\left\{67, \left(\sqrt{2(\nu(\varrho)+3)}+2\right)^2, \left(\tfrac{\nu(\varrho)(1+\varrho)}{\nu(\varrho)-(1+\varrho)}\right)^2\right\}. \tag{4.13}$$

It can be seen that the minimal possible code rate increases with $\nu(\varrho)$. On the other hand, the lower bound $\left(\frac{\nu(\varrho)(1+\varrho)}{\nu(\varrho)-(1+\varrho)}\right)^2$ on n decreases in $\nu(\varrho)$. Hence, one must find a trade-off between the desired rate and the minimal n by choosing $\nu(\varrho)$ accordingly.

As an example, let $\nu(\varrho) = \max\{4, 2(1+\varrho)\}$. In this case, we must use a code length of at least $n \in \Omega(\varrho^2)$ by (4.13). The minimal dimension grows linearly in ϱ, which means that arbitrarily small rates are possible. For $\varrho \leq 3$, the minimal possible length is 68 in this case, and for $\varrho = 7$ and $\varrho = 10$, we have $n \geq 257$ and $n \geq 485$, respectively.

Again, we only require $\ell = \lfloor \frac{1}{R} \rfloor - 1$ twists asymptotically in order to obtain a maximal Schur square dimension. Note that the number of shortened positions ϱ only influences the minimal possible rate and length of the code, but not the smallest number of required twists.

4.6 Relation to Reed–Solomon Codes

In this section, we study the relation of RS and twisted RS codes. We first show that many twisted RS codes are not GRS codes, i.e., are not equivalent to RS codes. In Section 4.6.3, we consider a more general *separation* problem: Finding the dimensions of the smallest GRS code containing a given twisted RS code, and the largest GRS code that is contained in it.

4.6.1 Low-Rate Non-GRS Twisted RS Codes

In Section 4.5, we have seen that any GRS code with parameters $[n, k]$ has Schur square dimension $\min\{n, 2k-1\}$, which is the minimal value for an MDS code. If a twisted RS code has larger Schur square dimension, it cannot be a GRS code. Note that this is only possible for low-rate codes, i.e., $k \leq \frac{n}{2}$.

Hence, we can use the lower bound on the Schur square dimension developed in Theorem 4.16 to show the inequivalence of many twisted RS codes to RS codes. For instance, the following statement shows that almost all low-rate twisted RS codes with one twist ($\ell = 1$) are not GRS codes.

Theorem 4.23. *Let $\ell = 1$ and $2k < n$. Choose $\boldsymbol{\alpha}, \boldsymbol{h}, \boldsymbol{t}, \boldsymbol{\eta}$ as in Definition 4.1 with the additional requirements*

$$\eta_1 \neq 0 \quad \text{and} \quad 1 < h_1 < k-2.$$

Then, the code $\mathcal{C}_{\boldsymbol{\alpha},\boldsymbol{t},\boldsymbol{h},\boldsymbol{\eta}}[n,k]$ has Schur square dimension $\dim(\mathcal{C}_{\boldsymbol{\alpha},\boldsymbol{t},\boldsymbol{h},\boldsymbol{\eta}}[n,k]^2) \geq 2k$ and is not a GRS code.

Proof. We apply Theorem 4.16. The set of evaluation polynomial degrees is given by

$$S_{k,\boldsymbol{h},\boldsymbol{t}} = \{0, 1, \ldots, h_1 - 1, h_1 + 1, \ldots, k-2, k-1, k-1+t_1\}.$$

Note that in particular $\{0, 1, k-2, k-1\} \subseteq S_{k,\boldsymbol{h},\boldsymbol{t}}$ due to the additional requirements. We show that $\{0, \ldots, 2k-2, \mu\} \subseteq \mathrm{D}(\mathcal{P}^{n,k}_{\boldsymbol{t},\boldsymbol{h},\boldsymbol{\eta}})$ for some $\mu \in \{2k-1, \ldots, n-1\}$.

- Let $0 \leq i \leq k-1$. Then, i can be written as the sum of two elements in $S_{k,\boldsymbol{h},\boldsymbol{t}}$ as follows:

$$i = \begin{cases} i + 0, & \text{if } i \neq h_1, \\ (h_1 - 1) + 1, & \text{if } i = h_1. \end{cases}$$

  Hence, $i \in \mathrm{D}(\mathcal{P}^{n,k}_{\boldsymbol{t},\boldsymbol{h},\boldsymbol{\eta}})$. Note that we require $0, 1, h_1 - 1 \in S_{k,\boldsymbol{h},\boldsymbol{t}}$, which is fulfilled by assumption.

- Let $k \leq k - 1 + i \leq 2k - 2$. Then,

$$k - 1 + i = \begin{cases} (k-1) + i, & \text{if } i \neq h_1, \\ (k-2) + (h_1 + 1), & \text{if } i = h_1. \end{cases}$$

  Hence, $k - 1 + i \in \mathrm{D}(\mathcal{P}^{n,k}_{\boldsymbol{t},\boldsymbol{h},\boldsymbol{\eta}})$. Note that we require $k-1, k-2, h_1+1 \in S_{k,\boldsymbol{h},\boldsymbol{t}}$.

- Due to $2k < n$, the set $\{2k-1, \ldots, n-1\}$ contains at least two elements. Hence, it intersects with the set $(k - 1 + t_i) + S_{k,\boldsymbol{h},\boldsymbol{t}}$ in at least one element μ.

The claim follows by Theorem 4.16 and $|\mathrm{D}(\mathcal{P}^{n,k}_{\boldsymbol{t},\boldsymbol{h},\boldsymbol{\eta}})| \geq 2k$. □

The arguments used in the proof of Theorem 4.23 can be generalized easily to apply to a much wider class of codes, but the proofs get very technical. In fact, it appears that most low-rate twisted RS codes have Schur square dimension larger than $2k-1$ and exceptional cases are rare.

Implication on Some High-Rate Twisted RS Codes

For high-rate codes ($k \geq \frac{n}{2}$), we can use the duality statements in Section 4.4 in cases where they apply. In particular, if the evaluation points $\boldsymbol{\alpha}$ form a multiplicative group (or if in addition 0 is an evaluation point and $t_i \neq n-k$ for all i or $h_i \neq 0$ for all i, cf. Remark 4.11), then we can apply the Schur square arguments to the (low-rate) dual code. Since the dual of a non-GRS code is non-GRS, this implies the inequivalence for the high-rate code.

4.6.2 A Combinatorial Inequivalence Argument

In this subsection, we approach the inequivalence problem from a combinatorial perspective. We show that for given code parameters $\boldsymbol{\alpha}$, $\boldsymbol{h}$, and $\boldsymbol{t}$, the choices of $\boldsymbol{\eta}$ for which the twisted RS code is MDS either all result in GRS codes or many of them result in non-GRS codes. We rely on the following well-known result about codes being GRS codes.

Lemma 4.24 ([RS85, RL89b])**.** *Let $\mathcal{C}$ be a linear code with a generator matrix of the form $\boldsymbol{G} = [\boldsymbol{I} \mid \boldsymbol{A}]$. Then, $\mathcal{C}$ is a GRS code if and only if, for $\boldsymbol{A}' \in \mathbb{F}_q^{k\times n-k}$ with $A'_{ij} = A_{ij}^{-1}$,*

(i) all entries of $\boldsymbol{A}$ are non-zero,

(ii) all 2×2 minors of $\boldsymbol{A}'$ are non-zero, and

(iii) all 3×3 minors of $\boldsymbol{A}'$ are zero.

An MDS code always has a unique generator matrix of the form $\boldsymbol{G} = [\boldsymbol{I} \mid \boldsymbol{A}]$ (since any k columns of a generator matrix are linearly independent) and such a matrix $\boldsymbol{A}$ fulfills Conditions (i) and (ii) of Lemma 4.24. Thus, Condition (iii) is the essential property that distinguishes a non-GRS MDS code from a GRS code.

For $k < 3$ or $n-k < 3$, the matrix $\boldsymbol{A}$ has no 3×3 minors, so any MDS code of these parameters is a GRS code. Hence, we only consider the case $\min\{k, n-k\} \geq 3$.

We start with a lemma that shows how the entries of a twisted RS code's generator matrix $\boldsymbol{G} = [\boldsymbol{I} \mid \boldsymbol{A}]$ depend on the twist coefficients $\boldsymbol{\eta}$.

Lemma 4.25. *Let $\boldsymbol{\alpha}, \boldsymbol{t}, \boldsymbol{h}$ be chosen as in Definition 4.1 and $\mathcal{H} \subseteq \mathbb{F}_q^\ell$ be the set of $\boldsymbol{\eta}$ such that $\mathcal{C}_{\boldsymbol{\alpha},\boldsymbol{t},\boldsymbol{h},\boldsymbol{\eta}}[n,k]$ is MDS. For any $\boldsymbol{\eta} \in \mathcal{H}$, let $\boldsymbol{G} = [\boldsymbol{I} \mid \boldsymbol{A}^{(\boldsymbol{\eta})}]$ be the generator matrix of $\mathcal{C}_{\boldsymbol{\alpha},\boldsymbol{t},\boldsymbol{h},\boldsymbol{\eta}}[n,k]$ in reduced row echelon form. Then, the entries of $\boldsymbol{A}^{(\boldsymbol{\eta})} \in \mathbb{F}_q^{k\times n-k}$ are given by*

$$A^{(\boldsymbol{\eta})}_{i,j} = p^{(i,j)}(\eta_1, \ldots, \eta_\ell) \quad \forall\, \boldsymbol{\eta} = [\eta_1, \ldots, \eta_\ell] \in \mathcal{H},$$

where $p^{(i,j)} = p_0^{(i,j)} + \sum_{\kappa=1}^{\ell} p_\kappa^{(i,j)} x_\kappa \in \mathbb{F}_q[x_1, \ldots, x_\ell]$ is a polynomial in ℓ variables of total degree at most 1, does not have a zero in $\mathcal{H}$, and whose coefficients do not depend on $\boldsymbol{\eta}$.

Proof. Let $\boldsymbol{\eta} \in \mathcal{H}$. W.l.o.g., let the t_i be ordered such that $t_1 < t_2 < \cdots < t_\ell$ (otherwise, re-order $\boldsymbol{t}$ and $\boldsymbol{h}$, respectively). For $i = 1, \ldots, k$, the i^{th} row of $\boldsymbol{G} = [\boldsymbol{I} \mid \boldsymbol{A}^{(\eta)}]$ is obtained by evaluating a polynomial $f^{(i)} \in \mathcal{P}^{n,k}_{\boldsymbol{t},\boldsymbol{h},\boldsymbol{\eta}}$ that fulfills

$$f^{(i)}(\alpha_j) = \begin{cases} 1, & i = j, \\ 0, & i \neq j, \end{cases} \quad \forall\, i, j = 1, \ldots, k. \tag{4.14}$$

Thus, it factors into $f^{(i)} = g^{(i)} \cdot \sigma^{(i)}$, where $\sigma^{(i)} = \sum_{j=0}^{k-1} \sigma_j^{(i)} x^j := \prod_{j=1, j\neq i}^{k} (x - \alpha_j)$ and $g^{(i)} \in \mathbb{F}_q[x]_{\leq t_\ell}$. Using $\gamma_i := (\sigma^{(i)}(\alpha_i))^{-1} = \prod_{j=1, j\neq i}^{k} (\alpha_i - \alpha_j)^{-1} \in \mathbb{F}_q$, we obtain the equation

$$1 = f^{(i)}(\alpha_i) = g^{(i)}(\alpha_i) \cdot \sigma^{(i)}(\alpha_i) = g^{(i)}(\alpha_i) \cdot \gamma_i^{-1} \quad \Leftrightarrow \quad \gamma_i = \sum_{j=0}^{t} g_j^{(i)} \alpha_i^j. \tag{4.15}$$

For $\kappa \in \{1, \ldots, t_\ell\} \setminus \{t_i : i = 1, \ldots, \ell\}$, the $(k + \kappa - 1)$-th coefficient of f is zero, so we obtain the $t_\ell - \ell$ many equations in the unknowns $\boldsymbol{g}^{(i)} := [g_{t_\ell}^{(i)}, g_{t_\ell - 1}^{(i)}, \ldots, g_0^{(i)}]^\top$

$$[\sigma_{k-t_\ell-1+\kappa}^{(i)}, \ldots, \sigma_{k-1}^{(i)}, 1, \underbrace{0, \ldots, 0}_{\substack{\kappa-1 \\ \text{many}}}] \cdot \boldsymbol{g}^{(i)} = 0. \tag{4.16}$$

For $\kappa = 1, \ldots, \ell$, the $(k + t_\kappa - 1)$-th coefficient of f is

$$[\sigma_{k-t_\ell-1+t_\kappa}^{(i)}, \ldots, \sigma_{k-1}^{(i)}, 1, \underbrace{0, \ldots, 0}_{\substack{t_\kappa-1 \\ \text{many}}}] \cdot \boldsymbol{g}^{(i)} = \eta_\kappa \lambda_\kappa(a_0, \ldots, a_{k-1}) = \eta_\kappa c_\kappa \tag{4.17}$$

with $c_\kappa := \lambda_\kappa(a_0, \ldots, a_{k-1}) \in \mathbb{F}_q$. By combining (4.15), (4.16), and (4.17), we obtain a system of $t_\ell + 1$ many equations

$$\underbrace{\begin{bmatrix}
1 & & & & & & & \\
\sigma_{k-1}^{(i)} & 1 & & & & & & \\
\vdots & \vdots & \ddots & & & & & \\
\sigma_{k-t_\ell+t_{\ell-1}}^{(i)} & \sigma_{k-t_\ell+t_{\ell-1}+1}^{(i)} & \cdots & 1 & & & & \\
\sigma_{k-t_\ell+t_{\ell-1}-1}^{(i)} & \sigma_{k-t_\ell+t_{\ell-1}}^{(i)} & \cdots & \sigma_{k-1}^{(i)} & 1 & & & \\
\vdots & \vdots & \ddots & \vdots & \vdots & \ddots & & \\
\sigma_{k-t_\ell}^{(i)} & \sigma_{k-t_\ell+1}^{(i)} & \cdots & \sigma_{k-t_{\ell-1}+2}^{(i)} & \sigma_{k-t_{\ell-1}+3}^{(i)} & \cdots & 1 & \\
\alpha_i^{t_\ell} & \alpha_i^{t_\ell-1} & \cdots & \alpha_i^{t_{\ell-1}+1} & \alpha_i^{t_{\ell-1}} & \cdots & \alpha_i & 1
\end{bmatrix}}_{=: \boldsymbol{B}} \cdot \boldsymbol{g}^{(i)} = \begin{bmatrix}
\eta_\ell c_\ell \\ 0 \\ \vdots \\ 0 \\ \eta_{\ell-1} c_{\ell-1} \\ \vdots \\ 0 \\ \gamma_i
\end{bmatrix}.$$

Since the matrix $\boldsymbol{B}$ is lower-triangular, it is invertible and the solution $\boldsymbol{g}^{(i)}$ is of the form $g_j^{(i)} = g_{j,0}^{(i)} + \sum_{\kappa=1}^{\ell} g_{j,\kappa}^{(i)} \eta_i$, where $g_{j,\kappa}^{(i)} \in \mathbb{F}_q$ does not depend on $\boldsymbol{\eta}$. Hence, we obtain

$$\begin{aligned}
A_{i,j}^{(\boldsymbol{\eta})} &= f^{(i)}(\alpha_{k+j}) = \sigma^{(i)}(\alpha_{k+j}) \cdot g^{(i)}(\alpha_{k+j}) \\
&= \left[\prod_{\substack{\kappa=1 \\ \kappa\neq i}}^{k} (\alpha_{k+j} - \alpha_\kappa) \right] \cdot \left[\sum_{\mu=0}^{t_\ell} \left(g_{\mu,0}^{(i)} + \sum_{\kappa=1}^{\ell} g_{\mu,\kappa}^{(i)} \eta_i \right) \alpha_{k+j}^{\mu} \right] =: p^{(i,j)}(\eta_1, \ldots, \eta_\ell),
\end{aligned}$$

where the polynomial $p^{(i,j)}$ is of the claimed form. If $p^{(i,j)}$ had a zero $\boldsymbol{\eta} \in \mathcal{H}$, the i^{th} row of $\boldsymbol{G}$ would have Hamming weight less than $n-k+1$, contradicting the MDS property. □

Theorem 4.26. *Let* $\min\{k, n-k\} \geq 3$, $\boldsymbol{\alpha}$, $\boldsymbol{t}$, $\boldsymbol{h}$ *be chosen as in Definition 4.1 and* $\mathcal{H} \subseteq \mathbb{F}_q^\ell$ *be the set of* $\boldsymbol{\eta}$ *such that* $\mathcal{C}_{\boldsymbol{\alpha},\boldsymbol{t},\boldsymbol{h},\boldsymbol{\eta}}[n,k]$ *is MDS. Then, one of the following is true:*

i) $\mathcal{C}_{\boldsymbol{\alpha},\boldsymbol{t},\boldsymbol{h},\boldsymbol{\eta}}[n,k]$ *is GRS for all* $\boldsymbol{\eta} \in \mathcal{H}$.

ii) All $\boldsymbol{\eta} \in \mathcal{H}$ *for which* $\mathcal{C}_{\boldsymbol{\alpha},\boldsymbol{t},\boldsymbol{h},\boldsymbol{\eta}}[n,k]$ *is GRS are zeros of a non-zero multivariate polynomial* $p \in \mathbb{F}_q[x_1, \ldots, x_\ell]$ *of total degree at most* 6.

Proof. If $\mathcal{H}$ is empty or Case i) is fulfilled, we are done. Suppose both conditions are not fulfilled. Then, there is an $\boldsymbol{\eta}^* \in \mathcal{H}$ such that $\mathcal{C}_{\boldsymbol{\alpha},\boldsymbol{t},\boldsymbol{h},\boldsymbol{\eta}^*}[n,k]$ is non-GRS. Let $\boldsymbol{G} = [\boldsymbol{I} \mid \boldsymbol{A}^{(\boldsymbol{\eta}^*)}]$ the generator matrix of $\mathcal{C}_{\boldsymbol{\alpha},\boldsymbol{t},\boldsymbol{h},\boldsymbol{\eta}^*}[n,k]$ in systematic form. By Lemma 4.24, the element-wise inverse of $\boldsymbol{A}^{(\boldsymbol{\eta}^*)}$ has a non-vanishing 3×3 minor.

Now fix this minor (i.e., the row and column indices of the corresponding 3×3 submatrix), but consider all matrices $\boldsymbol{A}^{(\boldsymbol{\eta})}$ with $\boldsymbol{\eta} \in \mathcal{H}$ as an indeterminate. It follows from Lemma 4.25 that the entries of the element-wise inverse of $\boldsymbol{A}^{(\boldsymbol{\eta})}$ are rational functions of the form $\frac{1}{p_{i,j}}$, where the $p_{i,j} \in \mathbb{F}_q[x_1, \ldots, x_\ell] \setminus \{0\}$ are multivariate polynomials of total degree at most 1. Since the fixed minor is the determinant of a 3×3 submatrix of the element-wise inverse of $\boldsymbol{A}^{(\boldsymbol{\eta})}$, it is a rational function in $\eta_1, \ldots, \eta_\ell$ of total nominator degree at most six. Since the minor is non-zero for $\boldsymbol{\eta}^*$, the nominator polynomial, say p, cannot vanish.[7] □

Theorem 4.26 can be interpreted as follows: All values $\boldsymbol{\eta}$ that result in GRS codes are zeros of a multivariate polynomial of low total degree. If the polynomial does not vanish, this is quite a strong restriction as shown by the statements below. We rely on the following result, which was used in [LRMV14] to give the minimum distance of *affine Cartesian codes*[8].

Lemma 4.27 ([LRMV14, Theorem 3.8])**.** *Let* $p \in \mathbb{F}_q[x_1, \ldots, x_\ell] \setminus \{0\}$ *be a non-zero multivariate polynomial of total degree at most* 6*, and* $\mathcal{H} = \mathcal{H}_1 \times \cdots \times \mathcal{H}_\ell$*, where* $\mathcal{H}_i \subseteq \mathbb{F}_q$ *with*

$$7 \leq |\mathcal{H}_1| \leq |\mathcal{H}_2| \leq \cdots \leq |\mathcal{H}_\ell|.$$

Then, p *has at most* $6 \prod_{i=2}^{\ell} |\mathcal{H}_i|$ *zeros in* $\mathcal{H}$.

Proof. This statement is a special case of [LRMV14, Theorem 3.8]. □

Corollary 4.28. *Let* $\min\{k, n-k\} \geq 3$, $\boldsymbol{\alpha}$, $\boldsymbol{t}$, $\boldsymbol{h}$ *be chosen as in Definition 4.1. Furthermore, let* $\mathcal{H}_i \subseteq \mathbb{F}_q$ *with*

$$7 \leq |\mathcal{H}_1| \leq |\mathcal{H}_2| \leq \cdots \leq |\mathcal{H}_\ell|.$$

be such that the code $\mathcal{C}_{\boldsymbol{\alpha},\boldsymbol{t},\boldsymbol{h},\boldsymbol{\eta}}[n,k]$ *is MDS for all* $\boldsymbol{\eta} \in \mathcal{H} := \mathcal{H}_1 \times \cdots \times \mathcal{H}_\ell$*. If there is an* $\boldsymbol{\eta}^* \in \mathcal{H}$ *such that* $\mathcal{C}_{\boldsymbol{\alpha},\boldsymbol{t},\boldsymbol{h},\boldsymbol{\eta}^*}[n,k]$ *is non-GRS, then at least a fraction*

$$\frac{|\mathcal{H}| - 6\prod_{i=2}^{\ell} |\mathcal{H}_i|}{|\mathcal{H}|} = \frac{\prod_{i=1}^{\ell} |\mathcal{H}_i| - 6\prod_{i=2}^{\ell} |\mathcal{H}_i|}{\prod_{i=1}^{\ell} |\mathcal{H}_i|} = 1 - \frac{6}{|\mathcal{H}_1|} \tag{4.18}$$

of the codes $\mathcal{C}_{\boldsymbol{\alpha},\boldsymbol{t},\boldsymbol{h},\boldsymbol{\eta}}[n,k]$ *given by the* $\boldsymbol{\eta} \in \mathcal{H}$ *is non-GRS.*

[7] In our publication corresponding to this result, [BPR17], we mistakenly assumed that the polynomial p never vanishes, i.e., Case ii) is always fulfilled. Hence, [BPR17, Theorem 18] is not true in general.

[8] These are codes, where multivariate polynomials over a field K of bounded total degree are evaluated at the Cartesian product of subsets of K. For finite K, they are related to Reed–Muller codes, cf. [LRMV14].

Corollary 4.28 implies that if the η_i can be independently chosen from subsets of $\mathbb{F}_q$, then, either all resulting twisted RS codes are GRS or almost all of them are non-GRS. This assumption is, for instance, fulfilled by the constructive class of MDS twisted RS codes in Section 4.2. For most[9] low-rate twisted RS codes, we can drop the condition on the existence of a non-GRS code with the help of Theorem 4.23, and prove the following.

Corollary 4.29. *Let $2k < n$, $\min\{k, n-k\} \geq 3$, and $\boldsymbol{t}$, $\boldsymbol{h}$ be chosen as in Definition 4.1 with the additional requirement that one of the hook parameters h_j fulfills $1 < h_j < k-2$. As in Section 4.2, we choose a chain of fields*

$$\mathbb{F}_{s_0} \lneq \mathbb{F}_{s_1} \lneq \cdots \lneq \mathbb{F}_{s_\ell} = \mathbb{F}_q$$

and distinct evaluation points $\alpha_i \in \mathbb{F}_{s_0}$. Furthermore, let $\mathcal{H} := \mathcal{H}_1 \times \ldots \mathcal{H}_\ell$, where $\mathcal{H}_i = (\mathbb{F}_{s_i} \setminus \mathbb{F}_{s_{i-1}}) \cup \{0\}$. Then, at least

$$(|\mathcal{H}_1| - 6) \prod_{i=2}^{\ell} |\mathcal{H}_i| \geq (n^2 - n - 5) \prod_{i=2}^{\ell} \left(n^{2^i} - n^{2^{i-1}} + 1 \right) \tag{4.19}$$

choices of $\boldsymbol{\eta} \in \mathcal{H}$ are non-GRS MDS codes. The fraction of non-GRS MDS codes for values $\boldsymbol{\eta} \in \mathcal{H}$ is at least

$$1 - \tfrac{6}{|\mathcal{H}_1|} \geq 1 - \tfrac{6}{n^2-n+1} \to 1 \quad (n \to \infty).$$

Proof. By Section 4.2, all choices of $\boldsymbol{\eta} \in \mathcal{H}$ are MDS codes. Since one hook parameter h_j fulfills $1 < h_j < k-2$ and due to $2k < n$, the choice of $\boldsymbol{\eta}^* = [0, \ldots, 0, \eta_j, 0, \ldots, 0] \in \mathcal{H}$ with $\eta_j \neq 0$ results in a non-GRS code $\mathcal{C}_{\boldsymbol{\alpha},\boldsymbol{t},\boldsymbol{h},\boldsymbol{\eta}^*}[n,k]$ by Theorem 4.23. Note that in this case, the twisted RS code technically has only one twist.

Due to $s_0 \geq n$ and $s_i \geq s_{i-1}^2$ for all i, we must have $|\mathcal{H}_i| \geq n^{2^i} - n^{2^{i-1}} + 1$, so in particular $|\mathcal{H}_1| \geq n^2 - n + 1 \geq 7$, where the last inequality follows by $n \geq 3$ due to $\min\{k, n-k\} \geq 3$. Hence, there are at least $(|\mathcal{H}_1| - 6) \prod_{i=2}^{\ell} |\mathcal{H}_i| \geq (n^2 - n - 5) \prod_{i=2}^{\ell} (n^{2^i} - n^{2^{i-1}} + 1)$ many non-GRS MDS codes with $\boldsymbol{\eta} \in \mathcal{H}$ by Corollary 4.28, which implies the claim. □

Example 4.30. *Let all the code parameters be chosen as in Corollary 4.29. For $n = 256$, there are at least 65275, $\approx 4.3 \cdot 10^9$, and $\approx 1.8 \cdot 10^{19}$ non-GRS MDS codes for $\ell = 1, 2, 3$, respectively. The relative number of GRS codes with $\boldsymbol{\eta} \in \mathcal{H}$ is at most $\approx 9 \cdot 10^{-5}$ in this case, independent of ℓ.*

Remark 4.31. *For high-rate twisted RS codes for which the dual code is equivalent to a twisted RS code, we can apply a similar statement as Corollary 4.29 using the observations in Section 4.6.1.*

Summary

In summary, we have the following for codes fulfilling $\min\{k, n-k\} \geq 3$:

- For fixed $\boldsymbol{\alpha}, \boldsymbol{t}, \boldsymbol{h}$ and variable $\boldsymbol{\eta}$, either all MDS twisted RS codes are GRS or the number of non-GRS codes are given by the zeros of a non-zero multivariate polynomial of total degree at most 6.

[9] All except for those with $h_i \in \{0, 1, k-2, k-1\}$ for all $i = 1, \ldots, \ell$.

- In case when we can choose the η_i independently from subsets of $\mathbb{F}_q$, the latter case implies that most twisted RS codes are non-GRS.
- Given that the η_i are chosen from a chain of subfields as in Theorem 4.3 (sufficient MDS condition), the fraction of GRS codes decreases in the code length n with $O(\frac{1}{n^2})$.
- For low-rate codes for which at least one hook coefficient fulfills $1 < h_j < k-2$ (and high-rate codes whose dual is a twisted RS code of this kind), the existence of a non-GRS MDS code is guaranteed, so many (or in some cases most) of the codes are non-GRS.

4.6.3 Separation from GRS Codes Using Schur Squares

We now turn to the more general problem of showing how "far away" twisted RS codes $\mathcal{C}_{\boldsymbol{\alpha},\boldsymbol{t},\boldsymbol{h},\boldsymbol{\eta}}[n,k]$ are from being a GRS code. More precisely, our goal is to determine the smallest dimension of a GRS code $\mathcal{C}_{\mathrm{GRS,out}}$ and the largest dimension of a GRS code $\mathcal{C}_{\mathrm{GRS,inn}}$ such that

$$\mathcal{C}_{\mathrm{GRS,inn}} \subseteq \mathcal{C}_{\boldsymbol{\alpha},\boldsymbol{t},\boldsymbol{h},\boldsymbol{\eta}}[n,k] \subseteq \mathcal{C}_{\mathrm{GRS,out}}.$$

By construction, it is clear that any twisted RS code

- is contained in a large RS code of dimension $k + \max_i\{t_i\}$, since the coefficients with index $\geq k + \max_i\{t_i\}$ of the evaluation polynomials are zero, i.e.,

$$\dim(\mathcal{C}_{\mathrm{GRS,out}}) \leq k + \max_i\{t_i\}.$$

- and conversely contains a small RS code of dimension $\min_i\{h_i\}$, since the monomials $x^0, \ldots, x^{\min_i\{h_i\}-1}$ are evaluation polynomials, i.e.,

$$\dim(\mathcal{C}_{\mathrm{GRS,inn}}) \geq \min_i\{h_i\}.$$

Separation of Low-Rate Codes Based on Schur Squares

We start with general bounds on $\dim \mathcal{C}_{\mathrm{GRS,out}}$ and $\dim \mathcal{C}_{\mathrm{GRS,inn}}$ for low-rate codes.

Theorem 4.32. *Let $k < \frac{n}{2}$ and $\mathcal{C}[n,k]$ be a linear code with $\dim(\mathcal{C}^2) = 2k-1+\delta$ for some $\delta > 0$. Then, any GRS code $\mathcal{C}_{\mathrm{GRS,out}}$ that contains $\mathcal{C}$ has dimension*

$$\dim \mathcal{C}_{\mathrm{GRS,out}} \geq k + \tfrac{\delta}{2}.$$

Proof. Let k' be the dimension of $\mathcal{C}_{\mathrm{GRS,out}}$. Due to $\mathcal{C} \subseteq \mathcal{C}_{\mathrm{GRS,out}}$, we have

$$2k-1+\delta = \dim(\mathcal{C}^2) \leq \dim(\mathcal{C}^2_{\mathrm{GRS,out}}) \leq 2k'-1,$$

which implies $k' \geq k + \delta/2$. □

Theorem 4.33. *Let $k < \frac{n}{2}$ and $\mathcal{C}[n,k]$ be a linear code with $\dim(\mathcal{C}^2) = 2k-1+\delta$ for some $\delta > 0$. Then, any GRS code $\mathcal{C}_{\mathrm{GRS,inn}}$ that is contained in $\mathcal{C}$ has dimension*

$$\dim \mathcal{C}_{\mathrm{GRS,inn}} \leq \sqrt{(k-\tfrac{5}{2})^2 - 2\delta} + \tfrac{5}{2}.$$

Proof. Let k' be the dimension of $\mathcal{C}_{\mathrm{GRS,inn}}$ (note that $k' < \frac{n}{2}$). We choose a basis $\boldsymbol{c}_1, \ldots, \boldsymbol{c}_{k'}$ of $\mathcal{C}_{\mathrm{GRS,inn}}$ and extend it into a basis $\boldsymbol{c}_1, \ldots, \boldsymbol{c}_k$ of $\mathcal{C}$. The Schur products $\boldsymbol{c}_i \star \boldsymbol{c}_j$ for $i, j \leq k'$ span a space of dimension $2k' - 1$. The other Schur products can span a space of dimension at most $k'(k-k') + \binom{k-k'}{2}$, so we have

$$2k - 1 + \delta \leq \dim(\mathcal{C}^2) = 2k' - 1 + (k-k')k' + \tfrac{1}{2}(k-k')(k-k'-1),$$

which implies $(k' - \frac{5}{2})^2 \leq (k - \frac{5}{2})^2 - 2\delta$. □

Theorem 4.32 is quite strong since the difference $(\dim \mathcal{C}_{\mathrm{GRS,out}} - \dim \mathcal{C})$ grows linearly in the difference of the Schur square dimension to its lower bound $2k - 1$. The separation statement for the inner code, Theorem 4.33, provides a significant separation difference $\dim \mathcal{C} - \dim \mathcal{C}_{\mathrm{GRS,inn}}$ only for Schur squares of huge dimension (e.g., $\dim(\mathcal{C}^2) \in \Theta(k^2)$).

Separation of High-Rate Codes based on the Dual Code

For high-rate codes, we can apply the above statements on the dual code since the dual of a GRS code is also a GRS code and

$$\mathcal{C}_{\mathrm{GRS,inn}} \subseteq \mathcal{C} \subseteq \mathcal{C}_{\mathrm{GRS,out}} \quad \Longleftrightarrow \quad \mathcal{C}^{\perp}_{\mathrm{GRS,out}} \subseteq \mathcal{C}^{\perp} \subseteq \mathcal{C}^{\perp}_{\mathrm{GRS,inn}}.$$

In this case, we obtain an upper bound on the dimension of the inner GRS code $\mathcal{C}_{\mathrm{GRS,inn}}$ by applying Theorem 4.32 on the Schur square of the dual code $\mathcal{C}^{\perp}$, and a lower bound on $\dim \mathcal{C}_{\mathrm{GRS,out}}$ using Theorem 4.33.

Separation of Twisted RS Codes

We have seen in Section 4.5 and Section 4.6.1 that many low-rate twisted RS codes have a larger Schur square as an RS code of the same dimension. Hence, we can directly apply the arguments above. In particular, low-rate twisted RS codes of maximal Schur square dimension can be separated from the smallest GRS code containing them using the following corollary.

Corollary 4.34. *Let $k < \frac{n}{2}$ and $\mathcal{C}[n,k]$ be a linear code with full Schur square dimension $\dim(\mathcal{C}^2) = n$. Then, any GRS code $\mathcal{C}_{\mathrm{GRS,out}}$ that contains $\mathcal{C}$ has dimension*

$$\dim \mathcal{C}_{\mathrm{GRS,out}} \geq k + \tfrac{n-2k+1}{2} = \tfrac{n+1}{2}.$$

This shows that the codes of the family constructed in Theorem 4.18 are not contained in a GRS code of dimension less than $\frac{n-2k+1}{2}$ larger than their own dimension. For a fixed rate $R = \frac{k}{n} < \frac{1}{2}$, this separation difference grows linearly in n.

4.7 Subclasses of Long MDS Twisted RS Codes

In this section, we show that there are MDS twisted RS codes of length $\approx \frac{q}{2}$. We consider two constructive subclasses with one twist. For notational convenience, we slightly change the notation in the case $\ell = 1$ into

$$t := t_1, \quad h := h_1, \quad \eta := \eta_1, \quad \mathcal{C}_{\boldsymbol{\alpha},t,h,\eta}[n,k] := \mathcal{C}_{\boldsymbol{\alpha},\boldsymbol{t},\boldsymbol{h},\boldsymbol{\eta}}[n,k].$$

For certain parameters, we also provide upper bounds on the maximal length of such codes, proving that the two constructive classes of MDS twisted RS codes achieve the maximal length in these cases. We use the concept of k-sum generators in finite abelian groups, which was introduced in [RL89a, RL92] and originally used to construct non-RS MDS codes.

Definition 4.35 ([RL92])**.** Let $(A, \oplus)$ be a finite abelian group and $k \in \mathbb{N}$. A *k-sum generator* of A is a subset $S \subseteq A$ such that for any $a \in A$, there are distinct $s_1, \ldots, s_k \in S$ with $a = \bigoplus_{i=1}^{k} s_i$. The smallest integer such that any $S \subset A$ with $|S| > M(k, A)$ is a k-sum generator of A is denoted by $M(k, A)$.

Lemma 4.36 ([RL92, Theorem 3.1])**.** *Let $|A| = 2r$ for some $r \geq 6$. For any k with $3 \leq k \leq r-2$, we have*

$$M(k, A) = \begin{cases} r+1, & \text{if } A \in \{\mathbb{Z}_2^m, \mathbb{Z}_4 \times \mathbb{Z}_2^{m-1}\} \text{ for some } m > 1 \text{ and } k \in \{3, r-2\}, \\ r, & \text{else.} \end{cases}$$

In the remainder of this section, we apply this lemma to the even-order abelian groups $(\mathbb{F}_q^*, \cdot)$ with odd q and $(\mathbb{F}_q, +)$ with even q, respectively. An equivalent statement as Lemma 4.36 for $|A|$ odd can be found in [RL89a, RL92], but we do not make use of it here.

4.7.1 $(*)$-Twisted Reed–Solomon Codes

We first consider the case $[t, h] = [1, 0]$, which is illustrated in Figure 4.6. It can be seen as the $\mathbb{F}_q[x]$-analog of Sheekey's twisted Gabidulin codes [She16], though we use different techniques to analyze our codes.

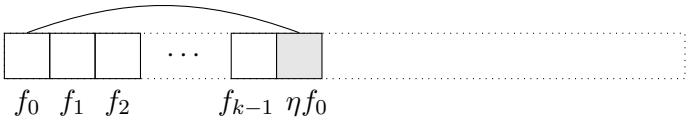


Figure 4.6: Illustration of the evaluation polynomials in the case $[t, h] = [1, 0]$.

In this case, the statement and proof of Lemma 4.2 simplifies significantly and we get the following condition for when the code $\mathcal{C}_{1,0,\eta,\boldsymbol{\alpha}}[n, k]$ is MDS.

Lemma 4.37. *Let $k, n, \boldsymbol{\alpha}$ be chosen as in Definition 4.1 and $\eta \in \mathbb{F}_q$. The code $\mathcal{C}_{\boldsymbol{\alpha},1,0,\eta}[n, k]$ is MDS if and only if*

$$\eta(-1)^k \prod_{i \in \mathcal{I}} \alpha_i \neq 1 \quad \forall \mathcal{I} \subseteq \{1, \ldots, n\} \text{ s.t. } |\mathcal{I}| = k. \tag{4.20}$$

Proof. Let $f = \sum_{i=0}^{k-1} f_i x^i + \eta f_0 x^k$ be an evaluation polynomial with $f_0 \neq 0$. If there is a subset $\mathcal{I} \subseteq \{1, \ldots, n\}$ with $|\mathcal{I}| = k$ and $f(\alpha_i) = 0$ for all $i \in \mathcal{I}$, then we can write $f = \eta f_0 \prod_{i \in \mathcal{I}} (x - \alpha_i)$ with constant term $f_0 = f(0) = \eta f_0 \prod_{i \in \mathcal{I}} (-\alpha_i)$, so it must fulfill

$$\eta(-1)^k \prod_{i \in \mathcal{I}} \alpha_i = 1.$$

Hence, f has at most $k-1$ roots among the α_i if and only if (4.20) is fulfilled. In the case $f_0 = 0$, any non-zero polynomial has degree at most $k-1$, implying that it always has at most $k-1$ roots among the α_i. We conclude that all non-zero evaluation polynomials have at most $k-1$ roots among the α_i if and only if (4.20) is satisfied. □

Condition (4.20) implies that if $(-1)^k\eta^{-1}$ cannot be represented by the product of k distinct evaluation points, then the resulting code is MDS. By choosing the evaluation points from a multiplicative group and $(-1)^k\eta^{-1}$ to be a value not in this group, we obtain the following class of MDS codes.

Definition 4.38. Let G be a proper subgroup of $(\mathbb{F}_q^*, \cdot)$, $\alpha_i \in G \cup \{0\}$ for all i, and $(-1)^k\eta^{-1} \in \mathbb{F}_q^* \setminus G$. Then, we call $\mathcal{C}_{\boldsymbol{\alpha},1,0,\eta}[n,k]$ a $(*)$-twisted code.

Theorem 4.39. *Any $(*)$-twisted code is MDS.*

Proof. For any $\mathcal{I} \subseteq \{1,\ldots,n\}$, we have $\prod_{i\in\mathcal{I}} \alpha_i \in G \cup \{0\}$. Since $(-1)^k\eta \notin G \cup \{0\}$, Condition (4.20) is fulfilled and the code is MDS. □

Theorem 4.39 implies that $(*)$-twisted codes can be rather long, as formally stated in the following corollary.

Corollary 4.40. *Let p be a prime divisor of the field size q. Then, there is a $(*)$-twisted code of length*

$$n = \tfrac{q-1}{p} + 1.$$

Proof. There is a proper subgroup G of $\mathbb{F}_q^*$ of cardinality $(q-1)/p$. We obtain the claimed length by choosing $\{\alpha_1,\ldots,\alpha_n\} = G \cup \{0\}$. □

If q is odd, then $(*)$-twisted codes can have length $n = \frac{q+1}{2}$.

Maximal Length of a $[1,0,\eta]$-Twisted Code

The construction using a multiplicative group is rather limited and one could raise the question whether different choices of $\boldsymbol{\alpha}$ would yield longer codes. The following analysis answers this negatively using k-sum generators.

Lemma 4.41. *Let $k, n, \boldsymbol{\alpha}, \boldsymbol{\eta}$ be chosen as in Definition 4.1 such that $S := \{\alpha_1,\ldots,\alpha_n\} \subseteq \mathbb{F}_q^*$ is a k-sum generator of $(\mathbb{F}_q^*, \cdot)$. Then, the code $\mathcal{C}_{\boldsymbol{\alpha},1,0,\eta}[n,k]$ is not MDS.*

Proof. Since S is a k-sum generator of $(\mathbb{F}_q^*, \cdot)$ and $(-1)^k\eta^{-1} \neq 0$, there is an index set $\mathcal{I} \subseteq \{1,\ldots,n\}$ with $|\mathcal{I}| = k$ such that $\prod_{i\in\mathcal{I}} \alpha_i = (-1)^k\eta^{-1}$, which implies that $\mathcal{C}_{\boldsymbol{\alpha},1,0,\eta}[n,k]$ is not MDS by Lemma 4.37. □

Theorem 4.42. *Let q be odd and $3 \leq k \leq \frac{q-1}{2} - 2$. If the code length satisfies $n > \frac{q+1}{2}$, then the code $\mathcal{C}_{\boldsymbol{\alpha},1,0,\eta}[n,k]$ is not MDS for any choice of $\boldsymbol{\alpha}$ as in Definition 4.1.*

Proof. The abelian group $(\mathbb{F}_q^*, \cdot)$ is cyclic and of even order $|\mathbb{F}_q^*| = q-1$ since q is odd. Thus, Lemma 4.36 implies

$$M(k, \mathbb{F}_q^*) = \tfrac{q-1}{2}.$$

For $n > \frac{q+1}{2}$, the set $S := \{\alpha_1, \ldots, \alpha_n\} \cap \mathbb{F}_q^*$ is therefore a k-sum generator of $\mathbb{F}_q^*$ since

$$|S| \geq n - 1 > \tfrac{q+1}{2} - 1 = \tfrac{q-1}{2} = M(k, \mathbb{F}_q^*)$$

By Lemma 4.41, the code $\mathcal{C}_{\boldsymbol{\alpha},1,0,\eta}[n,k]$ is not MDS. □

Remark 4.43. *The statements above only hold for odd q. For even $q > 4$, the $(*)$-twisted codes cannot attain length $\lfloor (q+1)/2 \rfloor$ since any proper subgroup of $(\mathbb{F}_q^*, \cdot)$ cannot have order greater than $\frac{q-1}{p}$, where $p > 2$ is the smallest prime divisor of $q-1$.*

When allowing arbitrary evaluation points and $\eta \in \mathbb{F}_q^*$, such long codes do exist: Our computer search (cf. Section 4.7.3) shows, e.g., for $q = 16$, there are many MDS $[1, 0, \eta]$-twisted RS codes of length $\lfloor (q+1)/2 \rfloor = 9$ for $k = 3, 4, 5$.

Duals of $(*)$-twisted Codes

Since the dual of an MDS code is MDS, we can use the results on $(*)$-twisted codes to predict the minimum distance of their duals, which are $[n-k, k-1, \eta]$-twisted codes (cf. 4.7).

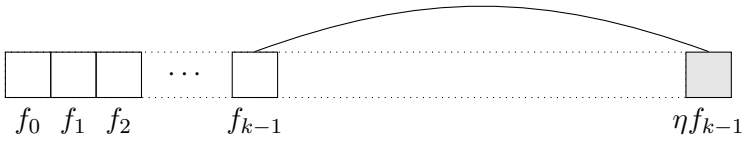


Figure 4.7: Illustration of the evaluation polynomials in the case $[t, h] = [n-k, k-1]$.

Theorem 4.44. *Let G be a proper subgroup of $(\mathbb{F}_q^*, \cdot)$ and $k, n, \boldsymbol{\alpha}$ be chosen as in Definition 4.1 such that $\{\alpha_1, \ldots, \alpha_n\} = G$ and $(-1)^{n-k+1}\eta^{-1} \notin G \cup \{0\}$. Then, the code $\mathcal{C}_{\boldsymbol{\alpha},n-k,k-1,\eta}[n,k]$ is MDS.*

Proof. By Theorem 4.10, the dual code of $\mathcal{C}_{\boldsymbol{\alpha},n-k,k-1,\eta}[n,k]$ is equivalent to

$$\mathcal{C}_{\boldsymbol{\alpha},1,0,-\eta}[n, n-k].$$

Due to $(-1)^{n-k+1}(-\eta)^{-1} = (-1)^{n-k}\eta^{-1} \notin G \cup \{0\}$, this code is a $(*)$-twisted code, which implies the claim. □

Non-GRS $(*)$-Twisted RS Codes

$(*)$-Twisted codes are only interesting if they are not equivalent to an RS code. The following statement shows that for $k \geq 3$, all low-rate $(*)$-twisted codes are non-GRS.

Theorem 4.45. *Let $3 \leq k \leq \frac{n-1}{2}$. Then, any $(*)$-twisted -code is non-GRS.*

Proof. We consider the Schur square dimension of a $(*)$-twisted -code $\mathcal{C}$. Since the evaluation polynomials of $\mathcal{C}$ have degrees $1, \ldots, k$, there are evaluation polynomials of degrees $2, \ldots, 2k$ of the Schur square code. Furthermore, since $x^1, x^2, x^{k-1}, \eta x^k + x^0 \in \mathcal{P}^{n,k}_{\boldsymbol{t},\boldsymbol{h},\boldsymbol{\eta}}$ (note that we require $2 \leq k-1$), the polynomial

$$x^1 \cdot (\eta x^k + x^0) - \eta x^{k-1} \cdot x^2 = x^1,$$

is among the evaluation points of the Schur square code. Due to $2k < n-1$, the polynomials $x^1, \ldots, x^{2k}$ form a $2k$-dimensional subspace of the Schur square evaluation points, so we have $\deg(\mathcal{C}^2) \geq 2k$, which proves the claim. □

For high-rate codes with $k \leq n-3$ and $\boldsymbol{\alpha}$ being the full multiplicative group, also all high-rate $(*)$-twisted codes are non-GRS.

Theorem 4.46. *Let $\frac{n-1}{2} < k \leq n-3$ and the $\boldsymbol{\alpha}$ form a multiplicative group. Then, any $(*)$-twisted -code with evaluation points $\boldsymbol{\alpha}$ is non-GRS.*

Proof. By Theorem 4.10, the dual code of the $(*)$-twisted code is equivalent to a low-rate twisted RS code $\mathcal{C}[n, n-k]$ with $[t, h] = [k, n-k-1]$, which is spanned the polynomials $x^0, \ldots, x^{n-k-2}, \eta' x^{n-1} + x^{n-k-1}$ for some $\eta' \neq 0$, and the same evaluation points $\boldsymbol{\alpha}$. Thus, there are evaluation polynomials of the Schur square code $\mathcal{C}^2$ of degrees $0, \ldots, 2(n-k)-4$, and $n-1$. Since $\boldsymbol{\alpha}$ forms a multiplicative group, we have $A = \prod_{i=1}^{n}(x - \alpha_i) = x^n - 1$, so

$$x^i \left(\eta' x^{n-1} + x^{n-k-1}\right) \equiv \eta' x^{i-1} + x^{n-k-1+i} \mod A \quad \forall\, i = 1, \ldots, k.$$

Due to $n-k \geq 3$, x^{n-k-2} is an evaluation polynomial of degree > 0 and we obtain a Schur square evaluation polynomial of degree $2(n-k)-3$ by $x^{n-k-2} \cdot (\eta' x^{n-1} + x^{n-k-1}) \bmod A$, and of degree $2(n-k)-2$ by $(\eta' x^{n-1} + x^{n-k-1})^2 \bmod A$.

Thus, the Schur square evaluation polynomial set is spanned by polynomials of degrees $0, \ldots, 2(n-k)-2, n-1$, so $\dim(\mathcal{C}^2) \geq 2(n-k)$ (note that we require $\frac{n-1}{2} < k$) and $\mathcal{C}$ is not a GRS code (recall that the dimension of $\mathcal{C}$ is $n-k \leq \frac{n-1}{2}$). □

We will see in the computer searches (cf. Section 4.7.3) that there are some $(*)$-twisted codes with $n = 6$ and $k = 3$ that are GRS codes. Note that in this case, we have $\frac{n-1}{2} < k$. Hence, the statement of Theorem 4.46 does not hold for arbitrary evaluation points $\boldsymbol{\alpha}$.

4.7.2 $(+)$-Twisted Reed–Solomon Codes

In the case $[t, h] = [1, k-1]$ (cf Figure 4.8), we obtain an analogous code class as the $(*)$-twisted codes, using additive subgroups of $\mathbb{F}_q$.

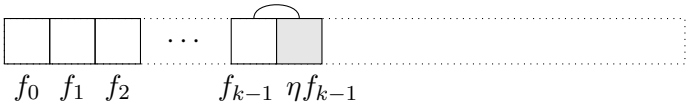


Figure 4.8: Illustration of the evaluation polynomials in the case $[t, h] = [1, k-1]$.

The following lemma is the analog of Lemma 4.37.

Lemma 4.47. *Let $k, n, \boldsymbol{\alpha}$ be chosen as in Definition 4.1 and $\eta \in \mathbb{F}_q$. Then, the code $\mathcal{C}_{\boldsymbol{\alpha},1,k-1,\eta}[n, k]$ is MDS if and only if*

$$\eta \sum_{i \in \mathcal{I}} \alpha_i \neq -1 \quad \forall\, \mathcal{I} \subseteq \{1, \ldots, n\} \text{ s.t. } |\mathcal{I}| = k. \tag{4.21}$$

Proof. The proof works as the one of Lemma 4.37. Any evaluation polynomial with $f_{k-1} \neq 0$ with k zeros at $\mathcal{I} \subseteq \{1, \ldots, n\}$ with $|\mathcal{I}| = k$ can be written as $\eta f_{k-1} \prod_{i \in \mathcal{I}}(x - \alpha_i)$, so its $(k-1)^{\text{th}}$ coefficient is

$$f_{k-1} = \eta f_{k-1} \prod_{i \in \mathcal{I}} (-\alpha_i).$$

Thus, no evaluation polynomials has more than $k-1$ zeros among the α_i if and only if (4.21) is fulfilled. □

Hence, the code is MDS if $-\eta^{-1}$ cannot be represented as the sum of any k evaluation points. This yields the following subclass of MDS twisted RS codes.

Definition 4.48. Let V be a proper subgroup of $(\mathbb{F}_q, +)$ and $\eta^{-1} \in \mathbb{F}_q \backslash V$. Then, $\mathcal{C}_{\boldsymbol{\alpha},1,0,\eta}[n,k]$ is called $(+)$-*twisted code* if the elements of $\boldsymbol{\alpha}$ are a subset of $V \cup \{\infty\}$.

Lemma 4.47 directly implies the following statement.

Theorem 4.49. *Any $(+)$-twisted code is an MDS code.*

Corollary 4.50. *Let p be the characteristic of $\mathbb{F}_q$. Then, there is a $(+)$-twisted code of length*

$$n = \tfrac{q}{p}.$$

Proof. There is a proper subgroup V of $(\mathbb{F}_q, +)$ with order q/p. □

Corollary 4.50 implies that for even q, $(+)$-twisted codes can have length $n = \frac{q}{2}$. The group V as in the proof of the corollary has the maximal size among all proper subgroups.

Maximal Length of an MDS $[1, k-1, \eta]$-Twisted Code

We again analyze the maximal achievable length if the restriction to evaluation points in subgroups is omitted.

Lemma 4.51. *Let $k, n, \boldsymbol{\alpha}$ be chosen as in Definition 4.1 such that $S := \{\alpha_1, \ldots, \alpha_n\} \in \mathbb{F}_q$ is a k-sum generator of $(\mathbb{F}_q, +)$. Then, the code $\mathcal{C}_{\boldsymbol{\alpha},1,k-1,\eta}[n,k]$ is not MDS.*

Theorem 4.52. *Let q be even and $3 \leq k \leq \frac{q}{2} - 2$. If the code length satisfies*

$$n > \begin{cases} \frac{q}{2}, & \text{if } 3 < k < \frac{q}{2} - 3, \\ \frac{q}{2} + 1, & \text{if } k \in \{3, \frac{q}{2} - 2\}, \end{cases}$$

then the twisted code $\mathcal{C}_{\boldsymbol{\alpha},1,k-1,\eta}[n,k]$ is not MDS for any choice of $\boldsymbol{\eta}$ as in Definition 4.1.

Extended $[t, k-1, \eta]$-Twisted Codes

For any $1 \leq t \leq n - k$, we can extend a $[1, k-1, \eta]$-twisted code by adding an evaluation point at infinity, whose evaluation is defined by

$$f(\infty) = f_{k-1}.$$

If the original code is MDS, so is the extended code since any non-zero evaluation polynomial with $f_{k-1} = 0$ has at most $k-2$ zeros among the original α_i, which implies that the maximal number of zero entries in the corresponding codeword is still $k-1$.

Using this observation, we obtain MDS twisted RS codes of length $n = \frac{q}{p} + 1$ using Corollary 4.50, and the bounds on the maximal length of an MDS $[1, k-1, \eta]$-twisted code in Theorem 4.52 become $\frac{q}{2} + 1$ and $\frac{q}{2} + 2$, respectively.

Non-GRS (+)-Twisted RS Codes

We also analyze the connection of (+)-twisted codes to RS codes. Since in this case, the evaluation points usually do not form a multiplicative group, we can apply the Schur square argument only to low-rate (+)-twisted codes. Our computer searches in the following subsection indicate that also most high-rate (+)-twisted codes are non-GRS.

Theorem 4.53. *Let $k < \frac{n}{2}$. Then any (+)-twisted code $\mathcal{C}$ is non-GRS.*

Proof. The evaluation polynomials of $\mathcal{C}$ have degrees $0, \ldots, k-2, k$. Thus, the Square code evaluation polynomial set obviously contains polynomials of degrees $0, \ldots, 2k-2, 2k$. Due to $2k < n$, we have $\dim(\mathcal{C}^2) \geq 2k$, so $\mathcal{C}$ is non-GRS. □

4.7.3 Computer Searches

In the following, we present computer searches for MDS twisted RS codes with one twist $\ell = 1$ over small fields. The searches were implemented in SageMath v7.4 [S+].[10] Since any twisted RS code with $\eta = \eta_1 = 0$ or $\min\{k, n-k\} \leq 2$ is a GRS code (cf. Lemma 4.24), we exclude these cases from the set of tested parameters.

Number of (∗)-Twisted Codes

We counted all (∗)-twisted codes over $\mathbb{F}_q$ for $q \leq 19$, i.e., the number of evaluation point sets $\{\alpha_1, \ldots, \alpha_n\}$ and corresponding $\eta \in \mathbb{F}_q$ that satisfy the conditions in Definition 4.38 for a given length n and dimension k. Furthermore, we determined the number of resulting equivalence classes of codes, as well as the number of non-GRS classes thereof.

For odd field sizes q, we can obtain MDS codes of length up to $\frac{q+1}{2}$, independent of k, which confirms Corollary 4.40. Furthermore, almost all found (∗)-twisted codes are non-GRS. Table 4.2 shows the results for $q = 19$.

