Technische Universität München

Fakultät für Elektrotechnik und Informationstechnik

Post-Quantum Cryptography in the Hamming Metric, the Rank Metric, and the Sum-Rank Metric

Julian Wilhelm Renner

Vollständiger Abdruck der von der Fakultät für Elektrotechnik und Informationstechnik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Ingenieurwissenschaften

genehmigten Dissertation.

Vorsitzender:		Prof. Dr.-Ing. Georg Sigl
Prüferinnen der Dissertation:	1.	Prof. Dr.-Ing. Antonia Wachter-Zeh
	2.	Prof. Dr. Anna-Lena Horlemann

Die Dissertation wurde am 17.11.2021 bei der Technischen Universität München eingereicht und durch die Fakultät für Elektrotechnik und Informationstechnik am 22.03.2022 angenommen.

Preface

This doctoral thesis is based on parts of the results that I developed during my time at the Institute for Communications Engineering of the Technical University of Munich (TUM) in the group for Coding and Cryptography. I want to take this opportunity to express my gratitude to all the people who were part of this great journey.

First, I am very grateful to Antonia Wachter-Zeh for giving me this unique opportunity, providing me with an inspiring and motivating environment, and being a remarkable doctoral adviser. The fruitful discussions with Antonia as well as the many conferences, research visits, and seminars in various parts of the world contributed to the quality and the presentation of my scientific contributions in this dissertation. Furthermore, I am very thankful to Antonia for the freedoms I had in all the aspects of my time at TUM.

Although he had no official duties with respect to my supervision, Sven Puchinger advised me throughout almost my entire doctoral program and worked with me on many different topics. I have always been very impressed by his extraordinarily fast comprehension and his ability to communicate complicated concepts in an easy-to-understand way. I want to thank Sven not only for our productive collaboration but also for the invitations to visit him and his great family both in Ulm and in Copenhagen.

Likewise, I am grateful to Pierre Loidreau for being my mentor and inviting me to Rennes as well as for the valuable collaboration and his advice. During my visits to Rennes, Pierre and Julien Lavauzelle ensured that I felt welcome and introduced me to the delicious Breton cuisine.

I am thankful to Anna-Lena Horlemann for agreeing to be a reviewer of this dissertation and the second examiner of my doctoral examination, and Georg Sigl for organizing and chairing my examination.

During my time at TUM, I had the opportunity to work with many brilliant researchers. These collaborations not only led to the results presented in this doctoral thesis, but also significantly shaped my approach to tackle new problems. To this end, I want to thank my co-authors Hannes Bartz, Ragnar Freij-Hollanti, Camilla Hollanti, Lukas Holzbaur, Anna-Lena Horlemann, Thomas Jerkovits, Johannes Kunz, Julien Lavauzelle, Lia Liu, Pierre Loidreau, Georg Maringer, Alessandro Neri, Sven Puchinger, Johan Rosenkilde, Thomas Schamberger, Georg Sigl, Antonia Wachter-Zeh, and Jens Zumbrägel. Furthermore, I am very thankful to Oliver De Candido, Lukas, Andreas Vollstädt, and Violetta Weger for their comments that greatly helped to improve this manuscript.

I am also very grateful to my office-mates Lukas, Andreas Lenz, and Violetta. Our countless technical and non-technical discussions, the symposiums we visited together, and the pleasant atmosphere in our office significantly contributed to making the past years so special and productive. Furthermore, I want to thank Lorenz Welter for our entertaining conversations, our fierce discussions, and his great support before, during, and after the oral defense of my dissertation. I am also very thankful to all the people who designed and built my doctoral hat – it turned out awesome!

Last but certainly not least, I want to thank my family and friends for supporting me. In particular, I am deeply grateful for the encouragement and support provided by my father Willi, my late mother Maria, my brother Daniel with his wife Claudia and their son Ludwig, and my aunt Hannerl.