Table 4.2: Number of (∗)-twisted codes over $\mathbb{F}_{19}$ (Total/Inequivalent/Non-GRS).

$n\backslash k$	3	4	5	6	7
6	1974/73/67				
7	1092/67/67	1092/63/63			
8	405/25/25	405/25/25	405/25/25		
9	90/7/7	90/6/6	90/6/6	90/7/7	
10	9/2/2	9/1/1	9/1/1	9/2/2	9/1/1

Relation to Roth–Lempel Codes

In [RL89a], Roth and Lempel presented a construction of non-GRS MDS codes with parameters $[n, k]$ based on finding subsets $S \subset \mathbb{F}_q$ with $|S| = n-2$ which are *not* $(k-1)$-sum

[10] The SageMath scripts were implemented by Johan Rosenkilde, one of the thesis author's co-authors in the publication corresponding to this result: [BPR17]. Full results and the source code can be downloaded from `http://jsrn.dk/code-for-articles`.

generators of $(\mathbb{F}_q, +)$. Similar to our $(+)$-twisted codes, subgroups of $\mathbb{F}_q$ are one way to obtain such non-$(k-1)$-sum generators. Hence, these codes can achieve similar parameters as our $(+)$-twisted codes, i.e., $\approx \frac{q}{2}$ for even q. For odd q, the results in [RL89a] imply that the code length is bounded asymptotically (for $q \to \infty$) by $\frac{q}{k}(1 + o(1))$, which is much shorter than our $(*)$-twisted codes in this case.

For small q, we determined all Roth–Lempel codes (i.e., the non-$(k-1)$-sum generators[11]) of given n and k, and counted in how many cases they were equivalent to our $(*)$-twisted codes or MDS (extended) $[1, k-1, \eta]$-twisted codes[12]. The results indicate that the three code classes have only little overlap. Some examples are given in Table 4.3.

Table 4.3: Number of inequivalent Roth–Lempel (RL), $(*)$-twisted , and MDS $[1, k-1, \eta]$-twisted codes / inequivalent to RL codes thereof.

q	n	k	RL codes	$(*)$-twisted codes	MDS $[1, k-1, \eta]$-twisted codes
13	7	2	35	2/1	8/6
16	8	5	186	0/0	83/73
23	12	5	0	1/1	0/0

All MDS Twisted Codes with one Twist up to Field of Size $q = 13$

Table 4.4 shows the number of MDS twisted RS codes for $q \leq 13$ (i.e., the number of evaluation point sets and parameters t, h, and η that result in MDS codes), the number of inequivalent codes, and the number of non-GRS equivalence classes. It can be observed that most cases, there are many fewer equivalence classes than parameter sets (e.g., $[q, n, k] = [13, 7, 4]$ each class can be obtained by ≈ 1200 parameters on average). Furthermore, most of the codes are non-GRS, as predicted by Section 4.6. We found three codes of length $n = q$:

- The code $[q, n, k] = [9, 9, 5]$ is equivalent to Glynn's code [Gly86], which is known as the first example of a code for odd q with $2 \leq k \leq q$ that is a non-GRS MDS code.
- $[q, n, k] = [9, 9, 4]$ is equivalent to the dual of Glynn's code.
- The code $[q, n, k] = [8, 8, 5]$ can be constructed using $t = 1$, $h = k - 1 = 4$, $\eta = 1 \in \mathbb{F}_8$, and $\{\alpha_1, \ldots, \alpha_8\} = (\mathbb{F}_8 \setminus \{1\}) \cup \{\infty\}$. It is equivalent to a Roth–Lempel code.

Although we do not have an explicit construction, except for the $(*)$-twisted and $(+)$-twisted codes, most choices of t and h admit codes of length $\approx \frac{q}{2}$ within the searched parameter range.

[11] In order to achieve codes of length n, the construction in [RL89a] always adds an evaluation point at infinity ($[0, \ldots, 0, 1]^\top$-column in G). For a fair comparison with our $(+)$-twisted codes, we allow to instead evaluate at one more $\alpha_{n-1} \in \mathbb{F}_q$ s.t. $S \cup \{\alpha_{n-1}\}$ with $|S \cup \{\alpha_{n-1}\}| = n - 1$ is not a $(k-1)$-sum generator of $\mathbb{F}_q$.

[12] By Section 4.7.2, these are exactly the codes with non-k-sum generators S of size $|S| = n$ as evaluation points, where η^{-1} cannot be represented as a sum of k elements in S, or an extension of such a code.

Table 4.4: Number of MDS Twisted Codes (Total/Inequivalent/Non-GRS. Blank = 0/0/0)

q	n	$k=3$	4	5	6	7	8	9
7	6	38/3/2						
8	6	406/5/4						
	7		63/2/1					
	8			14/2/1				
9	6	2374/7/5						
	7	216/3/3	332/3/2					
	8	4/1/1	40/1/1	36/1/1				
	9		4/1/1	4/1/1				
11	6	32518/15/11						
	7	6286/21/19	8554/20/18					
	8	585/15/15	160/7/6	960/9/7				
	9	40/3/3		20/1/0	135/1/0			
	10	2/1/1				22/2/1		
13	6	216722/26/21						
	7	71618/80/75	98430/80/75					
	8	11164/165/160	5176/98/93	26916/139/134				
	9	1110/32/31	41/4/3	381/8/5	5424/24/21			
	10	138/4/4			93/3/0	1167/4/1		
	11	24/1/1					254/1/0	
	12	2/1/1						26/2/1

4.8 Twisted RS Codes in the McEliece Cryptosystem

As a potential application of twisted RS codes, we analyze the McEliece [McE78] cryptosystem when used in combination with subclasses of the new codes. The system is a code-based public-key cryptosystem, which was introduced in 1978, in the same year as Rivest, Shamir, and Adleman proposed their famous RSA system [RSA78]. In 1986, Niederreiter [Nie86] introduced a second code-based public-key cryptosystem. Compared to other public-key cryptosystems, such as RSA or ElGamal [ElG85], code-based systems remained unfavorable for a long time due to large key sizes.

However, interest in code-based systems has significantly increased in the past years due to the potential thread by quantum computers: Systems based on the problem of integer factorization (e.g., RSA) or discrete logarithms (e.g., ElGamal) can be broken in polynomial time by such computers using Shor's algorithm [Sho99]. For the McEliece and Niederreiter cryptosystem, no such polynomial-time attack is known, see e.g. [OS09]. In 2017, the National Institute of Standards and Technology (NIST) of the US Department of Commerce started the first post-quantum cryptography standardization process [Nat17], and many variants of the McEliece and Niederreiter system are among the submitted proposals.

In the following, we describe the McEliece cryptosystem, which is based on publishing the generator matrix of an *obfuscated* code, in which it is easy to encode (= encryption) but hard to decode (= decryption) in general.[13] The private key enables the desired addressee to efficiently decrypt the cypher.

1. **Key Generation:**
   - Choose a linear code $\mathcal{C}[n, k, d]$ over $\mathbb{F}_q$, with generator matrix $\boldsymbol{G}$ and efficient decoding algorithm $\mathrm{dec}(\cdot)$ that can correct up to τ errors, from a given family $\mathcal{F}$ of codes at random.
   - Choose an invertible matrix $\boldsymbol{S} \in \mathbb{F}_q^{k\times k}$ at random.
   - Choose a permutation matrix[14] $\boldsymbol{P} \in \mathbb{F}_q^{n\times n}$ at random.
   - Compute $\widetilde{\boldsymbol{G}} = \boldsymbol{S} \cdot \boldsymbol{G} \cdot \boldsymbol{P}$ (*obfuscated generator matrix*).

   The secret key is $[\boldsymbol{S}, \boldsymbol{G}, \boldsymbol{P}, \mathrm{dec}(\cdot)]$ and the public key is $[\widetilde{\boldsymbol{G}}, \tau]$.
2. **Encryption:**
   - Message $\boldsymbol{m} \in \mathbb{F}_q^k$.
   - Choose $\boldsymbol{e} \in \mathbb{F}_q^n$ with $\mathrm{wt_H}(\boldsymbol{e}) = \tau$ at random.
   - Encrypt $\boldsymbol{r} = \boldsymbol{m} \cdot \widetilde{\boldsymbol{G}} + \boldsymbol{e}$.
3. **Decryption:**
   - Decode $\mathrm{dec}(\boldsymbol{r}\boldsymbol{P}^{-1}) = \mathrm{dec}((\boldsymbol{m}\boldsymbol{S})\boldsymbol{G} + \boldsymbol{e}\boldsymbol{P}^{-1})$ in order to obtain $\boldsymbol{m}' = \boldsymbol{m}\boldsymbol{S}$.
   - Compute $\boldsymbol{m} = \boldsymbol{m}'\boldsymbol{S}^{-1}$

Note that the decryption works since $(\boldsymbol{m}\boldsymbol{S})\boldsymbol{G} \in \mathcal{C}[n, k, d]$ and $\mathrm{wt_H}(\boldsymbol{e}\boldsymbol{P}^{-1}) = \mathrm{wt_H}(\boldsymbol{e}) = \tau$ since $\boldsymbol{P}^{-1}$ is a permutation matrix. The security of the system is based on the NP-completeness of the general decoding problem, cf. [BMVT78]. This means that an attacker faces an NP-complete problem unless she or he is able to retrieve the structure of an efficiently decodable code with generator matrix $\widetilde{\boldsymbol{G}}$. Attacks of the latter kind are called *structural attacks*.

The code family $\mathcal{F}$ which was proposed in the original McEliece cryptosystem consists of binary Goppa codes. For this code class, no polynomial-time structural attack has been found. Since then, many other code classes have been considered (cf. the overview in [OS09]) However, for most of these codes, structural attacks have been found. A comprehensive introduction to code-based cryptosystems can be found in [OS09].

For RS-like codes, structural attacks have been found by Sidelnikov and Shestakov [SS92], Wieschebrink [Wie06], Couvreur et al. [CGGU+14], and an improved attack by Wieschebrink [Wie10]. In the following, we argue that there is a subfamily of twisted RS codes that resist some of these structural attacks. More precisely, we can show that Sidelnikov–Shestakov and Couvreur et al.'s attack, as well as Wieschebrink's attack on the code itself, do not work for the codes. Wieschebrink's attack on the dual code, as well as the improved attack by Wieschebrink, do not apply in an obvious way.

[13] Niederreiter's cryptosystem is based on publishing an obfuscated parity-check matrix of an efficiently decodable code and encrypting the message as an error, whose syndrome is the cypher. We only consider the McEliece system here since the systems are equivalent in terms of structural attacks: one can easily obtain an obfuscated generator matrix from a parity-check matrix and vice versa.

[14] I.e., a matrix containing exactly one 1 per column and row, and zero otherwise.

4.8.1 Twisted RS Codes Resisting Some Known Structural Attacks

We analyze the following subfamily of twisted RS codes with respect to known structural attacks on the McEliece cryptosystem using RS-like codes. We combine the ideas of Theorem 4.3 (sufficient MDS condition) and Theorem 4.21 (shortened maximal Schur square codes) in order to obtain a subfamily of twisted RS codes that are MDS and have maximal Schur square when shortened by at most $\varrho = 2$ positions.

Definition 4.54. Let $k, n \in \mathbb{N}$, be such that $n > 67$ and $2\sqrt{n} + 6 < k \leq \frac{n}{2}$. Choose

$$\ell = \left\lfloor \tfrac{n+1}{k-\sqrt{n}} \right\rfloor - 1,$$
$$r = \left\lceil \tfrac{n+1}{\ell+2} \right\rceil + 2,$$

and define the hook and twist parameters $\boldsymbol{h}, \boldsymbol{t} \in \mathbb{N}_0^\ell$ as

$$h_i = r - 1 + i \quad \text{and} \quad t_i = (i+1)(r-2) - k + 2 \quad \text{for } i = 1, \ldots, \ell.$$

Furthermore, fix a chain of subfields $\mathbb{F}_{s_0} \leq \mathbb{F}_{s_1} \leq \cdots \leq \mathbb{F}_{s_\ell} =: \mathbb{F}_q$ such that $|\mathbb{F}_{s_0}| \geq n$. We define the following family of codes:

$$\mathcal{F}_{n,k} = \left\{ \mathcal{C}_{\boldsymbol{\alpha},\boldsymbol{t},\boldsymbol{h},\boldsymbol{\eta}}[n,k] \,:\, \alpha_1, \ldots, \alpha_n \in \mathbb{F}_{s_0} \setminus \{0\} \text{ distinct}, \, \eta_i \in \mathbb{F}_{s_i} \setminus \mathbb{F}_{s_{i-1}} \right\}.$$

Note that the parameters of the codes contained in the family are well-defined by the analysis in Section 4.5.2 (ℓ is chosen minimal fulfilling (4.8)).

Remark 4.55. *For a fixed chain of subfields, the number of distinct parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\eta}$ that result in codes in $\mathcal{F}_{n,k}$ is given by*

$$n! \binom{|\mathbb{F}_{s_0}| - 1}{n} \prod_{i=1}^{\ell} |\mathbb{F}_{s_i} \setminus \mathbb{F}_{s_{i-1}}| \geq n! \binom{|\mathbb{F}_{s_0}| - 1}{n} \prod_{i=1}^{\ell} \left(n^{2^i} - n^{2^{i-1}} \right).$$

which already for small n gives a sufficiently large number of possible parameters to prevent efficient naive brute-force structural attacks on the family. The number of inequivalent codes is presumably much smaller (cf. the examples for small field sizes in Section 4.7.3), but in order to utilize this fact, an attacker must determine the equivalence classes of the family.

In the following, we consider Sidelnikov and Shestakov's, Wieschebrink's, Couvreur et al.'s, and Wieschebrink's improved attacks on codes in the family $\mathcal{F}_{n,k}$. Since knowing the structure of the dual of a code might reveal the code's structure (we have an explicit formula for some codes in Section 4.4, but more general statements might be possible), we also analyze the attacks on the dual codes. The latter is also important if we want to use high-rate codes in the McEliece cryptosystem: We can simply use the dual codes of the codes in $\mathcal{F}_{n,k}$.

Sidelnikov–Shestakov Attack

The Sidelnikov–Shestakov attack [SS92] is a polynomial-time structural attack on the McEliece system with GRS codes as the code family. We describe the attack as in [Wie10]. Let $[\boldsymbol{I} \mid \boldsymbol{A}]$ be the unique systematic generator matrix of a GRS code with evaluation points $\alpha_1, \ldots, \alpha_n$,

where the positions of the codewords are multiplied by the non-zero field elements $v_1, \ldots, v_n$.[15] Then, row i of the matrix is obtained by evaluating the polynomial

$$f_i = c_i \prod_{\mu=1,\mu\neq i}^{k} (x - \alpha_\mu),$$

where $c_i \in \mathbb{F}_q$, and the entries of the matrix $\boldsymbol{A}$ are given by

$$A_{ij} = v_j f_i(\alpha_j) = \frac{c_i \cdot v_j \prod_{\mu=1}^{k} (\alpha_j - \alpha_\mu)}{\alpha_j - \alpha_i} =: \frac{c_i \cdot d_j}{\alpha_j - \alpha_i} \tag{4.22}$$

for $i = 1, \ldots, k$ and $j = k+1, \ldots, n$ (cf. [RS85]). By guessing the four values $\alpha_1, \alpha_2, c_1, c_2 \in \mathbb{F}_q$ ($\leq q^4$ possibilities[16]), one obtains the values $\alpha_{k+1}, \ldots, \alpha_n$ by solving the equations

$$\frac{A_{1j}}{A_{2j}} = \frac{c_1(\alpha_j - \alpha_2)}{c_2(\alpha_j - \alpha_1)} \quad \forall\, j = k+1, \ldots, n.$$

By a similar strategy, the remaining evaluation points $\alpha_3, \ldots, \alpha_k$ and $v_1, \ldots, v_n$ are found.

The attack relies on the structure of the matrix $\boldsymbol{A}$ given by (4.22). Such a matrix is called Cauchy matrix. Is is well-known that a matrix $\boldsymbol{A}$ is Cauchy if and only if $[\boldsymbol{I} \mid \boldsymbol{A}]$ generates a GRS code [RS85, RL89b]. Since all codes in the family $\mathcal{F}_{n,k}$ have Schur square dimension larger than $2k-1$, none of them is GRS, so the Sidelnikov–Shestakov is neither applicable to the codes in the family nor to their duals.

Wieschebrink Attack

Wieschebrink [Wie06] proposed a generalization of the Sidelnikov–Shestakov attack, which works for subcodes $\mathcal{C}[n,k]$ of a GRS code $\mathcal{C}_{\mathrm{GRS,out}}[n, k+\delta]$. Again, the i^{th} row of the systematic generator matrix $[\boldsymbol{I} \mid \boldsymbol{A}]$ of $\mathcal{C}$ is obtained from a polynomial

$$f_i = c_i(x) \prod_{\mu=1,\mu\neq i}^{k} (x - \alpha_\mu),$$

which is also an evaluation polynomial of $\mathcal{C}_{\mathrm{GRS,out}}$, and thus, $c_i(x) \in \mathbb{F}_q[x]$ is a polynomial of degree $\deg c_i(x) \leq \delta$. Compared to the Sidelnikov–Shestakov attack, an attacker need to guess two polynomials $c_1(x)$ and $c_2(x)$ instead of scalars, which costs about $\approx q^{2\delta}$ iterations, and is efficient only for small δ.

Hence, for $\mathcal{C} \in \mathcal{F}_{n,k}$, we need to ensure that the smallest GRS code containing $\mathcal{C}$ has much larger dimension than k. Fortunately, this is guaranteed due to the full Schur square dimension and Corollary 4.34: For each code in the family, we have $\delta \geq \frac{n+1}{2} - k$, which grows linearly in n for a fixed rate $R = \frac{k}{n} < \frac{1}{2}$.

Showing such a property for the dual code is more involved. We can only state the following:

- Since $\mathcal{C}_{\mathrm{GRS,inn}} \subseteq \mathcal{C}$ if and only if $\mathcal{C}^\top \subseteq \mathcal{C}^\top_{\mathrm{GRS,inn}}$, we can use Theorem 4.33, our result on the separation from an inner GRS code. For the code in $\mathcal{F}_{n,k}$, we obtain

  $$\delta \geq (k - \tfrac{5}{2}) - \sqrt{(k - \tfrac{5}{2})^2 - (n - 2k + 1)} =: \delta_{\min},$$

  which tends to 0 for a fixed rate and $n \to \infty$.

[15] The v_i are usually called *code multipliers*.
[16] In fact, one can assume w.l.o.g. that $\alpha_1 = 0$ and $\alpha_2 = 1$, cf. [Wie10].

- By construction, any $\mathcal{C}$ in $\mathcal{F}_{n,k}$ contains a GRS code of dimension r. It follows from the definition that $r \geq (k-\sqrt{n})\frac{1}{1+(k-\sqrt{n})/(n+1)} \geq (k-\sqrt{n})\frac{1}{1+R}$, so this gives a GRS code containing the dual code $\mathcal{C}^\top$ with

$$\delta \leq k - (k-\sqrt{n})\tfrac{1}{1+R} = \tfrac{k^2/n+\sqrt{n}}{1+R} =: \delta_{\max},$$

  where the upper bound $\delta_{\max}$ grows in n for any $k > 2\sqrt{n}+6$.

There is no obvious reason why $\mathcal{C}$ should contain a GRS code of dimension larger than r, so it seems likely that in most cases we would have $\delta = \delta_{\max}$. However, a proof of such a statement is an open problem. Assuming that it is true, then Wieschebrink's attack on $\mathcal{C}$ or $\mathcal{C}^\perp$ has work factor $\approx q^{2\delta}$ if

$$k \in [r+\delta, \tfrac{n+1}{2} - \delta] \approx [\tfrac{n}{\ell+1} - \delta, \tfrac{n}{2} - \delta].$$

Wieschebrink's attack can be refined if there are two codewords that have more than $k-1$ zero entries in common, cf. [Wie06, Section 4]. Since both $\mathcal{C}$ and $\mathcal{C}^\top$ are MDS, any two codewords differ in at least $n-k+1$ positions, so the refinement cannot be applied.

Couvreur et al.'s Attack Based on Schur Square Distinguishing

Couvreur et al. [CGGU$^+$14] proposed structural attacks on GRS-based McEliece systems and variants thereof, which are based on the fact that a low-rate GRS code, as well as any shortening of it at up to two positions, has an abnormally small Schur square dimension. In [CGGU$^+$14, Section 6], they present a polynomial-time attack based on finding codewords that result from evaluation polynomials that have certain evaluation points as zeros with high multiplicity. Such codewords are found by fixing two evaluation points, w.l.o.g., α_1 and α_2, and computing the codes $\mathcal{C}^{(i,j)}$ with $i,j > 0$ and $i+j < k$ which correspond to evaluation polynomials that vanish in α_1 with multiplicity at least i and in α_2 with multiplicity at least j. In [CGGU$^+$14, Section 6.2], they give a recursive formula that allows to compute all $\mathcal{C}^{(i,j)}$ from the four codes

$$\mathcal{C} = \mathcal{C}^{(0,0)}, \quad \mathcal{C}^{(0,1)}, \quad \mathcal{C}^{(1,0)}, \quad \text{and } \mathcal{C}^{(1,1)},$$

given that the Schur square dimensions of $\mathcal{C}^{(0,1)}$, $\mathcal{C}^{(1,0)}$, and $\mathcal{C}^{(1,1)}$ are not maximal.

Since $\mathcal{C} = \mathcal{C}^{(0,0)}$, and $\mathcal{C}^{(0,1)}$, $\mathcal{C}^{(1,0)}$, and $\mathcal{C}^{(1,1)}$ are simply shortenings of $\mathcal{C}$ at positions 1, 2, and $\{1,2\}$, respectively, all these codes have maximal Schur square dimension by construction. Since the shortenings are also MDS codes of low rate, their dual codes have maximal Schur square dimensions as well, which implies the following statement.

Theorem 4.56. *Let $\mathcal{C} \in \mathcal{F}_{n,k}$. Then, $\mathcal{C}^{(0,0)}$, $\mathcal{C}^{(0,1)}$, $\mathcal{C}^{(1,0)}$, and $\mathcal{C}^{(1,1)}$, as well as their duals, have maximal Schur square dimension.*

Hence, the attack in [CGGU$^+$14, Section 6] does not work for the McEliece cryptosystem using $\mathcal{F}_{n,k}$ as the code family.

Wieschebrink's Improved Attack and Possible Weakness

In [Wie10], Wieschebrink proposed an improved attack on the McEliece cryptosystem based on random subcodes of GRS codes. The key observation is that the Schur square of a low-rate

random subcode $\mathcal{C}$ of a GRS code $\mathcal{C}_{\mathrm{RS}}$ equals the Schur square of $\mathcal{C}_{\mathrm{RS}}$ with high probability. Hence, the Sidelnikov–Shestakov attack can be applied to a code's Schur square in order to retrieve the structure of the supercode $\mathcal{C}_{\mathrm{RS}}$. If the supercode $\mathcal{C}_{\mathrm{RS}}$ does not have low rate (i.e., its Schur square equals $\mathbb{F}_q^n$), then the code $\mathcal{C}$ can be shortened at many positions until it is a subcode of a low-rate GRS code (i.e., $\mathcal{C}_{\mathrm{RS}}$ shortened at the same positions). For random subcodes, the Schur square property is then still fulfilled with high probability.

Since twisted RS codes are subcodes of GRS codes (cf. Section 4.6.3), the codes must be analyzed in this setting. An initial test on the example in Section 4.8.2 below showed that the Schur square dimensions of its shortened codes are much smaller than the dimension of the corresponding shortenings of the obvious supercode $\mathcal{C}_{\mathrm{RS}}[n, k + t_\ell]$. Indeed, it was already mentioned in [Wie10] that it is possible to constructively choose subcodes that resist the attack. However, proving that the Schur squares of shortenings of the codes in $\mathcal{F}_{n,k}$ at many positions are not GRS codes is an open problem.

Generally, the idea of shortening the code at many positions and attacking the shortened code appears to be a possible weakness of code-based cryptosystems based on twisted RS codes. However, even if such an attack is possible, it is not clear how to utilize the structure of the shortened code in order to obtain the structure of the unshortened code. Also, using a few more twists than needed and arguments as in the proof of Theorem 4.21, it seems possible to construct codes that have maximal Schur squares even when shortened heavily. These observations need to be further analyzed in future work.

4.8.2 Example Parameters Resulting in Small Key Sizes

One of the main challenges in the McEliece cryptosystem is to decrease the public key size at a given *security level*, i.e., the work factor of the most efficient known generic or structural attack, compared to existing proposals. Since the codes in $\mathcal{F}_{n,k}$ are defined over a field of size $q \geq n^{2^\ell}$, the size of the public key (in bits) exponentially increases in the number of twists ℓ. For small ℓ, this effect is compensated by the fact that for given n, k, and number of errors, generic decoding attacks are much less efficient than for small fields (e.g., $\mathbb{F}_2$). In the following, we consider example parameters that significantly decrease the key size compared to state-of-the-art parameters of the original McEliece cryptosystem.

Consider $\mathcal{F}_{n,k}$ with $[n, k] = [255, 117]$, $\ell = 1$, and $q = s_1 = s_0^2 = (2^8)^2$. The codes in $\mathcal{F}_{n,k}$ can correct $\tau_{\mathrm{unique}} = 69$ errors uniquely, $\tau_{\mathrm{list}} = 83$ using a list decoder, and up to $\tau_{\mathrm{IRS}} = 104$ using an IRS decoder, cf. Section 4.3. In all cases, the key size is

$$K_{\mathrm{sys}} = k(n-k)\log_2(q)/8192 = 31.5\,\mathrm{KiB},$$

using a systematic generator matrix as the public key.[17]

The best-known structural attack on the code family is Wieschebrink's attack, which—under the assumption in the previous subsection—has work factor 2^{336}. For a given number of errors τ, the generic information set decoding attack [Pra62, McE78], has work factor $W_{\mathrm{I}} = \binom{n}{k}/\binom{n-\tau}{k} k^3 \log_2(q)^2$, which gives

$$W_{\mathrm{I,unique}} \approx 2^{105}, \quad W_{\mathrm{I,list}} \approx 2^{126}, \quad W_{\mathrm{I,IRS}} \approx 2^{165},$$

[17] Systematic generator matrices can decrease the semantic security if not carefully used, cf. [OS09, Section 5.1].

in these particular cases.[18] Thus, the security level in case of a unique decoder is $\approx 2^{100}$, and $\approx 2^{128}$ for a list decoder. For comparison, we consider the following parameter proposals for the original McEliece system based on binary Goppa codes:

- Security level $\approx 2^{100}$: Canteaut and Sendrier [CS98] suggest an $[n, k] = [2048, 1608]$ binary Goppa code that can correct $\tau = 40$ errors uniquely. This results in a key size of

$$K = 86.4\,\text{KiB},$$

  which is larger than our example by factor 2.7.

- Security level $\approx 2^{128}$: Barbier and Barreto [BB11] use an $[n, k] = [3262, 2482]$ binary Goppa code that can correct up to $\tau = 66$ errors using a list decoder. The key size is

$$K = 236.3\,\text{KB},$$

  which is larger than our list decoding example by factor 7.5.

4.9 Concluding Remarks

In this chapter, we have introduced twisted RS codes, a new class of evaluation codes that arise from Reed–Solomon codes by adding large-degree monomials, twists, to the evaluation polynomials whose coefficients linearly depend on the lower ones. We have shown that by choosing the evaluation polynomials in a suitable way, we obtain MDS codes, which sometimes are relatively short. Our proposed decoder is efficient for a small number of twists.

We have shown that there is a subfamily of the new codes that is closed under duality. Moreover, we have studied the Schur squares of the new codes and shown in many cases their dimension is much larger than for Reed–Solomon codes. This observation was used to show the inequivalence of many twisted RS codes to RS codes. We have also given a combinatorial inequivalence argument.

We studied two subclasses of "long" MDS twisted RS codes, which achieve lengths $n \approx \frac{q}{2}$. Finally, we have analyzed the new codes for the McEliece code-based cryptosystem. More precisely, a subfamily of twisted RS codes appears to be resistant against known structural attacks on the system based on RS-like codes. This shows that the codes should be considered for this application. There are some open problems:

- Decoding of twisted RS codes with a large number of twists (i.e., $\ell \geq 3$) is not yet feasible. We need to find a decoder that avoids brute-forcing the twist coefficients.
- The statements on the dual code in Section 4.4 should be extended—if possible—to cases where the evaluation points do not form a multiplicative group.
- The inequivalence statements in Section 4.6 do not hold for all twisted RS codes. Indeed, the computer searches in Section 4.7.3 indicate that there are twisted RS codes that are GRS codes. The question which twisted RS codes are GRS should be studied further.

[18] Peters [Pet10] proposed an improvement of the information-set decoding algorithm for non-binary fields, which decreases the expected number of required iterations. In contrast to the original algorithm, the cost of one iteration strongly depends on the field size, which is large in our case: $q = 2^{16}$. Hence, the algorithm appears to be slower than the original algorithm. However, this should be more carefully analyzed, e.g., by estimating the number of iterations using a Markov chain simulation as in [Pet10].

- Our computer searches show that there are more twisted RS codes of length $\approx \frac{q}{2}$ than the two subclasses that we studied in Section 4.7. It is an open problem to analytically describe these classes.
- Finding upper bounds on the dimension of the largest GRS code contained in a twisted RS code is not yet satisfactorily solved. Stronger bounds would directly yield a guarantee for the infeasability of Wieschebrink's attack on the dual code in Section 4.8.1.
- Experiments with small parameters indicate that if we do not restrict the evaluation points (as, e.g., in Section 4.2), we often obtain codes that are close to MDS (i.e., their minimum distance is close to $n - k + 1$), independent of the field size q or the number of twists ℓ. This observation should be analytically studied.
- In our analysis of subfamilies of twisted RS codes in Section 4.8, we considered known structural attacks. Certainly, this gives no guarantee that the codes are suitable for cryptographic applications. Hence, further attacks that are tailored to the new code class must be found and analyzed in order to get a better idea of their suitability in this scenario. A first idea would be to attack heavy shortenings of the codes, as discussed in Section 4.8.1. Furthermore, the resistance against Wieschebrink's improved attack needs to be analyzed analytically.

Part II

Codes in Rank Metric

5

Fast Decoding of Gabidulin Codes

Gabidulin codes can be decoded up to half the minimum distance in polynomial time based on similar decoders for Reed–Solomon codes. The first known algorithms are based on solving a key equation to find the so-called *error span polynomial*—the minimal subspace polynomial of the space spanned by the columns of the error matrix. A solution of this key equation is found by a skew polynomial equivalent of the extended Euclidean algorithm [Gab85] in $O(n^2)$, solving a system of linear equations [Rot91, Gab91] in $O(n^3)$, or a skew polynomial equivalent of the Berlekamp–Massey algorithm [PT91, RP04a, RP04b, HS10, SRB11, SB14] in $O(n^2)$ or $O^{\sim}(n^{1.69})$ operations over $\mathbb{F}_{q^m}$, where n is the code length. The methods were further accelerated with respect to the hidden constant in the asymptotic complexity expression in [GY08, SK09].

All of these decoding algorithms share a heavy second step: Computing a basis of the root space of the error span polynomial, resulting in a complexity of $O(n^2m)$ over $\mathbb{F}_q$, which asymptotically costs—up to logarithmic factors—as much as $O(n^2)$ operations over $\mathbb{F}_{q^m}$ using the bases in [CL09]. There is also an interpolation-based algorithm by Loidreau [Loi06] of quadratic complexity $O(n^2)$, which directly returns the message polynomial.

Recently, Wachter-Zeh, Afanassiev, and Sidorenko [WAS13] (see also [Wac13, Section 3.2.2]) introduced a key-equation-based algorithm that directly outputs the message polynomial without solving a linear system of equations, similar to Gao's algorithm for decoding Reed–Solomon codes [Gao03]. Its complexity is fully determined by the operations division, finding minimal subspace polynomials, and interpolation over the skew polynomial ring $\mathbb{F}_{q^m}[x;\sigma]$. The fastest known algorithms for these operations have quadratic complexity in the degrees of the involved polynomials (cf. Section 5.2).

In Section 5.1 of this chapter, we first recall and slightly generalize the key-equation-based decoding algorithm by Wachter-Zeh et al. and outline which skew-polynomial operations it uses. We then show in Section 5.2 that all of these operations can be implemented in $O^{\sim}(n^{\max\left\{\log_2(3),\min\{\frac{\omega+1}{2},1.635\}\right\}})$ operations over $\mathbb{F}_{q^m}$. In this way, we obtain the first subquadratic-time decoding algorithm for Gabidulin codes.

The results of this chapter were partly published in [MPMB16], [PW16], and [PW18].

5.1 Decoding Gabidulin Codes up to Half the Minimum Distance

In this section, we recall the decoding algorithm by Wachter-Zeh, Afanassiev, and Sidorenko [WAS13] (see also [Wac13, Section 3.2.2]) and phrase it in skew polynomial language. This slight reformulation not only fits the notation of this dissertation, but also admits a direct

generalization of the algorithm to Gabidulin codes over fields of characteristic zero (cf. Section 5.3). Let the received word be of the form

$$\boldsymbol{r} = \boldsymbol{c} + \boldsymbol{e} \in \mathbb{F}_{q^m}^n,$$

where $\boldsymbol{c} = [f(\alpha_1), \dots, f(\alpha_n)] \in \mathcal{C}_{\mathrm{G}}[n,k]$ and $\boldsymbol{e}$ is an error word of rank weight $\mathrm{wt}_{\mathrm{R}}(\boldsymbol{e})$. The goal of the algorithm is to recover the message polynomial f from the received word $\boldsymbol{r}$.

It is obtained by solving a key equation (see Lemma 5.1 below), which also includes the unknown *error span polynomial*

$$\Lambda = \mathcal{M}_{\langle e_1,\dots,e_n\rangle} \in \mathbb{F}_{q^m}[x;\sigma].$$

By definition, the error span polynomial satisfies $\Lambda(u) = 0$ for all $u \in \langle e_1, \dots, e_n\rangle$ and its degree equals the rank weight of the error $\deg \Lambda = \mathrm{wt}_{\mathrm{R}}(\boldsymbol{e})$. The key equation consists of two further polynomials R and G that can be directly obtained from the received word and evaluation points. The polynomial R is the interpolation polynomial

$$R = \mathcal{I}_{\{(\alpha_i,r_i)\}_{i=1}^n} \in \mathbb{F}_{q^m}[x;\sigma].$$

Recall that R has degree $\deg R < n$ and satisfies $R(\alpha_i) = r_i$ for all $i = 1, \dots, n$. Also, we can compute the minimal subspace polynomial

$$G = \mathcal{M}_{\langle \alpha_1,\dots,\alpha_n\rangle} \in \mathbb{F}_{q^m}[x;\sigma],$$

which satisfies $G(u) = 0$ for all $u \in \langle \alpha_1, \dots, \alpha_n\rangle$ and has degree $\deg G = n$. Using these polynomials, we can prove the following key equation.[1]

Lemma 5.1 ($\mathbb{F}_{q^m}[x;\sigma]$-analog of the key equation in [WAS13, Wac13])**.** *The polynomials Λ, R, f, and G, defined as above, fulfill*

$$\Lambda R \equiv \Lambda f \mod_{\mathrm{r}} G.$$

Proof. Due to the properties of the evaluation map and the involved polynomials, we have

$$\big[\Lambda \cdot (R - f)\big](\alpha_i) = \Lambda(R(\alpha_i) - f(\alpha_i)) = \Lambda(r_i - f_i) = \Lambda(e_i) = 0.$$

Hence, G must right-divide $\Lambda(R - f)$ since otherwise the remainder of the right division of $\Lambda(R - f)$ by G would be a polynomial with n-dimensional root space $\langle \alpha_1, \dots, \alpha_n\rangle$ of degree smaller than n, contradicting the properties of skew polynomials. □

As most key equations, Lemma 5.1 provides a non-linear relation in the indeterminates Λ and f. Hence, we solve the following linearized problem, whose solution equals the one of the key equation if the rank weight of the error is less than half the minimum distance (cf. Lemma 5.3 below). In Section 6.1.1, we will study a generalization of Problem 5.2.

Problem 5.2. *Let k, R and G be given as above. Find $[\lambda, \omega] \in (\mathbb{F}_{q^m}[x;\sigma]^*)^2$ such that*

$$\lambda R \equiv \omega \mod_{\mathrm{r}} G, \tag{5.1}$$

$$\deg \lambda > \deg \omega + k, \tag{5.2}$$

$$\deg \lambda \textit{ minimal}. \tag{5.3}$$

[1] It can be seen as the rank-metric analog of Theorem 3.6 on page 29 in the special case $h = \ell = s = 1$.

Lemma 5.3. *If* $\mathrm{wt_R}(\boldsymbol{e}) < \frac{d}{2}$*, then Problem 5.2 has a solution. Furthermore, any solution* $[\lambda, \omega]$ *of the problem satisfies*

$$[\Lambda, \Lambda f] = \alpha[\lambda, \omega],$$

where $\alpha \in \mathbb{F}_{q^m}^*$*, and* f *and* Λ *are defined as above.*

Proof. Problem 5.2 has a solution since $[\Lambda, \Lambda f]$ satisfies (5.1) by Lemma 5.1, and (5.2) by $\deg \Lambda f < \deg \Lambda + k$. Due to (5.3), any solution $[\lambda, f]$ must fulfill $\deg \lambda \leq \deg \Lambda =: \tau$. By the properties of the MSP Λ, we know that $\dim\langle e_1, \dots, e_n\rangle = \tau$ and therefore

$$\deg \mathcal{M}_{\langle \lambda(e_1), \dots, \lambda(e_n)\rangle} = \dim\langle \lambda(e_1), \dots, \lambda(e_n)\rangle \leq \tau.$$

Similar to [Loi06], we show that $\mathcal{M}_{\langle \lambda(e_1), \dots, \lambda(e_n)\rangle}(\omega - \lambda f)$ is the all-zero polynomial. The degree bounds above imply that the polynomial has degree $< k + 2\tau$, which is at most n for $\tau = \mathrm{wt_R}(\boldsymbol{e}) \leq \frac{d-1}{2} = \frac{n-k}{2}$. Due to (5.1), we obtain

$$\begin{aligned}\big[\mathcal{M}_{\langle \lambda(e_1), \dots, \lambda(e_n)\rangle}(\omega - \lambda f)\big](\alpha_i) &= \mathcal{M}_{\langle \lambda(e_1), \dots, \lambda(e_n)\rangle}(\lambda(R(\alpha_i)) - \lambda(f(\alpha_i))) \\ &= \mathcal{M}_{\langle \lambda(e_1), \dots, \lambda(e_n)\rangle}(\lambda(e_i)) = 0.\end{aligned}$$

Hence, the polynomial evaluates to zero at n linearly independent points, but has degree less than n, and therefore is the all-zero polynomial. Since $\mathbb{F}_{q^m}[x;\sigma]$ is an integral domain, we obtain $\omega = \lambda f$. Together with (5.1), we get $\lambda(e_i) = 0$ for all $i = 1, \dots, n$. Since $\deg \lambda \leq \deg \Lambda$, the polynomial λ must be a scalar multiple of the MSP Λ, which proves the claim. □

If $[\Lambda, \Lambda f] = \alpha[\lambda, \omega]$, we obtain the message polynomial by right-dividing ω by λ. Thus, Lemma 5.3 implies a half-the-minimum distance decoder for Gabidulin codes, which is outlined in Algorithm 4. The algorithm's complexity depends on the cost of several skew polynomial operations. In the next section, we will provide new cost bounds on these operations, proving that Algorithm 4 can be implemented in sub-quadratic time. This result is formally stated in the following theorem.

Theorem 5.4. *Let* $\boldsymbol{r} = \boldsymbol{c} + \boldsymbol{e}$ *be the received word and* $\boldsymbol{c} = [f(\alpha_1), \dots, f(\alpha_n)]$ *for* $\deg f < k$*. If* $\mathrm{wt_R}(\boldsymbol{e}) < \frac{d}{2}$*, then Algorithm 4 finds the correct message polynomial* $f \in \mathbb{F}_{q^m}[x;\sigma]$ *in*

$$O\Big(n^{\max\left\{\log_2(3), \min\left\{\frac{\omega+1}{2}, 1.635\right\}\right\}} \log(n)\Big)$$

operations over $\mathbb{F}_{q^m}$ *using the algorithms for fast skew polynomial arithmetic in Section 5.2 and Algorithm 13 in Section 6.1.1 for solving Problem 5.2.*[2]

Proof. Correctness directly follows from Lemma 5.3. The core step of the algorithm is to solve Problem 5.2. We will thoroughly study a generalization of this problem in Chapter 6. There, we show that Problem 5.2 can be solved by Algorithm 13 on page 114 in $O(\mathcal{M}_{q^m}(n) \log(n))$ operations over $\mathbb{F}_{q^m}$, where $\mathcal{M}_{q^m}(n) \in O(n^{\min\{\frac{\omega+1}{2}, 1.635\}})$ is the cost of multiplying two skew polynomials of degree at most n (cf. Algorithm 5 in Section 5.2.1).

[2] Alternatively, Problem 5.2 can be solved in $O(n^{\min\{\frac{\omega+1}{2}, 1.635\}} \log^2(n))$ by a skew-variant of the linearized extended Euclidean algorithm in [WSB10] (see also [Wac13, Section 3.1.4]) using the division algorithm in Section 5.2.2.

Algorithm 4: Half-the-Minimum-Distance Decoder for Gabidulin Codes ($\mathbb{F}_{q^m}[x;\sigma]$-analog of [WAS13, Algorithm 4], see also [Wac13, Algorithm 3.6])

Input: Received word $\boldsymbol{r} \in \mathbb{F}_{q^m}^n$

Output: $f \in \mathbb{F}_{q^m}[x;\sigma]_{<k}$ such that $\boldsymbol{c} = [f(g_1), \dots, f(g_n)]$ with $\mathrm{wt_R}(\boldsymbol{r} - \boldsymbol{c}) < \frac{d}{2}$ or "decoding failure".

1 $R \leftarrow \mathcal{I}_{\{[\alpha_i, r_i]\}_{i=1}^n}$ // $O(n^{\max\{\log_2(3), \min\{\frac{\omega+1}{2}, 1.635\}\}} \log(n))$

2 $G \leftarrow \mathcal{M}_{\langle \alpha_1, \dots, \alpha_n \rangle}$ // $O(n^{\max\{\log_2(3), \min\{\frac{\omega+1}{2}, 1.635\}\}} \log(n))$

3 $[\lambda, \omega] \leftarrow$ Solve Problem 5.2 with input R, G by Algorithm 13 // $O(n^{\min\{\frac{\omega+1}{2}, 1.635\}} \log(n))$

4 $[\chi, \varrho] \leftarrow$ Right-divide ω by λ // $O(n^{\min\{\frac{\omega+1}{2}, 1.635\}} \log(n))$

5 **if** $\varrho = 0$ **then**

6 **return** χ

7 **else**

8 **return** *"decoding failure"*

The other steps of Algorithm 4 consist of the following skew polynomial operations. The complexity statements are derived in Section 5.2.

- The interpolation at n point tuples $[\alpha_1, r_1], \dots, [\alpha_n, r_n]$ in Line 1 can be implemented in $O(n^{\max\{\log_2(3), \min\{\frac{\omega+1}{2}, 1.635\}\}} \log(n))$ operations over $\mathbb{F}_{q^m}$ using Algorithm 10 in Section 5.2.4.
- The minimal subspace polynomial of the n-dimensional subspace $\langle \alpha_1, \dots, \alpha_n \rangle$ in Line 2 can be computed in $O(n^{\max\{\log_2(3), \min\{\frac{\omega+1}{2}, 1.635\}\}} \log(n))$ operations over $\mathbb{F}_{q^m}$ using Algorithm 8 in Section 5.2.1.
- In Line 4, the algorithm divides ω by λ. If $t := \mathrm{wt_R}(\boldsymbol{e}) < d/2$, we have $\deg \lambda = t \leq n$ and $\deg \omega < t + k \leq n$, so the division can be implemented in $O(n^{\min\{\frac{\omega+1}{2}, 1.635\}} \log(n))$ operations over $\mathbb{F}_{q^m}$ using Algorithm 6 in Section 5.2.3.

Summarized, we obtain the claimed complexity. □

Remark 5.5. *4We can achieve the same asymptotic complexity as in Theorem 5.4 for the error-erasure decoders in [SKK08] and [Wac13], which compute two MSPs, an interpolation, a few multiplications and divisions, and once solve a variant of Algorithm 13, all on polynomials of degree in the order of n.*

5.2 Fast Operations on Skew Polynomials

We have seen in the previous section that the complexity of decoding of Gabidulin codes directly depends on the cost of multiplication, division, interpolation, and computing minimal subspace polynomials in $\mathbb{F}_{q^m}[x;\sigma]$. In addition, encoding of Gabidulin codes can be accomplished by a multi-point evaluation (MPE). If the evaluation points of a Gabidulin code form a normal basis of $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$ (w.r.t. σ), then MPE and interpolation can be replaced by a σ-transform and its inverse, respectively. In this section, we therefore prove new cost bounds on the following operations.

Definition 5.6. Let $s \in \mathbb{N}$. We define the following formal complexity measures in operations over $\mathbb{F}_{q^m}$, i.e., the infimum of the worst-case complexities of algorithms that solve the given operation.

Measure	Operation
$\mathcal{M}_{q^m}(s)$	Multiplication of two polynomials $a, b \in \mathbb{F}_{q^m}[x;\sigma]_{\leq s}$
$\mathcal{D}_{q^m}(s)$	Left or right division of $a \in \mathbb{F}_{q^m}[x;\sigma]_{\leq s}$ by $b \in \mathbb{F}_{q^m}[x;\sigma]_{\leq s}$
$\mathcal{MSP}_{q^m}(s)$	Computation of the MSP $\mathcal{M}_{\langle U\rangle}$ for a generating set $U = \{u_1, \ldots, u_s\}$ of a subspace of $\mathbb{F}_{q^m}$
$\mathcal{MPE}_{q^m}(s)$	Multi-point evaluation of $a \in \mathbb{F}_{q^m}[x;\sigma]_{\leq s}$ at s points $\alpha_1, \ldots, \alpha_s \in \mathbb{F}_{q^m}$
$\mathcal{I}_{q^m}(s)$	Finding the interpolation polynomial of s point tuples.
$\mathcal{ST}_{q^m}(s)$	Computing the σ-transform or its inverse of a polynomial $a \in \mathbb{F}_{q^m}[x;\sigma]_{<s}$, with respect to a normal basis $\mathcal{B}_{\mathrm{N}} = \{\sigma^0(\beta), \ldots, \sigma^{s-1}(\beta)\}$ of $\mathbb{F}_{q^s} \subseteq \mathbb{F}_{q^m}$.

We briefly summarize previous best known cost bounds on these operations and outline our contribution in the following. Table 5.1 provides an overview of all bounds.

Previous Work

Silva et al. [SK09] and Wachter-Zeh et al. [WAS13] proposed fast algorithms of several operations in the case $\sigma = \cdot^q$, $s \leq m$, and given the existence of a low-complexity normal basis of $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$. This includes algorithms for the σ-transform w.r.t. a basis of $\mathbb{F}_{q^s} \subseteq \mathbb{F}_{q^m}$ ($O(ms^2)$ over $\mathbb{F}_q$), multi-point evaluation ($O(m^2 s)$ over $\mathbb{F}_q$), and multiplication of skew polynomials modulo $x^m - 1$ ($O(m^2 s)$ over $\mathbb{F}_q$). The runtime of the latter two methods is determined by a σ-transform w.r.t. a basis of $\mathbb{F}_{q^m}$ with complexity $O(m^2 s)$.

For $\sigma = \cdot^q$ and arbitrary $s > 0$, [Wac13, Section 3.1.2] showed that $\mathcal{M}_{q^m}(s) \in O(s^{\frac{\omega+1}{2}})$. It is also known that $\mathcal{MSP}_{q^m}(s)$ [LSS14], $\mathcal{MPE}_{q^m}(s)$ [Wac13], and $\mathcal{ST}_{q^m}(s)$ [SK09] are in $O(s^2)$. Interpolation can be implemented in $\mathcal{I}_{q^m}(s) \in O(s^3)$, cf. [SK07]

In the general case, [CL17a] proposed fast algorithms for multiplication in $O(ms)$ and division in $O(ms\log(s))$ operations over $\mathbb{F}_{q^m}$. If $m \in o(s)$, then these algorithms are sub-quadratic in s. For the case $m \in \Omega(s)$, which is relevant for decoding Gabidulin codes, it has been an open problem whether division algorithms of sub-quadratic complexity exist.

Operations over $\mathbb{F}_{q^m}$ cost $O^{\sim}(m)$ operations over $\mathbb{F}_q$ using the bases in [CL09]. A quadratic complexity $O(s^2)$ over $\mathbb{F}_{q^m}$ therefore costs $O^{\sim}(ms^2)$ over $\mathbb{F}_q$, and the algorithms for fast operations that directly operate on $\mathbb{F}_q$ in [SK09, WAS13] are—asymptotically—not faster than the known ones over $\mathbb{F}_{q^m}$. Thus, we compare our results to the cost bounds over $\mathbb{F}_{q^m}$.

Our Contribution

In Section 5.2.1, we generalize the multiplication algorithm in [Wac13] from linearized to skew polynomials and slightly improve its complexity using fast rectangular matrix multiplication algorithms, resulting in $\mathcal{M}_{q^m}(s) \in O(s^{\min\{\frac{\omega+1}{2}, 1.635\}})$. The generalization enables us to apply the reduction of division in $\mathbb{F}_{q^m}[x;\sigma]$ to multiplication in both $\mathbb{F}_{q^m}[x;\sigma]$ and $\mathbb{F}_{q^m}[x;\sigma^{-1}]$, which was presented in [CL17a], in order to prove $\mathcal{D}_{q^m}(s) \in O^{\sim}(s^{\min\{\frac{\omega+1}{2}, 1.635\}})$, cf. Section 5.2.2.

Our fast methods for MPE and computing MSPs in Section 5.2.3 use ideas of well-known divide-&-conquer algorithms for MPE and so-called sub-product-tree computation over $\mathbb{F}_{q^m}[x]$, where, compared to the latter case, the algorithms need to call each other recursively. The divide-&-conquer interpolation algorithm in Section 5.2.4 is based on the new MPE and MSP algorithms. All three algorithms cost $O^{\sim}(s^{\max\{\log_2(3),\min\{\frac{\omega+1}{2},1.635\}\}})$ operations over $\mathbb{F}_{q^m}$.

In Section 5.2.5, we speed up the σ-transform and its inverse using fast algorithms for Toeplitz-matrix-vector multiplication and solving a system of equations given by a Toeplitz matrix, respectively. The resulting cost bound is quasi-linear in s, i.e., $\mathcal{ST}_{q^m}(s) \in O^{\sim}(s)$.

Table 5.1: Previous and new cost bounds in operations over $\mathbb{F}_{q^m}$ for the complexity measures of Definition 5.6. The new cost bounds are proved in Sections 5.2.1-5.2.5.

Operation	Previous cost bound (ref.: see text)	New cost bound
$\mathcal{M}_{q^m}(s)$	$\begin{cases} O\left(\min\left\{ms, s^{\frac{\omega+1}{2}}\right\}\right), & \text{if } \sigma = \cdot^q, \\ O(ms), & \text{else.} \end{cases}$	$O\left(\min\left\{ms, s^{\min\{\frac{\omega+1}{2},1.635\}}\right\}\right)$
$\mathcal{D}_{q^m}(s)$	$O(ms\log(s))$	$O\left(\min\left\{ms, s^{\min\{\frac{\omega+1}{2},1.635\}}\right\}\log(s)\right)$
$\mathcal{MSP}_{q^m}(s)$	$O(s^2)$	$O\left(s^{\max\left\{\log_2(3),\min\{\frac{\omega+1}{2},1.635\}\right\}}\log(s)\right)$
$\mathcal{MPE}_{q^m}(s)$	$O(s^2)$	$O\left(s^{\max\left\{\log_2(3),\min\{\frac{\omega+1}{2},1.635\}\right\}}\log(s)\right)$
$\mathcal{I}_{q^m}(s)$	$O(s^3)$	$O\left(s^{\max\left\{\log_2(3),\min\{\frac{\omega+1}{2},1.635\}\right\}}\log(s)\right)$
$\mathcal{ST}_{q^m}(s)$	$O(s^2)$	$O(s\log(s)\log(\log(s)))$

5.2.1 Multiplication

In this section, we provide a sub-quadratic cost bound on $\mathcal{M}_{q^m}(s)$ by generalizing the fast multiplication algorithm for linearized polynomials from [Wac13, Theorem 3.1] to skew polynomial rings. The generalization follows without complications by reformulating the statements in skew polynomial language. However, it is necessary to obtain other fast algorithms for skew or even linearized polynomials.[3] In addition, we achieve a speed-up compared to [Wac13] for the best-known ω using specialized rectangular matrix multiplication algorithms.

The idea of the algorithm is to multiply small fragments of the involved polynomials and then put together the constituent results to a whole. This approach was first presented for calculating compositions of power series in [BK78, PS73] and is related to the baby-steps giant-steps method.

We say that polynomials $a, b \in \mathbb{F}_{q^m}[x;\sigma]$ *overlap* at k positions if their supports, i.e., $\operatorname{supp}(a) := \{i : a_i \neq 0\}$ and $\operatorname{supp}(b) := \{i : b_i \neq 0\}$, intersect in at most k elements:

$$|\operatorname{supp}(a) \cap \operatorname{supp}(b)| \leq k.$$

If $k = 0$, we say that a and b are *non-overlapping*. If the overlapping positions are known, the sum of two polynomials overlapping at k positions can be calculated with k additions in $\mathbb{F}_{q^m}$. We use this notation to prove the following result.

[3] For instance, the fast division algorithm in [CL17a] does not yield a fast linearized polynomial ($\sigma = \cdot^q$) division algorithm since it reduces division in $\mathbb{F}_{q^m}[x;\cdot^q]$ to multiplication in both $\mathbb{F}_{q^m}[x;\cdot^q]$ and $\mathbb{F}_{q^m}[x;\cdot^{1/q}]$.

Theorem 5.7 ($\mathbb{F}_{q^m}[x;\sigma]$-analog of [Wac13, Lemma 3.1/Theorem 3.1])**.** *Let $a, b \in \mathbb{F}_{q^m}[x;\sigma]_{\leq s}$ and $s^* := \lceil \sqrt{s+1} \rceil$. Then, Algorithm 5 computes the multiplication $c = a \cdot b$ in $O(s^{3/2})$ field operations, plus the cost of multiplying an $s^* \times s^*$ with an $s^* \times (s + s^*)$ matrix over $\mathbb{F}_{q^m}$.*

Algorithm 5: Multiplication ($\mathbb{F}_{q^m}[x;\sigma]$-analog of [Wac13, Algorithm 3.1])

Input: $a, b \in \mathbb{F}_{q^m}[x;\sigma]_{\leq s}$
Output: $c = a \cdot b$

1 Set up matrices A and B as in (5.4) // $s^{3/2} \cdot O(1)$
2 $C \leftarrow A \cdot B$ // $s^* \cdot O((s^*)^\omega)$ or $(s^*)^{3.2699}$
3 Extract the $c^{(i)}$'s from C as in (5.4) // $s^{3/2} \cdot O(1)$
4 **return** $c \leftarrow \sum_{i=0}^{s^*-1} c^{(i)}$ // $O\left(s^{3/2}\right)$

Proof. The idea of Algorithm 5 is to fragment a into s^* non-overlapping polynomials $a^{(i)}$ by

$$a = \sum_{i=0}^{s^*-1} a^{(i)} = \sum_{i=0}^{s^*-1} \left(\sum_{j=0}^{s^*-1} a_{is^*+j} x^{is^*+j} \right).$$

Similarly, the result c of the multiplication $c = a \cdot b$ can be written as

$$c = a \cdot b = \left(\sum_{i=0}^{s^*-1} a^{(i)} \right) \cdot b = \sum_{i=0}^{s^*-1} \underbrace{(a^{(i)} \cdot b)}_{=: c^{(i)}}.$$

Similar to [Wac13, Lemma 3.1], we can rewrite the polynomials $c^{(i)}$ as follows.

$$\begin{aligned} c^{(i)} &= \left(\sum_{j=0}^{s^*-1} a_{is^*+j} x^{is^*+j} \right) \cdot \left(\sum_{k=0}^{s} b_k x^k \right) = \sum_{j=0}^{s^*-1} \sum_{k=0}^{s} \left(a_{is^*+j} x^{is^*+j} \cdot b_k x^k \right) \\ &= \sum_{j=0}^{s^*-1} \sum_{k=0}^{s} \left(a_{is^*+j} \sigma^{is^*+j}(b_k) x^{is^*+j+k} \right) = \sum_{h=0}^{s+s^*-1} \underbrace{\left(\sum_{j=0}^{h} a_{is^*+j} \sigma^{is^*+j}(b_{h-j}) \right)}_{=: c_h^{(i)}} x^{is^*+h} \end{aligned}$$

Thus, the $c^{(i)}$'s pairwise overlap at not more than s positions, which we know. Since σ^{-is^*} is an automorphism, we obtain the coefficients of the polynomials $c^{(i)}$ by a vector multiplication

$$\begin{aligned} \sigma^{-is^*}(c_h^{(i)}) &= \sum_{j=0}^{h} \sigma^{-is^*}(a_{is^*+j}) \sigma^{-is^*+is^*+j}(b_{h-j}) = \sum_{j=0}^{h} \sigma^{-is^*}(a_{is^*+j}) \sigma^j(b_{h-j}) \\ &= \left[\sigma^{-is^*}(a_{is^*}), \ldots, \sigma^{-is^*}(a_{is^*+s^*-1}) \right] \cdot \left[\sigma^0(b_h), \sigma^1(b_{h-1}), \ldots, \sigma^h(b_0), 0, \ldots, 0 \right]^\top. \end{aligned}$$

Since the left and right vector are independent of h and i, respectively, we can express the

coefficients $\sigma^{-is^*}(c_h^{(i)})$ by a matrix multiplication $C = A \cdot B$ with

$$\begin{aligned} C &= \left[C_{ij}\right]_{i=0,\dots,s^*-1}^{j=0,\dots,s+s^*-1}, \; C_{ij} = \sigma^{-is^*}(c_j^{(i)}), \\ A &= \left[A_{ij}\right]_{i=0,\dots,s^*-1}^{j=0,\dots,s^*-1}, \quad A_{ij} = \sigma^{-is^*}(a_{is^*+j}), \\ B &= \left[B_{ij}\right]_{i=0,\dots,s^*-1}^{j=0,\dots,s+s^*-1}, \; B_{ij} = \begin{cases} \sigma^j(b_{i-j}), & 0 \le i-j \le s, \\ 0, & \text{else.} \end{cases} \end{aligned} \tag{5.4}$$

We analyze the complexity. The matrices A and B can be set up using $s^* \cdot s + s^* \cdot s^* \approx s^{3/2}$ many computations of automorphisms to $\mathbb{F}_{q^m}$ elements. We can compute the matrix C by a multiplying an $s^* \times s^*$ with an $s^* \times (s + s^*)$ matrix. The coefficients of the polynomials $c^{(i)}$ are obtained from C in $\approx s^{3/2}$ automorphism computations. Finally, c is computed by adding the polynomials $c^{(i)}$. For any $k < s^*$, the polynomials $\sum_{i=0}^{k-1} c^{(i)}$ and $c^{(k)}$ overlap at s known positions, so the sum of all $c^{(i)}$'s can be computed in $O(s^* \cdot s) = O(s^{3/2})$ time. Since any $\mathbb{F}_{q^m}$-automorphism can be computed in $O(1)$, the lines of Algorithm 5 except for the matrix multiplication have overall complexity $O(s^{3/2})$. □

Corollary 5.8. *The multiplication of the $s^* \times s^*$ with the $s^* \times (s+s^*)$ matrix in Algorithm 5 can be implemented in different ways, resulting in the following cost bounds on skew polynomial multiplication.*

i) As described in [Wac13], the multiplication can be carried out by $s^* + 1$ many multiplications of square $s^* \times s^*$ matrices, resulting in

$$\mathcal{M}_{q^m}(s) \in O(s^* \cdot (s^*)^\omega) \subseteq O\left(s^{\frac{\omega+1}{2}}\right).$$

By plugging in different values of the matrix multiplication exponent ω, we obtain

$$\mathcal{M}_{q^m}(s) \in \begin{cases} O\left(s^{1.91}\right), & \omega \approx 2.8074 \text{ [Str69]}, \\ O\left(s^{1.69}\right), & \omega \approx 2.376 \text{ [CW90]}. \end{cases}$$

ii) Alternatively, we can use tailored multiplication algorithms for rectangular matrices, such as the ones in [HP98] and [KZHP08], resulting in

$$\mathcal{M}_{q^m}(s) \in O\left((s^*)^{3.2699}\right) \subseteq O\left(s^{1.635}\right),$$

where 3.2699 [KZHP08, Example 1] is the exponent of multiplying a square matrix with a matrix whose number of columns is approximately the square of its row count.

Corollary 5.8 provides an asymptotic statement. The following remark assesses the practicality of Algorithm 5 for finite input size s.

Remark 5.9. *Using its defining formula, i.e.,* (2.3)*, skew polynomial multiplication can be performed in approximately $2s^2$ many field operations. If we instead consider case* (i) *of Corollary 5.8 with naive matrix multiplication (each costing approximately $2(s^*)^3$ operations), skew polynomial multiplication takes $\approx s^* \cdot (2(s^*)^3) = 2(s^*)^4 = 2s^2$ operations in total. Algorithm 5 with case* (i) *of Corollary 5.8 is therefore faster than a naive implementation if and*

only if the algorithm for multiplying two matrices of dimension $s^* \times s^*$ is faster than $2(s^*)^3$. Strassen's algorithm [Str69] costs $\approx 4.7(s^*)^{\log_2(7)}$ field operations, which is smaller than $2(s^*)^3$ for $s^* \geq 85$. Hence, the resulting algorithm is only faster than the naive implementation for an input size of $s \geq 7225$. The Coppersmith–Winograd algorithm [CW90] has a much larger "hidden constant" and improves upon the naive case only for much larger values of s.

The rectangular matrix multiplication algorithms in [HP98] and [KZHP08] considered in case (ii) *of Corollary 5.8 might improve upon the naive implementation for smaller values of s than case* (i) *with Strassen's algorithm. However, this must be analyzed thoroughly since [HP98] and [KZHP08] only provide an asymptotic analysis.*

On the other hand, there are efficiently implemented linear algebra libraries that provide fast matrix multiplication algorithms for small input sizes, optimized for the used programming language or hardware. With the help of these libraries, Algorithm 5 can be faster than a naive implementation for smaller values of s than given above.

5.2.2 Division

In this section, we show that division of skew polynomials of degree at most s can be implemented in sub-quadratic time for arbitrary s. So far, the best-known cost bound on division is $O^{\sim}(sm)$ [CL17a], which is quasi-linear for $s \gg m$, but quadratic if s is in the order of m—the case relevant for decoding Gabidulin codes. Our result improves upon this bound in the latter case.

We obtain the new cost bound by relying on a reduction of division in $\mathbb{F}_{q^m}[x;\sigma]$ to multiplication in another skew polynomial ring $\mathbb{F}_{q^m}[x;\sigma^{-1}]$, which was presented Caruso and Le Borgne in [CL17a]. Together with the fast multiplication algorithm in the previous subsection, the cost bound immediately follows. For the sake of completeness, we briefly recall Caruso and Le Borgne's reduction. As in [CL17a], we only consider right division since the other case follows analogously.

Lemma 5.10 ([CL17a])**.** *Right division of two polynomials in $\mathbb{F}_{q^m}[x;\sigma]_{\leq s}$ can be implemented in $O(\log(s))$ multiplications in $\mathbb{F}_{q^m}[x;\sigma^{-1}]_{\leq s}$ and two multiplications in $\mathbb{F}_{q^m}[x;\sigma]_{\leq s}$ using Algorithm 6.*

Proof idea. The idea of Algorithm 6 is that the quotient $\chi \in \mathbb{F}_{q^m}[x;\sigma]$ of the right division of a skew polynomial $a \in \mathbb{F}_{q^m}[x;\sigma]$ by $b \in \mathbb{F}_{q^m}[x;\sigma]$ satisfies

$$\tau_s(a) \equiv \tau_{s-\ell}(\chi) \cdot \tau_\ell(b^{(s-\ell)}) \mod_{\mathrm{r}} x^{s-\ell+1},$$

where $s = \deg a \geq \deg b = \ell$ and

$$\tau_s : \mathbb{F}_{q^m}[x;\sigma]_{\leq s} \to \mathbb{F}_{q^m}[x;\sigma^{-1}]_{\leq s}, \quad \textstyle\sum_{i=0}^{s} a_i x^i \mapsto \sum_{i=0}^{s} a_{s-i} x^i \tag{5.5}$$

maps a polynomial to $\mathbb{F}_{q^m}[x;\sigma^{-1}]$ and reverts its coefficients. Hence, the polynomial $\tau_{s-\ell}(\chi)$ can be computed using the right inverse of $\tau_\ell(b^{(s-\ell)})$ modulo $x^{s-\ell+1}$, which is obtained in $O(\log(s))$ multiplications in $\mathbb{F}_{q^m}[x;\sigma^{-1}]$ using a Hensel-like lifting in Algorithm 7. Afterwards, the remainder ϱ is computed from the quotient. □

It is clear from Lemma 5.10 that we require an efficient skew polynomial multiplication method that works for any automorphism σ in order to obtain a fast division algorithm.

Our generalization of Wachter-Zeh's algorithm to arbitrary skew polynomials in Section 5.2.1 therefore provides the following new cost bound.

Algorithm 6: RightDiv(a, b) [CL17a, Algorithm 5]

Input: $a, b \in \mathbb{F}_{q^m}[x;\sigma]$, $s = \deg a \geq \deg b = \ell$
Output: $\chi, \varrho \in \mathbb{F}_{q^m}[x;\sigma]$ s.t. $a = \chi \cdot b + \varrho$ and $\deg \varrho < \ell$.

1 $c \leftarrow \tau_\ell(b^{(s-\ell)})$; $\widetilde{a} \leftarrow \tau_s(a)$, where τ_ℓ and τ_s are as in (5.5) // $O(s)$
2 $c^{-1} \leftarrow \text{RightInv}(c, s-\ell+1)$ // $O(\mathcal{M}_{q^m}(s)\log s)$
3 $\chi \leftarrow \tau_{s-\ell}^{-1}(\widetilde{a} \cdot c^{-1} \mod_{\mathrm{r}} x^{\ell-1})$ // $\mathcal{M}_{q^m}(s)$
4 $\varrho \leftarrow a - \chi \cdot b$ // $\mathcal{M}_{q^m}(s)$
5 **return** $[\chi, \varrho]$

Algorithm 7: RightInv(c, η) [CL17a, Algorithm 5]

Input: $c \in \mathbb{F}_{q^m}[x;\sigma^{-1}]$ with $c_0 \neq 0$, $\eta \in \mathbb{N}$.
Output: $d \in \mathbb{F}_{q^m}[x;\sigma^{-1}]$ s.t. $c \cdot d \equiv 1 \mod_{\mathrm{r}} x^\eta$

1 $h_0 \leftarrow 1/c_0$ // $O(1)$
2 **for** $i = 1, \ldots, \lceil \log_2(\eta) \rceil$ **do**
3 $\quad h_i \leftarrow 2h_{i-1} - h_{i-1} \cdot c \cdot h_{i-1} \mod_{\mathrm{r}} x^{2^i}$ // $2 \cdot \mathcal{M}_{q^m}(2^i)$
4 **return** $h_{\lceil \log_2(\eta) \rceil}$

Theorem 5.11. *Right and left division of two skew polynomials in $\mathbb{F}_{q^m}[x;\sigma]_{\leq s}$ costs*

$$\mathcal{D}_{q^m}(s) \in O\Big(s^{\min\{\frac{\omega+1}{2}, 1.635\}} \log s\Big)$$

operations over $\mathbb{F}_{q^m}$ using Algorithm 6 and the multiplication algorithm in Section 5.2.1.

5.2.3 Minimal Subspace Polynomials and Multi-Point Evaluation

In this section, we present an efficient algorithm for multi-point evaluation and computing minimal subspace polynomials in skew polynomial rings.

In the polynomial ring $\mathbb{F}_{q^m}[x]$, fast multi-point evaluation at a set $S \subseteq \mathbb{F}_{q^m}$ typically uses a pre-computed sub-product tree in combination a divide-&-conquer trick [GG99, Section 10.1]. Such a sub-product tree consists of polynomials $M_U = \prod_{u \in U}(x-u)$, where U is a subset $U \subseteq S$. The M_U can be efficiently computed since for any partition $U = A \cup B$, $A \cap B = \emptyset$, the polynomial M_U can be written as the product of the polynomials M_A and M_B

$$M_U = M_A \cdot M_B. \tag{5.6}$$

Over skew polynomials, we can apply a similar trick, where the analog of the sub-product tree consists of minimal subspace polynomials $\mathcal{M}_{\langle U \rangle}$ spanned by a subset U of the evaluation set S. The analog statement of (5.6) in the skew polynomial case is given in Lemma 5.12 (see below). In contrast to the $\mathbb{F}_{q^m}[x]$ case, the left factor depends on a multi-point evaluation of the right factor. Hence, the sub-product tree cannot be efficiently pre-computed without a fast multi-point evaluation and vice versa using this method.

We therefore treat the two problems and their cost bounds jointly. The following two lemmata lay the foundation to algorithms that compute MSPs and MPEs by convoluted recursive calls of each other.

Lemma 5.12 ($\mathbb{F}_{q^m}[x;\sigma]$-analog of [LSS14, Lemma 1])**.** *Let $U = \{u_1, \dots, u_s\}$ be a generating set of a subspace $\mathcal{U} \subseteq \mathbb{F}_{q^m}$, $A, B \subseteq \mathbb{F}_{q^m}$ such that $U = A \cup B$. Then,*

$$\mathcal{M}_{\mathcal{U}} = \mathcal{M}_{\langle U\rangle} = \mathcal{M}_{\langle \mathcal{M}_{\langle A\rangle}(B)\rangle} \cdot \mathcal{M}_{\langle A\rangle}.$$

If $s = 1$, then

$$\mathcal{M}_{\mathcal{U}} = \begin{cases} 1, & \text{if } u_1 = 0, \\ x - \frac{\sigma(u_1)}{u_1}, & \text{else.} \end{cases} \tag{5.7}$$

Proof. The proof works as in [LSS14], using properties of $\mathbb{F}_{q^m}[x;\sigma]$. Let $u \in \mathcal{U}$. Since the union of A and B spans $\mathcal{U}$, there are $a \in \langle A\rangle$ and $b \in \langle B\rangle$ such that $u = a + b$. We have

$$\begin{aligned}(\mathcal{M}_{\langle \mathcal{M}_{\langle A\rangle}(B)\rangle} \cdot \mathcal{M}_{\langle A\rangle})(u) &= \mathcal{M}_{\langle \mathcal{M}_{\langle A\rangle}(B)\rangle}(\mathcal{M}_{\langle A\rangle}(u)) = \mathcal{M}_{\langle \mathcal{M}_{\langle A\rangle}(B)\rangle}(\mathcal{M}_{\langle A\rangle}(a+b)) \\ &= \mathcal{M}_{\langle \mathcal{M}_{\langle A\rangle}(B)\rangle}(\mathcal{M}_{\langle A\rangle}(a) + \mathcal{M}_{\langle A\rangle}(b)) = \mathcal{M}_{\langle \mathcal{M}_{\langle A\rangle}(B)\rangle}(\mathcal{M}_{\langle A\rangle}(b)) = 0\end{aligned}$$

Hence, the polynomial $\mathcal{M}_{\langle \mathcal{M}_{\langle A\rangle}(B)\rangle} \cdot \mathcal{M}_{\langle A\rangle}$ vanishes on $\mathcal{U}$. Furthermore, it is monic and, since $\dim\langle \mathcal{M}_{\langle A\rangle}(B)\rangle = \dim\langle B\rangle - \dim(\langle A\rangle \cap \langle B\rangle)$, has degree

$$\deg\left(\mathcal{M}_{\langle \mathcal{M}_{\langle A\rangle}(B)\rangle} \cdot \mathcal{M}_{\langle A\rangle}\right) = \dim\langle \mathcal{M}_{\langle A\rangle}(B)\rangle + \dim\langle A\rangle = \dim\langle A \cup B\rangle = \dim\langle U\rangle$$

Due to the minimality of its degree, $\mathcal{M}_{\langle \mathcal{M}_{\langle A\rangle}(B)\rangle} \cdot \mathcal{M}_{\langle A\rangle}$ must be the MSP of $\mathcal{U}$. The base case $M_{\langle u_i\rangle}$ follows by the same argument. □

Lemma 5.13. *Let $a \in \mathbb{F}_{q^m}[x;\sigma]$ and $U \subseteq \mathbb{F}_{q^m}$. Furthermore, let $A, B \subseteq \mathbb{F}_{q^m}$ be a partition of U, i.e., $U = A \cup B$ and $A \cap B = \emptyset$. Let ϱ_A, ϱ_B be the respective remainders of the right divisions of the polynomial a by $\mathcal{M}_{\langle A\rangle}$ and $\mathcal{M}_{\langle B\rangle}$. Then, the multi-point evaluation of a at the set U is given by the recursive formula*

$$a(U) = \varrho_A(A) \cup \varrho_B(B). \tag{5.8}$$

If $U = \{u\}$ and $\deg a \leq 1$, say $a = a_1 x + a_0$ for $a_1, a_0 \in \mathbb{F}_{q^m}$, then

$$a(U) = \{a_1\sigma(u) + a_0 u\}. \tag{5.9}$$

Proof. Let $u \in U$. If $u \in A$, we have

$$a(u) = (\chi_A \cdot \mathcal{M}_{\langle A\rangle} + \varrho_A)(u) = \chi_A(\overbrace{\mathcal{M}_{\langle A\rangle}(u)}^{=0}) + \varrho_A(u) = \overbrace{\chi_A(0)}^{=0} + \varrho_A(u) = \varrho_A(u).$$

Likewise, we can prove that $a(u) = \varrho_B(u)$ if $u \in B$. Since $A \cup B$ is a partition of U, we obtain (5.8). Equation (5.9) follows directly from the definition of the evaluation map. □

By combining the ideas of the preceding lemmata, we obtain Algorithm 8 and Algorithm 9, which provide the following cost bounds on MSP computation and MPE.

Theorem 5.14. *Finding the MSP of an $\mathbb{F}_q$-subspace spanned by s elements of $\mathbb{F}_{q^m}$ and an MPE of a polynomial of degree at most s at s many points can be implemented in*

$$\begin{aligned}\mathcal{MSP}_{q^m}(s), \mathcal{MPE}_{q^m}(s) &\in O\Big(\max\Big\{s^{\log_2(3)}\log(s), \mathcal{M}_{q^m}(s), \mathcal{D}_{q^m}(s)\Big\}\Big)\\ &\subseteq O\Big(s^{\max\left\{\log_2(3), \min\{\frac{\omega+1}{2}, 1.635\}\right\}}\log(s)\Big).\end{aligned}$$

operations over $\mathbb{F}_{q^m}$ using Algorithms 8 and 9, respectively.

Algorithm 8: MSP(U)

Input: Generating set $U = \{u_1, \ldots, u_s\}$ of a subspace $\mathcal{U} \subseteq \mathbb{F}_{q^m}$.
Output: MSP $\mathcal{M}_{\langle U\rangle}$.

1 **if** $s = 1$ **then**
2 **return** $\mathcal{M}_{\langle u_1\rangle}(x)$ *according to* (5.7) // $O(1)$
3 **else**
4 $A \leftarrow \{u_1, \ldots, u_{\lfloor s/2\rfloor}\}$, $B \leftarrow \{u_{\lfloor s/2\rfloor+1}, \ldots, u_s\}$ // $O(1)$
5 $\mathcal{M}_{\langle A\rangle} \leftarrow \mathrm{MSP}(A)$ // $\mathcal{MSP}_{q^m}(s/2)$
6 $\mathcal{M}_{\langle A\rangle}(B) \leftarrow \mathrm{MPE}\Big(\mathcal{M}_{\langle A\rangle}, B\Big)$ // $\mathcal{MPE}_{q^m}(s/2)$
7 $\mathcal{M}_{\langle \mathcal{M}_{\langle A\rangle}(B)\rangle} \leftarrow \mathrm{MSP}\Big(\mathcal{M}_{\langle A\rangle}(B)\Big)$ // $\mathcal{MSP}_{q^m}(s/2)$
8 **return** $\mathcal{M}_{\langle \mathcal{M}_{\langle A\rangle}(B)\rangle} \cdot \mathcal{M}_{\langle A\rangle}$ // $\mathcal{M}_{q^m}(s)$

Algorithm 9: MPE($a, \{u_1, \ldots, u_s\}$)

Input: $a \in \mathbb{F}_{q^m}[x;\sigma]_{\leq s}$, $\{u_1, \ldots, u_s\} \in \mathbb{F}_{q^m}^s$
Output: Evaluation of a at all points u_i

1 **if** $s = 1$ **then**
2 **return** $\{a_1\sigma(u_1) + a_0u_1\}$ *according to* (5.9) // $O(1)$
3 **else**
4 $A \leftarrow \{u_1, \ldots, u_{\lfloor s/2\rfloor}\}$, $B \leftarrow \{u_{\lfloor s/2\rfloor+1}, \ldots, u_s\}$ // $O(1)$
5 $\mathcal{M}_{\langle A\rangle} \leftarrow \mathrm{MSP}(A)$ // $\mathcal{MSP}_{q^m}(s/2)$
6 $\mathcal{M}_{\langle B\rangle} \leftarrow \mathrm{MSP}(B)$ // $\mathcal{MSP}_{q^m}(s/2)$
7 $[\chi_A, \varrho_A] \leftarrow \mathrm{RightDiv}\Big(a, \mathcal{M}_{\langle A\rangle}\Big)$ // $\mathcal{D}_{q^m}(s)$
8 $[\chi_B, \varrho_B] \leftarrow \mathrm{RightDiv}\Big(a, \mathcal{M}_{\langle B\rangle}\Big)$ // $\mathcal{D}_{q^m}(s)$
9 **return** $\mathrm{MPE}(\varrho_A, A) \cup \mathrm{MPE}(\varrho_B, B)$ // $2 \cdot \mathcal{MPE}_{q^m}(s/2)$

Proof. We analyze the correctness and complexity of Algorithm 8 and Algorithm 9 jointly since they call each other recursively on smaller input size respectively. These convoluted calls are illustrated in Figure 5.1.

First, we show the correctness by induction. In the base case $s = 1$, both algorithms return the correct result by Lemma 5.12 and Lemma 5.13. The induction step also follows by Lemma 5.12 and Lemma 5.13, and due to $|A|$, $|B|$, $\deg \mathcal{M}_{\langle A\rangle}$, $\deg \varrho_A$, and $\deg \varrho_B$ being smaller than s.

As for the complexity, we analyze the non-negligible lines of both algorithms. Algorithm 8:

- Lines 5 and 7 both have complexity $\mathcal{MSP}_{q^m}\left(\frac{s}{2}\right)$ because $|A| \approx |B| \approx |U|/2 = \frac{s}{2}$.
- Line 6 computes the result in $\mathcal{MPE}_{q^m}\left(\frac{s}{2}\right)$ time because $\deg \mathcal{M}_{\langle A \rangle} \leq |B| \approx |U|/2 = \frac{s}{2}$.
- Line 8 multiplies two polynomials of degree $< s$ and therefore costs $\mathcal{M}_{q^m}(s)$.

In total, we obtain

$$\mathcal{MSP}_{q^m}(s) = 2 \cdot \mathcal{MSP}_{q^m}\left(\tfrac{s}{2}\right) + \mathcal{MPE}_{q^m}\left(\tfrac{s}{2}\right) + \mathcal{M}_{q^m}(s)\,. \tag{5.10}$$

The lines of Algorithm 9 have the following complexities:

- Lines 5 and 6 compute MSPs with input size $|A| \approx |B| \approx \frac{s}{2}$, so they cost $\mathcal{MSP}_{q^m}\left(\frac{s}{2}\right)$.
- Lines 7 and 8 divide polynomials from $\mathbb{F}_{q^m}[x;\sigma]_{\leq s}$ and have complexity $\mathcal{D}_{q^m}(s)$ each.
- Line 9 performs two MPEs of polynomials (remainders, cf. Theorem 2.18) with degree $< |B| \approx \frac{s}{2}$ at $|A| \leq |B| \approx \frac{s}{2}$ positions. Thus, the line has complexity $2 \cdot \mathcal{MPE}_{q^m}\left(\frac{s}{2}\right)$.

If both algorithms are called with the same set U, we can save one computation of $\mathrm{MSP}(A)$, which is computed both in Algorithm 8 (Line 5) and Algorithm 9 (Line 5), cf. Figure 5.1. At further recursion depths, the function is always first called by Algorithm 8. Hence, the recursive expression of the MPE cost is given by

$$\mathcal{MPE}_{q^m}(s) = \mathcal{MSP}_{q^m}\left(\tfrac{s}{2}\right) + 2 \cdot \mathcal{MPE}_{q^m}\left(\tfrac{s}{2}\right) + 2 \cdot \mathcal{D}_{q^m}(s)\,. \tag{5.11}$$

We define $C(s) := \max\left\{\mathcal{MPE}_{q^m}(s)\,, \mathcal{MSP}_{q^m}(s)\right\}$ and derive an upper bound on $C(s)$. Using (5.10) and (5.11), we obtain

$$C(s) \leq 3 \cdot C\left(\tfrac{s}{2}\right) + \max\left\{\mathcal{M}_{q^m}(s)\,, 2 \cdot \mathcal{D}_{q^m}(s)\right\}.$$

For $f(s) = \max\left\{\mathcal{M}_{q^m}(s)\,, \mathcal{D}_{q^m}(s)\right\}$, we distinguish three cases and use the master theorem:

- If $f(s) \in O\left(s^{\log_2(3)-\varepsilon}\right)$ for some $\varepsilon > 0$, then $C(s) \in O(s^{\log_2(3)})$.
- If $f(s) \in \Theta\left(s^{\log_2(3)}\right)$, then $C(s) \in O(s^{\log_2(3)} \log(s))$.
- If $f(s) \in \Omega\left(s^{\log_2(3)+\varepsilon}\right)$ for some $\varepsilon > 0$, then $C(s) \in O(\max\left\{\mathcal{M}_{q^m}(s)\,, \mathcal{D}_{q^m}(s)\right\})$.

Using the results of Section 5.2.1 and Section 5.2.2, the claim follows. □

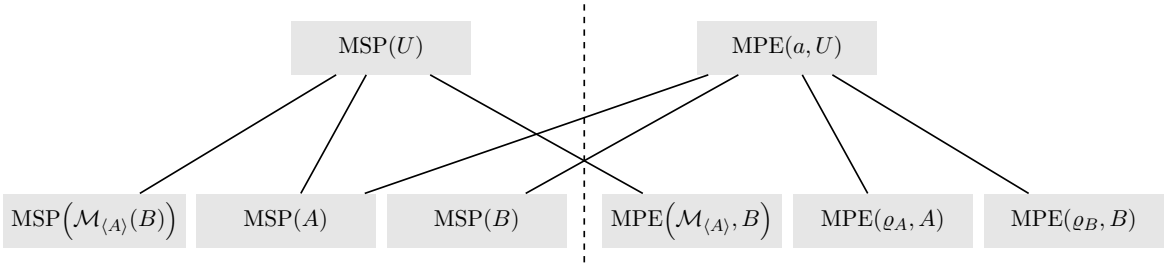


Figure 5.1: Illustration of the convoluted recursive calls of Algorithm 8 and Algorithm 9. Note that $\mathrm{MSP}(A)$ is called by both $\mathrm{MSP}(U)$ and $\mathrm{MPE}(a, U)$.

5.2.4 Interpolation

In the following, we use the fast MSP and MPE algorithms of the preceding section to obtain a fast divide-&-conquer interpolation algorithm for skew polynomials. Compared to existing fast interpolation algorithms for $\mathbb{F}_{q^m}[x]$ (cf. [GG99, Section 10.2]), we need to consider the properties of the skew polynomial evaluation map. The new algorithm is based on the following lemma.

Lemma 5.15. *Let $[x_i, y_i] \in \mathbb{F}_{q^m}^2$, for $i = 1, \ldots, s$, be such that the x_i are linearly independent over $\mathbb{F}_q$. The corresponding interpolation polynomial fulfills the recursion*

$$\mathcal{I}_{\{[x_i,y_i]\}_{i=1}^{s}} = \mathcal{I}_{\{[\widetilde{x}_i,y_i]\}_{i=1}^{\lfloor s/2 \rfloor}} \cdot \mathcal{M}_{\langle x_{\lfloor s/2 \rfloor+1},\ldots,x_s\rangle} + \mathcal{I}_{\{[\widetilde{x}_i,y_i]\}_{i=\lfloor s/2 \rfloor+1}^{s}} \cdot \mathcal{M}_{\langle x_1,\ldots,x_{\lfloor s/2 \rfloor}\rangle} \tag{5.12}$$

where we define

$$\widetilde{x}_i := \begin{cases} \mathcal{M}_{\langle x_{\lfloor s/2 \rfloor+1},\ldots,x_s\rangle}(x_i), & \text{if } i = 1, \ldots, \lfloor s/2 \rfloor \\ \mathcal{M}_{\langle x_1,\ldots,x_{\lfloor s/2 \rfloor}\rangle}(x_i), & \text{otherwise.} \end{cases}$$

If $s = 1$ (base case), we have $\mathcal{I}_{\{[x_i,y_i]\}_{i=1}^{1}} = \frac{y_1}{x_1}$.

Proof. The base case ($s = 1$) is clear due to $\frac{y_1}{x_1}(x_1) = \frac{y_1}{x_1}\sigma^0(x_1) = y_1$. The interpolation polynomial of the $[\widetilde{x}_i, y_i]$ is well-defined since $\widetilde{x}_1, \ldots, \widetilde{x}_{\lfloor s/2 \rfloor}$ are linearly independent. This is fulfilled because the $x_1, \ldots, x_s$ are linearly independent and $\mathcal{M}_{\langle x_{\lfloor s/2 \rfloor+1},\ldots,x_s\rangle}(\cdot)$ is a linear map whose kernel is the span of $x_{\lfloor s/2 \rfloor+1}, \ldots, x_s$ and does not include any of the $x_1, \ldots, x_{\lfloor s/2 \rfloor}$ since the x_i are linearly independent by assumption. Moreover,

$$\begin{aligned} \mathcal{I}_{\{[x_i,y_i]\}_{i=1}^{s}}(x_i) &= \mathcal{I}_{\{[\widetilde{x}_i,y_i]\}_{i=1}^{\lfloor s/2 \rfloor}}\big(\underbrace{\mathcal{M}_{\langle x_{\lfloor s/2 \rfloor+1},\ldots,x_s\rangle}(x_i)}_{=\widetilde{x}_i}\big) + \mathcal{I}_{\{[\widetilde{x}_i,y_i]\}_{i=\lfloor s/2 \rfloor+1}^{s}}\big(\underbrace{\mathcal{M}_{\langle x_1,\ldots,x_{\lfloor s/2 \rfloor}\rangle}(x_i)}_{=0}\big) \\ &= \mathcal{I}_{\{[\widetilde{x}_i,y_i]\}_{i=1}^{\lfloor s/2 \rfloor}}(\widetilde{x}_i) + \mathcal{I}_{\{[\widetilde{x}_i,y_i]\}_{i=\lfloor s/2 \rfloor+1}^{s}}(0) = y_i + 0 = y_i. \end{aligned}$$

Analogously, also $\mathcal{I}_{\{[\widetilde{x}_i,y_i]\}_{i=\lfloor s/2 \rfloor+1}^{s}}$ is well-defined and $\mathcal{I}_{\{[x_i,y_i]\}_{i=1}^{s}}(x_i) = y_i$ for $i > \lfloor \frac{s}{2} \rfloor$. The degree constraint holds due to

$$\begin{aligned} &\deg \mathcal{I}_{\{[x_i,y_i]\}_{i=1}^{s}} \leq \\ &\max\Big\{\underbrace{\deg \mathcal{I}_{\{[\widetilde{x}_i,y_i]\}_{i=1}^{\lfloor s/2 \rfloor}}}_{<\frac{s}{2}} + \underbrace{\deg \mathcal{M}_{\langle x_{\lfloor s/2 \rfloor+1},\ldots,x_s\rangle}}_{\leq\frac{s}{2}}, \underbrace{\deg \mathcal{I}_{\{[\widetilde{x}_i,y_i]\}_{i=\lfloor s/2 \rfloor+1}^{s}}}_{<\frac{s}{2}} + \underbrace{\deg \mathcal{M}_{\langle x_1,\ldots,x_{\lfloor s/2 \rfloor}\rangle}}_{\leq\frac{s}{2}}\Big\} < s, \end{aligned}$$

Hence, the right-hand side of (5.12) is the desired interpolation polynomial (cf. Lemma 2.24). □

Lemma 5.15 directly yields a recursive interpolation algorithm (Algorithm 10) that relies on the fast algorithms for MSP computation and MPE of Section 5.2.3. We can prove the following cost bound.

Theorem 5.16. *The interpolation polynomial of s point tuples can be found in*

$$\begin{aligned} \mathcal{I}_{q^m}(s) &\in O(\max\{\mathcal{MSP}_{q^m}(s), \mathcal{MPE}_{q^m}(s), \mathcal{M}_{q^m}(s)\}) \\ &\subseteq O(s^{\max\left\{\log_2(3), \min\{\frac{\omega+1}{2}, 1.635\}\right\}} \log(s)) \end{aligned}$$

operations over $\mathbb{F}_{q^m}$ using Algorithm 10.

Algorithm 10: $\mathrm{IP}(\{[x_i, y_i]\}_{i=1}^{s})$

Input: $[x_1, y_1], \ldots, [x_s, y_s] \in \mathbb{F}_{q^m}^2$, x_i linearly independent
Output: Interpolation polynomial $\mathcal{I}_{\{[x_i,y_i]\}_{i=1}^{s}}$

1 **if** $s = 1$ **then**
2 **return** $\frac{y_1}{x_1}$ // $O(1)$
3 **else**
4 $A \leftarrow \{x_1, \ldots, x_{\lfloor s/2 \rfloor}\}$, $B \leftarrow \{x_{\lfloor s/2 \rfloor + 1}, \ldots, x_s\}$ // $O(1)$
5 $\mathcal{M}_{\langle A \rangle} \leftarrow \mathrm{MSP}(A)$ // $\mathcal{MSP}_{q^m}(s/2)$
6 $\mathcal{M}_{\langle B \rangle} \leftarrow \mathrm{MSP}(B)$ // $\mathcal{MSP}_{q^m}(s/2)$
7 $\{\widetilde{x}_1, \ldots, \widetilde{x}_{\lfloor s/2 \rfloor}\} \leftarrow \mathrm{MPE}\big(\mathcal{M}_{\langle B \rangle}, A\big)$ // $\mathcal{MPE}_{q^m}(s/2)$
8 $\{\widetilde{x}_{\lfloor s/2 \rfloor + 1}, \ldots, \widetilde{x}_s\} \leftarrow \mathrm{MPE}\big(\mathcal{M}_{\langle A \rangle}, B\big)$ // $\mathcal{MPE}_{q^m}(s/2)$
9 $\mathcal{I}_1 \leftarrow \mathrm{IP}\Big(\{[\widetilde{x}_i, y_i]\}_{i=1}^{\lfloor s/2 \rfloor}\Big)$ // $\mathcal{I}_{q^m}(s/2)$
10 $\mathcal{I}_2 \leftarrow \mathrm{IP}\Big(\{[\widetilde{x}_i, y_i]\}_{i=\lfloor s/2 \rfloor + 1}^{s}\Big)$ // $\mathcal{I}_{q^m}(s/2)$
11 **return** $\mathcal{I}_1 \cdot \mathcal{M}_{\langle B \rangle} + \mathcal{I}_2 \cdot \mathcal{M}_{\langle A \rangle}$ // $2 \cdot \mathcal{M}_{q^m}(s/2)$

Proof. Correctness results from Lemma 5.15. Its non-negligible lines have the following costs:

- The complexities of Lines 5 and 6 are $\mathcal{MSP}_{q^m}\left(\frac{s}{2}\right)$.
- Lines 7 and 8 take $\mathcal{MPE}_{q^m}\left(\frac{s}{2}\right)$ time each.
- The algorithm calls itself recursively with input size $\approx \frac{s}{2}$ in Lines 9 and 10.
- Two multiplications of cost $2 \cdot \mathcal{M}_{q^m}\left(\frac{s}{2}\right)$ are required in Line 11.

This gives the recursive complexity expression

$$\mathcal{I}_{q^m}(s) = 2 \cdot \mathcal{I}_{q^m}\left(\tfrac{s}{2}\right) + 2 \cdot \left(\mathcal{MSP}_{q^m}\left(\tfrac{s}{2}\right) + \mathcal{MPE}_{q^m}\left(\tfrac{s}{2}\right) + \mathcal{M}_{q^m}\left(\tfrac{s}{2}\right)\right),$$

which implies the claim by the master theorem. □

5.2.5 σ-Transform

The following theorem shows that the σ-transform can be computed by a vector-matrix multiplication and its inverse by solving a Toeplitz linear system, respectively. This observation implies that both problems can be solved in quasi-linear time over $\mathbb{F}_{q^m}$.

Theorem 5.17. *Let $s \mid m$. The σ-transform of a polynomial $a \in \mathbb{F}_{q^m}[x;\sigma]_{<s}$ w.r.t a normal basis $\mathcal{B}$ of $\mathbb{F}_{q^s}$ can be implemented by multiplying a vector with a Toeplitz matrix (cf. Algorithm 11). Similarly, computing the inverse σ-transform of $\hat{a} \in \mathbb{F}_{q^m}[x;\sigma]_{<s}$ w.r.t $\mathcal{B}$ consists of solving a Toeplitz system (cf. Algorithm 12). Thus, both operations can be implemented in*

$$\mathcal{ST}_{q^m}(s) \in O(s \log(s) \log(\log(s)))$$

operations over $\mathbb{F}_{q^m}$.

Algorithm 11: $\mathrm{ST}(a, \mathcal{B})$

Input: $a = \sum_{i=0}^{s-1} a_i x^i \in \mathbb{F}_{q^m}[x;\sigma]_{<s}$ and normal basis $\mathcal{B} = \{\beta, \sigma(\beta), \ldots, \sigma^{s-1}(\beta)\}$
Output: σ-transform $\hat{a} \in \mathbb{F}_{q^m}[x;\sigma]_{<s}$ of a w.r.t. $\mathcal{B}$

1 Set up the Toeplitz matrix $\boldsymbol{B}$ as in (5.13)
2 $[\hat{a}_0, \ldots, \hat{a}_{s-1}] \leftarrow [a_{s-1}, \ldots, a_0] \cdot \boldsymbol{B}$ using (ordinary) polynomial multiplication (see e.g. [BP12, Chapter 2, Problem 5.1]) // $O(s \log(s) \log(\log(s)))$
3 **return** $\hat{a} = \sum_{i=0}^{s-1} \hat{a}_i x^i$

Algorithm 12: $\mathrm{IST}(\hat{a}, \mathcal{B})$

Input: $\hat{a} = \sum_{i=0}^{s-1} \hat{a}_i x^i \in \mathbb{F}_{q^m}[x;\sigma]_{<s}$ and normal basis $\mathcal{B} = \{\beta, \sigma(\beta), \ldots, \sigma^{s-1}(\beta)\}$
Output: Inverse σ-transform $a \in \mathbb{F}_{q^m}[x;\sigma]_{<s}$ of $\hat{a}$ w.r.t. $\mathcal{B}$

1 Set up the Toeplitz matrix $\boldsymbol{B}$ as in (5.13)
2 Solve $[\hat{a}_0, \ldots, \hat{a}_{s-1}] = [a_{s-1}, \ldots, a_0] \cdot \boldsymbol{B}$ for $[a_{s-1}, \ldots, a_0]$ using the algorithm for solving Toeplitz systems in [BGY80] // $O(s \log^2(s) \log(\log(s)))$
3 **return** $a = \sum_{i=0}^{s-1} a_i x^i$

Proof. Correctness of the two algorithms follows by reformulating the definition of the σ-transform and its inverse as

$$[\hat{a}_0, \ldots, \hat{a}_{s-1}] = [a_{s-1}, \ldots, a_0] \cdot \underbrace{\begin{bmatrix} \sigma^{s-1}(\beta) & \sigma^s(\beta) & \sigma^{s+1}(\beta) & \ldots & \sigma^{2s-1}(\beta) \\ \sigma^{s-2}(\beta) & \sigma^{s-1}(\beta) & \sigma^s(\beta) & \ldots & \sigma^{2s-2}(\beta) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \sigma^0(\beta) & \sigma^1(\beta) & \sigma^2(\beta) & \ldots & \sigma^{s-1}(\beta) \end{bmatrix}}_{=: \boldsymbol{B}}, \tag{5.13}$$

where the matrix $\boldsymbol{B}$ is an $s \times s$ Toeplitz matrix over $\mathbb{F}_{q^m}$, which is invertible since it is also a σ-Vandermonde matrix (cf. [LN97, Lemma 3.5.1]).

The multiplication of $[a_{s-1}, \ldots, a_0]$ with the Toeplitz matrix $\boldsymbol{B}$ equals computing the coefficients $s-1, \ldots, 2s-1$ of the ordinary polynomial multiplication of $\sum_{i=0}^{s-1} a_{s-1-i} x^i \in \mathbb{F}_{q^m}[x]$ with $\sum_{i=0}^{2s-1} \sigma^i(\beta) x^i \in \mathbb{F}_{q^m}[x]$, see e.g. [BP12, Chapter 2, Problem 5.1]. Thus, it can be implemented in $O(s \log(s) \log(\log(s)))$ operations over $\mathbb{F}_{q^m}$ using the Schönhage–Strassen polynomial multiplication algorithm.

It was shown by Brent, Gustavson, and Yun in [BGY80] that solving a linear Toeplitz system can be reduced to a Padé approximation problem, which again can be solved using the extended Euclidean algorithm with stopping condition over $\mathbb{F}_{q^m}[x]$. A fast version of the latter algorithm was introduced by Aho and Hopcroft in [AH74] based on existing algorithms over $\mathbb{Z}$, and corrected in [BGY80]. Its complexity is $O(\mathcal{P}(s) \log(s))$, where $\mathcal{P}(s) \in O(s \log(s) \log(\log(s)))$ is the cost of multiplying two ordinary polynomials of degree at most s. Hence, we obtain the claimed complexity. □

Since the input sizes of Algorithm 11 and Algorithm 12 are s, we have $\mathcal{ST}_{q^m}(s) \in \Omega(s)$, so the cost bound in Theorem 5.17 is tight up to log-factors.

Remark 5.18. *We can also perform the σ-transform with respect to an incomplete basis $\{\sigma^i(\beta), \dots, \sigma^{i+s'-1}(\beta)\} \subseteq \mathcal{B}$ for some $i \in \mathbb{Z}$ and $s' \leq s$. In this case, the σ-transform and its inverse correspond to a vector-matrix multiplication and solving a system with an $(s' \times s')$-submatrix of $\boldsymbol{B}$, indexed by consecutive rows and columns. This costs $O^{\sim}(s')$ over $\mathbb{F}_{q^m}$.*

Implication: Optimal Skew Polynomial Multiplication for $s = m$

It was already mentioned in [Wac13, Section 3.1.3, page 43] that multiplication of skew polynomials in $\mathbb{F}_{q^m}[x;\sigma]_{<m}$ and $(m \times m)$-matrices over $\mathbb{F}_q$ are closely connected through the σ-transform. However, an efficient reduction of one problem to the other was not yet possible due to the non-existence of efficient (inverse) σ-transform computation.

Our speed-up of the σ-transform from $O(s^2)$ to $O^{\sim}(s)$, in operations over $\mathbb{F}_{q^m}$, implies the following two important observations, which we prove in Appendix A.3:

- For $s = m$, the cost of skew polynomial multiplication is lower-bounded by

$$\mathcal{M}_{q^m}(m) \in \Omega(m^{\omega-1}),$$

  where ω is the matrix multiplication exponent.

- There is a multiplication algorithm for skew polynomials of degree $s \in \Theta(m)$ that costs $O(m^{\omega})$ operations over $\mathbb{F}_q$ (which is—up to log-factors—as fast as a hypothetical algorithm that costs $O(m^{\omega-1})$ over $\mathbb{F}_{q^m}$).

5.3 Concluding Remarks

In this chapter, we have shown how to decode Gabidulin codes in sub-quadratic time in the code length over $\mathbb{F}_{q^m}$. We have achieved this result by improving the best-known cost bound of many operations over skew polynomials, such as multiplication, division, multi-point evaluation, minimal subspace polynomial computation, interpolation, and the σ-transform in cases relevant for decoding Gabidulin codes. An overview of new and known cost bounds is given in Table 5.1 on page 98.

After the first publication of the results in this chapter, Caruso and Le Borgne [CL17b] proposed a fast algorithm for multiplication of skew polynomials of complexity $O(s^{\omega-2}m^2)$ over $\mathbb{F}_q$, which improves upon the multiplication algorithm in Section 5.2.2 for $m^{2/(5-\omega)} \leq s \leq m$. As an immediate consequence, also the cost bound for division is improved to $O^{\sim}(s^{\omega-2}m^2)$ over $\mathbb{F}_q$ in this range. The result does not influence our cost bounds for the σ-transform, MSP, MPE, and interpolation since the first is already quasi-linear and the other bounds would be dominated by the $s^{\log_2(3)}$ factor.

As open questions, it remains to find asymptotically optimal algorithms for computing the mentioned operations. We have partly answered this question for skew polynomial multiplication in the case when the involved polynomials have degree $s \in \Theta(m)$ and—up to log-factors—for the σ-transformation, cf. Section 5.2.5. Faster multiplication algorithms would directly imply faster division. It is also an interesting question whether one can eliminate the $\log_2(3)$ lower bound on the exponent in our MSP, MPE, and interpolation cost bounds.

Remarks on Generality

This chapter considers skew polynomial rings over a finite field $\mathbb{F}_{q^m}$. However, most results immediately translate to skew polynomials over an arbitrary Galois extension L/K with cyclic Galois group, which is generated by the automorphism σ (cf. Section 2.3.1). Table 5.2 outlines which statements hold in this case and states—if required—which additional conditions must be satisfied.

Table 5.2: List of statements that hold for skew polynomials over any field extension L/K of cyclic Galois group with generator σ. Additional requirements: *Automorphism $\sigma^{(i)}$ can be computed in $O(1)$ for any $i = 1, \ldots, [L : K]$. $^+$Two elements in $L[x]$ of degree at most s can be multiplied in $O(s \log(s) \log(\log(s)))$ operations over L.

Section	Statements
Decoding (Section 5.1)	Lemma 5.1, Lemma 5.3, Theorem 5.4*
Multiplication (Section 5.2.1)	Theorem 5.7*, Corollary 5.8*, Remark 5.9*
Division (Section 5.2.2)	Lemma 5.10*, Corollary 5.11*
MSP & MPE (Section 5.2.3)	Lemma 5.12, Lemma 5.13, Theorem 5.14*
Interpolation (Section 5.2.4)	Lemma 5.15, Theorem 5.16*
σ-Transform (Section 5.2.5)	Theorem 5.17$^{*,+}$, Remark 5.18$^{*,+}$

Note that the complexities are given in operations over the field L. If the field is of characteristic zero, one must be careful about the increasing size of field element representations of intermediate results (e.g. growing nominator and denominator in extensions of $\mathbb{Q}$). This effect might yield an increased overall bit complexity compared to a naive implementation.

6

Decoding Interleaved Gabidulin Codes Using Row Reduction

Interleaved Gabidulin codes allow to correct most errors of rank weight beyond half the minimum distance by assuming a synchronized error model for the constituent parts of a codeword. This fact can be utilized in network coding [KK08, SKK08, SB10] and to attack cryptosystems based on rank-metric codes [LO06, GOT17, WPR18].

The first such decoder was presented by Loidreau and Overbeck [LO06], a generalization of Loidreau's Welch–Berlekamp-like algorithm [Loi06]. Alternatively, decoding can be formulated as a set of simultaneous key equations by the algorithm of Sidorenko, Jiang, and Bossert [SB10, SJB11]. Both algorithms are partial unique decoders. Recently, Wachter-Zeh and Zeh [WZ14] presented an interpolation-based decoding method. The algorithm can be seen as a list decoder with exponential worst-case, but comparably small average list size, or as a partial unique decoder with small failure probability.

In Section 6.1, we show that both the main task of the algorithm by Sidorenko, Jiang, and Bossert and the interpolation step in the algorithm by Wachter-Zeh and Zeh can be solved by row reduction of a specific skew polynomial module basis. The approach is inspired by many recent publications that have unified the core of various decoding algorithms, such as the Guruswami–Sudan list decoder [Ale05, LO08, BB10, CH10, NB15] and power decoding [Nie13a, Nie14, NB15, Ros18] (see also Chapter 3/Appendix A.1), both for Reed–Solomon and Hermitian codes, based on $\mathbb{F}_{q^m}[x]$-row reduction. Using well-known row reduction algorithms, these decoding algorithms have become flexible, easy to prove, and partly faster.

In Section 6.2, we adapt two of these $\mathbb{F}_{q^m}[x]$-row reduction methods, the Mulders–Storjohann [MS03] and Alekhnovich [Ale05] algorithm, to skew polynomial rings and show that most of their properties translate to the $\mathbb{F}_{q^m}[x;\sigma]$ case with only a few modifications. The complexity analysis is more involved due to the non-commutativity of the skew polynomial ring. Compared to previous works on computing matrix normal forms over so-called Ore rings [AB01, BCL06, Mid11], which are mostly concerned in coefficient growth due to infinite base fields, our algorithms are faster when considered on skew polynomials over finite fields.

Finally, we present two row reduction algorithms that are specialized for the two decoding problems in Section 6.3. The first is an $\mathbb{F}_{q^m}[x;\sigma]$-adaption of Rosenkilde's Demand–Driven algorithm [Nie13a] and the other one, weak Popov walk, is a conceptually new row reduction method inspired by Gröbner walks [CKM97].

The results of this chapter were partly published in [LNPS15], [PRLS17], and [PMM+17].

6.1 Implementing Known Decoding Algorithms Using Row Reduction

In this section, we show how to implement both key-equation- and interpolation-based decoding methods for interleaved Gabidulin codes to row reduction of skew polynomial matrices. For notational convenience, we sometimes start matrix indices at 0 for the decoding problems and some related row reduction algorithms in the succeeding sections of this chapter.

6.1.1 Key-Equation-Based Decoding

We recall the key-equation-based decoding algorithm for *horizontally* interleaved Gabidulin codes by Sidorenko, Jiang, and Bossert [SB10, SJB11] and rephrase it slightly using the notation of Wachter-Zeh et al.'s decoding algorithm for non-interleaved Gabidulin codes [WAS13, Wac13] (see also Section 5.1).

A System of Key Equations

In the case of interleaved Gabidulin codes, we obtain h recevied words $\boldsymbol{r}_i = \boldsymbol{c}_i + \boldsymbol{e}_i \in \mathbb{F}_{q^m}^n$ for $i = 1, \dots, h$, where $\boldsymbol{c}_i = [f_i(\alpha_1), \dots, f_i(\alpha_n)]$ is a codeword of the i^{th} constituent Gabidulin code $\mathcal{C}_{\text{G}}^{(i)}[n, k_i]$ (recall that $\deg f_i < k_i$).

Similar to the approach in Section 5.1, we can obtain interpolation polynomials $R_i \in \mathbb{F}_{q^m}[x;\sigma]_{<n}$ that satisfy $R(\alpha_i) = r_i$. Furthermore, let $G := \mathcal{M}_{\langle \alpha_1, \dots, \alpha_n \rangle}$ be the minimal subspace polynomial of the space spanned by the evaluation points.

We consider *horizontally* interleaved Gabiulin codes, cf. Section 2.3.4, i.e., the spans of the error word components $\langle e_{i,1}, \dots, e_{i,n} \rangle \subseteq \mathbb{F}_{q^m}$ are contained in an error space $\mathcal{E} \subseteq \mathbb{F}_{q^m}$ of dimension τ. We would like to obtain a joint error locator $\Lambda := \mathcal{M}_{\mathcal{E}}$ and the message polynomials f_i from the following system of key equations.

Lemma 6.1 (Key Equations)**.** *The polynomials R_i, f_i, Λ, and G, defined as above, satisfy*

$$\Lambda R_i \equiv \Lambda f_i \mod_{\text{r}} G \quad \forall\, i = 1, \dots, h.$$

Since the polynomials Λf_i have small degree $\tau + k_i - 1$, one can find the unknown polynomial Λ, which is the same in all h congruences, by solving a large linear system of equations in the top $n - \tau - k_i$ coefficients of $\Lambda R_i \mod_{\text{r}} G$. For random errors of rank weight τ, the system has a unique monic solution with large probability[1] (cf. [SB10, SJB11]) if

$$\tau \leq \frac{h}{h+1}\left(n - \frac{1}{h}\sum_{i=1}^{h} k_i\right)$$

many errors occurred. In the following, we show how to solve the problem more efficiently.

Remark 6.2. *Sidorenko, Jiang, and Bossert [SB10, SJB11] formulated an equivalent syndrome-based key equation. In this form, the algorithm can be rephrased to work for the error model considered for* vertically *interleaved Gabidulin codes, and is closely connected to Loidreau's decoding algorithm [LO06], see [Wac13, Section 4.1]. Furthermore, it can be solved like the above key equation as shown below.*

[1] This failure behavior is sometimes desirable: It was shown in [WPR18] that the attack of [GOT17] on the rank-metric-based Faure–Loidreau cryptosystem [FL06] is equivalent to decoding interleaved Gabidulin codes. Thus, the system can be repaired by choosing the public key such that all known decoders fail.

Solving the System of Key Equations by a Shift Register Synthesis Problem

Lemma 6.1, together with the observation that $\deg \Lambda + n > \deg(\Lambda f_i) - (n - k_i)$, motivates the following natural generalization of Problem 5.2 in Section 5.1, which we call "Multi-Sequence generalized Linear Skew-Feedback Shift Register" (MgLSSR) synthesis problem.[2]

Problem 6.3 (MgLSSR)**.** *Let $r_i \in \mathbb{F}_{q^m}[x;\sigma]$, $g_i \in \mathbb{F}_{q^m}[x;\sigma] \setminus \{0\}$, and $\gamma_i \in \mathbb{N}_0$ for $i = 1, \dots, h$ be given. Furthermore, let $h \in O(\mu)$, where $\mu := \min_i\{\gamma_i + \deg g_i\}$. Find skew polynomials $\lambda, \omega_1, \dots, \omega_h \in \mathbb{F}_{q^m}[x;\sigma]$, with λ of minimal degree such that the following holds:*

$$\lambda r_i \equiv \omega_i \mod_{\mathrm{r}} g_i \tag{6.1}$$

$$\deg \lambda + \gamma_0 > \deg \omega_i + \gamma_i \tag{6.2}$$

The significance of the MgLSSR problem extends beyond the key equation in Lemma 6.1: It is able to solve a variety of key equations arising from decoding Gabidulin codes ($h = 1$) and interleaved Gabidulin codes, both for correcting errors and erasures. An overview of these key equations can be found in Appendix A.4. In all of these key equations, the parameter μ is in the order of the code length n and usually h is much smaller than μ. We will measure the complexity of algorithms solving Problem 6.3 by h and μ.

We show how to solve Problem 6.3 by row reduction of a skew polynomial module basis that depends on the input of the problem, analogously to the approach for $\mathbb{F}_{q^m}[x]$-shift register synthesis presented by Rosenkilde in [Nie13a]. Compared to the latter case, we need to take care of the non-commutativity of $\mathbb{F}_{q^m}[x;\sigma]$.

Finding a Solution of the MgLSSR Problem by Row Reduction

In the following, let $h \in O(\mu)$, $r_i \in \mathbb{F}_{q^m}[x;\sigma]$, $g_i \in \mathbb{F}_{q^m}[x;\sigma] \setminus \{0\}$, and $\gamma_i \in \mathbb{N}_0$ for $i = 1, \dots, h$ be arbitrary but fixed. W.l.o.g., we assume that $\deg r_i \leq \deg g_i$ for all i since replacing r_i by $(r_i \mod_{\mathrm{r}} g_i)$ would yield the same solution of Problem 6.3. We define the set $\mathcal{M}^{(\mathrm{M})}$ of vectors $\boldsymbol{v} \in \mathbb{F}_{q^m}[x;\sigma]^{h+1}$ satisfying the congruence relations (6.1), i.e.,

$$\mathcal{M}^{(\mathrm{M})} := \left\{[\lambda, \omega_1, \dots, \omega_h] \in \mathbb{F}_{q^m}[x;\sigma]^{h+1} \;:\; \lambda r_i \equiv \omega_i \mod_{\mathrm{r}} g_i \;\forall i = 1, \dots, h\right\}. \tag{6.3}$$

Lemma 6.4. *Consider an instance of Problem 6.3 and $\mathcal{M}^{(\mathrm{M})}$ as in (6.3). The set $\mathcal{M}^{(\mathrm{M})}$ with component-wise addition and left multiplication by elements of $\mathbb{F}_{q^m}[x;\sigma]$ forms a free left module over $\mathbb{F}_{q^m}[x;\sigma]$. A basis of $\mathcal{M}^{(\mathrm{M})}$ is given by the rows of the matrix*

$$\boldsymbol{M}^{(\mathrm{M})} = \begin{bmatrix} 1 & r_1 & r_2 & \dots & r_h \\ 0 & g_1 & 0 & \dots & 0 \\ 0 & 0 & g_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & g_h \end{bmatrix} \in \mathbb{F}_{q^m}[x;\sigma]^{(h+1)\times(h+1)} \,. \tag{6.4}$$

Proof. The first part of the claim follows directly since $\mathcal{M}^{(\mathrm{M})} \subseteq \mathbb{F}_{q^m}[x;\sigma]^{h+1}$ is closed under addition and left scalar multiplication. Similar to [Nie13a, Lemma 1], by (6.1), any vector

[2] Over $\mathbb{F}_{q^m}[x]$, this problem is also known as a multi-sequence shift-register synthesis [FT91, SS11], simultaneous Padé approximation [BGM96], or vector rational function reconstruction [OS06]. It has further generalizations (e.g., [BL92, BL94]), some with applications in decoding algebraic codes [RR00, ZGA11, NB15].

$\boldsymbol{v} \in \mathcal{M}^{(\mathrm{M})}$ satisfies $v_0 r_i = v_i + p_i g_i$ for some $p_i \in \mathbb{F}_{q^m}[x;\sigma]$. Hence, the rows $\boldsymbol{m}_0, \dots, \boldsymbol{m}_h$ of $\boldsymbol{M}^{(\mathrm{M})}$ are a generator set of $\mathcal{M}^{(\mathrm{M})}$ since we can write $\boldsymbol{v} = v_0\boldsymbol{m}_0 + \sum_{i=1}^{h} p_i \boldsymbol{m}_i$ (note that the p_i are multiplied from the left, which is essential due to the non-commutativty of $\mathbb{F}_{q^m}[x;\sigma]$). They are linearly independent since $\mathcal{M}^{(\mathrm{M})}$ is a triangular matrix with non-zero diagonal elements. □

Lemma 6.4 provides a convenient description of all vectors satisfying the congruence relations of Problem 6.3. In order to obtain a solution of the problem, we need to find an element in the left row space of $\boldsymbol{M}^{(\mathrm{M})}$ that has minimal $\deg \lambda$ and also satisfies the degree inequality (6.2). The following lemma shows that both properties are fulfilled by a particular row of a shifted weak Popov form of $\boldsymbol{M}^{(\mathrm{M})}$ (recall the notation from Section 2.3.2).

Lemma 6.5. *Consider an instance of Problem 6.3 and the module $\mathcal{M}^{(\mathrm{M})}$ as in (6.3). Let $\boldsymbol{w}^{(\mathrm{M})} = [\gamma_0, \dots, \gamma_h] \in \mathbb{N}_0^{h+1}$ and $\boldsymbol{V}$ be a basis of $\mathcal{M}^{(\mathrm{M})}$ in $\boldsymbol{w}^{(\mathrm{M})}$-shifted weak Popov form. Then, the row $\boldsymbol{v}$ of $\boldsymbol{V}$ with $\mathrm{LP}(\Phi_{\boldsymbol{w}^{(\mathrm{M})}}(\boldsymbol{v})) = 0$ is a solution of Problem 6.3.*

Proof. Since $\boldsymbol{v} \in \mathcal{M}^{(\mathrm{M})}$, it fulfills the congruences of Problem 6.3. We have $\mathrm{LP}(\Phi_{\boldsymbol{w}^{(\mathrm{M})}}(\boldsymbol{u})) = 0$ if and only if $\deg u_0 + \gamma_0 = \deg(\Phi_{\boldsymbol{w}^{(\mathrm{M})}}(\boldsymbol{u})_0) > \deg(\Phi_{\boldsymbol{w}^{(\mathrm{M})}}(\boldsymbol{u})_i) = \deg u_i + \gamma_i$ for all $i > 0$., i.e. $\boldsymbol{u}$ satisfies the degree restriction (6.2). The minimality of u_0 is ensured by Lemma 2.29. □

Lemma 6.5 implies a solution strategy for Problem 6.3, which is summarized in Algorithm 13. The complexity of the algorithm is determined by a skew polynomial row reduction. In the following sections, we analyze how and in which complexity we can row-reduce $\mathbb{F}_{q^m}[x;\sigma]$-matrices. The results of these sections are summarized in the following theorem.

Theorem 6.6. *Algorithm 13 is correct and can be implemented using*

i) Algorithm 15 on page 118 in $O(h^2\mu^2)$,

ii) Algorithm 17 on page 125 in $O(h^3 \mathcal{M}_{q^m}(\mu)\log(\mu)) \subseteq O(h^3 \mu^{\min\{\frac{\omega+1}{2}, 1.635\}} \log(\mu))$, and

iii) Algorithm 19 on page 129 in $O(h\mu^2)$, if $g_i = x^{t_i} + c_i$ with $t_i \in \Theta(\mu)$ and $c_i \in \mathbb{F}_{q^m}$ $\forall$ i,

operations over $\mathbb{F}_{q^m}$.

Algorithm 13: Solve Problem 6.3 (MgLSSR) by Row Reduction

Input: Instance of Problem 6.3
Output: Solution $\boldsymbol{v} = [\lambda, \omega_1, \dots, \omega_h]$ of Problem 6.3

1 Set up $\boldsymbol{M}^{(\mathrm{M})}$ as in (6.4)
2 Compute $\boldsymbol{V}$ as a $\boldsymbol{w}^{(\mathrm{M})}$-shifted weak Popov form of $\boldsymbol{M}^{(\mathrm{M})}$, with $\boldsymbol{w}^{(\mathrm{M})}$ as in Lemma 6.5
3 **return** *the row $\boldsymbol{v}$ of $\boldsymbol{V}$ having* $\mathrm{LP}(\Phi_{\boldsymbol{w}^{(\mathrm{M})}}(\boldsymbol{v})) = 0$

Proof. Correctness is clear by Lemma 6.5. A $\boldsymbol{w}^{(\mathrm{M})}$-shifted weak Popov form is obtained by transforming $\Phi_{\boldsymbol{w}^{(\mathrm{M})}}(\boldsymbol{M}^{(\mathrm{M})})$ into weak Popov form using the algorithms in Section 6.2 and 6.3, and then applying $\Phi^{-1}_{\boldsymbol{w}^{(\mathrm{M})}}$ it. The complexities follow by Corollary 6.28 in Section 6.2.2, Corollary 6.41 in Section 6.2.3, and Theorem 6.48 in Section 6.3.1, respectively. □

Remark 6.7. *Case iii) of Theorem 6.6 applies to many key equations arising from decoding Gabidulin and interleaved Gabidulin codes, cf. Appendix A.4. For instance, the g_i of the key equation in Lemma 5.1 and its generalized version in Lemma 6.1 can be chosen of the form $x^n - 1$ if $n \mid m$. Also, in syndrome-based key equations, the g_i are simply powers of x. Hence, we achieve the complexity as the Berlekamp–Massey-like algorithm in [SJB11], but for a more general class of shift register synthesis problems.*

6.1.2 Interpolation-Based Decoding

In the following, we recall the interpolation-based decoding algorithm for *vertically* interleaved Gabidulin codes by Wachter-Zeh and Zeh [WZ13], and show that its interpolation step can be solved by row reduction of a specific skew polynomial module basis. The algorithm is inspired by the Guruswami–Sudan algorithm for Reed–Solomon codes [GS99], and our row reduction approach similarly resembles the fast module-based algorithms for implementing the Guruswami–Sudan, see e.g. [LO08, BB10].

Decoding Using Interpolation

Problem 6.8 (Interpolation Step)**.** *Let $n, k_1, \dots, k_h, \tau \in \mathbb{N}$. Given $\boldsymbol{r}_i = [r_{i,1}, \dots, r_{i,n}] \in \mathbb{F}_{q^m}^n$ for $i = 1, \dots, h$, and $\alpha_1, \dots, \alpha_n \in \mathbb{F}_{q^m}$ that are linearly independent over $\mathbb{F}_{q^m}$. Find a non-zero skew polynomial vector $[Q_0, \dots, Q_h] \in \mathbb{F}_{q^m}[x;\sigma]^{h+1}$ such that*

$$Q_0(\alpha_i) + \sum_{i=1}^{h} Q_j(r_{i,j}) = 0 \qquad \forall\, i = 1, \dots, n \tag{6.5}$$

$$\deg Q_0 < n - \tau \tag{6.6}$$

$$\deg Q_j < n - \tau - k_i + 1 \qquad \forall\, j = 1, \dots, h \tag{6.7}$$

It was shown in [WZ14] that Problem 6.8 has a solution if

$$\tau < \frac{h}{h+1}\left(n + 1 - \frac{1}{h}\sum_{i=1}^{h} k_i\right), \tag{6.8}$$

Also, if the number of errors, i.e., the rank of the error matrix $\boldsymbol{E}_{\mathrm{v}}$ (cf. (2.5) in Section 2.3.4), is at most τ, the message polynomials $f_i \in \mathbb{F}_{q^m}[x;\sigma]_{<k_i}$ satisfy

$$Q_0 + \sum_{i=1}^{h} Q_i f_i = 0,$$

so they are the roots of the formal multi-variate skew polynomial $Q = Q_0(x) + \sum_{i=0}^{h} Q_i(y_i)$ in the indeterminates $x, y_1, \dots, y_h$. Wachter-Zeh and Zeh [WZ14] presented a root-finding algorithm that returns a basis of the affine root space of Q in $O(h^3 n^2)$ operations over $\mathbb{F}_{q^m}$. The number of roots of the multivariate polynomial Q may be exponential in $\sum_i k_i$. However, [WZ14] showed that the expected dimension of this affine space is close to 0, so with large probability only the actual message polynomials $f_1, \dots, f_h$ are contained in the list.

Solving the Interpolation Problem Using Row Reduction

Consider an instance of Problem 6.8. Let $\mathcal{M}^{(\mathrm{I})} \subseteq \mathbb{F}_{q^m}[x;\sigma]^{h+1}$ be the set of vectors $[Q_0,\dots,Q_h]$ satisfying (6.5), i.e.,

$$\mathcal{M}^{(\mathrm{I})} = \left\{ [Q_0,\dots,Q_h] \in \mathbb{F}_{q^m}[x;\sigma]^{h+1} \,:\, Q_0(\alpha_i) + \sum_{i=1}^{h} Q_j(r_{i,j}) = 0 \;\forall\, i = 1,\dots,n \right\}. \tag{6.9}$$

Lemma 6.9. *The set $\mathcal{M}^{(\mathrm{I})}$ as in (6.9) with component-wise addition and left multiplication with elements from $\mathbb{F}_{q^m}[x;\sigma]$ is a free left $\mathbb{F}_{q^m}[x;\sigma]$-module. A basis of $\mathcal{M}^{(\mathrm{I})}$ is given by the rows of the matrix*

$$\boldsymbol{M}^{(\mathrm{I})} = \begin{bmatrix} G & 0 & 0 & \dots & 0 \\ -R_1 & 1 & 0 & \dots & 0 \\ -R_2 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -R_h & 0 & 0 & \dots & 1 \end{bmatrix}, \tag{6.10}$$

where $G = \mathcal{M}_{\langle \alpha_1,\dots,\alpha_n \rangle}$ is the annihilator polynomial of the α_i and for each $i = 1,\dots,h$, $R_i = \mathcal{I}_{\{[\alpha_j, r_{i,j}]\}_{j=1}^{n}}$ is the unique interpolation polynomial of degree $< n$ satisfying $R_i(\alpha_j) = r_{i,j}$ for all $j = 1,\dots,n$.

Proof. The set $\mathcal{M}^{(\mathrm{I})}$ is closed under addition since $(a+b)(\alpha) = a(\alpha) + b(\alpha)$ for all $a, b \in \mathbb{F}_{q^m}[x;\sigma]$ and $\alpha \in \mathbb{F}_{q^m}$. It is closed under multiplication since for $[Q_0,\dots,Q_h]$ and $f \in \mathbb{F}_{q^m}[x;\sigma]$, we have $f \cdot [Q_0,\dots,Q_h] \in \mathcal{M}^{(\mathrm{I})}$ due to

$$(f \cdot Q_0)(\alpha_j) + \sum_{i=1}^{h} (f \cdot Q_i)(\alpha_j) = f \cdot \left(Q_0(\alpha_j) + \sum_{i=1}^{h} Q_i(\alpha_j) \right) = f(0) = 0 \quad \forall\, j = 1,\dots,n.$$

The rows $\boldsymbol{m}_1,\dots,\boldsymbol{m}_h$ of $\boldsymbol{M}^{(\mathrm{I})}$ are linearly independent since $\boldsymbol{M}^{(\mathrm{I})}$ is in lower triangular form and has non-zero diagonal elements. Also, they are in $\mathcal{M}^{(\mathrm{I})}$ since

$$\begin{aligned} \boldsymbol{m}_0 &: && G(\alpha_j) = 0, && \forall j = 1,\dots,n, \\ \boldsymbol{m}_i &: && 1(r_{i,j}) - R_i(\alpha_j) = r_{i,j} - r_{i-j}, && \forall j = 1,\dots,n. \end{aligned}$$

Let $[Q_0,\dots,Q_h] \in \mathcal{M}^{(\mathrm{I})}$. We show that it is a linear combination of the $\boldsymbol{m}_i$. We have

$$[Q_0,\dots,Q_h] - \sum_{i=1}^{h} Q_i \boldsymbol{m}_i = [a,0,\dots,0] \in \mathcal{M}^{(\mathrm{I})},$$

where $a \in \mathbb{F}_{q^m}[x;\sigma]$ satisfies $a(\alpha_j) = 0$ for all $j = 1,\dots,n$. In the division $a = \chi G + \varrho$, we must have $\varrho = 0$ since otherwise ϱ would be a polynomial of degree $< \deg G$ vanishing at all α_i, contradicting the minimality of G. Hence, we can write $[Q_0,\dots,Q_h] = \chi \boldsymbol{m}_0 + \sum_{i=1}^{h} Q_i \boldsymbol{m}_i$. □

Lemma 6.9 provides a description of all vectors satisfying the evaluation condition (6.5) of Problem 6.8. The following lemma shows how to find a vector in the row space of the matrix $\mathcal{M}^{(\mathrm{I})}$ satisfying also the degree constraints (6.6) and (6.7) using row reduction.

Lemma 6.10. *Consider an instance of Problem 6.8 satisfying* (6.8) *(i.e., the problem has a solution), and* $\mathcal{M}^{(\mathrm{I})}$ *as in* (6.9)*. Let* $\boldsymbol{w}^{(\mathrm{I})} = [0, k_1 - 1, \ldots, k_h - 1]$ *and* $\boldsymbol{V}$ *be a basis of* $\mathcal{M}^{(\mathrm{I})}$ *in* $\boldsymbol{w}^{(\mathrm{I})}$*-shifted weak Popov form. Then, a row* $\boldsymbol{v}$ *of* $\boldsymbol{V}$ *with minimal* $\boldsymbol{w}^{(\mathrm{I})}$*-shifted degree* $\deg \Phi_{\boldsymbol{w}^{(\mathrm{I})}}(\boldsymbol{v})$ *is a solution of Problem 6.8.*

Proof. Since (6.8) is fulfilled, the problem has a solution $\boldsymbol{u} = [Q_0, \ldots, Q_h] \in \mathcal{M}^{(\mathrm{I})}$ satisfying the degree constraints (6.6) and (6.7), which—by the choice of $\boldsymbol{w}^{(\mathrm{I})}$—can be re-written as

$$\deg(\Phi_{\boldsymbol{w}^{(\mathrm{I})}}(\boldsymbol{u})) = \max_i \{\deg(u_i) + {w^{(\mathrm{I})}}_i\} < n - \tau.$$

Let $t := \mathrm{LP}(\Phi_{\boldsymbol{w}^{(\mathrm{I})}}(\boldsymbol{u}))$ and $\boldsymbol{v}'$ be the row of $\boldsymbol{V}$ with leading position t. Any row $\boldsymbol{v}$ in $\boldsymbol{V}$ of minimal $\boldsymbol{w}^{(\mathrm{I})}$-shifted degree satisfies the degree constraints

$$\deg(\Phi_{\boldsymbol{w}^{(\mathrm{I})}}(\boldsymbol{v})) \leq \deg(\Phi_{\boldsymbol{w}^{(\mathrm{I})}}(\boldsymbol{v}')) \leq \deg \deg(\Phi_{\boldsymbol{w}^{(\mathrm{I})}}(\boldsymbol{u})) < n - \tau,$$

where the second inequality follows by Lemma 2.29. □

Theorem 6.11. *If Problem 6.8 satisfies* (6.6) *(i.e., has a solution), Algorithm 14 is correct and can be implemented using*

i) Algorithm 15 on page 118 in $O(h^3 n^2)$*,*

ii) Alg. 17 (page 125) in $O(h^3 \mathcal{M}_{q^m}(hn) \log(hn)) \subseteq O(h^{\min\{\frac{\omega+7}{2}, 4.635\}} n^{\min\{\frac{\omega+1}{2}, 1.635\}} \log(hn))$*,*

iii) and Algorithm 21 on page 132 in $O(h^2 n^2)$

operations over $\mathbb{F}_{q^m}$*.*

Algorithm 14: Solve Problem 6.8 (interpolation step) by Row Reduction

Input: Instance of Problem 6.8
Output: Solution $\boldsymbol{v} = [Q_0, \ldots, Q_h]$ of Problem 6.8

1 Set up $\boldsymbol{M}^{(\mathrm{I})}$ as in (6.10)
2 Compute $\boldsymbol{V}$ as a $\boldsymbol{w}^{(\mathrm{I})}$-shifted weak Popov form of $\boldsymbol{M}^{(\mathrm{I})}$, with $\boldsymbol{w}^{(\mathrm{I})}$ as in Lemma 6.10
3 **return** *a row* $\boldsymbol{v}$ *of* $\boldsymbol{V}$ *having minimal* $\boldsymbol{w}^{(\mathrm{I})}$*-shifted degree* $\deg(\Phi_{\boldsymbol{w}^{(\mathrm{I})}}(\boldsymbol{v}))$*.*

Proof. Correctness is clear by Lemma 6.10. Again, a $\boldsymbol{w}^{(\mathrm{I})}$-shifted weak Popov form is obtained by transforming $\Phi_{\boldsymbol{w}^{(\mathrm{I})}}(\boldsymbol{M}^{(\mathrm{I})})$ into wPf using the algorithms in Section 6.2 and Section 6.3, and then applying $\Phi^{-1}_{\boldsymbol{w}^{(\mathrm{I})}}$ to it. The complexities follow by Corollary 6.29 in Section 6.2.1, Corollary 6.44 in Section 6.2.3, and Theorem 6.50 in Section 6.3.2, respectively. □

6.2 Row Reduction of Skew Polynomial Matrices

6.2.1 The Mulders–Storjohann Algorithm over Skew Polynomials

In this section, we introduce an algorithm for obtaining a row-reduced basis of a left $\mathbb{F}_{q^m}[x;\sigma]$-module $\mathcal{V} \subseteq \mathbb{F}_{q^m}[x;\sigma]^r$ spanned by the rows of a full-rank square matrix in $\mathrm{GL}_r(\mathbb{F}_{q^m}[x;\sigma])$. The algorithm is an $\mathbb{F}_{q^m}[x;\sigma]$-variant of the Mulders–Storjohann algorithm [MS03] for row reduction of $\mathbb{F}_{q^m}[x]$-matrices. By taking care of the non-commutativity, the algorithm and its

correctness proof translate to the $\mathbb{F}_{q^m}[x;\sigma]$-case with only a few modifications. The complexity analysis, however, is significantly more difficult due to the non-existence of a determinant of $\mathbb{F}_{q^m}[x;\sigma]$-matrices, cf. Section 6.2.2. The algorithm consists of the elementary operation given by the following definition.

Definition 6.12. A *simple transformation i on j at position h* on a matrix $\boldsymbol{V} \in \mathrm{GL}_r(\mathbb{F}_{q^m}[x;\sigma])$ with $\deg v_{i,h} \leq \deg v_{j,h}$ replaces the row $\boldsymbol{v}_j$ by $\boldsymbol{v}_j - \alpha x^\delta \boldsymbol{v}_i$, where $\delta = \deg v_{j,h} - \deg v_{i,h}$ and $\alpha = \mathrm{LC}(v_{j,h})/\sigma^\delta(\mathrm{LC}(v_{i,h}))$. For two rows $\boldsymbol{v}_i$ and $\boldsymbol{v}_j$ with $\mathrm{LP}(\boldsymbol{v}_i) = \mathrm{LP}(\boldsymbol{v}_j)$, a *simple LP-transformation i on j* is a simple transformation i on j at position $\mathrm{LP}(\boldsymbol{v}_i)$.

Remark 6.13. *A simple transformation i on j at position h cancels the leading term of the skew polynomial $v_{j,h}$. In general, a simple transformation is a sequence of elementary row operations, so it neither changes the row space nor the rank of a matrix.*

The Mulders–Storjohann algorithm, Algorithm 15, applies simple LP-transformations to a matrix until all leading positions are different. We prove its correctness below.

Algorithm 15: Mulders–Storjohann Algorithm for $\mathbb{F}_{q^m}[x;\sigma]$-Matrices

Input: A matrix $\boldsymbol{V} \in \mathrm{GL}_r(\mathbb{F}_{q^m}[x;\sigma])$, whose rows span a module $\mathcal{V} \subseteq \mathbb{F}_{q^m}[x;\sigma]^r$.
Output: A basis of $\mathcal{V}$ in weak Popov form.

```
while there are two rows v_i and v_j of V with the same leading position do
    Apply a simple LP-transformation i on j
return V
```

The following tool enables us to prove that only finitely many simple LP-transformations are needed to transform a basis into weak Popov form.

Definition 6.14. The *value function* of a vector in $\mathbb{F}_{q^m}[x;\sigma]^r$ is given by

$$\begin{aligned}\psi : \mathbb{F}_{q^m}[x;\sigma]^r &\to \mathbb{N}_0 \\ \boldsymbol{v} &\mapsto \begin{cases} 0, & \text{if } \boldsymbol{v} = \boldsymbol{0}, \\ r \deg \boldsymbol{v} + \mathrm{LP}(\boldsymbol{v}) + 1, & \text{otherwise.} \end{cases}\end{aligned}.$$

The *sum value* of a matrix in $\mathbb{F}_{q^m}[x;\sigma]^{r\times r}$ is the sum of the rows' values.

Lemma 6.15. *Let $\boldsymbol{V} \in \mathrm{GL}_r(\mathbb{F}_{q^m}[x;\sigma])$. Consider a simple LP-transformation i on j, where $\boldsymbol{v}_j$ is replaced by $\boldsymbol{v}'_j$. Then, $\psi(\boldsymbol{v}'_j) < \psi(\boldsymbol{v}_j)$.*

Proof. The proof resembles the one of [Nie13a, Lemma 8], but we include it for completeness. We must have $\deg \boldsymbol{v}'_j \leq \deg \boldsymbol{v}_j$ since $\deg \boldsymbol{v}_j = \deg(\alpha x^\delta \boldsymbol{v}_i)$, where α and δ are chosen as in Definition 6.12. The entries of $\boldsymbol{v}_j$ and $\alpha x^\delta \boldsymbol{v}_i$ right of the leading position $h = \mathrm{LP}(\boldsymbol{v}_j)$ have degree $< \deg \boldsymbol{v}_j$ and so is their sum. The simple LP-transformation ensures that $\deg v_{jh} < \deg v'_{jh}$, which implies $\mathrm{LP}(\boldsymbol{v}'_j) < \mathrm{LP}(\boldsymbol{v}_j)$ or $\deg \boldsymbol{v}'_j < \deg \boldsymbol{v}_j$, proving the claim. □

Theorem 6.16. *Algorithm 15 is correct.*

Proof. The values $\psi(\boldsymbol{v}_i)$ of the rows $\boldsymbol{v}_i$ of the input matrix $\boldsymbol{V}$ are finite. Since the sum of the row values decreases with each simple LP-transformation (cf. Lemma 6.15) and cannot become negative by definition, the algorithm terminates. At termination, all rows have distinct leading positions, i.e., $\boldsymbol{V}$ is in weak Popov form. □

Remark 6.17. *The proof of Theorem 6.16 implies a rough complexity estimate for Algorithm 15: The sum value of $\boldsymbol{V}$ is in $O(r \deg \boldsymbol{V})$ and each simple LP-transformation costs $O(r \operatorname{maxdeg} \boldsymbol{V})$ operations over $\mathbb{F}_{q^m}$, so the total complexity is at most $O(r^2 \deg \boldsymbol{V} \operatorname{maxdeg} \boldsymbol{V})$.*

There might be several possibilities for simple LP-transformations during the execution of Algorithm 15. Theorem 6.16 shows that any choice yields a basis in weak Popov form.

Shifted Weak Popov Form

We can use Algorithm 15 to transform a matrix $\boldsymbol{V} \in \mathrm{GL}_r(\mathbb{F}_{q^m}[x;\sigma])$ into $\boldsymbol{w}$-shifted weak Popov form for some $\boldsymbol{w} \in \mathbb{N}_0^r$. One simply applies the algorithm to the shifted basis $\Phi_{\boldsymbol{w}}(\boldsymbol{V})$ and then inverts $\Phi_{\boldsymbol{w}}$ on the result. Since all elements in the i^{th} column of $\Phi_{\boldsymbol{w}}(\boldsymbol{V})$ are in the left module $\mathbb{F}_{q^m}[x;\sigma]x^{w_i}$, simple LP-transformations, which are elementary row operations, result again in entries in $\mathbb{F}_{q^m}[x;\sigma]x^{w_i}$, so the inversion is possible.

The strategy works for any algorithm that is based on simple LP-transformations, in particular all row reduction algorithms in the remainder of this chapter.

6.2.2 Orthogonality Defect and Cost of the Mulders–Storjohann Algorithm

The complexity analysis of the Mulder–Storjohann algorithm over $\mathbb{F}_{q^m}[x;\sigma]$ is more involved than in the commutative case. We use similar arguments as in [Nie13a], which proceeds analogous to the original paper [MS03], but finer-grained with the help of the so-called *orthogonality defect*, which was introduced by Lenstra in [Len85] for measuring how far a matrix is from being in weak Popov form. The measure is defined using the degree of the determinant of an $\mathbb{F}_{q^m}[x]$-matrix.

Over $\mathbb{F}_{q^m}[x;\sigma]$, a determinant in the classical sense does not exist due to non-commutativity. We therefore use the so-called *Dieudonné determinant* for $\mathbb{F}_{q^m}[x;\sigma]$-matrices and show that the resulting notion of orthogonality defect behaves exactly as in the $\mathbb{F}_{q^m}[x]$-case.

The Determinant Degree and Orthogonality Defect

The Dieudonné determinant is a generalization of the commutative determinant function to square matrices over a skew field, cf. [Die43] or [Dra83, § 20]. It can be used for $\mathbb{F}_{q^m}[x;\sigma]$-matrices by embedding $\mathbb{F}_{q^m}[x;\sigma]$ into its left field of fractions. For our purposes, we bypass the technical definition of this generalized determinant and only consider the following observations by Taelman [Tae06] about its degree.

Lemma 6.18 ([Tae06])**.** *There is a unique function* $\deg\det : \mathbb{F}_{q^m}[x;\sigma]^{r\times r} \to \mathbb{N}_0 \cup \{-\infty\}$ *s.t.:*

- $\deg\det(\boldsymbol{AB}) = \deg\det(\boldsymbol{A}) + \deg\det(\boldsymbol{B})$ *for all* $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{F}_{q^m}[x;\sigma]^{r\times r}$.
- $\deg\det \boldsymbol{A} = 0$ *for all* $\boldsymbol{A} \in \mathrm{GL}_r(\mathbb{F}_{q^m}[x;\sigma])$.
- *If* $\boldsymbol{A}$ *is a diagonal matrix with diagonal entries* $d_0, \dots, d_{m-1}$*, then* $\deg\det \boldsymbol{A} = \sum_i \deg d_i$.

Definition 6.19. We call the function $\deg\det(\cdot)$ of Lemma 6.18 the *determinant degree*.

The following properties of the determinant degree are immediate consequences of the statements above.

Corollary 6.20. *For any $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{F}_{q^m}[x;\sigma]^{r\times r}$, the determinant degree satisfies:*

- *If $\boldsymbol{B}$ is obtained from $\boldsymbol{A}$ by elementary row operations, then* $\deg\det \boldsymbol{B} = \deg\det \boldsymbol{A}$.
- *If $\boldsymbol{B}$ is obtained from $\boldsymbol{A}$ by multiplying one row with a non-zero skew polynomial $f \in \mathbb{F}_{q^m}[x;\sigma]^*$ from the left, then* $\deg\det \boldsymbol{B} = \deg f + \deg\det \boldsymbol{A}$.
- *If $\boldsymbol{A}$ is triangular with diagonal entries $d_0, \dots, d_{m-1}$, then* $\deg\det \boldsymbol{A} = \sum_i \deg d_i$.
- *For any shift $\boldsymbol{w} \in \mathbb{N}_0^r$, we have* $\deg\det(\Phi_{\boldsymbol{w}}(\boldsymbol{A})) = \deg\det(\boldsymbol{A}) + \sum_i w_i$.

The description of $\deg\det(\cdot)$ above is not constructive. However, it does allow us to directly compute the determinant degree for certain special cases, such as the following example.

Example 6.21. *In Algorithm 13, we need to row-reduce a matrix $\Phi_{\boldsymbol{w}^{(\mathrm{M})}}(\boldsymbol{M}^{(\mathrm{M})})$, where $\boldsymbol{M}^{(\mathrm{M})}$ is of the form (6.4). For an interleaved Gabidulin code of parameters $n = 100$, $h = 2$, $k_1 = 58$, $k_2 = 31$, exemplary entries of $\boldsymbol{M}^{(\mathrm{M})}$ could have degrees $\deg g_1 = \deg g_2 = 100$, $\deg r_1 = \deg r_2 = 99$, and the shifts would be chosen as $\boldsymbol{w}^{(\mathrm{M})} = (100, 42, 69)$ (cf. Section 6.1.1 or Table A.1). We obtain*

$$\deg\det \Phi_{\boldsymbol{w}^{(\mathrm{M})}}(\boldsymbol{M}^{(\mathrm{M})}) = \deg\det \begin{bmatrix} 1 & r_1 & r_2 \\ & g_1 & \\ & & g_2 \end{bmatrix} \begin{bmatrix} x^{\gamma_0} & & \\ & x^{\gamma_1} & \\ & & x^{\gamma_2} \end{bmatrix} = \deg g_1 + \deg g_2 + \sum_i \gamma_i = 411$$

by Corollary 6.20. ∎

We use the following definition and observation to bound the complexity of the Mulders–Storjohann algorithm over $\mathbb{F}_{q^m}[x;\sigma]$. Lemma 6.23 and Corollary 6.24 are trivial statements over $\mathbb{F}_{q^m}[x]$, but more difficult to prove in the non-commutative case.

Definition 6.22. The orthogonality defect of $\boldsymbol{V} \in \mathbb{F}_{q^m}[x;\sigma]^{r\times r}$ is $\Delta(\boldsymbol{V}) = \deg \boldsymbol{V} - \deg\det \boldsymbol{V}$.

Lemma 6.23. *Let $\boldsymbol{V} \in \mathrm{GL}_r(\mathbb{F}_{q^m}[x;\sigma])$ be in weak Popov form. Then, $\Delta(\boldsymbol{V}) = 0$.*

Proof. Since $\boldsymbol{V}$ has full rank, all its rows are non-zero. W.l.o.g., we assume that the leading positions are on the diagonal since otherwise, we could re-order the rows of $\boldsymbol{V}$ without changing $\deg\det \boldsymbol{V}$ or $\deg \boldsymbol{V}$, cf. Corollary 6.20. Using elementary row operations, we construct a sequence of matrices $\boldsymbol{V} = \boldsymbol{V}^{(0)}, \boldsymbol{V}^{(1)}, \dots, \boldsymbol{V}^{(r-1)}$ such that the first i columns of $\boldsymbol{V}^{(i)}$ are zero below the diagonal and the leading positions, marked by ▒ here, stay on the diagonal, i.e.,

$$\boldsymbol{V}^{(i)} = \left[\begin{array}{ccc|ccc} v_{1,1}^{(i)} & \dots & v_{1,i}^{(i)} & v_{1,i+1}^{(i)} & \dots & v_{1,r}^{(i)} \\ & \ddots & \vdots & \vdots & \ddots & \vdots \\ & & v_{i,i}^{(i)} & v_{i,i+1}^{(i)} & \dots & v_{i,r}^{(i)} \\ \hline & & & v_{i+1,i+1}^{(i)} & \dots & v_{i+1,r} \\ & & & \vdots & \ddots & \vdots \\ & & & v_{r,i+1}^{(i)} & \dots & v_{r,r}^{(i)} \end{array}\right]$$

The row operations that we apply to $\boldsymbol{V}^{(i-1)}$ in order to obtain $\boldsymbol{V}^{(i)}$ are as follows. For $j \leq i$, we keep $\boldsymbol{v}_j^{(i)} = \boldsymbol{v}_j^{(i-1)}$. Let $j > i$ and denote by $p_{j,i} = a_{j,i} v_{i,i}^{(i-1)} = b_{j,i} v_{j,i}^{(i-1)} \in \mathbb{F}_{q^m}[x;\sigma]$ the right union of $v_{i,i}^{(i-1)}$ and $v_{j,i}^{(i-1)}$, where $a_{j,i}, b_{j,i} \in \mathbb{F}_{q^m}[x;\sigma]$ (cf. Section 2.3.1). We compute

$$\boldsymbol{v}_j^{(i)} = b_{j,i} \boldsymbol{v}_j^{(i-1)} - a_{j,i} \boldsymbol{v}_i^{(i-1)}.$$

By definition of the union, we have $v_{j,i}^{(i)} = 0$ and the leading position remains $\operatorname{LP}(\boldsymbol{v}_j^{(i)}) = j$ since $\deg(a_{j,i}\boldsymbol{v}_i^{(i-1)}) \leq \deg(b_{j,i}\boldsymbol{v}_j^{(i-1)})$ and $\operatorname{LP}(\deg(a_{j,i}\boldsymbol{v}_i^{(i-1)})) = i < j = \deg(b_{j,i}\boldsymbol{v}_j^{(i-1)})$. Since we are scaling j^{th} row by $b_{j,i}$, the degree and determinant degree of $\boldsymbol{V}^{(i)}$ are given by

$$\deg \boldsymbol{V}^{(i)} = \deg \boldsymbol{V}^{(i-1)} + \textstyle\sum_{j>i} b_{j,i} \quad \text{and} \quad \deg\det \boldsymbol{V}^{(i)} = \deg\det \boldsymbol{V}^{(i-1)} + \textstyle\sum_{j>i} b_{j,i},$$

respectively (cf. Corollary 6.20). Thus, inductively, $\boldsymbol{V}^{(r-1)}$ is an upper triangular matrix and

$$\begin{aligned}\deg\det \boldsymbol{V} &= \deg\det \boldsymbol{V}^{(r-1)} - \textstyle\sum_i \sum_{j>i} b_{j,i} = \sum_i \deg v_{i,i}^{(r-1)} - \sum_i \sum_{j>i} b_{j,i} \\ &= \deg \boldsymbol{V}^{(r-1)} - \textstyle\sum_i \sum_{j>i} b_{j,i} = \deg \boldsymbol{V},\end{aligned}$$

which proves the claim. □

Corollary 6.24. *Let $\boldsymbol{V} \in \mathrm{GL}_r(\mathbb{F}_{q^m}[x;\sigma])$. Then, $\deg\det \boldsymbol{V} \leq \deg \boldsymbol{V}$ and $\Delta(\boldsymbol{V}) \geq 0$.*

Proof. Simple LP-transformations on $\boldsymbol{V}$ neither increase the degree of any row nor change the determinant degree of $\boldsymbol{V}$. By Lemma 6.23, Algorithm 15 therefore computes a matrix $\boldsymbol{U}$ with $\deg\det \boldsymbol{V} = \deg\det \boldsymbol{U} = \deg \boldsymbol{U} \leq \deg \boldsymbol{V}$, which proves the claim. □

Remark 6.25. *Lemma 6.23 implies an algorithm for computing $\deg\det \boldsymbol{V}$ of any full-rank matrix $\boldsymbol{V}$: If $\boldsymbol{V}$ is transformed into weak Popov form using Algorithm 15, then $\deg\det(\boldsymbol{V})$ equals the degree of the result since $\deg\det(\cdot)$ is invariant under simple LP-transformations.*

Complexity of the Algorithm

Using the tools developed above, we can bound the complexity of Algorithm 15 analogously to [Nie13a] for the $\mathbb{F}_{q^m}[x]$ case.

Theorem 6.26. *Let $\boldsymbol{V} \in \mathrm{GL}_r(\mathbb{F}_{q^m}[x;\sigma])$ with $r \in O(\Delta(\boldsymbol{V}))$. Then, Algorithm 15 with input matrix $\boldsymbol{V}$ performs at most $r\big(\Delta(\boldsymbol{V})+r\big)$ simple LP-transformations and has complexity $O(r^2 \Delta(\boldsymbol{V}) \operatorname{maxdeg}(\boldsymbol{V}))$ in operations over $\mathbb{F}_{q^m}$.*

Proof. Let $\boldsymbol{U}$ be the output of the algorithm. By Lemma 6.15, every simple LP-transformation reduces the sum of the ψ-values of a matrix, so their number is upper-bounded by

$$\begin{aligned}\textstyle\sum_{i=0}^{r-1} \psi(\boldsymbol{v}_i) - \sum_{i=0}^{r} \psi(\boldsymbol{u}_i) &= \textstyle\sum_{i=0}^{r-1} \big[r \deg \boldsymbol{v}_i + \operatorname{LP}(\boldsymbol{v}_i) - (r \deg \boldsymbol{u}_i + \operatorname{LP}(\boldsymbol{u}_i))\big] \\ &\leq r \textstyle\sum_{i=0}^{r-1} \big[\deg \boldsymbol{v}_i - \deg \boldsymbol{u}_i + 1\big] = r\big[\deg \boldsymbol{V} - \deg \boldsymbol{U} + r\big] \\ &= r\big[\Delta(\boldsymbol{V}) + r\big],\end{aligned}$$

where the last equality uses $\deg \boldsymbol{U} = \deg\det \boldsymbol{U}$ from Lemma 6.23 and the invariance of $\deg\det$ under elementary row operations.

Each simple LP-transformation consists of calculating $\boldsymbol{v}_j - \alpha x^\beta \boldsymbol{v}_i$, so we must apply an automorphism σ^β to the at most $r \operatorname{maxdeg}(\boldsymbol{V})$ many coefficients of $\boldsymbol{v}_i$, multiply them with α from the left, and subtract $\boldsymbol{v}_i$ from $\boldsymbol{v}_j$, which costs $O(r \operatorname{maxdeg}(\boldsymbol{V}))$ operations over $\mathbb{F}_{q^m}$. □

Remark 6.27. *Due to $\Delta(\boldsymbol{V}) \leq \deg \boldsymbol{V}$, the cost bound in Theorem 6.26 is at least as good as the rough complexity estimate in Remark 6.17. Corollary 6.28 below shows that it can improve upon the latter (which amounts to $O(h^3\mu^2)$ in this case).*

Implications on the Cost of the Decoding Problems

The matrices used for decoding interleaved Gabidulin codes in Section 6.1 can be transformed into weak Popov form using Algorithm 15 with the following costs.

Corollary 6.28. *Consider an instance of Problem 6.3 (MgLSSR). Let $\boldsymbol{M}^{(\mathrm{M})}$ be as in (6.4) and choose $\boldsymbol{w}^{(\mathrm{M})}$ as in 6.5. The orthogonality defect of $\Phi_{\boldsymbol{w}^{(\mathrm{M})}}(\boldsymbol{M}^{(\mathrm{M})})$ is $\Delta(\Phi_{\boldsymbol{w}^{(\mathrm{M})}}(\boldsymbol{M}^{(\mathrm{M})})) \leq \mu$ and we can transform $\Phi_{\boldsymbol{w}^{(\mathrm{M})}}(\boldsymbol{M}^{(\mathrm{M})})$ into weak Popov form in $O(h^2\mu^2)$ operations over $\mathbb{F}_{q^m}$ using Algorithm 15.*

Proof. The matrix $\Phi_{\boldsymbol{w}^{(\mathrm{M})}}(\boldsymbol{M}^{(\mathrm{M})})$ is of the form

$$\Phi_{\boldsymbol{w}^{(\mathrm{M})}}(\boldsymbol{M}^{(\mathrm{M})}) = \begin{bmatrix} x^{\gamma_0} & r_1 x^{\gamma_1} & r_2 x^{\gamma_2} & \dots & r_h x^{\gamma_h} \\ & g_1 x^{\gamma_1} & & & \\ & & g_2 x^{\gamma_2} & & \\ & & & \ddots & \\ & & & & g_h x^{\gamma_h} \end{bmatrix}. \tag{6.11}$$

W.l.o.g., we can assume that $\gamma_0 \leq \max_{i\geq 1}\{\gamma_i + \deg r_i\} \leq \mu$ since otherwise $\boldsymbol{M}^{(\mathrm{M})}$ is already in $\boldsymbol{w}^{(\mathrm{M})}$-shifted weak Popov form. Since $\Phi_{\boldsymbol{w}^{(\mathrm{M})}}(\boldsymbol{M}^{(\mathrm{M})})$ is in triangular form, its determinant degree is

$$\deg\det \Phi_{\boldsymbol{w}^{(\mathrm{M})}}(\boldsymbol{M}^{(\mathrm{M})}) = \textstyle\sum_{i=1}^{h} \deg g_i + \sum_{i=0}^{h} \gamma_i \ .$$

Let $\boldsymbol{m}_i$ be the i^{th} row of $\boldsymbol{M}^{(\mathrm{M})}$. The degree of the matrix is given by

$$\begin{aligned} \deg \Phi_{\boldsymbol{w}^{(\mathrm{M})}}(\boldsymbol{M}^{(\mathrm{M})}) &= \textstyle\sum_{i=0}^{h} \deg \Phi_{\boldsymbol{w}^{(\mathrm{M})}}(\boldsymbol{m}_i) = \max_i\{\gamma_i + \deg s_i\} + \sum_{i=1}^{h} (\gamma_i + \deg g_i) \\ &\leq \mu + \deg\det \Phi_{\boldsymbol{w}^{(\mathrm{M})}}(\boldsymbol{M}^{(\mathrm{M})}), \end{aligned}$$

so the orthogonality defect is upper-bounded by $\Delta(\Phi_{\boldsymbol{w}^{(\mathrm{M})}}(\boldsymbol{M}^{(\mathrm{M})})) \leq \mu$. By Theorem 6.26, Algorithm 15 row-reduces the matrix in $O(h^2\mu^2)$ operations over $\mathbb{F}_{q^m}$. □

Corollary 6.29. *Consider an instance of Problem 6.8. Let $\boldsymbol{M}^{(\mathrm{I})}$ be as in (6.10) and $\boldsymbol{w}^{(\mathrm{I})}$ be as in Lemma 6.10. Then, the orthogonality defect of $\Phi_{\boldsymbol{w}^{(\mathrm{I})}}(\boldsymbol{M}^{(\mathrm{I})})$ is upper-bounded by $\Delta(\Phi_{\boldsymbol{w}^{(\mathrm{I})}}(\boldsymbol{M}^{(\mathrm{I})})) \leq hn$ and we can row-reduce the matrix $\Phi_{\boldsymbol{w}^{(\mathrm{I})}}(\boldsymbol{M}^{(\mathrm{I})})$ in $O(h^3\mu^2)$ time using Algorithm 15.*

Proof. The degrees of the non-zero entries of $\Phi_{\boldsymbol{w}^{(\mathrm{I})}}(\boldsymbol{M}^{(\mathrm{I})})$ are component-wise given by

$$\begin{bmatrix} n & & & & \\ < n & k_1 - 1 & & & \\ < n & & k_2 - 1 & & \\ \vdots & & & \ddots & \\ < n & & & & k_h - 1 \end{bmatrix}$$

Thus, $\Delta(\Phi_{\boldsymbol{w}^{(\mathrm{I})}}(\boldsymbol{M}^{(\mathrm{I})})) \leq (h+1)n - n - \sum_{i=1}^{h}(k_i - 1) \leq hn$. Using Theorem 6.26 and $\operatorname{maxdeg}(\Phi_{\boldsymbol{w}^{(\mathrm{I})}}(\boldsymbol{M}^{(\mathrm{I})})) = n$, the complexity in operations over $\mathbb{F}_{q^m}$ becomes $O(h^3n^2)$. □

6.2.3 A Divide-&-Conquer Variant: Alekhnovich's Algorithm

Over ordinary polynomial rings, Alekhnovich [Ale05] proposed an algorithm for fast row reduction based on efficient $\mathbb{F}_{q^m}[x]$-multiplication algorithms. The algorithm can be seen as a generalization of Knuth–Schönhage's GCD algorithm (cf. [Ale05]) or as a divide-&-conquer variant of the Mulders–Storjohann algorithm (cf. [Nie13b]).

In this section, we show that—by taking care of non-commutativity—Alekhnovich's algorithm can be adapted for row reduction of skew polynomial matrices. The method relies on the fast $\mathbb{F}_{q^m}[x;\sigma]$-multiplication algorithm in Section 5.2.1. Similar to [Ale05], we use the following notation.

Definition 6.30. Let $v \in \mathbb{F}_{q^m}[x;\sigma]$, $\boldsymbol{v} \in \mathbb{F}_{q^m}[x;\sigma]^r$, $\boldsymbol{V} \in \mathbb{F}_{q^m}[x;\sigma]^{r\times r}$, and $t \in \mathbb{N}_0$. The *length* $|v|$ of v is defined by

$$|v| := \begin{cases} 0, & \text{if } v = 0, \\ \deg v - \max\{i \in \mathbb{N}_0 \,:\, v_j = 0 \;\forall\, 0 \le j \le i\}, & \text{else.} \end{cases}$$

Similarly, the length of a vector $\boldsymbol{v}$ is $|\boldsymbol{v}| = \max_i\{\deg \boldsymbol{v} - \deg v_i + |v|_i\}$ and the length of a matrix $\boldsymbol{V}$ is $|\boldsymbol{V}| = \max_i\{|\boldsymbol{v}_i|\}$.

We define the *accuracy approximation to depth* t of v as $v|_t := \sum_{i=\deg v - t+1}^{\deg v} v_i x^i$, of $\boldsymbol{v}$ as

$$\boldsymbol{v}|_t = \left[v_1|_{\min\{0,t-(\deg \boldsymbol{v}-\deg v_1)\}}, \ldots, v_r|_{\min\{0,t-(\deg \boldsymbol{v}-\deg v_r)\}}\right],$$

and extend it row-wise to matrices $\boldsymbol{V}$.

Example 6.31. *For instance, we have*

$$\boldsymbol{V} = \begin{bmatrix} x^2+x & x^2+1 \\ x^4 & x^3+x^2+x+1 \end{bmatrix},\ \boldsymbol{V}|_2 = \begin{bmatrix} x^2+x & x^2 \\ x^4 & x^3 \end{bmatrix},\ \textit{and}\ \boldsymbol{V}|_1 = \begin{bmatrix} x^2 & x^2 \\ x^4 & 0 \end{bmatrix}.$$

Note that the accuracy approximation of a polynomial, vector, or matrix to depth t always results in a length at most t, i.e., $|a|_t| \le t$, $|\boldsymbol{v}|_t| \le t$, and $|\boldsymbol{V}|_t| \le t$. The algorithm uses the following observation that two matrices containing polynomials of small length can be multiplied efficiently although their degrees might be large.

Lemma 6.32. *Let $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{F}_{q^m}[x;\sigma]^{r\times r}$ such that each entry of the two matrices has length at most t. The multiplication $\boldsymbol{A}\cdot\boldsymbol{B}$ can be implemented in $O(r^3\mathcal{M}_{q^m}(t))$ time.*

Proof. Consider two polynomials $a, b \in \mathbb{F}_{q^m}[x;\sigma]$ of length $\le t$. There are $\widetilde{a}, \widetilde{b} \in \mathbb{F}_{q^m}[x;\sigma]$ of degree at most t such that $a = \widetilde{a}x^{\deg a - |a|+1}$ and $b = \widetilde{b}x^{\deg b - |b|+1}$. Thus, the multiplication $a\cdot b$ can be re-written as

$$a\cdot b = \left[\widetilde{a}\cdot\sigma^{\deg a-|a|+1}\big(\widetilde{b}\big)\right] x^{\deg a+\deg a-|a|-|b|+1}$$

and implemented using at most t automorphism computations, which is negligible, and a multiplication of two polynomials of degree t, costing $O(\mathcal{M}_{q^m}(t))$. Naive multiplication[3] of $\boldsymbol{A}$ by $\boldsymbol{B}$ consists of r^3 many multiplications as above, which cost $O(r^3\mathcal{M}_{q^m}(t))$, plus r^3 additions of polynomials of length $\le 2t$, which is negligible in comparison. □

[3] In faster matrix multiplication algorithms, the length of polynomials in intermediate computations might be much larger than t. Thus, we compute it naively in cubic time.

Similar the original paper [Ale05], the final algorithm (see Algorithm 17 below) is recursive and calls a base case given by Algorithm 16. We first establish the correctness and complexity of the latter one.

Lemma 6.33. *Algorithm 16 is correct and returns a matrix $\boldsymbol{U}$ of the form $U_{ij} = u_{ij}x^{d_{ij}}$, where $u_{ij} \in \mathbb{F}_{q^m}$ and $d_{ij} = \deg \boldsymbol{v}_i - \deg \boldsymbol{v}_j$. If $|\boldsymbol{V}| \leq 1$, the algorithm costs $O(r^3)$ operations over $\mathbb{F}_{q^m}$.*

Algorithm 16: $\mathrm{R}(\boldsymbol{V})$

Input: Basis $\boldsymbol{V} \in \mathrm{GL}_r(\mathbb{F}_{q^m}[x;\sigma])$ of a module $\mathcal{V} \subseteq \mathbb{F}_{q^m}[x;\sigma]^r$ of degree η
Output: $\boldsymbol{U} \in \mathbb{F}_{q^m}[x;\sigma]^{r\times r}$ such that $\boldsymbol{U}\cdot\boldsymbol{V}$ is in wPf or $\deg(\boldsymbol{U}\cdot\boldsymbol{V}) \leq \deg \boldsymbol{V} - 1$

1 $\boldsymbol{U} \leftarrow \boldsymbol{I}$
2 **while** $\deg \boldsymbol{V} = \eta$ *and* $\boldsymbol{V}$ *is not in wPf* **do**
3 Find i, j such that $\mathrm{LP}(\boldsymbol{v}_i) = \mathrm{LP}(\boldsymbol{v}_j)$ and $\deg \boldsymbol{v}_i \geq \deg \boldsymbol{v}_j$
4 $\delta \leftarrow \deg \boldsymbol{v}_i - \deg \boldsymbol{v}_j$
5 $\alpha \leftarrow \mathrm{LC}(\mathrm{LT}(\boldsymbol{v}_i))/\sigma^\delta(\mathrm{LC}(\mathrm{LT}(\boldsymbol{v}_j)))$
6 $\boldsymbol{U} \leftarrow (\boldsymbol{I} - \alpha x^\delta \boldsymbol{E}_{i,j})\cdot\boldsymbol{U}$ and $\boldsymbol{V} \leftarrow (\boldsymbol{I} - \alpha x^\delta \boldsymbol{E}_{i,j})\cdot\boldsymbol{V}$
7 **return** $\boldsymbol{U}$

Proof. Inside the while loop, the algorithm performs a simple LP-transformation on $\boldsymbol{V}$, which can be written as a multiplication of $(\boldsymbol{I} - \alpha x^\delta \boldsymbol{E}_{i,j})$ from the left. By multiplying these matrices $(\boldsymbol{I} - \alpha x^\delta \boldsymbol{E}_{i,j})$ in the correct order, one obtains the matrix $\boldsymbol{U}$ such that $\boldsymbol{U}\boldsymbol{V}$ equals the result of the sequence of simple LP-transformations on $\boldsymbol{V}$.

The matrix $\boldsymbol{U}$ is of the claimed form during the entire algorithm. This is obviously fulfilled at the start since $\boldsymbol{U} = \boldsymbol{I}$. As long as the condition of the while loop is fulfilled, the row degrees of $\boldsymbol{V}$ are constant. Hence, $\delta = \deg \boldsymbol{v}_i - \deg \boldsymbol{v}_j$ and Line 6 corresponds to subtracting the row $\alpha x^{\deg \boldsymbol{v}_i - \deg \boldsymbol{v}_j}\boldsymbol{v}_j$ from $\boldsymbol{v}_i$, where μ^{th} entry of the first vector is of the form

$$\alpha x^{\deg \boldsymbol{v}_i - \deg \boldsymbol{v}_j} u_{j\mu} x^{\deg \boldsymbol{v}_j - \deg \boldsymbol{v}_\mu} = \alpha\sigma^{\deg \boldsymbol{v}_i - \deg \boldsymbol{v}_j}(u_{j\mu}) x^{\deg \boldsymbol{v}_i - \deg \boldsymbol{v}_\mu} =: u'_{i\mu} x^{\deg \boldsymbol{v}_i - \deg \boldsymbol{v}_\mu},$$

so the matrix' structure is preserved and updating $\boldsymbol{U}$ costs $O(r)$ operations over $\mathbb{F}_{q^m}$.

Since the degree of a row $\boldsymbol{v}_i$ is decreased after at most r simple LP-transformations on it, the while loop runs at most r^2 iterations. If $|(\boldsymbol{V})| \leq 1$, each row of $\boldsymbol{V}$ consist of monomials of the same degree, but possibly different coefficient. Thus, each simple LP-transformation, i.e., computing $\boldsymbol{V}$ as in Line 6, consists of at most r many automorphism computations, scalar multiplications and additions, respectively, and does not change length of the matrix. Together with the cost of updating $\boldsymbol{U}$, we obtain the claimed complexity. □

Remark 6.34. *The original algorithm [Ale05] over $\mathbb{F}_{q^m}[x]$ operates on the columns of $\boldsymbol{V}$ by multiplying the returned matrix $\boldsymbol{U}$ from the right. In contrast, it is necessary to consider the "transposed" problem (operating on rows) here since we are dealing with left $\mathbb{F}_{q^m}[x;\sigma]$-modules.*

We can apply Algorithm 16 iteratively on a matrix $\boldsymbol{V}$ in order to decrease its degree by at least $t \in \mathbb{N}$ or transform it into weak Popov form. We call this algorithm $\mathrm{R}(\boldsymbol{V}, t)$. As in [Ale05], we speed up $\mathrm{R}(\boldsymbol{V}, t)$ in a divide-&-conquer fashion using two main observations:

- We can recursively compute $\mathrm{R}(\boldsymbol{V}, t)$ by two calls of the algorithm with input size $t' \approx \frac{t}{2}$.
- The involved matrices can be (accuracy) approximated without loss, so re-assembling the results of the recursive calls (= matrix multiplications) has small cost.

We formalize these observations in the following lemmata.

Lemma 6.35. *Let $\boldsymbol{V} \in \mathrm{GL}_r(\mathbb{F}_{q^m}[x;\sigma])$, $t, t' \in \mathbb{N}$, $t' < t$, and $\boldsymbol{U} = \mathrm{R}(\boldsymbol{V}, t')$. Then,*

$$\mathrm{R}(\boldsymbol{V}, t) = \mathrm{R}\big[\boldsymbol{U} \cdot \boldsymbol{V}, t - (\deg \boldsymbol{V} - \deg(\boldsymbol{U} \cdot \boldsymbol{V}))\big] \cdot \boldsymbol{U}.$$

Proof. The matrix $\boldsymbol{U}$ reduces $\deg \boldsymbol{V}$ by at least t' or transforms $\boldsymbol{V}$ into wPf. Multiplication by $\mathrm{R}[\boldsymbol{U} \cdot \boldsymbol{V}, t - (\deg \boldsymbol{V} - \deg(\boldsymbol{U} \cdot \boldsymbol{V}))]$ further reduces the degree of $\boldsymbol{U} \cdot \boldsymbol{V}$ by $t - (\deg \boldsymbol{V} - \deg(\boldsymbol{U} \cdot \boldsymbol{V})) \geq t - t'$ (or $\boldsymbol{U} \cdot \boldsymbol{V}$ in wPf). □

Lemma 6.36. *Let $\boldsymbol{V} \in \mathrm{GL}_r(\mathbb{F}_{q^m}[x;\sigma])$ and $t \in \mathbb{N}$. Then, $\mathrm{R}(\boldsymbol{V}, t) = \mathrm{R}(\boldsymbol{V}|_t, t)$.*

Proof. The proof works exactly as the one of [Ale05, Lemma 2.7], but we give its idea for the sake of completeness. If a sequence of simple LP-transformations on the same rows is applied to two matrices $\boldsymbol{V}$ and $\boldsymbol{V}'$ whose top t coefficients agree (i.e., $\boldsymbol{V}|_t = \boldsymbol{V}'|_t$), then as long as the degree of the matrices does not drop by at least t, the coefficients of the simple LP-transformations are the same since they depend only on the top coefficients of the matrices. If after some transformations, modeled by $\boldsymbol{U}$, the degree is dropped by $t' < t$, then we have $(\boldsymbol{U}\boldsymbol{V})|_{t-t'} = (\boldsymbol{U}\boldsymbol{V}')|_{t-t'}$, so the claim follows by an inductive argument. □

Lemma 6.37. *Let $\boldsymbol{V} \in \mathrm{GL}_r(\mathbb{F}_{q^m}[x;\sigma])$ and $t \in \mathbb{N}$. The entries of $\boldsymbol{U} = \mathrm{R}(\boldsymbol{V}, t)$ are of the form $U_{ij} = \sum_{\mu=-t+1}^{t-1} u_{ij\mu} x^{d_{ij}+\mu}$, where $u_{ij\mu} \in \mathbb{F}_{q^m}$ and $d_{ij} := \deg \boldsymbol{v}_i - \deg \boldsymbol{v}_j$.*

Proof. The proof is again similar to [Ale05, Lemma 2.8]. We prove it by induction, where the case $t = 1$ is given by Lemma 6.33. If we choose $t' = t - 1$ in Lemma 6.35 and write $\boldsymbol{U} = \mathrm{R}(\boldsymbol{V})$, $\boldsymbol{V}' = \boldsymbol{U} \cdot \boldsymbol{V}$, and $\Delta t = \deg \boldsymbol{V} - \deg \boldsymbol{V}'$, we obtain

$$\mathrm{R}(\boldsymbol{V}, t) = \mathrm{R}(\boldsymbol{V}', t - \Delta t) \cdot \boldsymbol{U}$$

By induction hypothesis, all monomials in $\mathrm{R}(\boldsymbol{V}', t - \Delta t)$ have degree $d'_{ij} + \mu'$ with $|\mu'| \leq t - \Delta t - 1$. Also, $|d'_{ij} - d_{ij}| \leq \Delta t$ by definition. Since the entries of $\boldsymbol{U}$ are monomials of degree d_{ij}, we obtain the claimed degree by $|\mu| = |d'_{ij} + \mu' - d_{ij}| \leq |d'_{ij} - d_{ij}| + |\mu'| \leq t - 1$. □

Combining the ideas above, we obtain Algorithm 17, $\hat{\mathrm{R}}(\boldsymbol{V}, t)$, whose correctness and complexity is proved in the following theorem.

Theorem 6.38. *Algorithm 17 is correct and costs $O(r^3 \mathcal{M}_{q^m}(t))$ operations over $\mathbb{F}_{q^m}$.*

Algorithm 17: $\hat{\mathrm{R}}(\boldsymbol{V}, t)$

Input: Basis $\boldsymbol{V} \in \mathrm{GL}_r(\mathbb{F}_{q^m}[x;\sigma])$ of a module $\mathcal{V} \subseteq \mathbb{F}_{q^m}[x;\sigma]^r$ of degree η, and $t \in \mathbb{N}$
Output: $\boldsymbol{U} \in \mathbb{F}_{q^m}[x;\sigma]^{r \times r}$ such that $\boldsymbol{U} \cdot \boldsymbol{V}$ is in wPf or $\deg(\boldsymbol{U} \cdot \boldsymbol{V}) \leq \deg \boldsymbol{V} - t$

```
if t = 1 then
    return R(V|_1)                                   // O(r^3)
U_1 ← R̂(V|_t, ⌊t/2⌋)                                 // f(t/2) (cf. proof)
V_1 ← U_1 · V|_t                                     // O(r^3 M_{q^m}(t))
U_2 ← R̂(V_1, t − (deg V|_t − deg V_1))               // f(t/2) (cf. proof)
return U_2 · U_1                                     // O(r^3 M_{q^m}(t))
```

Proof. Correctness follows from $\mathrm{R}(\boldsymbol{V}, t) = \hat{\mathrm{R}}(\boldsymbol{V}, t)$ by induction, where the base case is given in Lemma 6.33. For $t > 1$, with $\hat{\boldsymbol{U}} = \hat{\mathrm{R}}(\boldsymbol{V}|_t, \lfloor \frac{t}{2} \rfloor)$ and $\boldsymbol{U} = \mathrm{R}(\boldsymbol{V}|_t, \lfloor \frac{t}{2} \rfloor)$, the claim follows by

$$\begin{aligned}\hat{\mathrm{R}}(\boldsymbol{V}, t) &= \hat{\mathrm{R}}(\hat{\boldsymbol{U}} \cdot \boldsymbol{V}|_t, t - (\deg \boldsymbol{V}|_t - \deg(\hat{\boldsymbol{U}} \cdot \boldsymbol{V}|_t))) \cdot \hat{\boldsymbol{U}} \\ &\overset{\text{(i)}}{=} \mathrm{R}(\boldsymbol{U} \cdot \boldsymbol{V}|_t, t - (\deg \boldsymbol{V}|_t - \deg(\boldsymbol{U} \cdot \boldsymbol{V}|_t))) \cdot \boldsymbol{U} \overset{\text{(ii)}}{=} \mathrm{R}(\boldsymbol{V}|_t, t) \overset{\text{(iii)}}{=} \mathrm{R}(\boldsymbol{V}, t),\end{aligned}$$

where (i) is the induction hypothesis, (ii) uses Lemma 6.35, and (iii) equals Lemma 6.36.

Algorithm 17 calls itself twice on inputs of sizes $\approx \frac{t}{2}$ and in addition, performs two matrix multiplications in Lines 4 and 6. Since the involved matrices contain polynomials of length $O(t)$ (cf. Lemma 6.37), these multiplications cost $O(r^3 \mathcal{M}_{q^m}(t))$ operations over $\mathbb{F}_{q^m}$ (cf. Lemma 6.32). Recursively, the cost bound $f(t)$ in operations over $\mathbb{F}_{q^m}$ reads

$$f(t) = 2 \cdot f(\tfrac{t}{2}) + O(r^3 \mathcal{M}_{q^m}(t)),$$

and can be resolved as $f(t) \in O(r^3 \mathcal{M}_{q^m}(t) \log(t))$ by the master theorem since the base case has complexity $O(r^3)$ (cf. Lemma 6.33). □

We have seen in Section 6.2.2 that the degree of any matrix $\boldsymbol{V}'$ in weak Popov form, obtained from a full-rank matrix $\boldsymbol{V}$ by a sequence of simple LP-transformations, is exactly $\deg \boldsymbol{V}' = \deg \boldsymbol{V} - \Delta(\boldsymbol{V})$. Hence, it suffices to choose $t = \Delta(\boldsymbol{V})$ in Algorithm 17 to ensure that the resulting matrix transforms $\boldsymbol{V}$ into weak Popov form. We formalize this in the main statement of this subsection:

Theorem 6.39. *Let $\boldsymbol{V} \in \mathrm{GL}_r(\mathbb{F}_{q^m}[x;\sigma])$ be a basis of a module $\mathcal{V} \subseteq \mathbb{F}_{q^m}[x;\sigma]^r$. Then, $\hat{\mathrm{R}}(\boldsymbol{V}, \Delta(\boldsymbol{V})) \cdot \boldsymbol{V}$ is a basis of $\mathcal{V}$ in weak Popov form. It can be computed in*

$$O(r^3 \mathcal{M}_{q^m}(\Delta(\boldsymbol{V})) \log(\Delta(\boldsymbol{V})))$$

operations over $\mathbb{F}_{q^m}$ using Algorithm 17.

Remark 6.40. *If the used multiplication algorithm is in $\Omega(t^{1+\varepsilon})$ for some $\varepsilon > 0$, e.g., the one in Section 5.2.1 with $\omega > 2$, we can omit the* log*-factors in the complexity expressions of Theorem 6.38 and Theorem 6.39.*

Implications on the Cost of the Decoding Problems

The matrices used for decoding interleaved Gabidulin codes in Section 6.1 can be transformed into weak Popov form using Algorithm 17 with the following costs.

Corollary 6.41. *Consider an instance of Problem 6.3 (MgLSSR). Let $\boldsymbol{M}^{(\mathrm{M})}$ be as in* (6.4) *and $\boldsymbol{w}^{(\mathrm{M})}$ be as in Lemma 6.5. Then, Algorithm 17 can row-reduce $\Phi_{\boldsymbol{w}^{(\mathrm{M})}}(\boldsymbol{M}^{(\mathrm{M})})$ in*

$$O(h^3 \mathcal{M}_{q^m}(\mu) \log(\mu)) \subseteq O(h^3 \mu^{\min\{\frac{\omega+1}{2}, 1.635\}} \log(\mu))$$

operations over $\mathbb{F}_{q^m}$, where $\mu := \max_i\{\gamma_i + \deg g_i\}$.

Proof. This follows from Theorem 6.39 and $\Delta(\Phi_{\boldsymbol{w}^{(\mathrm{M})}}(\boldsymbol{M}^{(\mathrm{M})})) \leq \mu$ (cf. Corollary 6.28). □

Remark 6.42. *In [SB14], Sidorenko and Bossert give an algorithm for solving the MgLSSR problem in the special case arising from syndrome-based key equations. In their paper, the interleaving parameter h is considered to be a constant and the cost is given as $O(\mathcal{M}_{q^m}(n)\log(n))$. If h is included in the complexity analysis, one obtains $O(h^3\mathcal{M}_{q^m}(n)\log(n))$.[4] Hence, the algorithm's cost bound equals the one from Corollary 6.41 in this case.*

Remark 6.43. *If we apply Algorithm 17 to the shifted basis $\Phi_{\boldsymbol{w}^{(\mathrm{M})}}(\boldsymbol{M}^{(\mathrm{M})})$ of Algorithm 13 in the special case $h = 1$ and $\sigma = \cdot^q$, we obtain an alternative to the* Linearized Extended Euclidean algorithm *(LEEA) from [WSB10] (see also [Wac13, Section 3.1.4]) of the same complexity. The algorithms are equivalent up to the implementation of a simple LP-transformation. This is not surprising since on 2×2 matrices, Alekhnovich's algorithm over $\mathbb{F}_{q^m}[x]$ is equivalent to the fast Knuth–Schönhage-like GCD algorithm introduced by Aho and Hopcroft [AH74] and Brent, Gustavson, and Yun [BGY80] (see also [GG99, Chapter 11]). The LEEA in [Wac13] is based on the latter one.*

Corollary 6.44. *Consider an instance of Problem 6.8 (interpolation step). Let $\boldsymbol{M}^{(\mathrm{I})}$ be as in* (6.10) *and $\boldsymbol{w}^{(\mathrm{I})}$ be as in Lemma 6.10. Then, Algorithm 17 can row-reduce $\Phi_{\boldsymbol{w}^{(\mathrm{I})}}(\boldsymbol{M}^{(\mathrm{I})})$ in*

$$O(h^3\mathcal{M}_{q^m}(hn)\log(hn)) \subseteq O(h^{\min\{\frac{\omega+7}{2},4.635\}}n^{\min\{\frac{\omega+1}{2},1.635\}}\log(hn))$$

operations over $\mathbb{F}_{q^m}$.

Proof. The claim follows from Theorem 6.39 using $\Delta(\Phi_{\boldsymbol{w}^{(\mathrm{I})}}(\boldsymbol{M}^{(\mathrm{I})})) \leq hn$ (cf. Corollary 6.29). □

6.3 Specialized Row Reduction Algorithms for the Decoding Problems

In this section, we present two algorithms for row reduction of skew polynomial matrices that are particularly fast on the matrices arising in the decoding problems given in Section 6.1. The first one is a skew polynomial variant of Rosenkilde's Demand-Driven algorithm and is fast for key-equation-based methods. The second procedure is inspired by Collart et al.'s Gröbner walk algorithm and efficiently row-reduces the module basis for interpolation-based decoding.

6.3.1 Key-Equation Based Decoding: Rosenkilde's Demand-Driven Algorithm

In the following, we adapt Rosenkilde's Demand-Driven algorithm [Nie13a] to skew polynomials. The algorithm achieves a speed-up compared to the Mulders–Storjohann algorithm (Algorithm 15) in case of the MgLSSR problem by utilizing the module's structure, which enables to remember only the first column of the module basis during a series of simple LP-transformations and to compute the remaining entries on demand. We prove the method using an auxiliary algorithm, which is given by Algorithm 18 below.

[4] Line 8 of [SB14, Algorithm 4] computes h many matrix-vector multiplications of dimension h, containing polynomials of length η. This can be implemented in $O(h^3\mathcal{M}_{q^m}(\eta))$ operations over $\mathbb{F}_{q^m}$ and—by the same arguments as in Lemma 6.32—cannot be easily accelerated. The arguments as in [SB14, Theorem 2] imply the given complexity.

Lemma 6.45. *Algorithm 18 is correct. The while loop is executed at most $O(h\mu)$ times.*

Algorithm 18: Auxiliary Algorithm

Input: $\boldsymbol{V} \leftarrow \Phi_{\boldsymbol{w}^{(\mathrm{M})}}(\boldsymbol{M}^{(\mathrm{M})})$ with $\boldsymbol{M}^{(\mathrm{M})}$ as in (6.4) and $\boldsymbol{w}^{(\mathrm{M})}$ as in Lemma 6.5
Output: Basis $\boldsymbol{V}'$ of $\mathcal{M}^{(\mathrm{M})}$ as in (6.3) in $\boldsymbol{w}^{(\mathrm{M})}$-shifted weak Popov form

1 $[\eta, h] \leftarrow [\deg \boldsymbol{v}_0, \mathrm{LP}(\boldsymbol{v}_0)]$
2 **while** $\deg v_{0,0} \leq \eta$ **do**
3 $\alpha \leftarrow$ coefficient of x^η in $v_{0,h}$
4 **if** $\alpha \neq 0$ **then**
5 $\eta_h \leftarrow \deg \boldsymbol{v}_h$
6 $\alpha_h \leftarrow$ coefficient of x^{η_h} in $v_{h,h}$
7 **if** $\eta < \eta_h$ **then**
8 swap $[\boldsymbol{v}_0, \alpha, \eta]$ and $[\boldsymbol{v}_h, \alpha_h, \eta_h]$
9 $\boldsymbol{v}_0 \leftarrow \boldsymbol{v}_0 - \alpha/\sigma^{\eta-\eta_h}(\alpha_h) x^{\eta-\eta_h} \boldsymbol{v}_h$
10 $[\eta, h] \leftarrow \begin{cases} [\eta, h-1], & \text{if } h > 1 \\ [\eta-1, h], & \text{else.} \end{cases}$
11 **return** $\Phi^{-1}_{\boldsymbol{w}^{(\mathrm{M})}}(\boldsymbol{V})$

Proof. Algorithm 18 performs a sequence of simple transformations h on 0 on $\boldsymbol{V}$, which are executed by Line 9. During the algorithm, we always have $\mathrm{LP}(v_i) = i$ for all $i \geq 0$. This is obviously fulfilled at the start by the structure of $\boldsymbol{M}^{(\mathrm{M})}$ as in (6.4), and ensured in the remainder of the execution by Line 8 that swaps two rows with the same leading position h in order to perform the simple transformation always on the 0^{th} row.

At the start of the algorithm, we have $\eta = \deg \boldsymbol{v}_0$ and $h = \mathrm{LP}(\boldsymbol{v}_0)$, so $\psi(\boldsymbol{v}_0) = \eta(h+1)+h$. During the algorithm, whenever a simple transformation is performed, we reduce the value $\psi(\boldsymbol{v}_0)$ of the 0^{th} row. Unless $\mathrm{LP}(\boldsymbol{v}_0) = 0$, we have $\psi(\boldsymbol{v}_0) \leq \eta(h+1)+h$ since Line 10 corresponds to a decrement of $\eta(h+1)+h$ by one if $h > 1$ and by two if $h = 1$. Note that $\psi(\boldsymbol{v}_0) = \eta(h+1)+h$ if and only if the algorithm performs the next simple transformation.

Hence, the number of iterations of the while loop is a lower bound on the total reduction of the sum value of the matrix and is therefore upper bounded by $(h+1)(\Delta(\Phi_{\boldsymbol{w}^{(\mathrm{M})}}(\boldsymbol{M}^{(\mathrm{M})}))+h+1) \in O(h\mu)$, cf. Theorem 6.26. At termination, we have $\deg v_{0,0} > \eta$, which means $\mathrm{LP}(\boldsymbol{v}_0) = 0$ and $\boldsymbol{V}$ is in wPf. □

We speed up Algorithm 18 using the following two observations.

Lemma 6.46. *Consider an instance of Problem 6.3 with $\boldsymbol{M}^{(\mathrm{M})}$ as in (6.4), $\boldsymbol{w}^{(\mathrm{M})}$ as in Lemma 6.5, and $\widetilde{g}_i := g_i x^{\gamma_i}$. Let $\boldsymbol{V}$ be a matrix that originates from $\boldsymbol{M}^{(\mathrm{M})}$ by a sequence of the following modified simple transformation j on i: Instead of replacing $\boldsymbol{v}_i$ by $\boldsymbol{v}'_i$, we replace it by*

$$\boldsymbol{v}''_i := [v'_{i,0}, v'_{i,1} \bmod_{\mathrm{r}} \widetilde{g}_1, \ldots, v'_{i,h} \bmod_{\mathrm{r}} \widetilde{g}_1].$$

The resulting matrix $\boldsymbol{V}$ is still a basis of the same module and we have $\psi(\boldsymbol{v}''_i) \leq \psi(\boldsymbol{v}'_i) < \psi(\boldsymbol{v}_i)$.

Proof. The row space is not changed since for all $h > 0$, the h^{th} modulo reduction can be realized instead by elementary row operations on the matrix after replacing $\boldsymbol{v}_i$ by $\boldsymbol{v}'_i$. In order

to prove this, we define $\boldsymbol{g}_h := [0, \ldots, 0, \widetilde{g}_h, 0, \ldots, 0]$ and J_h to be the set of rows of $\boldsymbol{V}$ that span $\boldsymbol{g}_h$ and fulfill $\psi(\boldsymbol{v}) \leq \psi(\boldsymbol{g}_h)$ for all $\boldsymbol{v} \in J_h$. Initially, we have $J_h = \{\boldsymbol{v}_h\}$. We distinguish two cases. *Case 1:* Let $\boldsymbol{v}_i \in J_h$. Then, $\psi(\boldsymbol{v}_i') < \psi(\boldsymbol{v}_i) \leq \psi(\boldsymbol{g}_h)$ and the h^{th} modulo reduction has no effect. We update $J_h = (J_h \setminus \{\boldsymbol{v}_i\}) \cup \{\boldsymbol{v}_i', \boldsymbol{v}_j\}$. *Case 2:* If $\boldsymbol{v}_i \notin J_h$, the vector $\boldsymbol{g}_h$ is a (left) linear combination of rows in J_h, so the h^{th} modulo reduction can be realized by elementary row operations after the replacement of $\boldsymbol{v}_i$ by $\boldsymbol{v}_i''$. The set J_h is not changed. *In both cases*, J_h does not become empty, so we can apply the argument inductively.

Due to the properties of a simple transformation and since the modulo operation does not increase any of the degrees in the vector, we have $\psi(\boldsymbol{v}_i'') \leq \psi(\boldsymbol{v}_i') < \psi(\boldsymbol{v}_i)$. □

Lemma 6.47. *Let $r_i, g_i \in \mathbb{F}_{q^m}[x;\sigma]$ and $\gamma_i \in \mathbb{N}_0$ for all $i = 1, \ldots, h$ be chosen as in Problem 6.3 with the additional property that the g_i are of the form $g_i = x^{t_i} + c_i$ for some $t_i \in \Theta(\mu)$ and $c_i \in \mathbb{F}_{q^m}$. For any $\lambda \in \mathbb{F}_{q^m}[x;\sigma]$ of degree $\deg \lambda \in O(\mu)$, $h \in \{1, \ldots, h\}$ and $\eta \in \mathbb{N}_0$, we can compute the coefficient of x^η in $(\lambda r_h x^{\gamma_h} \mod_{\mathrm{r}} g_h x^{\gamma_h})$ in $O(\mu)$ operations over $\mathbb{F}_{q^m}$.*

Proof. Since $(\lambda r_h x^{\gamma_h} \mod_{\mathrm{r}} g_h x^{\gamma_h})$ is divisible from the right by x^{γ_h} and has degree at most $t_i + \gamma_h - 1$, all coefficients of x^η with $\eta < \gamma_h$ and $\eta \geq t_i + \gamma_h$ are zero. For all other values of η, the coefficient of x^η in $(\lambda r_h x^{\gamma_h} \mod_{\mathrm{r}} g_h x^{\gamma_h})$ is the one of $x^{\eta-\gamma_h}$ in $(\lambda r_h \mod_{\mathrm{r}} g_h)$. By the structure of g_h, this coefficient is a linear combination of the coefficients with indices

$$\eta - \gamma_h, \eta - \gamma_h + t_h, \ldots, \eta - \gamma_h + t \cdot t_h,$$

of the polynomial λr_h, where $t = \lfloor \frac{\deg(\lambda r_h) - \eta + \gamma_h}{t_h} \rfloor \in O(1)$ (i.e., the number of coefficients is constant compared to μ). Each of these coefficients can be computed in $O(\mu)$ automorphism computations, multiplications and additions over $\mathbb{F}_{q^m}$, so the overall cost is as stated. □

Theorem 6.48. *Algorithm 19 is correct and costs $O(h\mu^2)$ operations over $\mathbb{F}_{q^m}$.*

Algorithm 19: Skew Polynomial Variant of Rosenkilde's Demand-Driven Algorithm

Input: Instance of Problem 6.3 with $g_i = x^{t_i} + c_i$, where $t_i \in \Theta(\mu)$ and $c_i \in \mathbb{F}_{q^m}$. Furthermore, $\widetilde{g}_i = g_i x^{\gamma_i}$ and $\widetilde{r}_i = r_i x^{\gamma_i}$ for $i = 1, \ldots, h$

Output: Row $\boldsymbol{v}$ with $\mathrm{LP}(\boldsymbol{v}) = 0$ of a basis $\boldsymbol{V}$ of $\mathcal{M}^{(\mathrm{M})}$ as in (6.3) in $[\gamma_0, \ldots, \gamma_h]$-shifted weak Popov form.

$[\eta, h] \leftarrow [\deg \boldsymbol{v}_0, \mathrm{LP}(\boldsymbol{v}_0)]$, where $\boldsymbol{v}_0 = [x^{\gamma_0}, \widetilde{r}_1, \ldots, \widetilde{r}_h]$
$[\lambda_0, \ldots, \lambda_h] \leftarrow [1, 0, \ldots, 0]$
$\alpha_i x^{\eta_j} \leftarrow$ leading monomial of $\widetilde{g}_i$ for $i = 1, \ldots, h$
while $\deg \lambda_0 \leq \eta$ **do**
  $\alpha \leftarrow$ coefficient of x^η in $(\lambda_0 \widetilde{r}_h \mod_{\mathrm{r}} \widetilde{g}_h)$ `// O(μ) by Lemma 6.47`
  if $\alpha \neq 0$ **then**
    if $\eta < \eta_h$ **then**
      swap $[\lambda_0, \alpha, \eta]$ and $[\lambda_h, \alpha_h, \eta_h]$
    $\lambda_0 \leftarrow \lambda_0 - \alpha/\sigma^{\eta-\eta_h}(\alpha_h) x^{\eta-\eta_h} \lambda_h$ `// O(μ)`
  $[\eta, h] \leftarrow \begin{cases} [\eta, h-1], & \text{if } h > 1 \\ [\eta - 1, h], & \text{else.} \end{cases}$
return $[\lambda_0, (\lambda_0 r_1 \mod_{\mathrm{r}} g_1), \ldots, (\lambda_0 r_h \mod_{\mathrm{r}} g_h)]$ `// O(hμ²)`

Proof. Algorithm 19 arises from Algorithm 18 by the following modifications:

- Instead of the entire matrix $\boldsymbol{V}$, we only remember the first column $[\lambda_0, \dots, \lambda_h]$ of $\Phi^{-1}_{\boldsymbol{w}^{(\mathrm{M})}}(\boldsymbol{V})$, which is $[1, 0, \dots, 0]$ at the beginning. Since $\Phi^{-1}_{\boldsymbol{w}^{(\mathrm{M})}}(\boldsymbol{V})$ is a basis of the module $\mathcal{M}^{(\mathrm{M})}$ as in (6.3), each row $\boldsymbol{v} = [\lambda x^{\gamma_0}, \omega_1 x^{\gamma_1}, \dots, \omega_h x^{\gamma_h}]$ of $\boldsymbol{V}$ is determined by λ using
$$\omega_i x^{\gamma_i} = (\lambda \widetilde{r}_i \bmod_{\mathrm{r}} \widetilde{g}_i).$$
- After every simple transformation, we reduce the coefficients of the affected row modulo $\widetilde{g}_h$ as in Lemma 6.46. This neither changes the correctness of the algorithm nor the upper bound $O(h\mu)$ on the number of iterations of the while loop since the line's value cannot increase.
- Combining both modifications above, we can compute the coefficient of x^η in $v_{0,0}$ as in Line 5 on demand.[5]

Hence, the two algorithms are computationally equivalent. Algorithm 19 returns the first row of the computed basis, which has $\boldsymbol{w}^{(\mathrm{M})}$-shifted leading position 0.

As for the complexity, the while loop is iterated at most $O(h\mu)$ times by Lemma 6.45, where Line 5 has complexity $O(\mu)$ over $\mathbb{F}_{q^m}$ by Lemma 6.47, and Line 9 costs $O(\mu)$ additions, multiplications and automorphism computations. Line 11 of the algorithm consists of h multiplications and modulo reductions, which cost $O(h\mu^2)$ in total.[6] □

6.3.2 Interpolation-Based Decoding: Weak Popov Walk

In this section, we describe an algorithm for computing a $\boldsymbol{w}$-shifted weak Popov form of a basis that is already reduced over a different shift $\boldsymbol{w}'$. By applying this algorithm iteratively to the module basis of the interpolation problem (cf. Algorithm 14), we obtain a complexity reduction from $O(h^3 n^2)$ to $O(h^2 n^2)$ compared to the Mulders–Storjohann. The idea is inspired by "Gröbner walks" [CKM97], which is why we call it "weak Popov walk". To the best of our knowledge, there is no known $\mathbb{F}_{q^m}[x]$-equivalent of this algorithm.

The base of the method is Algorithm 20, which "walks" from a weak Popov form w.r.t. the shift $\boldsymbol{w}$ into one with shift $\boldsymbol{w} + [1, 0, \dots, 0]$.

Lemma 6.49. *Algorithm 20 is correct. It performs at most*
$$O(r \deg \det(\boldsymbol{V}) + \textstyle\sum_{i,j} |w_i - w_j| + r^2))$$
operations over $\mathbb{F}_{q^m}$.

Proof. Let $\boldsymbol{V}$ be the input and $\hat{\boldsymbol{V}}$ the output matrix of the algorithm. The algorithm performs at most one simple transformation on each row $\boldsymbol{v}_i$ with $i \in I$ and keeps all other rows untouched. The set I contains the indices of the rows $\boldsymbol{v}_i$ with $\mathrm{LP}_{\boldsymbol{w}}(\boldsymbol{v}_i) \neq \mathrm{LP}_{\hat{\boldsymbol{w}}}(\boldsymbol{v}_i)$ and the row with $\mathrm{LP}_{\boldsymbol{w}}(\boldsymbol{v}_i) = 0$ since the leading position either remains the same or becomes 0 under the new shift. We show that the leading positions of the rows $\hat{\boldsymbol{v}}_{i_1}, \dots, \hat{\boldsymbol{v}}_{i_s}$ are a permutation of $h_{i_1}, \dots, h_{i_s}$. We distinguish two cases given by the if condition.

[5] This is the origin of the name "demand-driven".

[6] The cost of Line 11 can be further reduced using the fast multiplication and division algorithms of Section 5.2. However, this would not change the algorithm's overall complexity.

Algorithm 20: WeakPopovWalkStep

Input: Shift $\boldsymbol{w} \in \mathbb{N}_0^m$ and matrix $\boldsymbol{V} \in \mathrm{GL}_r(\mathbb{F}_{q^m}[x;\sigma])$ in $\boldsymbol{w}$-shifted weak Popov form
Output: Matrix in $\hat{\boldsymbol{w}}$-shifted weak Popov form spanning the same $\mathbb{F}_{q^m}[x;\sigma]$-row space as V, where $\hat{\boldsymbol{w}} = \boldsymbol{w} + [1, 0, \ldots, 0]$

$h_i \leftarrow \mathrm{LP}_{\boldsymbol{w}}(\boldsymbol{v}_i)$, for $i = 0, \ldots, m-1$
$I \leftarrow$ indexes i such that $\mathrm{LP}_{\hat{\boldsymbol{w}}}(\boldsymbol{v}_i) = 0$
$[i_1, \ldots, i_s] \leftarrow I$ sorted such that $h_{i_1} < h_{i_2} < \ldots < h_{i_s}$
$t \leftarrow i_1$
for $i = i_2, \ldots, i_s$ **do**
  if $\deg v_{t,0} \leq \deg v_{i,0}$ **then**
    Apply a simple transformation t on i at position 0 in $\mathcal{V}$
  else
    Apply a simple transformation i on t at position 0 in $\mathcal{V}$
    $t \leftarrow i$
return V

Case 1: During the algorithm, let i and t be such that $\deg v_{t,0} \leq \deg v_{i,0}$. We show that the resulting row $\hat{\boldsymbol{v}}_i = \boldsymbol{v}_i + \alpha x^{\delta} \boldsymbol{v}_t$, where $\delta = \deg v_{i,0} - \deg v_{t,0} \geq 0$ and suitable $\alpha \in \mathbb{F}_{q^m}$, has $\mathrm{LP}_{\hat{\boldsymbol{w}}}(\hat{\boldsymbol{v}}_i) = h_i$. Since we are performing a simple transformation at position 0 under the shift $\hat{\boldsymbol{w}}$, we know that the $\hat{\boldsymbol{w}}$-shifted degree of the row decreases to

$$\deg_{\hat{\boldsymbol{w}}} \hat{\boldsymbol{v}}_i \leq \deg_{\hat{\boldsymbol{w}}} \boldsymbol{v}_i - 1 = \deg_{\boldsymbol{w}} \boldsymbol{v}_i = \deg v_{i,h_i} + w_{h_i}.$$

Due to $h_t < h_i$, for $\boldsymbol{u} := \alpha x^{\delta} \boldsymbol{v}_t$ we have

$$\deg u_j + w_j < \delta + \deg \boldsymbol{v}_t + w_j = \deg \boldsymbol{v}_i + w_j = \deg v_{i,h_i} + w_{h_i}.$$

Thus, $\deg \hat{v}_{i,h_i} = \deg v_{i,h_i}$ and $\deg \hat{v}_{i,j} + w_j < \deg v_{i,h_i} + w_{h_i}$ for all $j > h_i$, so $\mathrm{LP}_{\hat{\boldsymbol{w}}}(\hat{\boldsymbol{v}}_i) = h_i$.

Case 2: Using exactly the same arguments, we obtain $\mathrm{LP}_{\hat{\boldsymbol{w}}}(\hat{\boldsymbol{v}}_t) = h_i$ in the other case ($\deg v_{t,0} > \deg v_{i,0}$).

Combined, each row $\boldsymbol{v}_i$ with index $i \in I \setminus \{i_1\}$ either keeps its leading position h_i or gives it away to another row $\boldsymbol{v}_t$ and waits for getting the initial LP h_{i_j} for $j > i$ of again another row v_{i_j} (i.e, $t \leftarrow i$). At termination, the row with index t is the only one on which no simple transformation was performed, so it keeps the leading position 0. Hence, the $\mathrm{LP}_{\hat{\boldsymbol{w}}}(\hat{\boldsymbol{v}}_{i_1}), \ldots, \mathrm{LP}_{\hat{\boldsymbol{w}}}(\hat{\boldsymbol{v}}_{i_s})$ are a permutation of the initial leading position $h_{i_1}, \ldots, h_{i_s}$, so $\hat{\boldsymbol{V}}$ is in $\hat{\boldsymbol{w}}$-shifted weak Popov form.

As for the complexity, since Algorithm 20 performs at most one simple transformation on each row using another row of the original matrix, we can bound the cost by the number of non-zero monomials in the original matrix. W.l.o.g., we assume that $\mathrm{LP}_{\boldsymbol{w}}(\boldsymbol{v}_i) = i$ for all i (otherwise, permute the rows). Since $\Phi_{\boldsymbol{w}}(\boldsymbol{V})$ is in weak Popov form, we have

$$\deg_{\boldsymbol{w}}(\boldsymbol{V}) = \deg \det \Phi_{\boldsymbol{w}}(\boldsymbol{V}) = \deg \det \boldsymbol{V} + \sum_{i=0}^{r-1} w_i$$

Among all matrices having this weighted degree, a matrix with the maximal possible number of monomials is one with maximal shifted degree of the last row, i.e., $\deg_{\boldsymbol{w}} \boldsymbol{v}_{r-1} = \deg_{\boldsymbol{w}}(\boldsymbol{V}) -$

$\sum_{i=0}^{r-2} \deg_{\boldsymbol{w}} \boldsymbol{v}_i$, where the degrees of the $\boldsymbol{v}_i$ are chosen as small as possible. Due to $\mathrm{LP}_{\boldsymbol{w}}(\boldsymbol{v}_i) = i$, we must have $\deg_{\boldsymbol{w}} \boldsymbol{v}_i \geq w_i$, so we choose

$$\deg_{\boldsymbol{w}} \boldsymbol{v}_i = w_i \text{ for } i = 0, \ldots, r-2, \text{ and } \deg_{\boldsymbol{w}} \boldsymbol{v}_{r-1} = w_{r-1} + \deg \det \boldsymbol{V}.$$

Hence, we have $\deg v_{i,j} \leq \max\{-1, \deg \boldsymbol{v}_i - w_j\}$, and the total number of monomials involved in the simple transformations is at most

$$\begin{aligned}\sum_{i,j} (\deg v_{i,j} + 1) &\leq \sum_{i,j} \max\{0, \deg \boldsymbol{v}_i - w_j + 1\} \leq \sum_j \deg \det \boldsymbol{V} + \sum_{i,j} (|w_i - w_j| + 1) \\ &\leq r^2 + r \deg \det \boldsymbol{V} + \sum_{i,j} |w_j - w_i|,\end{aligned}$$

which implies the complexity statement. □

Iteratively applying Algorithm 20 leads to a method for efficiently row-reducing the module basis given in Section 6.1.2 for solving the interpolation step of decoding interleaved Gabidulin codes. The procedure is outlined in Algorithm 21.

Theorem 6.50. *Algorithm 21 is correct and performs at most $O(h^2 n^2)$ operations over $\mathbb{F}_{q^m}$.*

Algorithm 21: WeakPopovWalk

Input: Instance of Problem 6.8, the matrix $\boldsymbol{V} \leftarrow \boldsymbol{M}^{(\mathrm{I})}$ as in (6.10), and $\boldsymbol{w} = [0, k_1 - 1, \ldots, k_h - 1]$ as in Lemma 6.10
Output: A $\boldsymbol{w}$-shifted weak Popov form of $\boldsymbol{M}^{(\mathrm{I})}$

```
w' ← w + [0, n, n, ..., n]
for i = 1, ..., n do
    V ← WeakPopovWalkStep(V, w')
    w' ← w' + [1, 0, ..., 0]
return V
```

Proof. At the start of the algorithm, $\boldsymbol{V}$ is in $\boldsymbol{w}'$-shifted weak Popov form due to the structure of $\boldsymbol{M}^{(\mathrm{I})}$ (cf. (6.10); note that the polynomials in the first column of $\boldsymbol{M}^{(\mathrm{I})}$ have degree at most n). By the correctness of Algorithm 20, at termination, the algorithm is in $(\boldsymbol{w} + [n, n, \ldots, n])$-shifted weak Popov form, which implies being in wPv w.r.t. $\boldsymbol{w}$. The algorithm calls *Algorithm* 20 n times, where $\sum_{i,j} |w_j - w_i| \in O(h^2 n)$ and $\deg \det \boldsymbol{V} = n$, resulting in the claimed complexity using Lemma 6.49. □

6.4 Concluding Remarks

We have shown that row reduction of skew polynomial matrices can be used for solving the core tasks of both various key-equation based (MgLSSR problem, see Appendix A.4) and interpolation-based (interpolation step) methods for decoding of interleaved Gabidulin codes. This implies flexible, efficient, and still conceptually simple decoding algorithms.

We have adapted several algorithms for row reduction of $\mathbb{F}_{q^m}[x]$-matrices to skew polynomials by taking care of the non-commutativity of $\mathbb{F}_{q^m}[x; \sigma]$. Furthermore, we have introduced

Algorithm	MgLSSR Problem	Interpolation Step
Skew Mulders–Storjohann (Sec. 6.2.1)	$O(h^2n^2)$	$O(h^3n^2)$
Skew Alekhnovich (Sec. 6.2.3)	$O(h^3\mathcal{M}_{q^m}(n)\log(n))$	$O(h^3\mathcal{M}_{q^m}(hn)\log(hn))$
Skew Demand-Driven (Sec. 6.3.1)	$O(hn^2)$	-
Weak Popov Walk (Sec. 6.3.2)	-	$O(h^2n^2)$
Skew Berlekamp–Massey [SJB11]	$O(hn^2)$	-
Sidorenko–Bossert [SB14]	$O(h^3\mathcal{M}_{q^m}(n)\log(n))$	-
Xie et al. [XLYS13]	-	$O(h^2n^2)$

Table 6.1: Complexity comparison of new and existing algorithms for solving the core tasks of decoding h-interleaved Gabidulin codes of length n: The MgLSSR problem (cf. Section 6.1.1) and the interpolation step (cf. Section 6.1.2), respectively. Motivated by Appendix A.4, we use $\mu \in \Theta(n)$ as the MgLSSR problem parameter. Complexities are given in operations over $\mathbb{F}_{q^m}$.

a new row-reduction algorithm, Weak Popov Walk, which might be of wider interest—also in the $\mathbb{F}_{q^m}[x]$-case. Table 6.1 summarizes the resulting complexities of the algorithms.

Usually, we have $h \ll n$, so the algorithms based on efficient multiplication (Skew Alekhnovich and Sidorenko–Bossert) are the fastest algorithms for solving the problems. Note that we improve upon the method by Xie et al. [XLYS13] in this case, which is the fastest-known algorithm for the interpolation step.

The row reduction method has many interesting properties that could be beneficial for decoding, in particular for key-equation-based methods. For instance, we can easily compute all "small" elements of the row space of a matrix, implying a description of all solutions of the MgLSSR problem. This again could be used in a Chase-like fashion [Cha72] to correct a few more errors beyond the error correction capability.

The row reduction framework introduced in this chapter can be directly applied to decoding Guruswami–Xing [GX13] and Guruswami–Wang [GW14] rank-metric codes, as well as the subspace codes by Mahdavifar and Vardy [MV13, Mah12], cf. [PRLS17].

Over $\mathbb{F}_{q^m}[x]$, row reduction based on order bases computation [GJV03, ZL12] sometimes results in more efficient algorithms than the ones we adapted in this chapter. Therefore, these methods should be considered in the skew polynomial case, where they promise a cost of $O^{\sim}(h^{\omega}\mathcal{M}_{q^m}(n))$ operations over $\mathbb{F}_{q^m}$ for both the MgLSSR and the interpolation problem.

Remarks on Generality

Skew Polynomials With Derivation

Boucher and Ulmer [BU14] constructed codes based on skew polynomial rings with so-called derivations [Ore33b], which is a generalization of the skew polynomials considered in this thesis.[7] The row reduction algorithms in this chapter—except for Algorithm 17—work over these rings without modification, though with a worse complexity.

[7] The multiplication rule is then $x \cdot f = \sigma(f) \cdot x + \delta(f)$ for $f \in \mathbb{F}_{q^m}$ and a specific map $\delta : \mathbb{F}_{q^m} \to \mathbb{F}_{q^m}$.

Skew Polynomials Over Fields of Characteristic Zero

The algorithms can also be applied for row reduction of skew polynomial matrices over arbitrary Galois extensions L/K with cyclic Galois group and generator σ, where we need to replace $\mathbb{F}_{q^m}$ by L and $\mathbb{F}_q$ by K. This, for instance, implies that we can decode Gabidulin codes over fields of characteristic zero in a quadratic or even sub-quadratic number of operations over L using the algorithm in Section 5.1. As in Section 5.3, the complexity statements are then given in operations over L and do not necessarily provide a good estimate of the bit complexity due to possible coefficient growth. Thus, the complexity should be more thoroughly analyzed, e.g., using the methods developed by [AB01, BCL06, KRS17].

7

Generalizations of Twisted Gabidulin Codes

Maximum rank distance (MRD) codes, and Gabidulin codes as their most famous subfamily, were independently introduced by Delsarte [Del78], Gabidulin [Gab85], and Roth [Rot91]. Originally, Gabidulin codes were defined as evaluation codes of skew polynomials with the Frobenius automorphism $\cdot^q$. The codes were later generalized to arbitrary automorphisms in [Rot96, KG05, ALR13]. Since the invention of Gabidulin codes, it had been an open question whether other MRD codes exist.

The first such families were introduced by Sheekey [She16], twisted Gabidulin codes, and Otal and Özbudak [OÖ16], where the latter is a special case of the first. Similar to Gabidulin codes, these codes were generalized using arbitrary automorphisms in [She16, Remark 9] and [LTZ15]. Recently, several further constructions were proposed, leading to new MRD codes for some parameters or general families of non-linear rank-metric codes [HM17, CMP16, OÖ17, CMPZ17, CMZ17, DS17]. However, since Gabidulin codes and their transposes provide MRD codes for any field size q and matrix dimensions m and n, the variety of MRD codes, and rank-metric codes in general, is still far behind their Hamming-metric counterparts.

In Section 7.1, we use a similar approach as for twisted Reed–Solomon codes (cf. Chapter 4) in order to introduce a new class of linear rank-metric codes by adding $\ell \in \mathbb{N}$ further monomials to the evaluation polynomials. The family is a generalization of Sheekey's twisted Gabidulin codes, whereas the methods for their analysis differ. We derive a sufficient condition for the new codes to be MRD in Section 7.2, which yields MRD codes over $\mathbb{F}_{q^m}$ of length up to $2^{-\ell}m$. In Section 7.3, we present a decoding algorithm for twisted Gabidulin codes whose evaluation points span a sub-field of $\mathbb{F}_{q^m}$, which is able to decode up to one third of the minimum distance for one twist $\ell = 1$, and up to $\approx \frac{d}{\ell+1}$ for large ℓ. Furthermore, we show that the new class contains MRD codes that are neither equivalent to Gabidulin nor Sheekey's twisted Gabidulin codes. Finally, we briefly analyze the new codes in the GPT cryptosystem in Section 7.5 and conclude the chapter in Section 7.6.

The results of this chapter were partly published in [PRS17].

7.1 Definition

The idea of the new codes is similar to Sheekey's twisted Gabidulin and the twisted Reed–Solomon codes in Chapter 4, using skew polynomial language. A Gabidulin code is MRD since the root space of a non-zero skew polynomial $f = \sum_{i=0}^{k-1} f_i x^i \in \mathbb{F}_{q^m}[x;\sigma]$ with degree less than k has dimension at most $k-1$. By the rank nullity theorem, the evaluation of this polynomial at n many $\mathbb{F}_q$-linearly independent evaluation points $\alpha_1, \ldots, \alpha_n \in \mathbb{F}_{q^m}$ therefore

spans an at least $(n-k+1)$-dimensional $\mathbb{F}_q$-subspace of $\mathbb{F}_{q^m}$—that is, the minimum rank weight of the resulting code is $n-k+1$.

Sheekey [She16] added one more monomial $\eta f_0 x^k$ to all evaluation polynomials whose coefficient depends linearly on the zeroth coefficient of f. By a suitable choice of $\eta \in \mathbb{F}_{q^m}$, it was shown that the root space of any such polynomial still has dimension at most $k-1$ (cf. Section 2.3.5).

Our construction further generalizes this approach, similar to the twisted Reed–Solomon codes in Chapter 4. We add $\ell \leq n-k$ many additional monomials, "twists", to f, which are of the form $\sum_{j=1}^{\ell} \eta_j f_{h_j} x^{k-1+t_j}$ with distinct $1 \leq t_j \leq n-k$, suitable $0 \leq h_j < k$ and $\eta_j \in \mathbb{F}_{q^m}$ (see also the illustrations of the $\mathbb{F}_q[x]$-analog of this construction in Section 4.1). The codes are formally defined as follows.

Definition 7.1. Let $n, k, \ell \in \mathbb{N}$ with $k < n$ and $\ell \leq n-k$. Choose a *hook vector* $\boldsymbol{h} \in \{0, \dots, k-1\}^\ell$ and a *twist vector* $\boldsymbol{t} \in \{1, \dots, n-k\}^\ell$ such that the t_i are distinct, and let $\boldsymbol{\eta} \in (\mathbb{F}_{q^m} \setminus \{0\})^\ell$. The set of $[k, \boldsymbol{t}, \boldsymbol{h}, \boldsymbol{\eta}]$-*twisted skew polynomials* over $\mathbb{F}_{q^m}$ is defined by

$$\mathcal{P}^{n,k}_{\boldsymbol{t},\boldsymbol{h},\boldsymbol{\eta}} = \left\{ f = \sum_{i=0}^{k-1} f_i x^i + \sum_{j=1}^{\ell} \eta_j f_{h_j} x^{k-1+t_j} : f_i \in \mathbb{F}_{q^m} \right\} \subseteq \mathbb{F}_{q^m}[x;\sigma].$$

Let $\alpha_1, \dots, \alpha_n \in \mathbb{F}_{q^m}$ be linearly independent over $\mathbb{F}_q$ and write $\boldsymbol{\alpha} = [\alpha_1, \dots, \alpha_n]$. The $[\boldsymbol{\alpha}, \boldsymbol{t}, \boldsymbol{h}, \boldsymbol{\eta}]$-*twisted Gabidulin code* of length n and dimension k is given by

$$\mathcal{C}_{\boldsymbol{\alpha},\boldsymbol{t},\boldsymbol{h},\boldsymbol{\eta}}[n,k] = \mathrm{ev}_{\boldsymbol{\alpha}}(\mathcal{P}^{n,k}_{\boldsymbol{t},\boldsymbol{h},\boldsymbol{\eta}}) \subseteq \mathbb{F}^n_{q^m}.$$

For brevity, we say *twisted skew polynomials* and *twisted Gabidulin codes*, respectively, and refer to Sheekey's codes as *Sheekey's twisted Gabidulin codes*.

The new codes are linear over $\mathbb{F}_{q^m}$ due to the linearity of the evaluation map. They indeed have dimension k since any distinct $f, g \in \mathcal{P}^{n,k}_{\boldsymbol{t},\boldsymbol{h},\boldsymbol{\eta}}$ fulfill

$$f \not\equiv g \mod_{\mathrm{r}} \mathcal{M}_{\langle \alpha_1, \dots, \alpha_n \rangle},$$

where $\mathcal{M}_{\langle \alpha_1, \dots, \alpha_n \rangle}$ is the minimal subspace polynomial of the space spanned by the evaluation points, which has degree n, so the evaluation map is injective on $\mathcal{P}^{n,k}_{\boldsymbol{t},\boldsymbol{h},\boldsymbol{\eta}}$.

7.2 A Sufficient Condition for Twisted Gabidulin Codes to be MRD

Not all codes as in Definition 7.1 are MRD. In the following, we give a sufficient condition for the new codes to be MRD. The strategy is to choose the evaluation polynomials $\alpha_1, \dots, \alpha_n$ and the twist coefficients $\boldsymbol{\eta}$ in such a way that the root space of any non-zero evaluation polynomial intersects with the space spanned by the evaluation points in at most k dimensions. We start with a lemma, which can be seen as the rank-metric analog of Lemma 4.2 on page 56.

Lemma 7.2. *Let $n, k, \boldsymbol{t}, \boldsymbol{h}, \boldsymbol{\eta}$ be chosen as in Definition 7.1 with $t_1 < t_2 < \cdots < t_\ell$. For any $\mathcal{S} \subset \langle \alpha_1, \dots, \alpha_n \rangle_{\mathbb{F}_q}$ with $\dim_{\mathbb{F}_q} \mathcal{S} = k$, consider the homogeneous linear system of equations in $g_0, \dots, g_{t_\ell - 1} \in \mathbb{F}_{q^m}$:*

$$\sum_{j=0}^{t_\ell - 1} g_j T^{(\mathcal{S})}_{i,j} = 0 \quad \text{for } i = k, \dots, k-1+t_\ell, \tag{7.1}$$

where for $\sum_{i=0}^{k} a_i x^i := \mathcal{M}_{\mathcal{S}}$ *and* $a_i := 0$ *for* $i < 0$ *or* $i \geq k$*, we have*

$$T_{i,j}^{(\mathcal{S})} = \begin{cases} \eta_\kappa^{-1}\sigma^j(a_{i-j}) - \sigma^j(a_{h_\kappa - j}), & \text{if } i = k-1+t_\kappa \text{ for } \kappa \in \{1,\dots,\ell\}, \\ \sigma^j(a_{i-j}), & \text{else.} \end{cases}$$

The code $\mathcal{C}_{\boldsymbol{\alpha},\boldsymbol{t},\boldsymbol{h},\boldsymbol{\eta}}[n,k]$ *is MRD if and only if there is no choice of* $\mathcal{S}$ *admitting a non-zero solution to the linear system.*

Proof. Let $f \in \mathcal{P}_{\boldsymbol{t},\boldsymbol{h},\boldsymbol{\eta}}^{n,k}$ be a polynomial whose root space fulfills

$$\dim\left(\text{roots}(f) \cap \langle \alpha_1,\dots,\alpha_n\rangle\right) \geq k. \tag{7.2}$$

Thus, there is a k-dimensional subspace $\mathcal{S} \subseteq \langle \alpha_1,\dots,\alpha_n\rangle$ whose minimal subspace polynomial $\mathcal{M}_{\mathcal{S}}$ right-divides f, i.e., $f = g \cdot \mathcal{M}_{\mathcal{S}}$ for some $g \in \mathbb{F}_{q^m}[x;\sigma]$. Write $g = \sum_{i=0}^{t_\ell - 1} g_i x^i$ and $\mathcal{M}_{\mathcal{S}} = \sum_{i=0}^{k} a_i x^i$ with $a_i := 0$ for $i < 0$ or $i \geq k$. By definition, the i^{th} coefficient of f is given by $f_i = \sum_{j=0}^{t_\ell - 1} g_j \sigma^j(a_{i-j})$. Since $f \in \mathcal{P}_{\boldsymbol{t},\boldsymbol{h},\boldsymbol{\eta}}^{n,k}$, we have for $i \geq k$

$$f_i = \begin{cases} \eta_\kappa f_{h_\kappa} & \text{if } i = k-1+t_\kappa \text{ for } \kappa \in \{1,\dots,\ell\}, \\ 0 & \text{otherwise .} \end{cases}$$

Hence, for $i = k - 1 + t_\kappa$, we get

$$0 = \sum_{j=0}^{t_\ell - 1} g_j \cdot \left(\eta_\kappa^{-1}\sigma^j\left(a_{i-j}\right) - \sigma^j\left(a_{h_\kappa - j}\right)\right).$$

A non-zero polynomial f satisfying (7.2) therefore yields a non-zero solution g_j of (7.1) for some subspace $\mathcal{S}$ and the other way round. Due to

$$\text{rank}(\text{ev}_{\boldsymbol{\alpha}}(f)) = n - \dim\left(\text{roots}(f) \cap \langle \alpha_1,\dots,\alpha_n\rangle\right),$$

the code $\mathcal{C}_{\boldsymbol{\alpha},\boldsymbol{t},\boldsymbol{h},\boldsymbol{\eta}}[n,k]$ is MRD if and only if (7.2) is not fulfilled for any non-zero $f \in \mathcal{P}_{\boldsymbol{t},\boldsymbol{h},\boldsymbol{\eta}}^{n,k}$, which implies the claim. □

For a given subspace $\mathcal{S}$, the system (7.1) is of a similar form as its twisted RS counterpart (cf. Figure 4.2 on page 56), i.e.,

$$\underbrace{\begin{bmatrix} \eta_{t_\ell}^{-1}+\square & \square & \dots & \square & \square & \square & \dots & \square & \square & \dots \\ \square & 1 & & & & & & & & \\ \vdots & \vdots & \ddots & & & & & & & \\ \square & \square & \dots & 1 & & & & & & \\ \eta_{t_{\ell-1}}^{-1}\square+\square & \eta_{t_{\ell-1}}^{-1}\square+\square & \dots & \eta_{t_{\ell-1}}^{-1}\square+\square & \eta_{t_{\ell-1}}^{-1}+\square & \square & \dots & \square & \square & \dots \\ \square & \square & \dots & \square & \square & 1 & & & & \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & & & \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & & & \\ \square & \square & \dots & \square & \square & \square & \dots & 1 & & \\ \eta_{t_1}^{-1}\square+\square & \eta_{t_1}^{-1}\square+\square & \dots & \eta_{t_1}^{-1}\square+\square & \eta_{t_1}^{-1}\square+\square & \eta_{t_1}^{-1}\square+\square & \dots & \eta_{t_1}^{-1}\square+\square & \eta_{t_1}^{-1}+\square & \dots \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots \end{bmatrix}}_{=:\boldsymbol{B}_{\mathcal{S}}} \begin{bmatrix} g_{t_\ell} \\ g_{t_\ell-1} \\ \vdots \\ g_1 \\ g_0 \end{bmatrix} = \mathbf{0}, \tag{7.3}$$

where the boxes □ represent elements obtained by $\mathbb{F}_q$-linear combinations, automorphism computations $\sigma(\cdot)$, and multiplications from $\alpha_1,\dots,\alpha_n$. If a row corresponds to a coefficient of

index $i = k-1+t_\kappa$ for some $\kappa = 1, \dots, \ell$, then the diagonal element of this row is $\eta_{t_\kappa}^{-1}+\square$, and it is 1 otherwise. This is due to $\sigma^{i-k}(a_{i-(i-k)}) = \sigma^{i-k}(a_k) = 1$ for all $i = k, k+1, \dots, k-1+t_\ell$ and since the diagonal elements are the $T_{i,j}^{(\mathcal{S})}$ of (7.1) with $j = i-k$. The entries above the diagonal are independent of the $\eta_{t_\kappa}^{-1}$ due to $a_{i-j} = 0$ for all $i > j+k$. We use this observation to prove the following sufficient condition for a twisted Gabidulin code to be MRD. The statement is the rank-metric equivalent of Theorem 4.3 on page 57 (see also Figure 4.3).

Theorem 7.3. *Let $s_0, \dots, s_\ell \in \mathbb{N}$ such that $\mathbb{F}_q \subseteq \mathbb{F}_{q^{s_0}} \subsetneq \mathbb{F}_{q^{s_1}} \subsetneq \cdots \subsetneq \mathbb{F}_{q^{s_\ell}} = \mathbb{F}_{q^m}$ is a chain of subfields. Furthermore, let $k < n \leq s_0$ and $\alpha_1, \dots, \alpha_n \in \mathbb{F}_{q^{s_0}}$ be linearly independent over $\mathbb{F}_q$, and let $\boldsymbol{t}$, $\boldsymbol{h}$, and $\boldsymbol{\eta}$ be chosen as in Definition 7.1 with the additional requirements*

$$\eta_i \in \mathbb{F}_{q^{s_i}} \setminus \mathbb{F}_{q^{s_{i-1}}}$$

and $t_1 < t_2 < \cdots < t_\ell$.[1] Then, the twisted Gabidulin code $\mathcal{C}_{\boldsymbol{\alpha},\boldsymbol{t},\boldsymbol{h},\boldsymbol{\eta}}[n,k]$ is MRD.

Proof. Let $\mathcal{S} \subseteq \langle \alpha_1, \dots, \alpha_n \rangle$ be a k-dimensional subspace. We show that the system (7.1) has only the zero solution. Lemma 7.2 then implies the claim. Since the a_i arise from $\alpha_1, \dots, \alpha_n$ by multiplications, automorphisms, and linear $\mathbb{F}_{q^{s_0}}$-combinations. Due to $\sigma(\mathbb{F}_{q^{s_0}}) = \mathbb{F}_{q^{s_0}}$, the boxes $\square$ of the system's matrix $\boldsymbol{B}_\mathcal{S}$ as in (7.3) represent elements from $\mathbb{F}_{q^{s_0}}$.

We consider the $\eta_{t_\kappa}^{-1}$ to be indeterminates. This means that $\det(\boldsymbol{B}_\mathcal{S}) \in \mathbb{F}_{q^{s_0}}[\eta_{t_1}^{-1}, \dots, \eta_{t_\ell}^{-1}]$ is a multivariate polynomial, where each $\eta_{t_\kappa}^{-1}$ appears at most of degree 1 in each monomial. Let $\boldsymbol{B}_\mathcal{S}^{(\mu)}$ be the $(\mu \times \mu)$-lower-right submatrix of $\boldsymbol{B}_\mathcal{S}$. We distinguish two cases:

(i) If $\mu \neq t_\kappa$ for all κ, then the first row of $\boldsymbol{B}_\mathcal{S}^{(\mu)}$ is $[1, 0, \dots, 0]$ and by Laplace's rule we get

$$\det\left(\boldsymbol{B}_\mathcal{S}^{(\mu)}\right) = \det\left(\boldsymbol{B}_\mathcal{S}^{(\mu-1)}\right).$$

(ii) If $\mu = t_\kappa$ for some κ, then the first row of $\boldsymbol{B}_\mathcal{S}^{(\mu)}$ is of the form $[\eta_{t_\kappa}^{-1}+\square, \square, \dots, \square]$ and the matrix contains only the indeterminates $\eta_{t_1}^{-1}, \dots, \eta_{t_{\kappa-1}}^{-1}$ in its rows 2 to μ, so

$$\det\left(\boldsymbol{B}_\mathcal{S}^{(\mu)}\right) = (\eta_{t_\kappa}^{-1} + T_\mu) \cdot \det\left(\boldsymbol{B}_\mathcal{S}^{(\mu-1)}\right) + U_\mu \in \mathbb{F}_{q^{s_0}}[\eta_{t_1}^{-1}, \dots, \eta_{t_\kappa}^{-1}],$$

where $T_\mu \in \mathbb{F}_{q^{s_0}}$ and $U_\mu \in \mathbb{F}_{q^{s_0}}[\eta_{t_1}^{-1}, \dots, \eta_{t_{\kappa-1}}^{-1}]$.

Combined, we get $\det(\boldsymbol{B}_\mathcal{S}^{(t_\kappa)}) = (\eta_{t_\kappa}^{-1} + T_{t_\kappa}) \det(\boldsymbol{B}_\mathcal{S}^{(t_\kappa-1)}) + U_{t_\kappa} \in \mathbb{F}_{q^{s_0}}[\eta_{t_1}^{-1}, \dots, \eta_{t_\kappa}^{-1}]$, where $\det(\boldsymbol{B}_\mathcal{S}^{(t_1)}) = \eta_{t_1}^{-1}$, and recursively substituting $\eta_\kappa \in \mathbb{F}_{q^{s_\kappa}} \setminus \mathbb{F}_{q^{s_{\kappa-1}}}$ for $\kappa = 1, \dots, \ell$ yields

$$\det\left(\boldsymbol{B}_\mathcal{S}^{(t_\kappa)}\right) \in \mathbb{F}_{q^{s_\kappa}} \setminus \{0\},$$

since $\eta_{t_\kappa}^{-1} \in \mathbb{F}_{q^{s_\kappa}} \setminus \mathbb{F}_{q^{s_{\kappa-i}}}$, $\det(\boldsymbol{B}_\mathcal{S}^{(t_\kappa-1)}) \in \mathbb{F}_{q^{s_{\kappa-1}}} \setminus \{0\}$, and $U_{t_\kappa} \in \mathbb{F}_{q^{s_{\kappa-1}}}$. In particular, we have $\det(\boldsymbol{B}_\mathcal{S}) = \det(\boldsymbol{B}_\mathcal{S}^{(t_\ell)}) \neq 0$ and the system (7.1) has only the zero solution. □

Theorem 7.3 implies a large constructive class of MRD twisted Gabidulin codes. For a given length n, the necessary extension degree m of these codes is given as follows.

[1] The entries of $\boldsymbol{t}$ and $\boldsymbol{h}$ can be permuted such that any twisted Gabidulin code fulfills the latter condition.

Corollary 7.4. *Let $\ell \in \mathbb{N}$ and $2^\ell \mid m$. There is an MRD twisted Gabidulin code with ℓ twists of length $n = 2^{-\ell}m$ over $\mathbb{F}_{q^m}$.*

Remark 7.5. *For MDS twisted Reed–Solomon codes, we were able to guarantee half the length of an RS code's maximal length ($n \leq q$) only in two special cases (cf. Section 4.7). In contrast, we achieve half the maximal length of a Gabidulin code ($n \leq m$) for MRD twisted Gabidulin codes with one twist $\ell = 1$ and any $\boldsymbol{t}$ and $\boldsymbol{h}$. Figure 7.1 illustrates the maximal dimensions of a codeword's matrix representation when constructed with Theorem 7.3.*

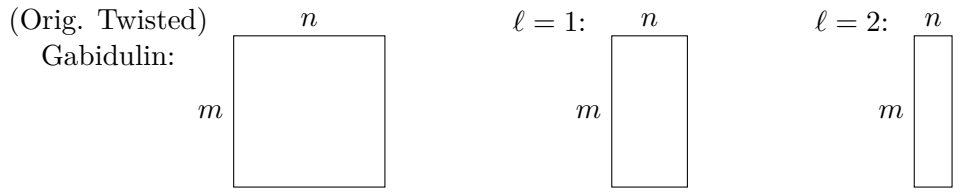


Figure 7.1: Maximal length n of Gabidulin codes (and Sheekey's twisted Gabidulin codes), and the new codes constructed with Theorem 7.3 for $\ell = 1$ and $\ell = 2$.

As for twisted RS codes, there can be MRD twisted Gabidulin codes of greater length than guaranteed by Theorem 7.3. One such class is given by Sheekey's twisted Gabidulin codes, which achieve the same maximal length as Gabidulin codes. Note the analogy to the "long" $(*)$-twisted RS codes in Section 4.7.

7.3 A Suboptimal Decoder

Decoding by guessing the twist coefficients as for twisted RS codes (cf. Section 4.3) is not possible for twisted Gabidulin codes since the number of possibilities is exponential in the code length. Clearly, a twisted Gabidulin code is contained in a Gabidulin code of dimension $k+\max_i\{t_i\}$, so we can always correct up to $\frac{n-k-\max_i\{t_i\}}{2}$ errors by decoding in this supercode.

In the special case of Sheekey's twisted Gabidulin codes, Rosenthal and Randrianarisoa [RR17] presented a decoder that can correct up to $\lfloor \frac{d-1}{2} \rfloor$ errors (instead of $\lfloor \frac{d-2}{2} \rfloor$ by decoding in the Gabidulin supercode). They obtain a system of linear equations with one additional non-linear equation, which they can solve in some cases. If their approach is generalized to one twist with arbitrary $t_1 > 1$ or $h_1 > 0$, then the number of non-linear equations becomes greater than one, and it is—at least—not obvious whether the system is solvable in polynomial time. The situation gets even more involved when multiple twists are considered.

In the following, we show how the codes can be decoded by partially utilizing the structure of the evaluation polynomials. The resulting decoder cannot decode up to half-the-minimum distance, but is efficient and achieves a decoding radius of about one third of the minimum distance in some important cases. We consider a received word

$$\boldsymbol{r} = \boldsymbol{c} + \boldsymbol{e} \in \mathbb{F}_{q^m}^n,$$

where $\boldsymbol{c} \in \mathcal{C}_{\boldsymbol{\alpha},\boldsymbol{t},\boldsymbol{h},\boldsymbol{\eta}}[n,k]$ and $\boldsymbol{e}$ is an error word of rank weight $\mathrm{wt_R}(\boldsymbol{e})$.

The following result implies a decoding algorithm for twisted Gabidulin codes with evaluation polynomials that span a subfield of $\mathbb{F}_{q^m}$. Its basic idea is to identify the largest "gap"

Δ_g, i.e., number of consecutive zero entry coefficients smaller than n, of the evaluation polynomials. The structure of the evaluation points implies that taking an automorphism of a codeword corresponds to a cyclic shift (and entry-wise automorphism computations) of the evaluation polynomial coefficients 0 to $n-1$. By a suitable shift of this kind, we obtain a codeword of a Gabidulin code of minimum distance $\Delta_\mathrm{g}+1$. The technique uses a similar approach as the proof of the BCH bound on the minimum distance of cyclic codes and only depends on the particular structure of the support of the evaluation polynomials. The parameters occurring in the statement and proof of Theorem 7.6 are illustrated in Figure 7.2.

Theorem 7.6. *Let $\boldsymbol{t}, \boldsymbol{h}, \boldsymbol{\alpha}, \boldsymbol{\eta}$ be chosen as in Definition 7.1 such that $t_1 < \cdots < t_\ell$ and the evaluation points span a subfield of $\mathbb{F}_{q^m}$, say $\langle \alpha_1, \ldots, \alpha_n \rangle_{\mathbb{F}_q} = \mathbb{F}_{q^n} \leq \mathbb{F}_{q^m}$. For $i = 0, \ldots, \ell+1$, we define*

$$\delta_i = \begin{cases} k-1, & \text{if } i = 0, \\ k-1+t_i, & \text{if } 1 \leq i \leq \ell, \\ n, & \text{if } i = \ell+1. \end{cases}$$

Furthermore, let $\Delta_\mathrm{g} = \max\{\delta_i - \delta_{i-1} - 1 \,:\, i = 1, \ldots, \ell+1\}$ and choose $\delta := \delta_i$ for some i with $\delta_i - \delta_{i-1} - 1 = \Delta_\mathrm{g}$. For a received word $\boldsymbol{r} = \boldsymbol{c} + \boldsymbol{e} \in \mathbb{F}_{q^m}^n$ with $\boldsymbol{c} \in \mathcal{C}_{\boldsymbol{\alpha},\boldsymbol{t},\boldsymbol{h},\boldsymbol{\eta}}[n,k]$, we have

$$\sigma^{n-\delta}(\boldsymbol{c}) \in \mathcal{C}_\mathrm{G}[n, n-\Delta_\mathrm{g}] \quad \text{and} \tag{7.4}$$

$$\mathrm{wt_R}\left(\sigma^{n-\delta}(\boldsymbol{e})\right) = \mathrm{wt_R}(\boldsymbol{e}), \tag{7.5}$$

where $\mathcal{C}_\mathrm{G}[n, n-\Delta_\mathrm{g}]$ is the Gabidulin code of dimension $n-\Delta_\mathrm{g}$ with evaluation points $\boldsymbol{\alpha}$. In particular, we can correct up to $\Delta_\mathrm{g}/2$ errors by decoding $\sigma^{n-\delta}(\boldsymbol{r})$ in $\mathcal{C}_\mathrm{G}[n, n-\Delta_\mathrm{g}]$.

Proof. First note that that $\sigma^{n-\delta}(\cdot) : \mathbb{F}_{q^m} \to \mathbb{F}_{q^m}$ is a bijective $\mathbb{F}_q$-linear map. Hence, it does not change the rank of $\boldsymbol{e}$, which proves (7.5). In order to show (7.4), we consider $\boldsymbol{c} \in \mathcal{C}_{\boldsymbol{\alpha},\boldsymbol{t},\boldsymbol{h},\boldsymbol{\eta}}[n,k]$, which corresponds to an evaluation polynomial

$$f = \sum_{j=0}^{k-1} f_j x^j + \sum_{j=1}^{\ell} \eta_j f_{h_j} x^{k-1+t_j}.$$

Thus, we have $c_i = f(\alpha_i)$ for all $i = 1, \ldots, n$. We can write the components of $\sigma^{n-\delta}(\boldsymbol{c})$ as

$$\sigma^{n-\delta}(c_i) = \sigma^{n-\delta}(f(\alpha_i)) = \left(x^{n-\delta} \cdot f\right)(\alpha_i) = \left(x^{n-\delta} \cdot f \mod_\mathrm{r} \mathcal{M}_{\langle \alpha_1, \ldots, \alpha_n \rangle}\right)(\alpha_i).$$

The polynomial $x^{n-\delta} \cdot f$ has a support (i.e., set of indices of non-zero coefficients)

$$\begin{aligned} \mathrm{supp}\left(x^{n-\delta} \cdot f\right) &\subseteq n - \delta + \{0, \ldots, k-1, k-1+t_1, \ldots, k-1+t_\ell\} \\ &= \{n-\delta, n-\delta+1, \ldots, n-\delta+\delta_0, n-\delta+\delta_1, \ldots, n-\delta+\delta_\ell\}. \end{aligned}$$

Since the α_i span a subspace $\mathbb{F}_{q^n}$, we have $\mathcal{M}_{\langle \alpha_1, \ldots, \alpha_n \rangle} = x^n - 1$, so taking a skew polynomial modulo $\mathcal{M}_{\langle \alpha_1, \ldots, \alpha_n \rangle}$ results in reducing each element of its support modulo n. This implies

$$\begin{aligned} &\mathrm{supp}\left(x^{n-\delta} \cdot f \mod_\mathrm{r} \mathcal{M}_{\langle \alpha_1, \ldots, \alpha_n \rangle}\right) \\ &\subseteq \{\delta_i - \delta, \ldots, \delta_\ell - \delta, n-\delta+1, \ldots, n-\delta+\delta_0, n-\delta+\delta_1, \ldots, n-\delta+\delta_{i-1}\}, \end{aligned}$$

so the degree of the evaluation polynomial corresponding to $\sigma^{n-\delta}(\boldsymbol{c})$ is

$$\deg\Big(x^{n-\delta} \cdot f \mod_{\mathrm{r}} \mathcal{M}_{\langle \alpha_1,\ldots,\alpha_n\rangle}\Big) \leq n - \delta + \delta_{i-1} = n - \Delta_{\mathrm{g}} - 1,$$

which proves that $\sigma^{n-\delta}(\boldsymbol{c}) \in \mathcal{C}_{\mathrm{G}}[n, n-\Delta_{\mathrm{g}}]$. □

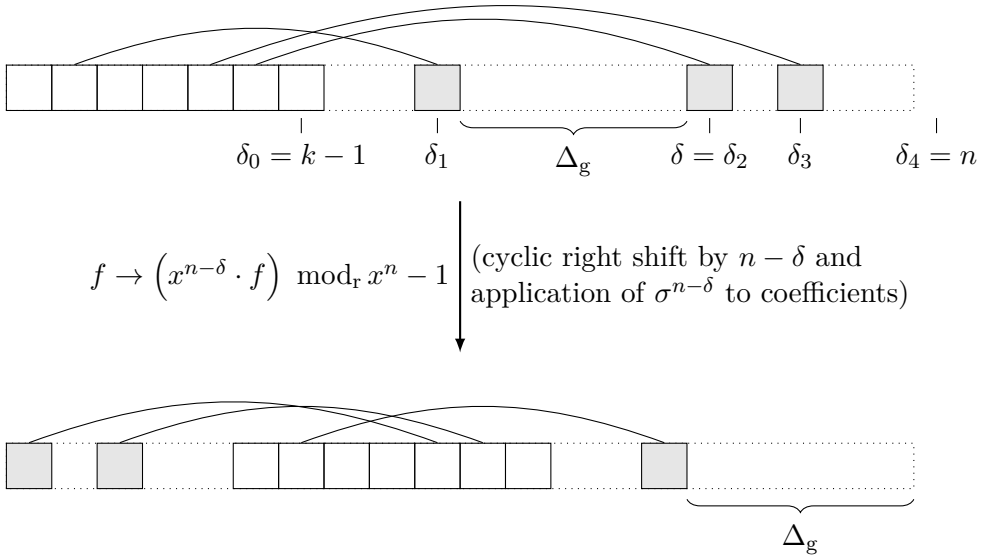


Figure 7.2: Illustration of the parameters $\delta_0, \ldots, \delta_{\ell+1}$, δ, and Δ_{g}, as well as the cyclic shift of the evaluation polynomials, in Theorem 7.6. Here, we use the example $n = 20$, $k = 7$, $\ell = 3$, $\boldsymbol{t} = [3, 9, 11]$, and $\boldsymbol{h} = [1, 5, 4]$.

Theorem 7.6 implies a simple decoding strategy, outlined in Algorithm 22, and the following two statements about the achievable decoding radius. The first statement is more general, but uses a BMD decoder for decoding in the Gabidulin code $\mathcal{C}_{\mathrm{G}}[n, n-\Delta_{\mathrm{g}}]$. The second statement holds for the MRD codes constructed in Section 7.2, and achieves a significantly larger decoding radius.

Algorithm 22: Decoder for twisted Gabidulin codes $\mathcal{C}_{\boldsymbol{\alpha},\boldsymbol{t},\boldsymbol{h},\boldsymbol{\eta}}[n,k]$

Input: Received word $\boldsymbol{r} \in \mathbb{F}_{q^m}^n$ and $\delta, \Delta_{\mathrm{g}}$ as in Theorem 7.6
Output: Codeword $\boldsymbol{c} \in \mathcal{C}_{\boldsymbol{\alpha},\boldsymbol{t},\boldsymbol{h},\boldsymbol{\eta}}[n,k]$ with minimal $\mathrm{wt}_{\mathrm{R}}(\boldsymbol{r}-\boldsymbol{c})$ or "decoding failure"

$\boldsymbol{r}' \leftarrow \sigma^{n-\delta}(\boldsymbol{r})$
$\boldsymbol{c}' \leftarrow$ Decode $\boldsymbol{r}'$ in $\mathcal{C}_{\mathrm{G}}[n, n-\Delta_{\mathrm{g}}]$ over $\mathbb{F}_{q^m}$ `// cf. Corollary 7.7 and 7.8`
if *decoder returned "decoding failure"* **then**
  return *"decoding failure"*
else
  return $\sigma^{-(n-\delta)}(\boldsymbol{c}')$

Corollary 7.7. *Let $\boldsymbol{t}, \boldsymbol{h}, \boldsymbol{\alpha}, \boldsymbol{\eta}$ be chosen as in Definition 7.1 such that the evaluation points span a subfield of $\mathbb{F}_{q^m}$. Then, we can correct up to*

$$\tau = \frac{n-k-\ell}{2(\ell+1)} \approx \frac{d}{2(\ell+1)}, \quad \text{with } d = n-k+1,$$

errors using Algorithm 22 in combination with a BMD decoder (cf. Section 5.1). The resulting complexity is $O^{\sim}(n^{\max\left\{\log_2(3), \min\{\frac{\omega+1}{2}, 1.635\}\right\}})$ in operations over $\mathbb{F}_{q^m}$.

Proof. Consider the notation in Theorem 7.6. Since the δ_i divide the interval $[k, n-1]$ into $\ell+1$ sub-intervals, the pigeonhole principle implies that we must have $(\ell+1)\Delta_{\mathrm{g}} + \ell \geq n-k$, so $\Delta_{\mathrm{g}} \geq \frac{n-k-\ell}{\ell+1}$. The code $\mathcal{C}_{\mathrm{G}}[n, n-\Delta_{\mathrm{g}}]$ can correct up to $\Delta_{\mathrm{g}}/2$ errors, which gives the claim. The complexity statement follows from the results in Chapter 5. □

Corollary 7.8. *Let $\boldsymbol{t}, \boldsymbol{h}$ be chosen as in Definition 7.1 and $\boldsymbol{\alpha}, \boldsymbol{\eta}$ as in Theorem 7.3 such that the evaluation points span a subfield $\mathbb{F}_{q^n} \leq \mathbb{F}_{q^m}$ with extension degree $h = [\mathbb{F}_{q^m} : \mathbb{F}_{q^n}]$ (note that $h \geq 2^\ell$). Then, the code $\mathcal{C}_{\mathrm{G}}[n, n-\Delta_{\mathrm{g}}]$ as in Theorem 7.6 can be interpreted as an interleaved Gabidulin code of interleaving degree h, and an error word $\boldsymbol{e} \in \mathbb{F}_{q^m}^n$ of rank weight t corresponds to t errors in the error model for vertically interleaved Gabidulin codes.*

This implies that for a random error $\boldsymbol{e} \in \mathbb{F}_{q^m}$ of rank weight t, the received word $\sigma^{n-\delta}(\boldsymbol{r})$ as in Theorem 7.6 can be corrected if

$$t < \tau := \tfrac{h}{h+1}\Delta_{\mathrm{g}},$$

with probability at least

$$1 - 3.5q^{-n[(2^\ell+1)(\tau-t)+1]},$$

which is $\geq 1 - \frac{4}{q^n}$ for $t < \tau$. The algorithm can be implemented in

$$O\Big(\min\Big\{h^3 \mathcal{M}_{q^m}(n)\log(n), hn^2\Big\}\Big)$$

operations over $\mathbb{F}_{q^n}$.

Proof. We choose a basis $\mathcal{B} = [\beta_1, \ldots, \beta_h]$ of $\mathbb{F}_{q^m}$ over $\mathbb{F}_{q^n}$ such that any element $f_i \in \mathbb{F}_{q^m}$ can be uniquely written as $f_i = \sum_{\mu=1}^{h} f_{i,\mu}\beta_\mu$, where $f_{i,\mu} \in \mathbb{F}_{q^n}$. Since the α_i span a subfield $\mathbb{F}_{q^n}$, the components of a codeword $\boldsymbol{c} \in \mathcal{C}_{\mathrm{G}}[n, n-\Delta_{\mathrm{g}}]$ over $\mathbb{F}_{q^m}$ can be written as

$$c_i = f(\alpha_i) = \sum_{j=0}^{n-\Delta_{\mathrm{g}}-1} f_j \sigma^j(\alpha_i) = \sum_{j=0}^{n-\Delta_{\mathrm{g}}-1} \left(\sum_{\mu=1}^{h} f_{j,\mu}\beta_\mu\right) \sigma^j(\alpha_i) = \sum_{\mu=1}^{h} \beta_\mu \left(\sum_{j=0}^{n-\Delta_{\mathrm{g}}-1} f_{j,\mu}\sigma^j(\alpha_i)\right),$$

where the $\sum_{j=0}^{n-\Delta_{\mathrm{g}}-1} f_{j,\mu}\sigma^j(\alpha_i)$ are entries of a codeword in the Gabidulin code $\mathcal{C}_{\mathrm{G}}[n, n-\Delta_{\mathrm{g}}]$ over the small field $\mathbb{F}_{q^n}$. Hence, we can interpret $\mathcal{C}_{\mathrm{G}}[n, n-\Delta_{\mathrm{g}}]$ over $\mathbb{F}_{q^m}$ as an h-interleaved Gabidulin code over $\mathbb{F}_{q^n}$. Since the error $\boldsymbol{e} \in \mathbb{F}_{q^m}$ is chosen at random of rank weight t, so is $\sigma^{n-\delta}(\boldsymbol{e})$ since $\sigma^{n-\delta}(\cdot)$ is a bijective rank-preserving mapping on $\mathbb{F}_{q^m}$. By expanding the error $\sigma^{n-\delta}(\boldsymbol{e})$ into a matrix of dimension $m \times n$ using a basis of $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$, we obtain an error matrix $\boldsymbol{E}_{\mathrm{v}}$ of rank t corresponding to t errors in the vertically interleaved error model.

Using Loidreau's decoding algorithm [LO06] or the adaption of the algorithm by Sidorenko, Jiang, and Bossert [SB10, SJB11] to the vertically interleaved error model in [Wac13, Section 4.1], we obtain the claimed decoding radius and failure probability.

The complexity follows by the results in Chapter 6. Note that since the α_i span a sub-field, their minimal subspace polynomial is of the form $x^n - 1$, so we can apply the skew-variant of Rosenkilde's demand-driven algorithm (Algorithm 19 on page 129). □

Remark 7.9. *The partial decoder implied by Corollary 7.8 has decoding radius*

$$\tau \geq \frac{2^\ell}{2^\ell+1}\frac{n-k-\ell}{\ell+1} = \frac{2^\ell}{2^\ell+1}\left(\frac{d}{\ell+1}-1\right), \quad \textit{with } d = n-k+1.$$

For $\ell = 1, 2, 3$, this is $\frac{2}{3}(\frac{d}{2}-1) \approx \frac{d}{3}$, $\frac{4}{5}(\frac{d}{3}-1)$, and $\frac{8}{9}(\frac{d}{4}-1)$. For large ℓ, we get $\tau \approx \frac{d}{\ell+1} - 1$.

7.4 Inequivalence to Other MRD Codes

The σ-sum of Evaluation Codes

It was observed in [HM17] that a linear code $\mathcal{C}[n,k]$ is a Gabidulin code with respect to an automorphism σ if and only if it fulfills $\dim(\mathcal{C} \cap \sigma(\mathcal{C})) = k-1$. Note that, for $k < n$, the latter condition implies

$$\dim(\mathcal{C} + \sigma(\mathcal{C})) = k+1,$$

which is a well-known property of Gabidulin codes and, e.g., is utilized to attack McEliece-like cryptosystems based on Gabidulin codes, cf. Section 7.5. Inspired by these observations, we consider the following notion in this section.

Definition 7.10. Let $\mathcal{C} \subseteq \mathbb{F}_{q^m}^n$ be a linear code and $\sigma \in \mathrm{Gal}(\mathbb{F}_{q^m}/\mathbb{F}_q)$. The *$\sigma$-sum of $\mathcal{C}$* is the subspace

$$\mathcal{C} + \sigma(\mathcal{C}) \subseteq \mathbb{F}_{q^m}^n.$$

We denote the dimension of the σ-sum by *σ-sum dimension*.

The following lemma implies that the σ-sum dimension of a linear code $\mathcal{C}$ is invariant under equivalence. We use the σ-sum for twisted Gabidulin codes analogously to the Schur square for twisted RS codes in Hamming metric (cf. Section 4.5 and Section 4.6), e.g., to show inequivalence of codes in rank metric.

Lemma 7.11. *Let $\mathcal{C}, \mathcal{C}' \subseteq \mathbb{F}_{q^m}^n$ be two linear codes that are equivalent w.r.t. the rank metric. Then, we have*

$$\dim(\mathcal{C} + \sigma(\mathcal{C})) = \dim(\mathcal{C}' + \sigma(\mathcal{C}'))$$

for any automorphism $\sigma \in \mathrm{Gal}(\mathbb{F}_{q^m}/\mathbb{F}_q)$.

Proof. Since $\mathcal{C}$ and $\mathcal{C}'$ are equivalent, by Definition 2.15 there is an automorphism $\sigma' \in \mathrm{Gal}(\mathbb{F}_{q^m}/\mathbb{F}_q)$ and a matrix $\boldsymbol{A} \in \mathrm{GL}_n(\mathbb{F}_q)$ such that[2]

$$\mathcal{C}' = \sigma'(\mathcal{C})\boldsymbol{A}.$$

Since $\sigma'(\cdot)\boldsymbol{A} : \mathbb{F}_{q^m} \to \mathbb{F}_{q^m}$ is a vector space isomorphism on $\mathbb{F}_{q^m}$, we know the following:

$$\sigma'(\mathcal{C} + \sigma(\mathcal{C}))\,\boldsymbol{A} = \sigma'(\mathcal{C})\boldsymbol{A} + \sigma'(\sigma(\mathcal{C}))\,\boldsymbol{A} = \sigma'(\mathcal{C})\boldsymbol{A} + \sigma(\sigma'(\mathcal{C})\boldsymbol{A}) = \mathcal{C}' + \sigma(\mathcal{C}').$$

This implies the claim. □

[2] In Definition 2.15, $\mathcal{C}$ is also multiplied by $\lambda \in \mathbb{F}_{q^m}^*$. This is not needed here since $\mathcal{C}, \mathcal{C}$ are linear over $\mathbb{F}_{q^m}$.

In order to apply Lemma 7.11 to an evaluation code $\mathcal{C}$, we need to easily compute $\sigma(\mathcal{C})$ for any automorphism σ. The following statement shows that $\sigma(\mathcal{C})$ is also an evaluation code and how its evaluation polynomials are connected to those of $\mathcal{C}$.

Lemma 7.12. *Let $\boldsymbol{\alpha} \in \mathbb{F}_{q^m}^n$ with linearly independent entries, $\mathcal{V} \subseteq \mathbb{F}_{q^m}[x;\sigma]_{<n}$ be an $\mathbb{F}_{q^m}$-subspace of evaluation polynomials, where σ is a generator of $\mathrm{Gal}(\mathbb{F}_{q^m}/\mathbb{F}_q)$, and define the evaluation code $\mathcal{C} = \mathrm{ev}_{\boldsymbol{\alpha}}(\mathcal{V})$. Furthermore, let $\sigma^i \in \mathrm{Gal}(\mathbb{F}_{q^m}/\mathbb{F}_q)$ for some $i \in \mathbb{N}_0$ (i.e., any automorphism in the group). Then, the code $\sigma^i(\mathcal{C})$ is an evaluation code with evaluation points $\boldsymbol{\alpha}$ and evaluation polynomials*

$$\mathcal{V}' = \Big\{ x^i f \mod_{\mathrm{r}} \mathcal{M}_{\langle \alpha_1,\dots,\alpha_n\rangle} \,:\, f \in \mathcal{V} \Big\} = \Big\{ \sigma^i(f) x^i \mod_{\mathrm{r}} \mathcal{M}_{\langle \alpha_1,\dots,\alpha_n\rangle} \,:\, f \in \mathcal{V} \Big\} \subseteq \mathbb{F}_{q^m}[x;\sigma]_{<n}.$$

In particular, if $\mathcal{V}$ is generated by $g_1,\dots,g_k \in \mathcal{V}$, then $\mathcal{V}'$ is also an $\mathbb{F}_{q^m}$-subspace of $\mathbb{F}_{q^m}[x;\sigma]_{<n}$ that is generated by the $\sigma^i(g_j)x^i \mod_{\mathrm{r}} \mathcal{M}_{\langle \alpha_1,\dots,\alpha_n\rangle}$ for $j = 1,\dots,k$.

Proof. For $f \in \mathbb{F}_{q^m}[x;\sigma]$, we have

$$\sigma^i(f(\alpha_j)) = (x^i \cdot f)(\alpha_j) = (\sigma^i(f)x^i)(\alpha_j) = (\sigma^i(f)x^i \mod_{\mathrm{r}} \mathcal{M}_{\langle \alpha_1,\dots,\alpha_n\rangle})(\alpha_j).$$

Let $\mathcal{V}$ be generated by $g_1,\dots,g_k \in \mathcal{V}$ and $f \in \mathcal{V}$. Then, there are $\beta_1,\dots,\beta_k \in \mathbb{F}_{q^m}$ such that $f = \sum_{\mu=1}^k \beta_\mu g_\mu$ and we have

$$\sigma^i(f)x^i = \sigma^i\Big(\sum_{\mu=1}^k \beta_\mu g_\mu\Big)x^i = \sum_{\mu=1}^k \sigma^i(\beta_\mu)\sigma^i(g_\mu)x^i.$$

Due to $\sigma^i(\beta_\mu) \in \mathbb{F}_{q^m}$, the evaluation polynomial space of $\mathcal{C}'$ is generated by the polynomials $\sigma^i(g_\mu)x^i \mod_{\mathrm{r}} \mathcal{M}_{\langle \alpha_1,\dots,\alpha_n\rangle}$. □

Remark 7.13. *If the α_i span a subfield of $\mathbb{F}_{q^m}$, then the minimal subspace polynomial $\mathcal{M}_{\langle \alpha_1,\dots,\alpha_n\rangle}$ is of the form $x^n - 1$. Thus, for a polynomial f of degree less than n, the polynomial $x^i f \mod_{\mathrm{r}} \mathcal{M}_{\langle \alpha_1,\dots,\alpha_n\rangle}$ is a cyclic shift of its coefficients, where the automorphism σ^i is applied to its coefficients.*

The following lemma shows how the sum of two linear evaluation codes is connected to the sum of their evaluation polynomial spaces.

Lemma 7.14. *Let $\boldsymbol{\alpha} \in \mathbb{F}_{q^m}$ with linearly independent entries, $\mathcal{V}, \mathcal{V}' \subseteq \mathbb{F}_{q^m}[x;\sigma]_{<n}$ be two $\mathbb{F}_{q^m}$-subspaces, and define the evaluation codes $\mathcal{C} = \mathrm{ev}_{\boldsymbol{\alpha}}(\mathcal{V})$ and $\mathcal{C}' = \mathrm{ev}_{\boldsymbol{\alpha}}(\mathcal{V}')$. Then, the sum of $\mathcal{C}$ and $\mathcal{C}'$ is also an evaluation code with*

$$\mathcal{C} + \mathcal{C}' = \mathrm{ev}_{\boldsymbol{\alpha}}(\mathcal{V} + \mathcal{V}').$$

Proof. This holds since $\mathrm{ev}_{\boldsymbol{\alpha}}(\cdot) : \mathbb{F}_{q^m}[x;\sigma]_{<n} \to \mathbb{F}_{q^m}^n$ is a vector space isomorphism. □

σ-sum of Known MRD Codes

Theorem 7.15. *Let $k < n$ and $\mathcal{C}_{\mathrm{G}}[n,k]$ be a Gabidulin code defined over the skew polynomial ring $\mathbb{F}_{q^m}[x;\sigma]$ with automorphism σ. Then,*

$$\dim(\mathcal{C}_{\mathrm{G}} + \sigma(\mathcal{C}_{\mathrm{G}})) = k + 1.$$

Proof. We know that $\mathcal{C}_{\mathrm{G}} = \mathrm{ev}_{\boldsymbol{\alpha}}(\mathcal{V})$, where $\mathcal{V} = \langle x^0, x^1, \ldots, x^{k-1} \rangle \subseteq \mathbb{F}_{q^m}[x;\sigma]_{<n}$. Furthermore, the evaluation polynomial space of $\sigma(\mathcal{C}_{\mathrm{G}})$ is given by $\mathcal{V}' = \langle x^1, \ldots, x^k \rangle \subseteq \mathbb{F}_{q^m}[x;\sigma]_{<n}$ (note that we require $k < n$). Thus, $\mathcal{V} + \mathcal{V}' = \langle x^0, \ldots, x^k \rangle$, and the claim follows by Lemma 7.14. □

Theorem 7.16. *Let $k < n-1$, $\eta \in \mathbb{F}_{q^m} \setminus \{0\}$, $\boldsymbol{\alpha} \in \mathbb{F}_{q^m}^n$ with linearly independent entries, and $\mathcal{C}_{\boldsymbol{\alpha},1,0,\eta}[n,k]$ be one of Sheekey's twisted Gabidulin codes defined over the skew polynomial ring $\mathbb{F}_{q^m}[x;\sigma]$ with automorphism σ. Then,*

$$\dim(\mathcal{C}_{\boldsymbol{\alpha},1,0,\eta} + \sigma(\mathcal{C}_{\boldsymbol{\alpha},1,0,\eta})) = k + 2.$$

Proof. The evaluation polynomials of $\mathcal{C}_{\boldsymbol{\alpha},1,0,\eta}$ are given by $\mathcal{V} = \langle x^1, \ldots, x^{k-1}, \eta x^k + x^0 \rangle$. Furthermore, $\sigma(\mathcal{C}_{\boldsymbol{\alpha},1,0,\eta})$ is obtained by evaluating $\mathcal{V}' = \langle x^2, \ldots, x^k, \sigma(\eta)x^{k+1} + x \rangle \subseteq \mathbb{F}_{q^m}[x;\sigma]_{<n}$ (note that we require $k < n-1$). Due to $(\eta x^k + x^0) - \eta x^k = x^0$, we have

$$\mathcal{V} + \mathcal{V}' = \langle x^0, \ldots, x^k, \sigma(\eta)x^{k+1} + x \rangle,$$

which implies the claim using Lemma 7.14. □

Inequivalence of Some Twisted Gabidulin Codes to Known MRD Codes

Theorem 7.17. *Let $4 < k < n-4$, $\ell = 1$, $\boldsymbol{t} = [t]$ with $2 < t < n-k-1$, $\boldsymbol{h} = [h]$ with $1 < h < k-2$, and $\boldsymbol{\eta} = [\eta]$ with $\eta \neq 0$. Furthermore, let $\boldsymbol{\alpha} \in \mathbb{F}_{q^m}$ such that the α_i are a basis of a subfield $\mathbb{F}_{q^n} \leq \mathbb{F}_{q^m}$. Let $\sigma \in \mathrm{Gal}(\mathbb{F}_{q^m}/\mathbb{F}_q)$ be a generator of the Galois group. Then, the corresponding twisted Gabdiulin code $\mathcal{C}_{\boldsymbol{\alpha},\boldsymbol{t},\boldsymbol{h},\boldsymbol{\eta}}[n,k]$ over $\mathbb{F}_{q^m}[x;\sigma]$, is not equivalent to any Gabidulin code or Sheekey twisted Gabidulin code ($\boldsymbol{t} = [1]$ and $\boldsymbol{h} = [0]$) defined over an arbitrary automorphism $\sigma' \in \mathrm{Gal}(\mathbb{F}_{q^m}/\mathbb{F}_q)$ that is not the identity map.*

Proof. Since σ is a generator of the Galois group, we can write $\sigma' = \sigma^i$ for some integer i. We show that

$$\dim\Big(\mathcal{C}_{\boldsymbol{\alpha},\boldsymbol{t},\boldsymbol{h},\boldsymbol{\eta}}[n,k] + \sigma^i\Big(\mathcal{C}_{\boldsymbol{\alpha},\boldsymbol{t},\boldsymbol{h},\boldsymbol{\eta}}[n,k]\Big)\Big) \geq k+3 \tag{7.6}$$

for all $i = 1, \ldots, n-1$, which proves the claim. Since the α_i form a subfield of $\mathbb{F}_{q^m}$, the unique evaluation polynomials of degree at most $n-1$, say $\mathcal{V}_i$, of $\sigma^i(\mathcal{C}_{\boldsymbol{\alpha},\boldsymbol{t},\boldsymbol{h},\boldsymbol{\eta}}[n,k])$ are cyclic shifts of the ones, say $\mathcal{V}$, of $\mathcal{C}_{\boldsymbol{\alpha},\boldsymbol{t},\boldsymbol{h},\boldsymbol{\eta}}[n,k]$ by i positions, where σ^i is applied to all coefficients, cf. Remark 7.13.

By symmetry, a shift by i gives the same dimension as a shift by $n-i$, so it suffices to consider only those shifts by $i = 1, \ldots, \lceil (n-1)/2 \rceil$. We distinguish the following cases:

- Case $i \geq 4$: The evaluation polynomial set $\mathcal{V}$ contains skew polynomials of degrees

  $$D = \{0, \ldots, k-1\} \setminus \{h\},$$

  which are $k-1$ distinct ones. Due to $4 \leq i \leq n/2$, $n-k \geq 5$, and $n \geq 10$ (this is implied by $4 < k < n-4$), there are polynomials of at least four distinct degrees $\geq k$ in $\mathcal{V}_i$:
  - Three monomials and $(\sigma^i(\eta)x^{i+k-1+t} + x^{h+i}) \bmod_{\mathrm{r}} x^n - 1$ if $k-1 < i+h < n$ or
  - four monomials otherwise.

In total, there are polynomials of at least $(k-1)+4 = k+3$ distinct degrees in $\mathcal{V}+\mathcal{V}_i$, so we have

$$\dim(\mathcal{V}+\mathcal{V}_i) \geq k+3.$$

- Case $i = 3$: The set $\mathcal{V}_i$ contains polynomials of at least three distinct degrees $\geq k$, i.e.,

$$\begin{aligned} D_i &= \{k, k+1, k+2\}, && \text{if } t \geq n-k-3, \text{ and} \\ D_i' &= \{k+1, k+2, k-1+t\}, && \text{if } t < n-k-3 \text{ (due to } h < k-2). \end{aligned}$$

  Furthermore, the set $\mathcal{V}$ contains all monomials of degree $D = \{0, \ldots, k-1\} \setminus \{h\}$, as well as the polynomial $g = \eta x^{k-1+t} + x^h$.

  - For $t > 3$, the degree of g is neither contained in D_i nor in D_i', so in total, there are polynomials of $k+3$ distinct degrees in $\mathcal{V}+\mathcal{V}_i$.
  - For $t = 3$, we have $x^{k+2} \in \mathcal{V}_i$, so $g - \eta x^{k-2} = x^h \in \mathcal{V}+\mathcal{V}_i$, which means that $\mathcal{V}+\mathcal{V}_i$ contain polynomials of $k+3$ distinct degrees.

- Case $i \in \{1, 2\}$: The sets $\mathcal{V}$ and $\mathcal{V}_i$ contain polynomials of degrees

$$\begin{aligned} D &= (\{0, \ldots, k-1\} \setminus \{h\}) \cup \{k-1+t\}, \quad \text{and} \\ D_i &= (\{i, \ldots, i+k-1\} \setminus \{i+h\}) \cup \{i+k-1+t\}, \end{aligned}$$

  respectively (note that $t \leq n-k-2$ implies $i+k-1+t < n$). Due to $1 < h < k-2$ and $t > 2$, we have $1 < h < i+h < k$ and $k-1+t > k+1$, which gives

$$|D \cup D_i| = |\{0, \ldots, k-1+i, k-1+t, k-1+t+i\}| \geq k+3.$$

In total, we have $\dim(\mathcal{V}+\mathcal{V}_i) \geq k+3$ for all $i = 1, \ldots, n-1$, which implies the claim using Lemma 7.14. □

Remark 7.18. *The conditions on t and h in Theorem 7.17 are sufficient but not necessary for (7.6) to hold. However, for fixed n and k, there are at most 4 values of t and 4 values of h, i.e., at most 16 pairs, that do not fulfill them. Hence, most twisted Gabidulin codes with one twist ($\ell = 1$) are inequivalent to both Gabidulin and Sheekey's twisted Gabidulin codes.*

It is technical but straightforward to generalize the statement of Theorem 7.17 to certain codes with multiple twists $\ell > 1$.

In any case (i.e., also if the evaluation points are arbitrary), Theorem 7.17 provides a powerful tool to prove inequivalence of a twisted Gabidulin code to Gabidulin or Sheekey's twisted Gabidulin codes in polynomial time.

The following example shows that there are twisted Gabidulin codes with $[t, h] \neq [1, 0]$ that have σ-sum dimension $k+2$ for a suitable automorphism σ.

Example 7.19. *Consider $3 < k < n-4$, $h = k-2$, $t = 2$, and $i = 2$: In this case, $\mathcal{V}$ and $\mathcal{V}_i$ are spanned by the polynomials*

$$\begin{aligned} B &= \left\{x^0, \ldots, x^{k-3}, x^{k-1}, \eta x^{k+1} + x^{k-2}\right\}, \quad \text{and} \\ B_i &= \left\{x^2, \ldots, x^{k-1}, x^{k+1}, \sigma^2(\eta x^{k+3}) + x^k\right\}, \end{aligned}$$

respectively. Hence, the basis elements $x^2, \ldots, x^{k-3}, x^{k-1}, \eta x^{k+1} + x^{k-2}$ *of* $\mathcal{V}$ *are all contained in* $\mathcal{V}_i$*, and we have*

$$\dim(\mathcal{V} + \mathcal{V}_i) = k + 2.$$

It is an open problem whether this code is equivalent to one of Sheekey's twisted Gabidulin codes with $\sigma' = \sigma^2$*.*

7.5 Twisted Gabidulin Codes in the GPT Cryptosystem

In 1991, Gabidulin, Paramonov, and Tretjakov proposed the GPT cryptosystem [GPT91], which can be seen as the rank-metric analog of the McEliece cryptosystem and uses Gabidulin codes. Since the general decoding problem in rank metric appears to be much harder than its Hamming-metric analog, the system has the potential to achieve significantly smaller key sizes than the McEliece cryptosystem at the same security level. Since its invention, many efficient attacks on the system have been found, e.g., [Gib95, Gib96, Ove05, Ove06, Ove08]. There are also several repairs of the systems and corresponding modifications of the mentioned attacks.

The most prominent structural attack is the polynomial-time method by Overbeck [Ove08], which is based on the fact that for a Gabidulin code $\mathcal{C}_{\mathrm{G}}[n,k]$ defined over $\mathbb{F}_{q^m}[x;\sigma]$, the dimension of the sum

$$\Upsilon_i(\mathcal{C}_{\mathrm{G}}) = \mathcal{C}_{\mathrm{G}} + \sigma(\mathcal{C}_{\mathrm{G}}) + \cdots + \sigma^i(\mathcal{C}_{\mathrm{G}})$$

is $\dim(\Upsilon_i(\mathcal{C}_{\mathrm{G}})) = \min\{k+i, n\}$. This also distinguishes a Gabidulin code from a random code $\mathcal{C}[n,k]$, which fulfills $\dim(\Upsilon_i(\mathcal{C})) = \min\{2k, n\}$ with high probability, cf. [Ove08]. Note the analogy to the Schur square dimension distinguisher for Reed–Solomon codes in [CGGU$^+$14].

Using similar arguments as in the previous section, we obtain an upper bound on the sum's dimension $\dim(\Upsilon_i(\mathcal{C}_{\boldsymbol{\alpha},\boldsymbol{t},\boldsymbol{h},\boldsymbol{\eta}}[n,k]))$ of a twisted Gabidulin code with ℓ twists as follows:

$$\dim\Big(\Upsilon_i(\mathcal{C}_{\boldsymbol{\alpha},\boldsymbol{t},\boldsymbol{h},\boldsymbol{\eta}}[n,k])\Big) \leq \min\{k + (\ell+1)i, n\}.$$

Hence, we would need at least $\ell \approx k$ twists in order to protect a twisted Gabidulin code against this distinguisher, resulting in codes over huge fields compared to their length. A few twists might nevertheless make the Overbeck attack infeasible for some parameters, which should be studied further.

7.6 Concluding Remarks

In this chapter, we have presented a generalization of Sheekey's twisted Gabidulin codes, where we added ℓ additional monomials to the low-degree evaluation polynomials. We have shown that the new code class contains MRD codes of length $2^{-\ell}m$ over the field $\mathbb{F}_{q^m}$.

We have proposed a partial decoder that is able to decode these MRD codes up to one third of their minimum distance, $d/3$, for $\ell = 1$, but only up to $d/(\ell+1)$ for $\ell \to \infty$. Hence, finding an algorithm for decoding up to or beyond half-the-minimum distance is an open problem. Supposedly, such an algorithm must utilize the algebraic structure of the evaluation codes, similar to the decoder for Sheekey's twisted Gabidulin codes in [RR17].

Furthermore, we have studied the σ-sums of twisted Gabidulin codes, which can be used as a tool—analog to the Schur square dimension of twisted RS codes—to determine whether a

code is equivalent to a Gabidulin or Sheekey's twisted Gabidulin code in polynomial time in the code length. We have used the σ-sum to show inequivalence of a large subclass of twisted Gabidulin codes to the latter classes. Since Gabidulin and Sheekey's twisted Gabidulin codes are—to the best of our knowledge—the only known classes of MRD codes that are linear and exist for more than a few parameters, this result is of fundamental theoretical interest.

Both Schur squares of twisted RS codes and the σ-sums of twisted Gabidulin codes serve as tools for showing inequivalence to other code classes, as well as distinguishers in cryptosystems, for the respective code classes. It is worth mentioning that the σ-sum is more powerful in both scenarios: The equivalence proofs are not restricted to low-rate codes (in contrast to arguments based on the Schur square). Moreover, one needs many more twists in order to hide a twisted Gabidulin code from the σ-sum distinguisher than a twisted RS code from the Schur square dimension distinguisher.

Finding subfamilies of MRD twisted Gabidulin codes of length $n = m$, other than Sheekey's codes, is an open problem. Since for twisted RS codes, we know two choices of the twist and hook vectors resulting in long MDS codes (cf. Section 4.7), where one case is the Hamming-metric analog of Sheekey's codes, a first idea would be to study the second parameter choice (i.e., the (+)-twisted codes) in the rank metric.

Moreover, the new twisted Gabidulin codes should be further analyzed in the context of the GPT cryptosystem. If they were used in the original GPT cryptosystem without modification, the codes would be vulnerable to Overbeck-like structural attacks, as discussed in Section 7.5. However, the codes might work well with modifications of the system, such as Loidreau's recent proposal [Loi16].

It is also an open question whether twisted Gabidulin codes can be used in other applications. Since they contain codes inequivalent to Gabidulin codes, we might obtain more degrees of freedom when constructing Ferrers diagram codes based on MRD codes, cf. [EGRW16]. Furthermore, we should analyze space-time codes based on twisted Gabidulin codes using the constructions in [GBL00, BGL02, LFT02, LGB03, PSBF16].

8

Conclusion

In this dissertation, we have covered constructions and decoders of evaluation codes, both in Hamming and rank metric. In the Hamming-metric part, the studied codes are interleaved Reed–Solomon, interleaved one-point Hermitian, and the new twisted Reed–Solomon codes. The considered rank-metric codes are Gabidulin, interleaved Gabidulin, and the new generalizations of twisted Gabidulin codes. Although the codes are defined over different metrics, their methods of construction and decoding closely resemble each other.

In Chapter 3, we have proposed new decoders for interleaved Reed–Solomon and interleaved one-point Hermitian codes by combining the ideas of several known algorithms (cf. Figure 3.2 on page 27). As all comparable algorithms, the new decoders are partial, i.e., fail for some codewords above half the minimum distance. While we do not have an analytic expression for the failure probability yet, various simulation results indicate that the new decoders achieve the derived maximal decoding radii. For the IRS decoder, this radius attains the same value as the previous-best algorithms, but at a better complexity due the missing root-finding step. In case of IH codes, the radius improves upon all known decoders.

In Chapter 4, we have constructed a new class of codes designed for the Hamming metric: Twisted RS codes. The code construction is inspired by Sheekey's rank-metric twisted Gabidulin codes, and is obtained from an RS code by adding ℓ many additional monomials to the evaluation polynomials whose coefficients depend on the k lowest coefficients. By choosing the evaluation points in a suitable way, we have given a large constructive subclass of MDS codes containing rather short codes, as well as two smaller subclasses containing "long" codes of length approximately half the field size. We have proposed a decoder based on RS decoders that is efficient for small ℓ and showed that certain subclasses of twisted RS codes are closed under duality. Furthermore, we have studied the Schur squares of the new codes, which revealed that many twisted RS codes are inequivalent to RS codes and that certain subclasses of twisted RS codes are resistant against some known structural attacks on the McEliece cryptosystem based on RS codes.

In the rank-metric part, we have first studied half-the-minimum-distance decoding of Gabidulin codes in Chapter 5. We have recalled the known decoder by Wachter-Zeh, Afanassiev, and Sidorenko, whose runtime is directly determined by the cost of the skew polynomial operations multiplication, division, multi-point evaluation, interpolation and minimal subspace polynomial computation. By accelerating these operations to sub-quadratic time in their input size (cf. Table 5.1 on page 98), independent of the field size, we have obtained the first decoder for Gabidulin codes of sub-quadratic runtime in the code length.

In Chapter 6, we have shown that the main steps of several known algorithms for decoding

interleaved Gabidulin codes, both key-equation- (cf. Table A.1) and interpolation-based, can be solved by row reduction of certain skew polynomial matrices. The approach continues the development of a unified row-reduction-based description of decoding algorithms by several recent works that have shown analogous results for many decoders of Reed–Solomon and one-point Hermitian codes. We have adapted several row reduction algorithms from ordinary polynomial rings to skew polynomials and in this way obtained simple, general, and in some cases faster (cf. Table 6.1 on page 133) algorithms.

In Chapter 7, we have proposed a generalization of Sheekey's twisted Gabidulin codes, analog to the construction of the twisted RS codes in Chapter 4. By giving a constructive subclass of MRD codes and showing that many of these codes are inequivalent to both Gabidulin and Sheekey's twisted Gabidulin codes, we have found a new class of MRD codes.

Open research problems have been given in the conclusion section of each chapter.

A

Appendices

A.1 Efficiently Decoding Interleaved Reed–Solomon and One-Point Hermitian Codes

We discuss the following problem and argue that, under certain conditions, Problem 3.9 (over $\mathbb{F} = \mathbb{F}_q$) and Problem 3.34 (over $\mathbb{F} = \mathbb{F}_{q^2}$), are instances of it. The problem is a slight generalization of the "2D key equation problem" in [Nie13b, Section 2.5], where the latter is obtained by setting $\mathcal{I}' = \mathcal{I}$ in the problem statement.

Problem A.1. *Let $\mathcal{I}, \mathcal{J}$ be index sets and $\mathcal{I}' \subseteq \mathcal{I}$ be a non-empty subset. Furthermore, let $\nu, \eta_{\boldsymbol{i}}, w_{\boldsymbol{j}} \in \mathbb{N}_0$, $S_{\boldsymbol{i},\boldsymbol{j}} \in \mathbb{F}[x]$ and $G_{\boldsymbol{j}} \in \mathbb{F}[x]$ for all $\boldsymbol{i} \in \mathcal{I}$ and $\boldsymbol{j} \in \mathcal{J}$. Find $\lambda_{\boldsymbol{i}}, \psi_{\boldsymbol{j}} \in \mathbb{F}[x]$ for $\boldsymbol{i} \in \mathcal{I}$ and $\boldsymbol{j} \in \mathcal{J}$, not all zero, such that*

$$\sum_{\boldsymbol{i} \in \mathcal{I}} \lambda_{\boldsymbol{i}} S_{\boldsymbol{i},\boldsymbol{j}} \equiv \psi_{\boldsymbol{j}} \mod G_{\boldsymbol{j}} \qquad \forall\, \boldsymbol{j} \in \mathcal{J}, \tag{A.1}$$

$$\nu \deg \lambda_{\boldsymbol{i}} + \eta_{\boldsymbol{i}} < \max_{\boldsymbol{i} \in \mathcal{I}'} \{\nu \deg \lambda_{\boldsymbol{i}} + \eta_{\boldsymbol{i}}\} \qquad \forall\, \boldsymbol{i} \in \mathcal{I}, \tag{A.2}$$

$$\nu \deg \psi_{\boldsymbol{j}} + w_{\boldsymbol{j}} < \max_{\boldsymbol{i} \in \mathcal{I}'} \{\nu \deg \lambda_{\boldsymbol{i}} + \eta_{\boldsymbol{i}}\} \qquad \forall\, \boldsymbol{j} \in \mathcal{J}, \tag{A.3}$$

and minimal $\max_{\boldsymbol{i} \in \mathcal{I}'} \{\nu \deg \lambda_{\boldsymbol{i}} + \eta_{\boldsymbol{i}}\}$.

Let $\boldsymbol{i}_1, \ldots, \boldsymbol{i}_{|\mathcal{I}|}$ and $\boldsymbol{j}_1, \ldots, \boldsymbol{j}_{|\mathcal{J}|}$ be ordered lists of the elements in $\mathcal{I}$ and $\mathcal{J}$, respectively. Any solution $[\lambda_{\boldsymbol{i}_1}, \ldots, \lambda_{\boldsymbol{i}_{|\mathcal{I}|}}, \psi_{\boldsymbol{j}_1}, \ldots, \psi_{\boldsymbol{j}_{|\mathcal{J}|}}]$ of the problem is in the row space of the following $\mathbb{F}[x]$-module basis:

$$\boldsymbol{M} = \begin{bmatrix} 1 & & & S_{\boldsymbol{i}_1,\boldsymbol{j}_1} & \cdots & S_{\boldsymbol{i}_1,\boldsymbol{j}_{|\mathcal{J}|}} \\ & \ddots & & \vdots & \ddots & \vdots \\ & & 1 & S_{\boldsymbol{i}_{|\mathcal{I}|},\boldsymbol{j}_1} & \cdots & S_{\boldsymbol{i}_{|\mathcal{I}|},\boldsymbol{j}_{|\mathcal{J}|}} \\ & & & G_{\boldsymbol{i}_1} & & \\ & & & & \ddots & \\ & & & & & G_{\boldsymbol{i}_{|\mathcal{J}|}} \end{bmatrix}$$

Similar to the $\mathbb{F}_{q^m}[x;\sigma]$-module definitions in Section 2.3.2, we say that an $\mathbb{F}[x]$-module basis is in *weak Popov form* if the *leading positions* of its rows, i.e., the right-most position of a maximal degree element in one row, are distinct. By the same arguments as in [Nie13b,

Section 2.5], we thus obtain a solution of Problem A.1 by transforming the matrix

$$\boldsymbol{M}' = \begin{bmatrix} 1 & & & S_{\boldsymbol{i}_1,\boldsymbol{j}_1}(x^\nu) & \dots & S_{\boldsymbol{i}_1,\boldsymbol{j}_{|\mathcal{J}|}}(x^\nu) \\ & \ddots & & \vdots & \ddots & \vdots \\ & & 1 & S_{\boldsymbol{i}_{|\mathcal{I}|},\boldsymbol{j}_1}(x^\nu) & \dots & S_{\boldsymbol{i}_{|\mathcal{I}|},\boldsymbol{j}_{|\mathcal{J}|}}(x^\nu) \\ & & & G_{\boldsymbol{i}_1}(x^\nu) & & \\ & & & & \ddots & \\ & & & & & G_{\boldsymbol{i}_{|\mathcal{J}|}}(x^\nu) \end{bmatrix} \cdot \begin{bmatrix} x^{\eta_{\boldsymbol{i}_1}} & & & & & \\ & \ddots & & & & \\ & & x^{\eta_{\boldsymbol{i}_{|\mathcal{I}|}}} & & & \\ & & & x^{w_{\boldsymbol{j}_1}} & & \\ & & & & \ddots & \\ & & & & & x^{w_{\boldsymbol{j}_{|\mathcal{J}|}}} \end{bmatrix} \tag{A.4}$$

into weak Popov form and extracting the row of minimal degree among all rows of leading positions in $\mathcal{I}'$. Such a row has the form

$$[\lambda_{\boldsymbol{i}_1}(x^\nu)x^{\eta_{\boldsymbol{i}_1}}, \dots, \lambda_{\boldsymbol{i}_{|\mathcal{I}|}}(x^\nu)x^{\eta_{\boldsymbol{i}_{|\mathcal{I}|}}}, \psi_{\boldsymbol{j}_1}(x^\nu)x^{w_{\boldsymbol{j}_1}}, \dots, \psi_{\boldsymbol{j}_{|\mathcal{J}|}}(x^\nu)x^{w_{\boldsymbol{j}_{|\mathcal{J}|}}}],$$

where in each polynomial the indeterminate is replaced by $x \mapsto x^\nu$ and the degrees are shifted by $[\eta_{\boldsymbol{i}_1}, \dots, \eta_{\boldsymbol{i}_{|\mathcal{I}|}}, w_{\boldsymbol{j}_1}, \dots, w_{\boldsymbol{j}_{|\mathcal{J}|}}]$. Thus, we can easily compute the $\lambda_{\boldsymbol{i}}$ and $\psi_{\boldsymbol{j}}$ from it. The complexity is determined by well-known algorithms for transforming $\mathbb{F}[x]$-matrices into weak Popov form, see e.g. [MS03, GJV03, Ale05, Nie13a, Nie13b]. The asymptotically fastest one, [GJV03], on the above input problem has asymptotic complexity

$$O^\sim\left((|\mathcal{I}| + |\mathcal{J}|)^\omega \, \nu^{-1} \operatorname{maxdeg}(\boldsymbol{M}')\right).$$

We therefore obtain the following complexities for solving Problem 3.9 and Problem 3.34, respectively.

Solving Problem 3.9 (Interleaved Reed–Solomon Codes) Efficiently

Theorem A.2. *Consider Problem 3.9 on page 31 with input $s \le \ell$, $A_{\boldsymbol{i},\boldsymbol{j}} \in \mathbb{F}_q[x]$ for all $\boldsymbol{i} \in \mathcal{I} := \{\boldsymbol{i} \in \mathbb{N}_0^h : 0 \le |\boldsymbol{i}| < s\}$ and $\boldsymbol{j} \in \mathcal{J} := \{\boldsymbol{j} \in \mathbb{N}_0^h : 1 \le |\boldsymbol{j}| \le \ell\}$ as in (3.3), and $G \in \mathbb{F}_q[x]$ as in Definition 3.1.*

Let $T = sn$. Then, $\lambda_{\boldsymbol{i}}, \psi_{\boldsymbol{j}}$ for $\boldsymbol{i} \in \mathcal{I}$ and $\boldsymbol{j} \in \mathcal{J}$ is a minimal solution of Problem 3.9 if and only if it is a minimal solution with monic $\lambda_{\boldsymbol{0}}$ of Problem A.1 with input

$$\begin{aligned} S_{\boldsymbol{i},\boldsymbol{j}} &= A_{\boldsymbol{i},\boldsymbol{j}}, \\ G_{\boldsymbol{j}} &= \begin{cases} x^{T+|\boldsymbol{j}|(n-1)+1}, & \text{if } |\boldsymbol{j}| < s, \\ G^s, & \text{else}, \end{cases} \\ \eta_{\boldsymbol{i}} &= \begin{cases} 1 + \ell(k-1), & \text{if } \boldsymbol{i} = \boldsymbol{0}, \\ |\boldsymbol{i}| + \ell(k-1), & \text{else}, \end{cases} \\ w_{\boldsymbol{j}} &= (\ell - |\boldsymbol{j}|)(k-1) \\ \nu &= 1, \\ \mathcal{I}' &= \{\boldsymbol{0}\}, \end{aligned}$$

for all $\boldsymbol{i} \in \mathcal{I}$ and $\boldsymbol{j} \in \mathcal{J}$. In this case, Problem A.1 can be solved in

$$O^\sim\left(\tbinom{h+\ell}{h}^\omega \ell n\right)$$

operations over $\mathbb{F}_q$, where ω is the matrix multiplication exponent.

Proof. The degree restrictions (3.6) and (3.7) coincide with (A.2) and (A.3). Note that $\ell(k-1)$ is added to all degree "shifts" of Problem A.1 in order to make them non-negative. For $|\boldsymbol{j}| \geq s$, also the congruence (3.5) coincides with (A.1).

The only point in which the two problems differ is the equality (3.4), which is a congruence modulo $G^{T+|\boldsymbol{j}|(n-1)+1}$ for $|\boldsymbol{j}| < s$ in (A.1). We show that a minimal solution is not affected by this modification. It is clear that Problem 3.9 has a solution of degree at most T: Take an arbitrary codeword and construct the solution corresponding to its error locator, which has degree at most n, so the corresponding solution has degree $\leq sn = T$. Thus, a minimal solution fulfills $\deg \lambda_{\boldsymbol{i}} \leq T + |\boldsymbol{i}|$ and

$$\deg\Big(\sum_{\boldsymbol{i}\in\mathcal{I}} \lambda_{\boldsymbol{i}} A_{\boldsymbol{i},\boldsymbol{j}}\Big) \leq T + |\boldsymbol{j}|(n-1) < \deg\left(x^{T+|\boldsymbol{j}|(n-1)+1}\right).$$

Since also $\deg \psi_{\boldsymbol{j}} \leq T + |\boldsymbol{j}|(k-1) \leq T + |\boldsymbol{j}|(n-1)$, taking (3.4) modulo $G^{T+|\boldsymbol{j}|(n-1)+1}$ has no effect, which proves that the minimal solutions of the two problems are the same.

As for the complexity, the matrix $\boldsymbol{M}'$ as in (A.4) has maximal degree $\operatorname{maxdeg} \boldsymbol{M}' \in O(\ell n)$ and $|\mathcal{I}|, |\mathcal{J}| \in O(\binom{h+\ell}{h})$ (cf. Lemma 3.15 on page 33), which implies the complexity statement. □

Solving Problem 3.34 (Interleaved One-Point Hermitian Codes) Efficiently

Theorem A.3. *Consider Problem 3.34 on page 43 with input $\ell, s \in \mathbb{N}$ with $s \leq \ell$, $\boldsymbol{R}, G \in \mathcal{R}$ as in Section 3.2.1, and*

$$\boldsymbol{A}^{(\boldsymbol{i},\boldsymbol{j})} := \boldsymbol{\mu}(A_{\boldsymbol{i},\boldsymbol{j}})\boldsymbol{\Xi} = \boldsymbol{\mu}\left(\tbinom{\boldsymbol{j}}{\boldsymbol{i}} \boldsymbol{R}^{\boldsymbol{j}-\boldsymbol{i}} G^{|\boldsymbol{i}|}\right)\boldsymbol{\Xi} \in \mathbb{F}_{q^2}[x]^{q\times q}$$

for all $\boldsymbol{i} \in \overline{\mathcal{I}} := \{\boldsymbol{i} \in \mathbb{N}_0^h : 0 \leq |\boldsymbol{i}| < s\}$ and $\boldsymbol{j} \in \overline{\mathcal{J}} := \{\boldsymbol{j} \in \mathbb{N}_0^h : 1 \leq |\boldsymbol{j}| \leq \ell\}$.

Let $T = sn + g$. Then, $\lambda_{\boldsymbol{i},\iota}, \psi_{\boldsymbol{j},\kappa} \in \mathbb{F}_{q^2}[x]$ for $[\boldsymbol{i},\iota] \in \mathcal{I} := \{[\boldsymbol{i},\iota] : \boldsymbol{i} \in \overline{\mathcal{I}}, \iota \in [q)\}$, $[\boldsymbol{j},\kappa] \in \mathcal{J} := \{[\boldsymbol{j},\kappa] : \boldsymbol{j} \in \overline{\mathcal{J}}, \kappa \in [q)\}$ is a minimal solution of Problem 3.34 if and only if it is a solution of Problem A.1 with input

$$\begin{aligned}
S_{[\boldsymbol{i},\iota],[\boldsymbol{j},\kappa]} &= A^{(\boldsymbol{i},\boldsymbol{j})}_{\iota,\kappa},\\
G_{[\boldsymbol{j},\kappa]} &= \begin{cases} x^{\lceil \frac{T+|\boldsymbol{j}|(n+2g-1)+1}{q} \rceil}, & \text{if } |\boldsymbol{j}| < s,\\ G^s, & \text{else,}\end{cases}\\
\eta_{\boldsymbol{i}} &= \begin{cases} \ell m_{\mathrm{H}} - |\boldsymbol{i}|(2g-1) + \iota(q+1) + 1, & \text{if } [\boldsymbol{i},\iota] \in \mathcal{I}',\\ \ell m_{\mathrm{H}} - |\boldsymbol{i}|(2g-1) + \iota(q+1), & \text{else,}\end{cases}\\
w_{\boldsymbol{j}} &= (\ell - |\boldsymbol{j}|)m_{\mathrm{H}} + \kappa(q+1)\\
\nu &= q,\\
\mathcal{I}' &= \{[\boldsymbol{i},\iota] \in \mathcal{I} : \boldsymbol{i} = \boldsymbol{0}\},
\end{aligned}$$

for all $[\boldsymbol{i},\iota] \in \mathcal{I}$ and $[\boldsymbol{j},\kappa] \in \mathcal{J}$. In this case, Problem 3.34 can be solved in

$$O^{\sim}\left(\tbinom{h+\ell}{h}^{\omega} \ell n^{\frac{\omega+2}{3}}\right)$$

operations over $\mathbb{F}_{q^2}$, where ω is the matrix multiplication exponent.

Proof. By definition, the congruences for $|\boldsymbol{j}| \geq s$ and all degree restrictions of the two problems agree. It is only left to show that for a minimal solution of either of the problem, we have

$$\psi_{\boldsymbol{j},\kappa} = \sum_{[\boldsymbol{i},\iota]\in\mathcal{I}} \lambda_{\boldsymbol{i},\iota} A_{\iota,\kappa}^{(\boldsymbol{i},\boldsymbol{j})} \quad \Leftrightarrow \quad \psi_{\boldsymbol{j},\kappa} \equiv \sum_{[\boldsymbol{i},\iota]\in\mathcal{I}} \lambda_{\boldsymbol{i},\iota} A_{\iota,\kappa}^{(\boldsymbol{i},\boldsymbol{j})} \mod x^{\lceil \frac{T+|\boldsymbol{j}|(n+2g-1)+1}{q} \rceil}$$

for any $[\boldsymbol{j},\kappa] \in \mathcal{J}$ with $1 \leq |j| < s$.

Problem 3.34 has a solution of degree $\leq T$ since any codeword corresponds to an error locator of multiplicity s of degree at most $\deg_{\mathcal{H}} \Lambda^{(s)} \leq sn + g = T$, which gives a solution of the problem by Theorem 3.33. Hence, there are $\lambda_{\boldsymbol{i}} \in \mathcal{R}$ with $[\lambda_{\boldsymbol{i},0}, \ldots, \lambda_{\boldsymbol{i},q-1}] = \boldsymbol{\nu}(\lambda_{\boldsymbol{i}})$ and

$$\deg_{\mathcal{H}} \Big(\sum_{\boldsymbol{i}\in\overline{\mathcal{I}}} \lambda_{\boldsymbol{i}} A_{\boldsymbol{i},\boldsymbol{j}} \Big) \leq T + |\boldsymbol{j}|(n+2g-1).$$

Since $\sum_{[\boldsymbol{i},\iota]\in\mathcal{I}} \lambda_{\boldsymbol{i},\iota} A_{\iota,\kappa}^{(\boldsymbol{i},\boldsymbol{j})}$ is the κ^{th} component of the vector representation of $\sum_{\boldsymbol{i}\in\overline{\mathcal{I}}} \lambda_{\boldsymbol{i}} A_{\boldsymbol{i},\boldsymbol{j}}$, we must have

$$\deg \Big(\sum_{[\boldsymbol{i},\iota]\in\mathcal{I}} \lambda_{\boldsymbol{i},\iota} A_{\iota,\kappa}^{(\boldsymbol{i},\boldsymbol{j})} \Big) \leq \tfrac{T+|\boldsymbol{j}|(n+2g-1)}{q} \quad \forall\, [\boldsymbol{j},\kappa] \in \mathcal{J}.$$

Since also $\deg \psi_{\boldsymbol{j},\kappa} \leq \frac{T+|\boldsymbol{j}|m_{\mathrm{H}}}{q}$, the modulo reduction does not influence the equality in case of a minimal solution.

Turning to complexity, the matrix $\boldsymbol{M}'$ as in (A.4) has maximal degree $\operatorname{maxdeg} \boldsymbol{M}' \in O(\ell n)$ and $|\mathcal{I}|, |\mathcal{J}| \in O(q\binom{h+\ell}{h})$ (cf. Lemma 3.15 on page 33). Thus, the problem can be solved in

$$O^{\sim}\Big(\Big(q\tbinom{h+\ell}{h}\Big)^{\omega} q^{-1}\ell n\Big) \subseteq O^{\sim}\Big(\tbinom{h+\ell}{h}^{\omega} \ell n^{\frac{\omega+2}{3}}\Big)$$

operations over $\mathbb{F}_{q^2}$. □

A.2 Proof of Theorem 4.21 (Schur Squares of Shortened Codes)

Proof of Theorem 4.21 on page 67. As in the proof of Theorem 4.18, we first show that the hook and twist parameters are well-defined.

- Due to $\ell < \frac{2n}{k} - 2$, we have $r > \frac{k}{2} + \varrho > 0$. Hence, $h_i > 0$, $t_i \geq 2(r-\varrho) - k + \varrho > 0$, and the t_i are distinct.
- Due to $\frac{n+1}{k-\sqrt{n}} - 2 < \ell$ and $\ell < \sqrt{n} - 2 - \varrho$, we have $\frac{n+1}{k-(\ell+2+\varrho)} < \frac{n+1}{k-\sqrt{n}} < \ell + 2$, and

$$h_i \leq r - 1 + \ell \leq \tfrac{n+1}{\ell+2} + \varrho + \ell < k.$$

- By $\ell < \sqrt{n} - 2 - \varrho$, we have $\ell^2 + (4+\varrho)\ell + 3 + 2\varrho \leq (\ell + 2 + \varrho)^2 < n$, so $\frac{n+1}{\ell+2} + 1 \leq \frac{n-\varrho}{\ell+1}$, and

$$t_i \leq (\ell+1)(r-\varrho) - k + \varrho \leq (\ell+1)(\tfrac{n+1}{\ell+2} + 1) - k + \varrho \leq (\ell+1)\tfrac{n-\varrho}{\ell+1} - k + \varrho = n - k.$$

Hence, the h_i and t_i are valid parameters by Definition 4.1.

Since shortening at one position i reduces the Schur square dimension by at least 1, it suffices to show the claim only for $\varrho' = \varrho$. Shortening an evaluation code at positions $\mathcal{I} \subseteq \{1, \ldots, n\}$

of cardinality $|\mathcal{I}| = \varrho$ means to take only those evaluation polynomials in $\mathcal{P}^{n,k}_{\boldsymbol{t},\boldsymbol{h},\boldsymbol{\eta}}$ that vanish on $\mathcal{I}$, and to evaluate them at all points α_i with $i \notin \mathcal{I}$. We first prove that this set of evaluation polynomials, say $\mathcal{P}'$, contains the degrees

$$S' := \{\varrho, \varrho+1, \ldots, r-1\} \cup \{(i+1)(r-\varrho)-1 \,:\, i = 1, \ldots, \ell\}\,.$$

The original set of evaluation polynomials $\mathcal{P}^{n,k}_{\boldsymbol{t},\boldsymbol{h},\boldsymbol{\eta}}$ contains $\langle 1, x, \ldots, x^{\varrho-1}\rangle = \mathbb{F}_q[x]_{<\varrho}$, as well as polynomials of degrees S' (cf. the arguments in the proof of Theorem 4.18). Take a polynomial of the latter kind, say f. By Lagrange interpolation, there is a polynomial $g \in \mathbb{F}_q[x]_{<\varrho}$ that evaluates to the same values as f at all shortened evaluation points α_i with $i \in \mathcal{I}$. Hence, $f - g$ vanishes at these α_i, has the same degree as f, and is contained in $\mathcal{P}'$.

Next, we prove that the evaluation polynomial set of Schur square of the shortened code contains polynomials of degrees

$$D' = \{2\varrho, \ldots, n-1+\varrho\}\,.$$

The argument is illustrated in Figure A.1. We distinguish the following cases:

- By multiplying the polynomials of degrees $\varrho, \varrho+1, \ldots, r-1$ with the one of degree ϱ, we obtain polynomials of degrees

$$D'_{-1} := \{2\varrho, \ldots, r+\varrho-1\}.$$

- By multiplying the polynomials of degrees $\varrho, \varrho+1, \ldots, r-1$ with the one of degree $r-1$, we obtain polynomials of degrees

$$D'_0 := \{r+\varrho-1, \ldots, \underbrace{2r-2}_{=2(r-\varrho)-2+2\varrho}\}.$$

- For $i = 1, \ldots, \ell$, by multiplying the polynomials of degrees $\varrho, \varrho+1, \ldots, r-1$ with the one of degree $(i+1)(r-\varrho)-1+\varrho$, we obtain polynomials of degrees

$$D'_i := \{(i+1)(r-\varrho)-1+2\varrho, \ldots, \underbrace{(i+1)(r-\varrho)-1+\varrho-r-1}_{=((i+1)+1)(r-\varrho)-2+2\varrho}\}.$$

Hence, the evaluation polynomials of the Schur square code contain polynomials of degrees

$$\bigcup_{i=-1}^{\ell} D'_i = \{2\varrho, \ldots, (\ell+2)(r-\varrho)-2+2\varrho\}.$$

By definition of r, we have $(\ell+2)(r-\varrho)-2 \geq n-1$, so $D' \subseteq \bigcup_{i=-1}^{\ell} D'_i$.

Consider the shortened evaluation point vector $\boldsymbol{\alpha}' \in \mathbb{F}_q^{n-\varrho}$ consisting only of the α_i with $i \notin \mathcal{I}$. Due to $\alpha_i \neq 0$, the polynomial $\prod_{i\notin\mathcal{I}}(x-\alpha_i)$ has a constant term, which implies that the evaluation map of the $\mathrm{ev}_{\boldsymbol{\alpha}'}(\cdot) : \langle x^j, \ldots, x^{j+n-\varrho}\rangle \to \mathbb{F}_q^{n-\varrho}$ is bijective for any $j \in \mathbb{N}_0$, in particular for $j = 2\varrho$. Hence, the Schur square dimension of $\mathcal{C}'$ is $\dim(\mathcal{C}'^2) \geq |D'| = n-\varrho$, which implies the claim. □

[1] As in Figure 4.5, these values are chosen for didactic reasons and do not fulfill (4.7) or (4.8).

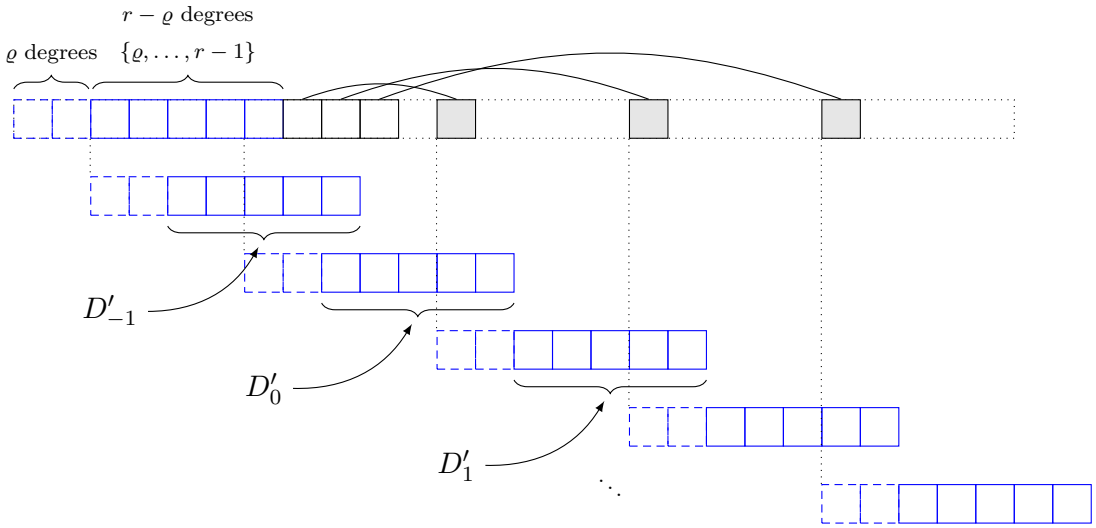


Figure A.1: Illustration of the proof of Theorem 4.21 for the example[1] $n = 26$, $k = 10$, $\varrho = 2$, $\ell = 3$, $r = 7$. The integers in $D' = \{2\varrho, \ldots, n - 1 + \varrho\}$ appear as evaluation polynomial degrees in $\cup_{i=-1}^{\ell} D'_i$.

A.3 Optimal Multiplication of Skew Polynomials of Degree m

The algorithms in Section 5.2 are all based on a fast skew polynomial multiplication algorithm. Therefore, it is an important question whether the multiplication algorithm in Section 5.2.1 can be accelerated and what the minimal cost of such an algorithm can be.

In this section, we answer both questions for the case when the polynomials have degree less than $s = m$: By combining the connection of skew polynomial and matrix multiplication observed in [Wac13, Section 3.1.3] and our fast σ-transformation algorithm in Section 5.2.5, we obtain a lower bound on the cost of a multiplication algorithm in $\mathbb{F}_{q^m}[x;\sigma]_{<m}$, as well as an algorithm that achieves this optimal complexity.

A.3.1 Relation of Skew Polynomial and Matrix Multiplication

In the following, we recall the connection of linearized polynomial and matrix multiplication from [Wac13, Section 3.1.3] and phrase it in skew polynomial language. The first lemma implies that there is a bijection between the set of skew polynomials of degree $< m$, $\mathbb{F}_{q^m}[x;\sigma]_{<m}$, and the set of $\mathbb{F}_q$-linear maps $\mathbb{F}_{q^m} \to \mathbb{F}_{q^m}$, $\mathrm{End}_{\mathbb{F}_q}(\mathbb{F}_{q^m})$, given by mapping a polynomial to its evaluation map.

Lemma A.4. *Let $a, b \in \mathbb{F}_{q^m}[x;\sigma]$. Then, their evaluation maps $a(\cdot), b(\cdot) : \mathbb{F}_{q^m} \to \mathbb{F}_{q^m}$ are the same if and only if $a \equiv b \mod_{\mathrm{r}} (x^m - 1)$.*

Proof. Suppose that the evaluation maps are the same. Then, $(a - b) \mod_{\mathrm{r}} (x^m - 1)$ is a skew polynomial of degree less than n that vanishes at all elements of the field $\mathbb{F}_{q^m}$, which implies $(a - b) \mod_{\mathrm{r}} (x^m - 1) = 0$, i.e., $a \equiv b \mod_{\mathrm{r}} (x^m - 1)$. The other direction follows from $(x^m - 1)(\cdot) = 0$. □

If two skew polynomials are multiplied modulo $x^m - 1$, denoted by $\cdot \bmod_{\mathrm{r}}(x^m-1)$, this corresponds to a composition of their evaluation maps, i.e.,

$$((a \cdot b) \bmod_{\mathrm{r}}(x^m - 1))(\cdot) = (a \cdot b)(\cdot) = a(b(\cdot)) = a(\cdot) \circ b(\cdot).$$

Let $\mathcal{B}$ be a basis of $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$ and denote by $[\psi]_{\mathcal{B}}^{\mathcal{B}}$ the matrix representation of a linear map $\psi \in \mathrm{End}_{\mathbb{F}_q}(\mathbb{F}_{q^m})$ with respect to $\mathcal{B}$. Then, the composition of two linear maps corresponds to a multiplication of their matrix representations, i.e., $[a(\cdot) \circ b(\cdot)]_{\mathcal{B}}^{\mathcal{B}} = [a(\cdot)]_{\mathcal{B}}^{\mathcal{B}} \cdot [b(\cdot)]_{\mathcal{B}}^{\mathcal{B}}$. This implies the following statement.

Lemma A.5. *Let $\mathcal{B}$ be a basis of $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$. Then, the mapping*

$$\varphi_{\mathcal{B}} : \left(\mathbb{F}_{q^m}[x;\sigma]_{<m}, \cdot \bmod_{\mathrm{r}}(x^m-1)\right) \to \left(\mathbb{F}_q^{m\times m}, \cdot\right), \quad a \mapsto [a(\cdot)]_{\mathcal{B}}^{\mathcal{B}}$$

is a monoid isomorphism.

Lemma A.5 implies that skew polynomial multiplication modulo $x^m - 1$ and matrix multiplication in $\mathbb{F}_q^{m\times m}$ are equivalent. If there is a fast implementation of the mapping $\varphi_{\mathcal{B}}$ and its inverse $\varphi_{\mathcal{B}}^{-1}$, the operations can be efficiently reduced to each other. The following lemma shows how this can be accomplished.

Lemma A.6. *Let $\mathcal{B}$ be a basis of $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$, $a \in \mathbb{F}_{q^m}[x;\sigma]_{<m}$, and $\boldsymbol{A} \in \mathbb{F}_q^{m\times m}$.*

- *If $\mathcal{B}$ is a normal basis, then $\varphi_{\mathcal{B}}(a)$ or $\varphi_{\mathcal{B}}^{-1}(A)$ can be computed by a σ-transform or an inverse σ-transform, respectively.*
- *Otherwise, $\varphi_{\mathcal{B}}(a)$ or $\varphi_{\mathcal{B}}^{-1}(A)$ can be computed by a σ-transform or an inverse σ-transform, respectively, plus two matrix multiplications in $\mathbb{F}_q^{m\times m}$.*

Proof. We can obtain $\varphi_{\mathcal{B}}(a)$ of a skew polynomial $a \in \mathbb{F}_{q^m}[x;\sigma]_{<m}$ by evaluating a at the m elements of $\mathcal{B}$ and representing the result in the basis $\mathcal{B}$. Computationally, this is a multipoint evaluation, or a σ-transform if $\mathcal{B}$ is a normal basis. Accordingly, the inverse $\varphi_{\mathcal{B}}^{-1}$ consists of an interpolation, or an inverse σ-transform in case of a normal basis.

If $\mathcal{B}$ is not a normal basis, we choose a normal basis $\mathcal{B}'$ and compute $\varphi_{\mathcal{B}}(a)$ using the (precomputed) change of basis matrices $\boldsymbol{T}_{\mathcal{B}'}^{\mathcal{B}}$ (from $\mathcal{B}'$ to $\mathcal{B}$) and $\boldsymbol{T}_{\mathcal{B}}^{\mathcal{B}'}$ by

$$\varphi_{\mathcal{B}}(a) = [a(\cdot)]_{\mathcal{B}}^{\mathcal{B}} = \boldsymbol{T}_{\mathcal{B}'}^{\mathcal{B}} \cdot [a(\cdot)]_{\mathcal{B}'}^{\mathcal{B}'} \cdot \boldsymbol{T}_{\mathcal{B}}^{\mathcal{B}'} = \boldsymbol{T}_{\mathcal{B}'}^{\mathcal{B}} \cdot \varphi_{\mathcal{B}'}(a) \cdot \boldsymbol{T}_{\mathcal{B}}^{\mathcal{B}'}.$$

Thus, $\varphi_{\mathcal{B}'}(a)$ can be obtained by a σ-transform, and $\varphi_{\mathcal{B}}(a)$ by two further matrix multiplications. The case $\varphi_{\mathcal{B}}^{-1}(A)$ works analogously. □

A.3.2 Faster Implementation of a Known Multiplication Algorithm

The observations of the previous subsection show that skew polynomial multiplication modulo $x^m - 1$ can be implemented using a few matrix multiplications and (inverse) σ-transforms. This was already observed in [Wac13, Section 3.1.3], where the σ-transforms were the bottleneck operations due to the lack of efficient algorithms. Our fast (inverse) σ-transform in Section 5.2.5, together with the fast bases of $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$ by Couveignes and Lercier [CL09], implies the following speed-up.

Theorem A.7. *Using the σ-transform as described in Theorem 5.17 and the $\mathbb{F}_q$-bases of $\mathbb{F}_{q^m}$ in [CL09], multiplication of $a, b \in \mathbb{F}_{q^m}[x;\sigma]_{<m}$ modulo $x^m - 1$ can be implemented in $O(m^\omega)$ operations over $\mathbb{F}_q$.*

Proof. Since $\varphi_\mathcal{B}$ is a monoid isomorphism, we can compute $c = a \cdot b \mod_\mathrm{r}(x^m - 1)$ by

$$c = \varphi_\mathcal{B}^{-1}\left(\varphi_\mathcal{B}(a) \cdot \varphi_\mathcal{B}(b)\right).$$

By Lemma A.6, computing c costs seven matrix multiplications, two σ-transforms and one inverse σ-transform. The fast σ-transform algorithm in Theorem 5.17 costs $O^\sim(m)$ operations over $\mathbb{F}_{q^m}$, and, using the fast bases from [CL09], in total $O^\sim(m^2)$ operations over $\mathbb{F}_q$. If the fast basis is not normal[2], each (inverse) σ-transform requires another multiplication with a change of basis matrix from the left in order to represent the result in a normal basis. Hence, the algorithm's complexity is dominated by the matrix multiplications, which cost $O(m^\omega)$ operations over $\mathbb{F}_q$. □

Remark A.8. *The proof of Theorem A.7 implies that any multi-point evaluation and interpolation w.r.t. a basis of $\mathbb{F}_{q^m}$ can be computed in $O^\sim(m^\omega)$ operations over $\mathbb{F}_q$.*

The algorithm implied by Theorem A.7 is a multiplication algorithm for skew polynomials of degree $s < m/2$ since the result of their multiplication has degree $< m$ and is not affected by the modulo operation. For polynomials of higher degree, we need to apply the following trick: Let $s = \mu \cdot m/2$ for some integer $\mu \geq 1$ and $a, b \in \mathbb{F}_{q^m}[x;\sigma]_{<s}$. If we fragment a and b into μ polynomials of degree $< m/2$, we can compute $c = a \cdot b$ by pairwise multiplying the fragments (μ^2 many multiplications in $\mathbb{F}_{q^m}[x;\sigma]_{<m/2}$) and adding the results (which is negligible since we know the overlapping positions). This costs $O\big(\max\left\{s^2 m^{\omega-2}, m^\omega\right\}\big)$ operations over $\mathbb{F}_q$.

Remark A.9. *Using the bases in [CL09], the skew polynomial multiplication algorithm in Section 5.2.1 (i.e., the $\mathbb{F}_{q^m}[x;\sigma]$-analog of [Wac13, Algorithm 3.1]) can be implemented in $O^\sim(ms^{\min\{\frac{\omega+1}{2}, 1.635\}})$ operations over $\mathbb{F}_q$. The algorithm described in this section is faster for*

$$\max\left\{m^{\frac{\omega-1}{1.635}}, m^{2\frac{\omega-1}{\omega+1}}\right\} < s < \min\left\{m^2, m^{\frac{3-\omega}{0.365}}\right\},$$

e.g., $m^{0.8152} < s < m^{1.7096}$ for the Coppersmith–Winograd exponent $\omega \approx 2.376$ [CW90]. In addition, the constant hidden by the O-notation might be smaller since matrices of dimension $m \times m$ are multiplied (instead of $\approx \sqrt{s}$ many $\sqrt{s} \times \sqrt{s}$ matrices in Section 5.2.1).

A.3.3 Optimality of the Multiplication Algorithm

In the following, we prove the optimality of the algorithm in Theorem A.7. If the basis in which we represent elements of $\mathbb{F}_{q^m}$ is a normal basis, then we can implement matrix multiplication in $\mathbb{F}_q^{m\times m}$ using one σ-transform, two inverse σ-transforms, and a skew polynomial multiplication modulo $x^m - 1$, cf. Lemma A.6. If in addition, the normal basis admits quasi-linear operations over $\mathbb{F}_q$, then the (inverse) σ-transforms cost $O^\sim(m^2)$ using Algorithm 11 and Algorithm 12. Hence, the skew polynomial multiplication becomes the bottleneck, which implies the following statement.

[2] In [CL09], two classes of bases with quasi-linear operations are presented, one of which contains only normal bases. However, these so-called *elliptic normal bases* do not exist for all extensions $\mathbb{F}_{q^m}/\mathbb{F}_q$.

Lemma A.10. *Let q and m be such that there is a normal basis of $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$ in which field operations[3] cost $O^{\sim}(m)$ over $\mathbb{F}_q$. Then, the multiplication of two matrices from $\mathbb{F}_q^{m\times m}$ can be implemented in*

$$O(m^\omega) \subseteq O\Big(\mathcal{M}_{q^m}^{(\mathbb{F}_q)}(m)\Big)$$

operations over $\mathbb{F}_q$, where $\mathcal{M}_{q^m}^{(\mathbb{F}_q)}(m)$ is the cost of an algorithm for multiplication of skew polynomials in $\mathbb{F}_{q^m}[x;\sigma]_{<m}$ (in operations over $\mathbb{F}_q$).

The so-called *elliptic normal bases* introduced by Couveignes and Lercier [CL09] fulfill both assumptions of Lemma A.10: They are normal bases that admit quasi-linear field operations. Although such bases do not exist for all field extensions $\mathbb{F}_{q^m}/\mathbb{F}_q$, the following is directly implied by the statements in [CL09].

Lemma A.11. *Let $(m_i)_{i\in\mathbb{N}}$ be a sequence of positive integers $m_i \in \mathbb{N}$ with $m_i \to \infty$ for $i \to \infty$. Then, there is a sequence of prime powers $(q_i)_{i\in\mathbb{N}}$ such that there is an elliptic normal basis of $\mathbb{F}_{q_i^{m_i}}$ over $\mathbb{F}_{q_i}$.*

Proof. Take an arbitrary sequence of prime powers $(\widetilde{q}_i)_{i\in\mathbb{N}}$. By [CL09, Section 5.2], there is a positive integer $f_i \in O(\log^2(m_i)(\log\log(m_i))^2)$ such that there is an elliptic normal basis of $\mathbb{F}_{q_i^{m_i}}$ over $\mathbb{F}_{q_i}$, where $q_i = \widetilde{q}_i^{f_i}$. □

Now suppose that, similar to the matrix multiplication exponent ω, there is a constant $\gamma \geq 2$ such that we have $\mathcal{M}_{q^m}(m) \in \Theta(m^\gamma)$, independent of the ground field $\mathbb{F}_q$. Then, Lemma A.10 and Lemma A.11 directly imply that we must have

$$\omega \leq \gamma.$$

Hence, the the algorithm for skew polynomial multiplication in Section A.3.2 is optimal

Remark A.12. *If we use the fast bases by Couveignes and Lercier [CL09], then the multiplication algorithm in Section 5.2.1 (i.e., the $\mathbb{F}_{q^m}[x;\sigma]$-analog of [Wac13, Algorithm 3.1]) costs at most $O^{\sim}(m^{\frac{\omega+1}{2}+1}) = O^{\sim}(m^{\frac{\omega+3}{2}})$ operations over $\mathbb{F}_q$, and differs from the lower cost bound by a factor $m^{\frac{3-\omega}{2}}$. Note that this only holds for $s = m$.*

The arguments in this section imply that if $\mathcal{M}_{q^m}(s) \in O^{\sim}(s)$, then there would be a quasi-quadratic matrix multiplication algorithm in the cases where an elliptic normal basis of $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$ exists. Hence, finding a quasi-linear skew polynomial multiplication algorithm is at least as hard as proving that there is a quasi-quadratic time matrix multiplication algorithm.

A.4 Key Equations Solvable with the MgLSSR Problem

Table A.1 gives an overview of key equations for decoding (interleaved) Gabidulin codes, both with and without erasures, that can be solved by the MgLSSR problem (Problem 6.3). A similar table for the well-studied $\mathbb{F}[x]$-analog of Problem 6.3 can be found in [Nie13b, Table 2.1].

[3] I.e., multiplication, addition, and computing inverses.

Table A.1: Selection of key equations solvable by the MgLSSR problem (Problem 6.3), both for decoding Gabidulin ($h = 1$) and interleaved Gabidulin ($h \geq 1$) codes. For each key equation, we give the input variables ($h \in \mathbb{N}$, $r_i, g_i \in \mathbb{F}_{q^m}[x;\sigma]$, and $\gamma_0, \gamma_i \in \mathbb{N}_0$ for $i = 1, \ldots, h$), output variables ($\lambda, \omega_i \in \mathbb{F}_{q^m}[x;\sigma]$), the parameter μ (for complexity measurement), and further restrictions. Variables are named as in their respective sources. Most key equations in the literature are written in linearized polynomial language ($\sigma = \cdot^q$), but might be generalized to arbitrary skew polynomials straightforwardly.

Key Equation	h	r_i	g_i	γ_0	γ_i	λ	ω_i	$\mu = \min_i\{\gamma_i + \deg g_i\}$	Restriction
[Gab85]	1	$S(x)$	x^{d-1}	0	0	$\Delta(x)$	$F(x)$	$d - 1 \leq n$	$\sigma = \cdot^q$
[SKK08, Theorem 12] (error-erasure decoding)	1	$S_{DU}(x)$	x^{d-1}	$\delta + \mu$	0	$\sigma_F(x)$	$\omega(x)$	$d - 1 + \delta + \mu \leq n$	$\sigma = \cdot^q$
[Wac13, Theorem 3.5] (row-space syndrome key equation)	1	$\widetilde{s}(x)$	x^{n-k}	0	0	$\Gamma(x)$	$\Phi(x)$	$n - k \leq n$	$\sigma = \cdot^q$
[Wac13, Theorem 3.6] and [WAS13] ("Gao-like" decoding)	1	$\hat{r}(x)$	$M_{\mathcal{G}}(x)$	k	0	$\Lambda(x)$	$\Lambda(f(x))$	$n + k \leq 2n$	$\sigma = \cdot^q$
[Wac13, Theorem 3.8] (error-erasure decoding)	1	$\Lambda^{(R)}(\hat{r}(\overline{\Gamma}^{(C)}(x^\gamma)))$	$x^m - 1$	$k + \varrho + \gamma$	0	$\Lambda^{(E)}$	$\Lambda^{(E)}(\Lambda^{(R)}(f(\overline{\Gamma}^{(C)}(x^\gamma))))$	$m + k + \varrho + \gamma \leq 2n$	$\sigma = \cdot^q$ and $\alpha_1, \ldots, \alpha_n$ normal basis of $\mathbb{F}_{q^m}$
Lemma 5.1 (skew-polynomial variant of [WAS13, Wac13])	1	R	G	k	0	Λ	Λf	$n + k \leq 2n$	-
[SB10, SJB11] (polynomial notation as in [SWC12])	≥ 1	$s^{(i)}(x)$	x^{n-k_i}	0	0	$\sigma(x)$	$\omega^{(i)}(x)$	$\max_i\{n - k_i\} \leq n$	$\sigma = \cdot^q$ and horizontally interleaved codes
[Wac13, Section 4.1] (row-space variant of [SB10, SJB11])	≥ 1	$\widetilde{s}^{(i)}(x)$	$x^{n-k^{(i)}}$	0	0	Γ	$\Phi^{(i)}(x)$	$\max_i\{n - k^{(i)}\} \leq n$	$\sigma = \cdot^q$ and vertically interleaved codes
Lemma 6.1	≥ 1	R_i	G	$\max_j\{k_j\}$	$\max_j\{k_j\} - k_i$	Λ	Λf_i	$\leq n + \max_j\{k_j\} \leq 2n$	horizontally interleaved codes

B

List of Symbols and Acronyms

The following is a list of symbols and acronyms that are commonly used inside this dissertation. We refrain from listing symbols that only appear close to their definitions. Due to the similarity of their meanings, some symbols are used for multiple notions (e.g., the evaluation points of Reed–Solomon and Gabidulin codes are both called α_i). Thus, we list some symbols by chapter.

Global Notions

Acronyms

AG	**A**lgebraic **G**eometry
BCH	**B**ose–**C**haudhuri–**H**ocquenghem
BMD	**B**ounded **M**inimum **D**istance
GRS	**G**eneralized **R**eed–**S**olomon
IH	**I**nterleaved (one-point) **H**ermitian
IRS	**I**nterleaved **R**eed–**S**olomon
MDS	**M**aximum **D**istance **S**eparable
MgLSSR	**M**ulti-Sequence **g**eneralized **L**inear **S**kew-Feedback **S**hift **R**egister
MPE	**M**ulti-**P**oint **E**valuation
MRD	**M**aximum **R**ank **D**istance
MSP	**M**inimal **S**ubspace **P**olynomial
RS	**R**eed–**S**olomon
wPf	**w**eak **P**opov **f**orm

Algebraic Notation

$\mathbb{F}$, K, L	Fields
$\mathbb{F}_q$ and $\mathbb{F}_{q^m}$	Finite fields of cardinality q and q^m (q prime power, $m \in \mathbb{N}$)
L/K	Field extension
$[L : K]$	Extension degree of a field extension L/K
$\mathrm{Gal}(L/K)$	Galois group of a field extension L/K
$\cdot^q$	Frobenius automorphism $\cdot^q : \mathbb{F}_{q^m} \to \mathbb{F}_{q^m}$, $\alpha \mapsto \alpha^q$
$K[x]$	Univariate polynomial ring over K
$K[x_1, \ldots, x_n]$	Multivariate polynomial ring over K

Codes

n	Code length
M	Cardinality of a code
k	Dimension of a linear code
d	Minimum distance (w.r.t. the given metric)
$\mathrm{d}(\cdot,\cdot)$	Metric $\mathrm{d}(\cdot,\cdot) : \mathbb{F}^n \times \mathbb{F}^n \to \mathbb{R}_{\geq 0}$
$\mathcal{C}(n, M)$	Code of length n and cardinality M
$\mathcal{C}(n, M, d)$	Code of length n, cardinality M, and minimum distance d
$\mathcal{C}[n, k]$	Linear code of length n and dimension k
$\mathcal{C}[n, k, d]$	Linear code of length n, dimension k, and minimum distance d
$\mathrm{ev}_{\boldsymbol{\alpha}}(\mathcal{P})$	Evaluation code with eval. polynomials $\mathcal{P}$, eval. points $\boldsymbol{\alpha}$, cf. Def. 2.2
$\boldsymbol{G}$	Generator matrix
$\boldsymbol{H}$	Parity-check matrix
$\mathcal{C}_{\mathrm{RS}}[n, k]$	Reed–Solomon code
$\mathcal{C}_{\mathrm{H}}^{(n,m_{\mathrm{H}})}$	One-point Hermitian code
$\mathcal{C}_{\mathrm{IRS}}(n, k_1, \ldots, k_h; h)$	Interleaved Reed–Solomon code ($\mathcal{C}_{\mathrm{IRS}}(n, k; h)$ if homogeneous)
$\mathcal{C}_{\mathrm{IH}}(n, m_{\mathrm{H}1}, \ldots, m_{\mathrm{H}h}; h)$	Interleaved one-point Hermitian code ($\mathcal{C}_{\mathrm{IH}}(n, m_{\mathrm{H}}; h)$ if homogeneous)
$\mathcal{C}_{\mathrm{G}}[n, k]$	Gabidulin code
$\mathcal{C}_{\mathrm{IG}}(n, k_1, \ldots, k_h; h)$	Interleaved Gabidulin code
$\mathcal{C}_{\mathrm{TG}}[n, k]$	(Sheekey's) Twisted Gabidulin code

Skew Polynomials

$\mathbb{F}_{q^m}[x;\sigma]$	Skew polynomials over $\mathbb{F}_{q^m}$ w.r.t. $\sigma \in \mathrm{Gal}(\mathbb{F}_{q^m}/\mathbb{F}_q)$, cf. Def. 2.16
$\mathbb{F}_{q^m}[x;\sigma]_{\leq s}, \mathbb{F}_{q^m}[x;\sigma]_{<s}$	Skew polynomials of degree at most/smaller than s
$\mathrm{LC}(f)$, $\mathrm{LT}(f)$	leading coefficient/term of a skew polynomial f
$f(\cdot)$	Evaluation map of a skew polynomial f, cf. Section 2.3.1
$\mathrm{roots}(f)$	Root space of a skew polynomial f
$\mathcal{M}_{\mathcal{U}}$	Minimal subspace polynomial of a subspace $\mathcal{U} \subseteq \mathbb{F}_{q^m}$, cf. Theorem 2.23
$\mathcal{I}_{\{[x_i,y_i]\}_{i=1}^s}$	Interpolation polynomial ($\in \mathbb{F}_{q^m}[x;\sigma]$), cf. Theorem 2.24
$\hat{\cdot}$	σ-transform, cf. Definition 2.25

Symbols by Chapter

Chapter 3: Improved Power Decoding of Interleaved Codes in Hamming Metric

General

h	Interleaving degree $h \in \mathbb{N}$
ℓ, s	Parameters of the decoders
$\mathcal{E}$	Set of error positions, cf. Section 2.2.3
R	Code rate $R = \frac{k}{n}$ of the (homogenenous) interleaved code
$\lvert\boldsymbol{i}\rvert$	Length of a vector $\lvert\boldsymbol{i}\rvert = \sum_{\mu=1}^{h} i_\mu$, cf. Definition 3.4
$\boldsymbol{i} \preceq \boldsymbol{j}$	Product partial order, cf. Definition 3.4
$\boldsymbol{a}^{\boldsymbol{i}}$	$= \prod_{\mu=1}^{m} a_\mu^{i_\mu}$, cf. Definition 3.4

Interleaved Reed–Solomon Codes

$\boldsymbol{f}$	Message polynomial vector $\boldsymbol{f} \in \mathbb{F}_q[x]^h_{<k}$, cf. Section 3.1.1
$\boldsymbol{R}$	Interpolation polynomial vector $\boldsymbol{R} \in \mathbb{F}_q[x]^h$, cf. Definition 3.1
G	$= \prod_{i=1}^{n}(x - \alpha_i) \in \mathbb{F}_q[x]$, cf. Definition 3.1
Λ	Error locator polynomial, cf. Definition 3.2
$\boldsymbol{\Omega}$	Vector of error evaluator polynomials, see below Definition 3.2
$\Psi_{\boldsymbol{j}}$	$= \Lambda^s \boldsymbol{f}^{\boldsymbol{j}}$, cf. (3.3)
$\Lambda_{\boldsymbol{i}}$	$= \Lambda^{s-\lvert\boldsymbol{i}\rvert}\boldsymbol{\Omega}^{\boldsymbol{i}}$, cf. (3.3)
$A_{\boldsymbol{i},\boldsymbol{j}}$	$= \binom{\boldsymbol{j}}{\boldsymbol{i}}\boldsymbol{R}^{\boldsymbol{j}-\boldsymbol{i}}G^{\lvert\boldsymbol{i}\rvert}$, cf. (3.3)

Interleaved One-Point Hermitian Codes

$\mathcal{H}$	Hermitian curve
$P_1, \ldots, P_{q^3}, P_\infty$	Points on the Hermitian curve (P_∞ point at infinity)
t_P	Local parameter in a point P on the curve
v_P	Discrete valuation in a point P on the curve
$\mathcal{L}(D)$	Riemann–Roch space of a divisor D
$\mathcal{R}$	Ring $\mathcal{R} = \mathcal{L}(\infty P_\infty)$, isomorphic to the set of bivariate polynomials of y-degree $< q$ with multiplication modulo the curve equation
$\deg_{\mathcal{H}}$	Degree of polynomials in $\mathcal{R}$
$\boldsymbol{f}$	Message polynomial vector $\boldsymbol{f} \in \mathcal{L}(m_{\mathrm{H}}P_\infty)^h$, cf. Section 3.2.1
$\boldsymbol{R}$	Interpolation polynomial vector $\boldsymbol{R} \in \mathcal{R}^h$, cf. Section 3.2.1
G	$= \prod_{\alpha \in \mathbb{F}_{q^2}}(x - \alpha) = x^{q^2} - x \in \mathcal{R}$, cf. Section 3.2.1
$\Lambda^{(s)}$	Error locator polynomial of multiplicity s, cf. Definition 3.25
$\Omega_{s,\boldsymbol{i}}$	Polynomial in $\mathcal{R}$, see below Lemma 3.28
$\Psi_{\boldsymbol{j}}$	$= \Lambda^{(s)}\boldsymbol{f}^{\boldsymbol{j}}$, cf. (3.16)
$\Lambda_{\boldsymbol{i}}$	$= \Omega_{s,\boldsymbol{i}}$, cf. (3.16)
$A_{\boldsymbol{i},\boldsymbol{j}}$	$= \binom{\boldsymbol{j}}{\boldsymbol{i}}\boldsymbol{R}^{\boldsymbol{j}-\boldsymbol{i}}G^{\lvert\boldsymbol{i}\rvert}$, cf. (3.16)
$[q)$	$= \{0, \ldots, q-1\}$

Chapter 4: Twisted Reed–Solomon Codes

ℓ	Number of twists $\ell \in \mathbb{N}$
$\boldsymbol{t}$	Twist vector $\boldsymbol{t} \in \mathbb{N}^\ell$ with distinct entries $1 \le t_i \le n-k$
$\boldsymbol{h}$	Hook vector $\boldsymbol{h} \in \mathbb{N}_0^\ell$ with entries $0 \le h_i < k$
$\boldsymbol{\eta}$	Vector of factors in the twist coefficients $\boldsymbol{\eta} \in (\mathbb{F}_q \setminus \{0\})^\ell$
$\boldsymbol{\alpha}$	Evaluation points $\boldsymbol{\alpha} \in \mathbb{F}_q^n$ with distinct entries
$\mathcal{P}^{n,k}_{\boldsymbol{t},\boldsymbol{h},\boldsymbol{\eta}}$	$[k, \boldsymbol{t}, \boldsymbol{h}, \boldsymbol{\eta}]$-twisted polynomials, cf. Definition 4.1
$\mathcal{C}_{\boldsymbol{\alpha},\boldsymbol{t},\boldsymbol{h},\boldsymbol{\eta}}[n,k]$	$[\boldsymbol{\alpha}, \boldsymbol{t}, \boldsymbol{h}, \boldsymbol{\eta}]$-twisted Reed–Solomon code, cf. Definition 4.1
$S_{k,\boldsymbol{h},\boldsymbol{t}}$	Degrees of the polynomials in $\mathcal{P}^{n,k}_{\boldsymbol{t},\boldsymbol{h},\boldsymbol{\eta}}$, cf. Theorem 4.16
$\mathrm{D}(\mathcal{V})$	$= \{\deg(f \cdot g) : f, g \in \mathcal{V}\}$, where $\mathcal{V} \subseteq \mathbb{F}_q[x]$, cf. Definition 4.13
$\mathrm{D}(\mathcal{V})_{<n}$	$= \{d \in \mathrm{D}(\mathcal{V}) : d < n\}$, where $\mathcal{V} \subseteq \mathbb{F}_q[x]$, cf. Definition 4.13

Chapter 5: Fast Decoding of Gabidulin Codes

s	Input size of a polynomial operation (e.g., degree)
$\mathcal{M}_{q^m}(s)$	Complexity of multiplication in $\mathbb{F}_{q^m}[x;\sigma]$, cf. Definition 5.6
$\mathcal{D}_{q^m}(s)$	Complexity of division in $\mathbb{F}_{q^m}[x;\sigma]$, cf. Definition 5.6
$\mathcal{MSP}_{q^m}(s)$	Complexity of computing minimal subspace polynomials, cf. Def. 5.6
$\mathcal{MPE}_{q^m}(s)$	Complexity of multi-point evaluation in $\mathbb{F}_{q^m}[x;\sigma]$, cf. Definition 5.6
$\mathcal{I}_{q^m}(s)$	Complexity of interpolation in $\mathbb{F}_{q^m}[x;\sigma]$, cf. Definition 5.6
$\mathcal{MPE}_{q^m}(s)$	Complexity of the (inverse) σ-transformn in $\mathbb{F}_{q^m}[x;\sigma]$, cf. Def. 5.6

Chapter 6: Decoding Interleaved Gabidulin Codes Using Row Reduction

h	Interleaving degree $h \in \mathbb{N}$
$\boldsymbol{E}_{\mathrm{h}}$	Error matrix ($\in \mathbb{F}_q^{m\times hn}$) of vertically int. Gabidulin codes, cf. (2.6)
$\boldsymbol{E}_{\mathrm{v}}$	Error matrix ($\in \mathbb{F}_q^{hm\times n}$) of horizontally int. Gabidulin codes, cf. (2.5)
$r_1,\dots,r_h,g_1,\dots,g_h$	Skew Polynomials in the MgLSSR problem, cf. Problem 6.3
$\gamma_0,\dots,\gamma_h$	Degree shifts ($\in \mathbb{N}_0$) in the MgLSSR problem, cf. Problem 6.3
$\lambda,\omega_1,\dots,\omega_h$	Solution ($\in \mathbb{F}_{q^m}[x;\sigma]$) of the MgLSSR problem, cf. Problem 6.3
μ	Input size $\mu = \min_i\{\gamma_i + \deg g_i\}$ of the MgLSSR problem, cf. Prob. 6.3
$\mathcal{M}^{(\mathrm{M})}$	Module containing the MgLSSR problem solution, cf. Equation 6.3
$\boldsymbol{M}^{(\mathrm{M})}$	Basis of the module $\mathcal{M}^{(\mathrm{M})}$, cf. Lemma 6.4
$\boldsymbol{w}^{(\mathrm{M})}$	Shift vector for solving the MgLSSR problem, cf. Lemma 6.5
$Q_0,\dots,Q_h$	Solution ($\in \mathbb{F}_{q^m}[x;\sigma]$) of the interpolation step, cf. Problem 6.8
$\mathcal{M}^{(\mathrm{I})}$	Module containing the interpolation step solutions, cf. Equation 6.9
$\boldsymbol{M}^{(\mathrm{I})}$	Basis of the module $\mathcal{M}^{(\mathrm{I})}$, cf. Lemma 6.9
$\boldsymbol{w}^{(\mathrm{I})}$	Shift vector for solving the interpolation step, cf. Lemma 6.10
ψ	Value function of a skew polynomial vector or matrix, cf. Def. 6.14
$\Delta(\boldsymbol{V})$	Orthogonality defect of a matrix $\boldsymbol{V} \in \mathrm{GL}_r(\mathbb{F}_{q^m}[x;\sigma])$, cf. Def. 6.22
$\lvert v\rvert, \lvert\boldsymbol{v}\rvert, \lvert\boldsymbol{V}\rvert$	Length of a skew polynomial v, vector $\boldsymbol{v}$, or matrix $\boldsymbol{V}$, cf. Def. 6.30
$v\rvert_t, \boldsymbol{v}\rvert_t, \boldsymbol{V}\rvert_t$	Accuracy approximation to depth $t \in \mathbb{N}_0$ of a skew polynomial v, vector $\boldsymbol{v}$, or matrix $\boldsymbol{V}$, cf. Definition 6.30
$\deg \boldsymbol{v}, \deg \boldsymbol{V}$	Degree of a skew polynomial vector $\boldsymbol{v}$ or matrix $\boldsymbol{V}$, cf. Definition 2.27
$\mathrm{maxdeg}\, \boldsymbol{V}$	Maximal degree of a matrix $\boldsymbol{V}$, cf. Definition 2.27
$\mathrm{LP}(\boldsymbol{v})$	Leading position of a vector $\boldsymbol{v}$, cf. Definition 2.27
$\mathrm{LT}(\boldsymbol{v})$	Leading term of a vector $\boldsymbol{v}$, cf. Definition 2.27
$\mathrm{GL}_r(\mathbb{F}_{q^m}[x;\sigma])$	Set of full-rank matrices in $\mathbb{F}_{q^m}[x;\sigma]^{r\times r}$
$\boldsymbol{w}$	(Degree) Shift vector $\boldsymbol{w} \in \mathbb{N}_0^r$
$\Phi_{\boldsymbol{w}}$	Shift mapping $\Phi_{\boldsymbol{w}} : \mathbb{F}_{q^m}[x;\sigma]^{r\times r} \to \mathbb{F}_{q^m}[x;\sigma]^{r\times r}$, cf. Section 2.3.2
$\mathrm{LP}_{\boldsymbol{w}}(\boldsymbol{v})$	Shifted leading position, cf. Section 2.3.2
$\deg_{\boldsymbol{w}}(\boldsymbol{v})$	Shifted degree, cf. Section 2.3.2

Chapter 7: Generalizations of Twisted Gabidulin Codes

ℓ	Number of twists $\ell \in \mathbb{N}$
$\boldsymbol{t}$	Twist vector $\boldsymbol{t} \in \mathbb{N}^\ell$ with distinct entries $1 \leq t_i \leq n-k$
$\boldsymbol{h}$	Hook vector $\boldsymbol{h} \in \mathbb{N}_0^\ell$ with entries $0 \leq h_i < k$
$\boldsymbol{\eta}$	Vector of factors in the twist coefficients $\boldsymbol{\eta} \in (\mathbb{F}_{q^m} \setminus \{0\})^\ell$
$\boldsymbol{\alpha}$	Evaluation points $\boldsymbol{\alpha} \in \mathbb{F}_{q^m}^n$ with $\mathbb{F}_q$-linearly independent entries
$\mathcal{P}_{\boldsymbol{t},\boldsymbol{h},\boldsymbol{\eta}}^{n,k}$	$[k, \boldsymbol{t}, \boldsymbol{h}, \boldsymbol{\eta}]$-twisted skew polynomials, cf. Definition 7.1
$\mathcal{C}_{\boldsymbol{\alpha},\boldsymbol{t},\boldsymbol{h},\boldsymbol{\eta}}[n,k]$	$[\boldsymbol{\alpha}, \boldsymbol{t}, \boldsymbol{h}, \boldsymbol{\eta}]$-twisted Gabidulin code, cf. Definition 7.1

Bibliography

References

[AB01] Sergei A. Abramov and Manuel Bronstein. On Solutions of Linear Functional Systems. In *International Symposium on Symbolic and Algebraic Computation*, pages 1–6, New York, NY, USA, 2001.

[AH74] Alfred V. Aho and John E. Hopcroft. *The Design and Analysis of Computer Algorithms*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1st edition, 1974.

[AL69] H Althaus and R Leake. Inverse of a Finite-Field Vandermonde Matrix (Corresp.). *IEEE Transactions on Information Theory*, 15(1):173–173, 1969.

[Ale05] Michael Alekhnovich. Linear Diophantine Equations Over Polynomials and Soft Decoding of Reed–Solomon Codes. *IEEE Transactions on Information Theory*, 51(7):2257–2265, July 2005.

[ALR13] Daniel Augot, Pierre Loidreau, and Gwezheneg Robert. Rank Metric and Gabidulin Codes in Characteristic Zero. In *IEEE International Symposium on Information Theory*, 2013.

[ALR17] Daniel Augot, Pierre Loidreau, and Gwezheneg Robert. Generalized Gabidulin Codes Over Fields of Any Characteristic. *Designs, Codes and Cryptography*, Oct 2017.

[Anv04] H Peter Anvin. The Mathematics of RAID-6, 2004. `http://kernel.org/pub/linux/kernel/people/hpa/raid6.pdf`.

[Arm08] Marc A Armand. Interleaved Reed–Solomon Codes Versus Interleaved Hermitian Codes. *IEEE Communications Letters*, 12(10), 2008.

[Art91] Michael Artin. *Algebra*. Prentice Hall, 1991.

[BB10] Peter Beelen and Kristian Brander. Key Equations for List Decoding of Reed–Solomon Codes and How to Solve Them. *Journal of Symbolic Computation*, 45(7):773–786, 2010.

[BB11] Morgan Barbier and Pauolo S. L.M̃. Barreto. Key Reduction of McEliece's Cryptosystem Using List Decoding. In *IEEE International Symposium on Information Theory*, July 2011.

[BCL06] Bernhard Beckermann, Howard Cheng, and George Labahn. Fraction-Free Row Reduction of Matrices of Ore Polynomials. *Journal of Symbolic Computation*, 41(5):513–543, 2006.

[Ber03] Thierry P Berger. Isometries for Rank Distance and Permutation Group of Gabidulin Codes. *IEEE Transactions on Information Theory*, 49(11):3016–3019, 2003.

[Ber15] Elwyn R Berlekamp. *Algebraic Coding Theory*. World Scientific Publishing Co, 2015.

[BGL02] Martin Bossert, Ernst M Gabidulin, and Paul Lusina. Space-Time Codes Based on Gaussian Integers. In *IEEE Int. Symp. Inf. Theory*, 2002.

[BGM96] George A Baker and Peter Graves-Morris. *Padé approximants*. Cambridge University Press, 1996.

[BGY80] Richard P Brent, Fred G Gustavson, and David YY Yun. Fast Solution of Toeplitz Systems of Equations and Computation of Padé Approximants. *Journal of Algorithms*, 1(3):259–295, 1980.

[BK78] Richard P Brent and Hsiang-Tsung Kung. Fast Algorithms for Manipulating Formal Power Series. *Journal of the ACM*, 25(4):581–595, October 1978.

[BKY03] Daniel Bleichenbacher, Aggelos Kiayias, and Moti Yung. Decoding of Interleaved Reed Solomon Codes Over Noisy Data. In *International Colloquium on Automata, Languages, and Programming*, pages 97–108. Springer, 2003.

[BL92] Bernhard Beckermann and George Labahn. A Uniform Approach for Hermite Padé and Simultaneous Padé Approximants and Their Matrix-Type Generalizations. *Numerical Algorithms*, 3(1):45–54, 1992.

[BL94] Bernhard Beckermann and George Labahn. A Uniform Approach for the Fast Computation of Matrix-Type Padé Approximants. *SIAM Journal on Matrix Analysis and Applications*, 15(3):804–823, July 1994.

[Bla79] Richard E. Blahut. Transform Techniques for Error Control Codes. *IBM Journal of Research and development*, 23(3):299–315, 1979.

[Bla83] Richard E Blahut. *Theory and Practice of Error Control Codes*. Addison Wesley Longman Publishing Co, 1983.

[BMS04] Andrew Brown, Lorenz Minder, and Amin Shokrollahi. Probabilistic Decoding of Interleaved RS-Codes on the q-Ary Symmetric Channel. In *IEEE International Symposium on Information Theory*, pages 326–326, 2004.

[BMS05] Andrew Brown, Lorenz Minder, and Amin Shokrollahi. Improved Decoding of Interleaved AG Codes. In *IMA International Conference on Cryptography and Coding*, pages 37–46. Springer, 2005.

[BMVT78] Elwyn Berlekamp, Robert McEliece, and Henk Van Tilborg. On the Inherent Intractability of Certain Coding Problems (Corresp.). *IEEE Transactions on Information Theory*, 24(3):384–386, 1978.

[Bos13] Martin Bossert. *Kanalcodierung*. Walter de Gruyter, 2013.

[BP12] Dario Bini and Victor Pan. *Polynomial and Matrix Computations: Fundamental Algorithms*. Springer Science & Business Media, 2012.

[Bra10] Kristian Brander. *Interpolation and List Decoding of Algebraic Codes*. PhD thesis, Technical University of Denmark, 2010.

[BU14] Delphine Boucher and Felix Ulmer. Linear Codes Using Skew Polynomials with Automorphisms and Derivations. *Designs, Codes and Cryptography*, 70(3):405–431, 2014.

[CB04] Yuval Cassuto and Jehoshua Bruck. A Combinatorial Bound on the List Size. Technical report, California Institute of Technology, 2004.

[CCMZ15] Ignacio Cascudo, Ronald Cramer, Diego Mirandola, and Gilles Zémor. Squares of Random Linear Codes. *IEEE Transactions on Information Theory*, 61(3):1159–1173, March 2015.

[CDN15] Ronald Cramer, Ivan Bjerre Damgård, and Jesper Buus Nielsen. *Secure Multiparty Computation and Secret Sharing*. Cambridge University Press, 2015.

[CGGU+14] Alain Couvreur, Philippe Gaborit, Valérie Gauthier-Umaña, Ayoub Otmani, and Jean-Pierre Tillich. Distinguisher-based Attacks on Public-key Cryptosystems Using Reed–Solomon Codes. *Designs, Codes and Cryptography*, 73(2):641–666, 2014.

[CH00] Stephen D Cohen and Dirk Hachenberger. The Dynamics of Linearized Polynomials. *Proceedings of the Edinburgh Mathematical Society (Series 2)*, 43(01):113–128, 2000.

[CH10] Henry Cohn and Nadia Heninger. Ideal forms of Coppersmith's Theorem and Guruswami–Sudan List Decoding. *arXiv*, 1008.1284, 2010.

[CH13] Henry Cohn and Nadia Heninger. Approximate Common Divisors via Lattices. *The Open Book Series*, 1(1):271–293, 2013.

[Cha72] David Chase. Class of Algorithms for Decoding Block Codes with Channel Measurement Information. *IEEE Transactions on Information Theory*, 18(1):170–182, 1972.

[CJC13] George C Clark Jr and J Bibb Cain. *Error-Correction Coding for Digital Communications*. Springer Science & Business Media, 2013.

[CJN+15] Muhammad F.Ĩ. Chowdhury, Claude-Pierre Jeannerod, Vincent Neiger, Éric Schost, and Gilles Villard. Faster Algorithms for Multivariate Interpolation with Multiplicities and Simultaneous Polynomial Approximations. *IEEE Transactions on Information Theory*, 61(5):2370–2387, May 2015.

[CKM97] Stéphane Collart, Michael Kalkbrener, and Daniel Mall. Converting Bases with the Gröbner Walk. *Journal of Symbolic Computation*, 24(3-4):465–469, 1997.

[CL09] Jean-Marc Couveignes and Reynald Lercier. Elliptic Periods for Finite Fields. *Finite Fields and Their Applications*, 15(1):1–22, 2009.

[CL17a] Xavier Caruso and Jérémy Le Borgne. A New Faster Algorithm for Factoring Skew Polynomials Over Finite Fields. *Journal of Symbolic Computation*, 79:411–443, 2017.

[CL17b] Xavier Caruso and Jérémy Le Borgne. Fast Multiplication for Skew Polynomials. *arXiv preprint arXiv:1702.01665*, 2017.

[CLR+01] Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, Clifford Stein, et al. *Introduction to Algorithms*, volume 2. MIT press Cambridge, 2001.

[CMP16] Antonio Cossidente, Giuseppe Marino, and Francesco Pavese. Non-Linear Maximum Rank Distance Codes. *Designs, Codes and Cryptography*, 79(3):597–609, 2016.

[CMPZ17] Bence Csajbók, Giuseppe Marino, Olga Polverino, and Corrado Zanella. A New Family of MRD-Codes. *arXiv preprint arXiv:1707.08487*, 2017.

[CMZ17] Bence Csajbók, Giuseppe Marino, and Ferdinando Zullo. New Maximum Scattered Linear Sets of the Projective Line. *arXiv preprint arXiv:1709.00926*, 2017.

[CS98] Anne Canteaut and Nicolas Sendrier. Cryptanalysis of the Original McEliece Cryptosystem. In *International Conference on the Theory and Application of Cryptology and Information Security*, volume 1514, pages 187–199. Springer, 1998.

[CS03] Don Coppersmith and Madhu Sudan. Reconstructing Curves in Three (and Higher) Dimensional Space from Noisy Data. In *ACM Symposium on the Theory of Computing*, 2003.

[CW90] Don Coppersmith and Shmuel Winograd. Matrix Multiplication via Arithmetic Progressions. *Journal of symbolic computation*, 9(3):251–280, 1990.

[DDKM16] Hoang Dau, Iwan Duursma, Han Mao Kiah, and Olgica Milenkovic. Repairing Reed–Solomon Codes with Multiple Erasures. *arXiv preprint arXiv:1612.01361*, 2016.

[Del78] Philippe Delsarte. Bilinear Forms over a Finite Field with Applications to Coding Theory. *Journal of Combinatorial Theory, Series A*, 25(3):226–241, 1978.

[Die43] Jean Dieudonné. Les déterminants sur un corps non commutatif. *Bulletin de la Société Mathématique de France*, 71:27–45, 1943.

[Dra83] Peter K. Draxl. *Skew Fields*. Cambridge University Press, 1983.

[DS17] Nicola Durante and Alessandro Siciliano. Non-Linear Maximum Rank Distance Codes in the Cyclic Model for the Field Reduction of Finite Geometries. *arXiv preprint arXiv:1704.02110*, 2017.

[EB97] Yves Edel and Jürgen Bierbauer. Twisted BCH-Codes. *Journal of Combinatorial Designs*, 5(5):377–389, 1997.

[EGN+92] Ronald J Evans, John Greene, Harald Niederreiter, et al. Linearized Polynomials and Permutation Polynomials of Finite Fields. *Michigan Mathematical Journal*, 39(3):405–413, 1992.

[EGRW16] Tuvi Etzion, Elisa Gorla, Alberto Ravagnani, and Antonia Wachter-Zeh. Optimal Ferrers Diagram Rank-Metric Codes. *IEEE Transactions on Information Theory*, 62(4):1616–1630, 2016.

[ElG85] Taher ElGamal. A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms. *IEEE Transactions on Information Theory*, 31(4):469–472, 1985.

[FL06] Cédric Faure and Pierre Loidreau. A New Public-Key Cryptosystem Based on the Problem of Reconstructing p-Polynomials. In *Coding and Cryptography*, pages 304–315. Springer, 2006.

[FS12] Michael A Forbes and Amir Shpilka. On Identity Testing of Tensors, Low-Rank Recovery and Compressed Sensing. In *ACM Symposium on the Theory of Computing*, pages 163–172, 2012.

[FT89] Gui-Liang Feng and Kenneth K. Tzeng. A Generalized Euclidean Algorithm for Multisequence Shift-Register Synthesis. *IEEE Transactions on Information Theory*, 35(3):584–594, 1989.

[FT91] Gui-Liang Feng and Kenneth K. Tzeng. A Generalization of the Berlekamp-Massey Algorithm for Multisequence Shift-Register Synthesis with Applications to Decoding Cyclic Codes. *IEEE Transactions on Information Theory*, 37(5):1274–1287, 1991.

[Gab85] Ernst M Gabidulin. Theory of Codes with Maximum Rank Distance. *Problems of Information Transmission*, 21(1):3–16, 1985.

[Gab91] Ernst M Gabidulin. A Fast Matrix Decoding Algorithm for Rank-Error-Correcting Codes. In *Workshop on Algebraic Coding*, pages 126–133. Springer, 1991.

[Gal68] Robert G Gallager. *Information Theory and Reliable Communication*, volume 2. Springer, 1968.

[Gao03] Shuhong Gao. A New Algorithm for Decoding Reed–Solomon Codes. In *Communications, Information and Network Security*, pages 55–68. Springer, 2003.

[GBL00] Ernst M Gabidulin, Martin Bossert, and P Lusina. Space-Time Codes Based on Rank Codes. In *IEEE International Symposium on Information Theory*, page 284, 2000.

[GG99] Joachim von zur Gathen and Jürgen Gerhard. *Modern Computer Algebra*. Cambridge university press, 1999.

[Gib95] JK Gibson. Severely Denting the Gabidulin Version of the McEliece Public Key Cryptosystem. *Designs, Codes and Cryptography*, 6(1):37–45, 1995.

[Gib96] Keith Gibson. The Security of the Gabidulin Public Key Cryptosystem. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 212–223. Springer, 1996.

[GJV03] Pascal Giorgi, Claude-Pierre Jeannerod, and Gilles Villard. On the Complexity of Polynomial Matrix Computations. In *International Symposium on Symbolic and Algebraic Computation*, pages 135–142, 2003.

[Gly86] David G Glynn. The Non-Classical 10-Arc of PG(4, 9). *Discrete Mathematics*, 59(1):43–51, 1986.

[Gol49] Marcel JE Golay. Notes on Digital Coding. *Proceedings of the Institute of Radio Engineers*, 37(6):657–657, 1949.

[Gop83] Valery Denisovich Goppa. Algebraico-Geometric Codes. *Izvestiya: Mathematics*, 21(1):75–91, 1983.

[GOT17] P. Gaborit, A. Otmani, and H. Talé Kalachi. Polynomial-Time Key Recovery Attack on the Faure–Loidreau Scheme based on Gabidulin Codes. *preprint*, April 2017. URL: `https://arxiv.org/abs/1606.07760`.

[GPT91] Ernst M Gabidulin, AV Paramonov, and OV Tretjakov. Ideals Over a Non-Commutative Ring and Their Application in Cryptology. In *Advances in Cryptology—EUROCRYPT'91*, pages 482–489. Springer, 1991.

[GR08] Venkatesan Guruswami and Atri Rudra. Explicit Codes Achieving List Decoding Capacity: Error-Correction with Optimal Redundancy. *IEEE Transactions on Information Theory*, 54(1):135–150, 2008.

[GS98] Venkatesan Guruswami and Madhu Sudan. Improved Decoding of Reed–Solomon and Algebraic-Geometric Codes. In *IEEE Foundations of Computer Science*, pages 28–37, 1998.

[GS99] Venkatesan Guruswami and Madhu Sudan. Improved Decoding of Reed–Solomon Codes and Algebraic-Geometric Codes. *IEEE Transactions on Information Theory*, 45(6):1757–1767, 1999.

[GW14] Venkatesan Guruswami and Carol Wang. Explicit Rank-Metric Codes List-Decodable with Optimal Redundancy. In *International Workshop on Randomization and Computation*, 2014. arXiv: 1311.7084.

[GX13] Venkatesan Guruswami and Chaoping Xing. List Decoding Reed–Solomon, Algebraic-Geometric, and Gabidulin Subcodes Up to the Singleton Bound. In *ACM Symposium on the Theory of Computing*, pages 843–852. ACM, 2013.

[GY08] Maximilien Gadouleau and Zhiyuan Yan. Complexity of Decoding Gabidulin Codes. In *IEEE Annual Conference on Information Science and Systems*, pages 1081–1085, 2008.

[Ham50] Richard W Hamming. Error Detecting and Error Correcting Codes. *Bell Labs Technical Journal*, 29(2):147–160, 1950.

[HM17] Anna-Lena Horlemann-Trautmann and Kyle Marshall. New Criteria for MRD and Gabidulin Codes and Some Rank-Metric Code Constructions. *Advances in Mathematics of Communications*, 11(3):533–548, 2017.

[HN99] Tom Høholdt and Rasmus Refslund Nielsen. Decoding Hermitian Codes with Sudan's Algorithm. In *International Symposium on Applied Algebra, Algebraic Algorithms, and Error-Correcting Codes*, volume 13, pages 260–270. Springer, 1999.

[HP98] Xiaohan Huang and Victor Y Pan. Fast Rectangular Matrix Multiplication and Applications. *Journal of Complexity*, 14(2):257–299, 1998.

[HS10] Yahia Hassan and Vladimir Sidorenko. Fast Recursive Linearized Feedback Shift Register Synthesis. In *International Workshop on Algebraic and Combinatorical Coding Theory*, pages 162–167, 2010.

[HTV82] H Hoeve, J Timmermans, and LJ Vries. Error Correction and Concealment in the Compact Disc System. *Origins and Successors of the Compact Disc*, page 82, 1982.

[Hua51] Loo-Keng Hua. A Theorem on Matrices over a Field and its Applications. *Acta Mathematica Sinica*, 1(2):109–163, 1951.

[Huf98] W Cary Huffman. Codes and Groups. *Handbook of Coding Theory*, 2(Part 2):1345–1440, 1998.

[HvLP98] Tom Høholdt, Jacobus H van Lint, and Ruud Pellikaan. Algebraic Geometry Codes. *Handbook of Coding Theory*, 1(Part 1):871–961, 1998.

[JH04] Jørn Justesen and Tom Høholdt. *A Course in Error-Correcting Codes*, volume 1. European Mathematical Society, 2004.

[Joh62] Selmer M. Johnson. A New Upper Bound for Error-Correcting Codes. *IEEE Transactions on Information Theory*, 46:203–207, 1962.

[JTH04] Jørn Justesen, Christian Thommesen, and Tom Høholdt. Decoding of Concatenated Codes with Interleaved Outer Codes. In *IEEE International Symposium on Information Theory*, pages 328–328, 2004.

[Jus76] Jørn Justesen. On the Complexity of Decoding Reed-Solomon Codes (Corresp.). *IEEE Transactions on Information Theory*, 22(2):237–238, March 1976.

[Kai80] Thomas Kailath. *Linear Systems*. Prentice-Hall, 1980.

[Kam14] Sabine Kampf. Bounds on Collaborative Decoding of Interleaved Hermitian Codes and Virtual Extension. *Designs, Codes and Cryptography*, 70(1-2):9–25, 2014.

[KG05] Alexander Kshevetskiy and Ernst Gabidulin. The New Construction of Rank Codes. In *IEEE International Symposium on Information Theory*, pages 2105–2108, 2005.

[KK08] Ralf Koetter and Frank R Kschischang. Coding for Errors and Erasures in Random Network Coding. *IEEE Transactions on Information Theory*, 54(8):3579–3591, 2008.

[KL97] V Yu Krachkovsky and Yuan Xing Lee. Decoding for Iterative Reed–Solomon Coding Schemes. *IEEE Transactions on Magnetics*, 33(5):2740–2742, 1997.

[KL98] V Yu Krachkovsky and Yuan Xing Lee. Decoding of Parallel Reed–Solomon Codes with Applications to Product and Concatenated Codes. In *IEEE International Symposium on Information Theory*, page 55, 1998.

[Kop12] Swastik Kopparty. List-Decoding Multiplicity Codes. In *Electronic Colloquium on Computational Complexity*, volume 19, page 2, 2012.

[KRS17] Mohamed Khochtali, Johan Rosenkilde né Nielsen, and Arne Storjohann. Popov Form Computation for Matrices of Ore Polynomials. In *International Symposium in Symbolic and Algebraic Computation*, 2017.

[KZHP08] ShanXue Ke, BenSheng Zeng, WenBao Han, and Victor Y Pan. Fast Rectangular Matrix Multiplication and Some Applications. *Science in China Series A: Mathematics*, 51(3):389–406, 2008.

[Lan02] Serge Lang. *Algebra*. Springer-Verlag New York, 2002.

[Len85] Arjen K. Lenstra. Factoring Multivariate Polynomials over Finite Fields. *Journal of Computer and System Sciences*, 30(2):235–246, 1985.

[LFT02] Youjian Liu, Michael P Fitz, and Oscar Yassuo Takeshita. A Rank Criterion for QAM Space-Time Codes. *IEEE Transactions on Information Theory*, 48(12):3062–3079, 2002.

[LGB03] Paul Lusina, Ernst Gabidulin, and Martin Bossert. Maximum Rank Distance Codes as Space-Time Codes. *IEEE Transactions on Information Theory*, 49(10):2757–2760, 2003.

[LN97] Rudolf Lidl and Harald Niederreiter. *Finite Fields*, volume 20. Cambridge University Press, 1997.

[LO06] Pierre Loidreau and Raphael Overbeck. Decoding Rank Errors Beyond the Error Correcting Capability. In *International Workshop on Algebraic and Combinatorical Coding Theory*, pages 186–190, 2006.

[LO08] Kwankyu Lee and Michael E. O'Sullivan. List Decoding of Reed–Solomon Codes from a Gröbner Basis Perspective. *Journal of Symbolic Computation*, 43(9):645 – 658, 2008.

[Loi06] Pierre Loidreau. A Welch–Berlekamp Like Algorithm for Decoding Gabidulin Codes. In *Coding and Cryptography*, pages 36–45. Springer, 2006.

[Loi10] Pierre Loidreau. Designing a Rank Metric Based McEliece Cryptosystem. In *Post-Quantum Cryptography*, pages 142–152, 2010.

[Loi16] Pierre Loidreau. An Evolution of GPT Cryptosystem. In *Workshop on Algebraic and Combinatorial Coding Theory*, 2016.

[LRMV14] Hiram H López, Carlos Rentería-Márquez, and Rafael H Villarreal. Affine Cartesian Codes. *Designs, Codes and Cryptography*, 71(1):5–19, 2014.

[LSS14] Wenhui Li, Vladimir Sidorenko, and Danilo Silva. On Transform-Domain Error and Erasure Correction by Gabidulin Codes. *Designs, Codes and Cryptography*, 73(2):571–586, 2014.

[LTZ15] Guglielmo Lunardon, Rocco Trombetti, and Yue Zhou. Generalized Twisted Gabidulin codes. *arXiv preprint arXiv:1507.07855*, 2015.

[Mah12] Hessam Mahdavifar. *List Decoding of Subspace Codes and Rank-Metric Codes*. PhD thesis, University of California, San Diego, 2012.

[McE78] Robert J. McEliece. A Public-Key Cryptosystem Based On Algebraic Coding Theory. *Deep Space Network Progress Report*, 42(44):114–116, 1978.

[Mid11] Johannes Middeke. *A Computational View on Normal Forms of Matrices of Ore Polynomials*. PhD thesis, Research Institute for Symbolic Computation (RISC), 2011.

[MS77] Florence Jessie MacWilliams and Neil James Alexander Sloane. *The Theory of Error Correcting Codes*. Elsevier, 1977.

[MS03] Thom Mulders and Arne Storjohann. On Lattice Reduction for Polynomial Matrices. *Journal of Symbolic Computation*, 35(4):377–401, 2003.

[MV13] Hessam Mahdavifar and Alexander Vardy. Algebraic List-Decoding of Subspace Codes. *IEEE Transactions on Information Theory*, 59(12):7814–7828, 2013.

[Nat17] National Institute of Standards and Technology (NIST). Post-Quantum Cryptography Standardization, 2017. URL: `https://csrc.nist.gov/Projects/Post-Quantum-Cryptography`.

[NB15] Johan S. R. Nielsen and Peter Beelen. Sub-Quadratic Decoding of One-Point Hermitian Codes. *IEEE Transactions on Information Theory*, 61(6):3225–3240, 2015.

[Nie86] Harald Niederreiter. Knapsack-Type Cryptosystems and Algebraic Coding Theory. *Problems of Control and Information Theory*, 15(2):159–166, 1986.

[Nie13a] Johan S. R. Nielsen. Generalised Multi-Sequence Shift-Register Synthesis Using Module Minimisation. In *IEEE International Symposium on Information Theory*, pages 882–886, 2013.

[Nie13b] Johan S. R. Nielsen. *List Decoding of Algebraic Codes*. PhD thesis, Technical University of Denmark, 2013.

[Nie14] Johan S. R. Nielsen. Power Decoding Reed–Solomon Codes up to the Johnson Radius. In *International Workshop on Algebraic and Combinatorical Coding Theory*, 2014.

[NRS17] Vincent Neiger, Johan Rosenkilde, and Eric Schost. Fast Computation of the Roots of Polynomials Over the Ring of Power Series. In *International Symposium on Symbolic and Algebraic Computation*, 2017.

[OM11] T. Onoda and K. Miwa. Hierarchized Two-Dimensional Code, Creation Method Thereof, and Read Method Thereof, May 25 2011. EP Patent 1,916,619.

[OÖ16] Kamil Otal and Ferruh Özbudak. Explicit Construction of Some Non-Gabidulin Linear Maximum Rank Distance Codes. *Advances in Mathematics of Communications*, 10(3), 2016.

[OÖ17] Kamil Otal and Ferruh Özbudak. Additive rank metric codes. *IEEE Transactions on Information Theory*, 63(1):164–168, 2017.

[Ore33a] Øystein Ore. On a Special Class of Polynomials. *Transactions of the American Mathematical Society*, 35(3):559–584, 1933.

[Ore33b] Øystein Ore. Theory of Non-Commutative Polynomials. *Annals of Mathematics*, pages 480–508, 1933.

[OS06] Zach Olesh and Arne Storjohann. The Vector Rational Function Reconstruction Problem. In *Waterloo Workshop in Computer Algebra*, pages 137–149, 2006.

[OS09] Raphael Overbeck and Nicolas Sendrier. Code-Based Cryptography. In *Post-Quantum Cryptography*, pages 95–145. Springer, 2009.

[Ove05] Raphael Overbeck. A New Structural Attack for GPT and Variants. In *International Conference on Cryptology in Malaysia*, pages 50–63. Springer, 2005.

[Ove06] Raphael Overbeck. Extending Gibson's Attacks on the GPT Cryptosystem. In *Coding and Cryptography*, pages 178–188. Springer, 2006.

[Ove07] Raphael Overbeck. *Public Key Cryptography Based on Coding Theory*. PhD thesis, TU Darmstadt, 2007.

[Ove08] Raphael Overbeck. Structural Attacks for Public Key Cryptosystems Based on Gabidulin Codes. *Journal of Cryptology*, 21(2):280–301, 2008.

[Par07] Farzad Parvaresh. *Algebraic List-Decoding of Error-Correcting Codes*. PhD thesis, University of California, San Diego, 2007.

[Pet10] Christiane Peters. Information-Set Decoding for Linear Codes over Fq. *International Conference on Post-Quantum Cryptography*, 2010:81–94, 2010.

[Pet12] Pete L. Clark. Non-Commutative Algebra. University of Georgia. `http://math.uga.edu/~pete/noncommutativealgebra.pdf`, 2012.

[Pra62] Eugene Prange. The Use of Information Sets in Decoding Cyclic Codes. *IRE Transactions on Information Theory*, 8(5):5–9, 1962.

[PS73] Michael S Paterson and Larry J Stockmeyer. On the Number of Nonscalar Multiplications Necessary to Evaluate Polynomials. *SIAM Journal on Computing*, 2(1):60–66, 1973.

[PT91] AV Paramonov and OV Tretjakov. An Analogue of Berlekamp–Massey Algorithm for Decoding Codes in Rank Metric. *Moscow Institute of Physics and Technology*, 1991.

[PV04] Farzad Parvaresh and Alexander Vardy. Multivariate Interpolation Decoding Beyond the Guruswami–Sudan Radius. In *Allerton Conference on Communication, Control and Computing*, 2004.

[PW72] William Wesley Peterson and Edward J Weldon. *Error-Correcting Codes*. MIT press, 1972.

[Ran15] Hugues Randriambololona. On Products and Powers of Linear Codes Under Componentwise Multiplication. In *International Conference on Arithmetic, Geometry, Cryptography, and Coding Theory*, 2015.

[RL89a] Ron M. Roth and Abraham Lempel. A Construction of Non-Reed-Solomon Type MDS Codes. *IEEE Transactions on Information Theory*, 35(3):655–657, May 1989.

[RL89b] Ron M. Roth and Abraham Lempel. On MDS Codes via Cauchy Matrices. *IEEE Transactions on Information Theory*, 35(6):1314–1319, 1989.

[RL92] Ron M. Roth and Abraham Lempel. t-Sum Generators of Finite Abelian Groups. *Discrete Mathematics*, 103(3):279–292, May 1992.

[Rob15a] Gwezheneg Robert. A New Constellation for Space-Time Coding. In *International Workshop on Coding and Cryptography*, 2015.

[Rob15b] Gwezheneg Robert. *Codes de Gabidulin en Caractéristique Nulle. Application au Codage Espace-Temps*. PhD thesis, Université Rennes 1, 2015.

[Ros18] Johan Rosenkilde. Power Decoding Reed–Solomon Codes up to the Johnson Radius. *Advances in Mathematics of Communications*, 12(1):81–106, 2018.

[Rot91] Ron M. Roth. Maximum-Rank Array Codes and their Application to Crisscross Error Correction. *IEEE Transactions on Information Theory*, 37(2):328–336, 1991.

[Rot96] Ron M Roth. Tensor Codes for the Rank Metric. *IEEE Transactions on Information Theory*, 42(6):2146–2157, 1996.

[Rot06] Ron Roth. *Introduction to Coding Theory*. Cambridge University Press, 2006.

[RP04a] Gerd Richter and Simon Plass. Error and Erasure Decoding of Rank-Codes with a Modified Berlekamp–Massey Algorithm. *ITG Fachbericht*, pages 203–210, 2004.

[RP04b] Gerd Richter and Simon Plass. Fast Decoding of Rank-Codes with Rank Errors and Column Erasures. In *IEEE International Symposium on Information Theory*, pages 398–398, 2004.

[RR00] Ron M. Roth and Gitit Ruckenstein. Efficient Decoding of Reed–Solomon Codes Beyond Half the Minimum Distance. *IEEE Transactions on Information Theory*, 46(1):246 –257, 2000.

[RR17] Joachim Rosenthal and Tovohery Randrianarisoa. A Decoding Algorithm for Twisted Gabidulin Codes. In *IEEE International Symposium on Information Theory*, pages 2771–2774, 2017.

[RS60] Irving S Reed and Gustave Solomon. Polynomial Codes over Certain Finite Fields. *Journal of the Society for Industrial and Applied Mathematics*, 8(2):300–304, 1960.

[RS85] Ron M Roth and Gadiel Seroussi. On Generator Matrices of MDS Codes (Corresp.). *IEEE Transactions on Information Theory*, 31(6):826–830, 1985.

[RSA78] Ronald L Rivest, Adi Shamir, and Leonard Adleman. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.

[RW15] Netanel Raviv and Antonia Wachter-Zeh. Some Gabidulin Codes Cannot be List Decoded Efficiently at Any Radius. In *IEEE International Symposium on Information Theory*, pages 6–10, June 2015.

[RW16] Netanel Raviv and Antonia Wachter-Zeh. Some Gabidulin Codes Cannot be List Decoded Efficiently at Any Radius. *IEEE Transactions on Information Theory*, 62(4):1605–1615, 2016.

[S$^+$] William A. Stein et al. SageMath Software. http://www.sagemath.org.

[SB10] Vladimir Sidorenko and Martin Bossert. Decoding Interleaved Gabidulin Codes and Multisequence Linearized Shift-Register Synthesis. In *IEEE International Symposium on Information Theory*, pages 1148–1152, 2010.

[SB14] Vladimir Sidorenko and Martin Bossert. Fast Skew-Feedback Shift-Register Synthesis. *Designs, Codes and Cryptography*, 70(1-2):55–67, 2014.

[Seg55] Beniamino Segre. Curve Razionali Normali Ek-Archi Negli Spazi Finiti. *Annali di Matematica Pura ed Applicata*, 39(1):357–379, 1955.

[Sha48] Claude Elwood Shannon. A Mathematical Theory of Communication. *Bell System Technical Journal*, 27(3):379–423, 1948.

[She16] John Sheekey. A New Family of Linear Maximum Rank Distance Codes. *Advances in Mathematics of Communications*, pages 475–488, 2016.

[Sho99] Peter W Shor. Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. *SIAM Review*, 41(2):303–332, 1999.

[Sil09] Danilo Silva. *Error Control for Network Coding*. PhD thesis, University of Toronto, 2009.

[Sin64] Richard Singleton. Maximum Distance q-nary Codes. *IEEE Transactions on Information Theory*, 10(2):116–118, 1964.

[SJB11] Vladimir Sidorenko, Lan Jiang, and Martin Bossert. Skew-Feedback Shift-Register Synthesis and Decoding Interleaved Gabidulin Codes. *IEEE Transactions on Information Theory*, 57(2):621–632, 2011.

[SK07] Danilo Silva and Frank R Kschischang. Rank-Metric Codes for Priority Encoding Transmission with Network Coding. In *Canadian Workshop on Information Theory*, pages 81–84. IEEE, 2007.

[SK09] Danilo Silva and Frank R. Kschischang. Fast Encoding and Decoding of Gabidulin Codes. In *IEEE International Symposium on Information Theory*, pages 2858–2862, June 2009.

[SKK08] Danilo Silva, Frank R Kschischang, and Ralf Koetter. A Rank-Metric Approach to Error Control in Random Network Coding. *IEEE Transactions on Information Theory*, 54(9):3951–3967, 2008.

[SRB11] Vladimir Sidorenko, Gerd Richter, and Martin Bossert. Linearized Shift-Register Synthesis. *IEEE Transactions on Information Theory*, 57(9):6025–6032, 2011.

[SRV12] Natalia Silberstein, Ankit S. Rawat, and Sriram Vishwanath. Error Resilience in Distributed Storage via Rank-Metric Codes. In *Allerton Conference on Communication, Control and Computing*, pages 1150–1157, October 2012.

[SS92] Vladimir M Sidelnikov and Sergey O Shestakov. On Insecurity of Cryptosystems Based on Generalized Reed–Solomon Codes. *Discrete Mathematics and Applications*, 2(4):439–444, 1992.

[SS11] V.R. Sidorenko and G. Schmidt. A Linear Algebraic Approach to Multisequence Shift-Register Synthesis. *Problems of Information Transmission*, 47(2):149–165, 2011.

[SSB05] Georg Schmidt, Vladimir R Sidorenko, and Martin Bossert. Interleaved Reed–Solomon Codes in Concatenated Code Designs. In *IEEE Information Theory Workshop*, pages 5–pp, 2005.

[SSB06] Georg Schmidt, Vladimir Sidorenko, and Martin Bossert. Decoding Reed-Solomon Codes Beyond Half the Minimum Distance Using Shift-register synthesis. In *IEEE International Symposium on Information Theory*, pages 459–463. IEEE, 2006.

[SSB07] Georg Schmidt, Vladimir Sidorenko, and Martin Bossert. Enhancing the Correcting Radius of Interleaved Reed–Solomon Decoding Using Syndrome Extension Techniques. In *IEEE International Symposium on Information Theory*, pages 1341–1345, 2007.

[SSB08] Vladimir Sidorenko, Georg Schmidt, and Martin Bossert. Decoding Punctured Reed–Solomon Codes up to the Singleton Bound. In *International ITG Conference on Source and Channel Coding*, pages 1–6, 2008.

[SSB09] Georg Schmidt, Vladimir R Sidorenko, and Martin Bossert. Collaborative Decoding of Interleaved Reed–Solomon Codes and Concatenated Code Designs. *IEEE Transactions on Information Theory*, 55(7):2991–3012, 2009.

[SSB10] Georg Schmidt, Vladimir R Sidorenko, and Martin Bossert. Syndrome Decoding of Reed–Solomon Codes Beyond Half the Minimum Distance Based on Shift-Register Synthesis. *IEEE Transactions on Information Theory*, 56(10):5245–5252, 2010.

[Sti88] Henning Stichtenoth. A Note on Hermitian Codes over GF (q/sup 2/). *IEEE Transactions on Information Theory*, 34(5):1345–1348, 1988.

[Sti09] Henning Stichtenoth. *Algebraic Function Fields and Codes*, volume 254. Springer Science & Business Media, 2009.

[Str69] Volker Strassen. Gaussian Elimination is not Optimal. *Numerische Mathematik*, 13(4):354–356, 1969.

[Sud97] Madhu Sudan. Decoding of Reed–Solomon Codes Beyond the Error-Correction Bound. *Journal of Complexity*, 13(1):180–193, 1997.

[SW99] Mohammad Amin Shokrollahi and Hal Wasserman. List Decoding of Algebraic-Geometric Codes. *IEEE Transactions on Information Theory*, 45(2):432–437, 1999.

[SWC12] Vladimir Sidorenko, Antonia Wachter-Zeh, and Di Chen. On Fast Decoding of Interleaved Gabidulin Codes. In *IEEE Problems of Redundancy in Information and Control Systems*, pages 78–83, 2012.

[Tae06] Lenny Taelman. Dieudonné Determinants for Skew Polynomial Rings. *Journal of Algebra and Its Applications*, 5(01):89–93, 2006.

[VL12] Jacobus Hendricus Van Lint. *Introduction to Coding Theory*, volume 86. Springer Science & Business Media, 2012.

[Wac13] Antonia Wachter-Zeh. *Decoding of Block and Convolutional Codes in Rank Metric*. PhD thesis, Ulm University and University of Rennes, 2013.

[WAS13] Antonia Wachter-Zeh, Valentin Afanassiev, and Vladimir Sidorenko. Fast Decoding of Gabidulin Codes. *Designs, Codes and Cryptography*, 66(1):57–73, 2013.

[WHPH87] William Wu, David Haccoun, Robert Peile, and Yasuo Hirata. Coding for Satellite Communication. *IEEE Journal on Selected Areas in Communications*, 5(4):724–748, 1987.

[Wie06] Christian Wieschebrink. An Attack on a Modified Niederreiter Encryption Scheme. In *Public Key Cryptography*, volume 3958, pages 14–26. Springer, 2006.

[Wie10] Christian Wieschebrink. Cryptanalysis of the Niederreiter Public Key Scheme Based on GRS Subcodes. In *International Workshop on Post-Quantum Cryptography*, pages 61–72. Springer, 2010.

[WL13] Baofeng Wu and Zhuojun Liu. Linearized Polynomials over Finite Fields Revisited. *Finite Fields and Their Applications*, 22:79–100, 2013.

[WSB10] Antonia Wachter, Vladimir Sidorenko, and Martin Bossert. A Fast Linearized Euclidean Algorithm for Decoding Gabidulin Codes. In *International Workshop on Algebraic and Combinatorical Coding Theory*, pages 298–303, 2010.

[Wu08] Yingquan Wu. New List Decoding Algorithms for Reed-Solomon and BCH Codes. *IEEE Transactions on Information Theory*, 54(8):3611–3630, 2008.

[WZ13] Antonia Wachter-Zeh and Alexander Zeh. Interpolation-Based Decoding of Interleaved Gabidulin Codes. In *International Workshop on Coding and Cryptography*, pages 528–538, 2013.

[WZ14] Antonia Wachter-Zeh and Alexander Zeh. List and Unique Error-Erasure Decoding of Interleaved Gabidulin Codes with Interpolation Techniques. *Designs, Codes and Cryptography*, 73(2):547–570, 2014.

[WZB14] Antonia Wachter-Zeh, Alexander Zeh, and Martin Bossert. Decoding Interleaved Reed–Solomon Codes Beyond Their Joint Error-Correcting Capability. *Designs, Codes and Cryptography*, 71(2):261–281, 2014.

[XLYS13] H. Xie, J. Lin, Z. Yan, and B. W. Suter. Linearized Polynomial Interpolation and Its Applications. *IEEE Transactions on Signal Processing*, 61(1):206–217, January 2013.

[YK92] Kyeongcheol Yang and P Vijay Kumar. On the True Minimum Distance of Hermitian Codes. In *Coding Theory and Algebraic Geometry*, pages 99–107. Springer, 1992.

[ZGA11] A. Zeh, C. Gentner, and D. Augot. An Interpolation Procedure for List Decoding Reed-Solomon Codes Based on Generalized Key Equations. *IEEE Transactions on Information Theory*, 57(9):5946–5959, 2011.

[ZL12] Wei Zhou and George Labahn. Efficient Algorithms for Order Basis Computation. *Journal of Symbolic Computation*, 47(7):793–819, July 2012.

Publications Containing Parts of this Thesis

[BBPR18] Peter Beelen, Martin Bossert, Sven Puchinger, and Johan Rosenkilde né Nielsen. Structural Properties of Twisted Reed–Solomon Codes with Applications to Code-Based Cryptography. In *IEEE International Symposium on Information Theory*, 2018.

[BPR17] Peter Beelen, Sven Puchinger, and Johan Rosenkilde né Nielsen. Twisted Reed–Solomon Codes. In *IEEE International Symposium on Information Theory*, 2017.

[LNPS15] Wenhui Li, Johan S.R. Nielsen, Sven Puchinger, and Vladimir Sidorenko. Solving Shift Register Problems over Skew Polynomial Rings Using Module Minimisation. In *International Workshop on Coding and Cryptography*, 2015.

[MPMB16] Sven Müelich, Sven Puchinger, David Mödinger, and Martin Bossert. An Alternative Decoding Method for Gabidulin Codes in Characteristic Zero. In *IEEE International Symposium on Information Theory*, pages 2549–2553, 2016.

[PBR17] Sven Puchinger, Irene Bouw, and Johan Rosenkilde né Nielsen. Improved Power Decoding of One-Point Hermitian Codes. In *International Workshop on Coding and Cryptography*, 2017.

[PMM+17] Sven Puchinger, Sven Müelich, David Mödinger, Johan Rosenkilde né Nielsen, and Martin Bossert. Decoding Interleaved Gabidulin Codes Using Alekhnovich's Algorithm. *Electronic Notes in Discrete Mathematics*, 57:175–180, 2017.

[PR17] Sven Puchinger and Johan Rosenkilde né Nielsen. Decoding of Interleaved Reed–Solomon Codes Using Improved Power Decoding. In *IEEE International Symposium on Information Theory*, 2017.

[PRLS17] Sven Puchinger, Johan Rosenkilde né Nielsen, Wenhui Li, and Vladimir Sidorenko. Row Reduction Applied to Decoding of Rank-Metric and Subspace Codes. *Designs, Codes and Cryptography*, 82(1-2):389–409, 2017.

[PRS17] Sven Puchinger, Rosenkilde né Nielsen, and John Sheekey. Further Generalisations of Twisted Gabidulin Codes. In *International Workshop on Coding and Cryptography*, 2017.

[PW16] Sven Puchinger and Antonia Wachter-Zeh. Sub-Quadratic Decoding of Gabidulin Codes. In *IEEE International Symposium on Information Theory*, pages 2554–2558, 2016.

[PW18] Sven Puchinger and Antonia Wachter-Zeh. Fast Operations on Linearized Polynomials and their Applications in Coding Theory. *Journal of Symbolic Computation*, 89:194–215, 2018.

Preprints Containing Parts of this Thesis

[PRB17] Sven Puchinger, Johan Rosenkilde né Nielsen, and Irene Bouw. Improved Power Decoding of Interleaved One-Point Hermitian Codes. *submitted to Designs, Codes, Cryptography, arXiv preprint arXiv:1801.07006*, 2017.

Other Publications and Preprints by the Author of this Thesis

[CHPB17] Yuval Cassuto, Evyatar Hemo, Sven Puchinger, and Martin Bossert. Multi-Block Interleaved Codes for Local and Global Read Access. In *IEEE International Symposium on Information Theory*, 2017.

[HKS+15] Matthias Hiller, Ludwig Kürzinger, Georg Sigl, Sven Müelich, Sven Puchinger, and Martin Bossert. Low-Area Reed Decoding in a Generalized Concatenated Code Construction for PUFs. In *IEEE Computer Society Annual Symposium on VLSI*, 2015.

[MPB+14] Sven Müelich, Sven Puchinger, Martin Bossert, Matthias Hiller, and Georg Sigl. Error Correction for Physical Unclonable Functions Using Generalized Concatenated Codes. In *International Workshop on Algebraic and Combinatorical Coding Theory*, 2014.

[MPB17a] Mostafa H. Mohamed, Sven Puchinger, and Martin Bossert. Guruswami–Sudan List Decoding for Complex Reed-Solomon Codes. In *International ITG Conference on Systems, Communications and Coding*, 2017.

[MPB17b] Sven Müelich, Sven Puchinger, and Martin Bossert. Low-Rank Matrix Recovery using Gabidulin Codes in Characteristic Zero. *Electronic Notes in Discrete Mathematics*, 57:161–166, 2017.

[PCF+15] Sven Puchinger, Michael Cyran, Robert F.H. Fischer, Martin Bossert, and Johannes B. Huber. Error Correction for Differential Linear Network Coding in Slowly Varying Networks. In *International ITG Conference on Systems, Communications and Coding*, 2015.

[PMB+15] Sven Puchinger, Sven Müelich, Martin Bossert, Matthias Hiller, and Georg Sigl. On Error Correction for Physical Unclonable Functions. In *International ITG Conference on Systems, Communications and Coding*, 2015.

[PMB17] Sven Puchinger, Sven Müelich, and Martin Bossert. On the Success Probability of Decoding (Partial) Unit Memory Codes. In *International Workshop on Optimal Codes and Related Topics*, 2017.

[PMIB17] Sven Puchinger, Sven Müelich, Karim Ishak, and Martin Bossert. Code-Based Cryptosystems Using Generalized Concatenated Codes. In Ilias S. Kotsireas and Edgar Martínez-Moro, editors, *Springer Proceedings in Mathematics & Statistics: Applications of Computer Algebra: Kalamata, Greece, July 20–23 2015*, volume 198, pages 397–423. Springer International Publishing, 2017. doi:10.1007/978-3-319-56932-1_26.

[PMWZB17] Sven Puchinger, Sven Müelich, Antonia Wachter-Zeh, and Martin Bossert. Timing Attack Resilient Decoding Algorithms for Physical Unclonable Functions. In *International ITG Conference on Systems, Communications and Coding*, 2017.

[PSBF16] Sven Puchinger, Sebastian Stern, Martin Bossert, and Robert F.H. Fischer. Space-Time Codes Based on Rank-Metric Codes and Their Decoding. In *IEEE International Symposium on Wireless Communication Systems*, pages 125–130, 2016.

[PWB14] Sven Puchinger, Antonia Wachter-Zeh, and Martin Bossert. Improved Decoding of Partial Unit Memory Codes Using List Decoding of Reed–Solomon Codes. In *International Zurich Seminar on Communications*, 2014.

[SPB17] Ulrich Speidel, Sven Puchinger, and Martin Bossert. Constraints for Coded Tunnels Across Long Latency Bottlenecks with ARQ-based Congestion Control. In *IEEE International Symposium on Information Theory*, 2017.

[WPR18] Antonia Wachter-Zeh, Sven Puchinger, and Julian Renner. Repairing the Faure–Loidreau Public-Key Cryptosystem. In *IEEE International Symposium on Information Theory*, 2018.

For data protection reasons, the curriculum vitae is not included in the online version.

For data protection reasons, the curriculum vitae is not included in the online version.