Julian Wilhelm Renner
München, April 2022

Abstract

Due to the rapid improvements of quantum supercomputers, several classical public-key cryptographic schemes could become insecure in the near future. Therefore, post-quantum cryptography has attracted increasing attention, and the National Institute of Standards and Technology launched the Post-Quantum Cryptography Standardization. Code-based schemes are considered to be post-quantum secure and play an important role in the aforementioned standardization process.

This dissertation studies code-based cryptography from three different perspectives. In the first part, the coding-theoretic problems *syndrome decoding in the sum-rank metric*, *syndrome decoding of high-order interleaved codes in the rank metric*, and *decoding Gabidulin codes beyond the unique decoding radius* are investigated. For all three problems, new algorithms are proposed and the complexities of the algorithms are analyzed. Furthermore, the algorithms are compared to existing strategies, and the impact of the findings on cryptography is discussed.

In the second part, two attacks on the code-based schemes *Twisted Reed–Solomon based McEliece* and *Hamming Quasi-Cyclic (HQC)* are developed. The attack on the former system is severe, as it exploits mathematical weaknesses in the cryptographic algorithms, and therefore demonstrates flaws in the security of the system. The attack on the latter scheme constitutes a power side-channel attack and targets an implementation proposed by the designers of the system. The described attack does *not* uncover any flaws in HQC but weaknesses in the proposed implementation.

In the third part, the new code-based encryption scheme LIGA is proposed. The scheme is based on the hardness of list decoding and interleaved decoding of Gabidulin codes, and it represents an improved variant of the broken *Faure–Loidreau* system. It is proven that the public-key encryption version is indistinguishable under chosen-plaintext attacks, and the key-encapsulation mechanism version achieves indistinguishability under adaptive chosen-ciphertext attacks, both under the assumption that the underlying problems are hard. The scheme is *not* based on hiding the structure of a code. It features short ciphertext sizes, small key sizes, and no decryption failures.

Contents

1

Introduction

Cryptographic techniques refer to methods for secure communication in the presence of adversaries and have been deployed for more than 3000 years [1]. Before the 1970s, they were largely an art, where constructing and breaking ciphers relied on creativity and experience [2]. Furthermore, the main users of cryptography were military organizations and governments, who mostly applied private-key encryption schemes, in which the same secret cryptographic key is used by both the sender and the receiver. In the early 1970s, the field of cryptography started to evolve from an art to a multidisciplinary research topic bringing together mathematics, computer science, electrical engineering, and physics. A strong theory has been developed that has allowed the thorough study of cryptography as a science and has influenced the mindset of scientists about the broad field of computer security [2]. Furthermore, in 1976, Diffie and Hellman published the first paper on public-key (or asymmetric) cryptography [3], which constituted a breakthrough in cryptography, as it guaranteed security objectives, such as data confidentiality, data integrity, authentication, and non-repudiation, for communication between parties who do not share a secret key [4]. These powerful tools are used by virtually all of us on a regular basis, e.g., every time we authenticate ourselves by a password, visit a https-based website, or conduct an e-commerce transaction. Scenarios in the future indicate that these methods will become even more important, as an ever increasing number of applications that are critical to our wellbeing (like cooperative driving vehicles) will rely on secure communication.

Today, the security of virtually all asymmetrically encrypted communication relies on the difficulty of computing discrete logarithms and factoring large integers, e.g., the Transport Layer Security (TLS) protocol version 1.3 [5]. In 1994, however, Peter Shor developed two quantum algorithms that solve the aforementioned problems efficiently [6], and it was shown that currently applied asymmetric schemes can be broken by a quantum computer with 20 million qubits within a few hours [7] . Although state-of-the-art quantum computers do not have even 100 qubits yet [8], companies aim to build 1000-plus qubit quantum computers by 2023 [9], and some researchers predict that within the next twenty years quantum computers will be powerful enough to break essentially all asymmetric schemes currently in use [10].

Due to this rapid development in quantum computing, research on post-quantum cryptography is already necessary now, as time is needed to improve the efficiency of post-quantum schemes, to build confidence in them, and to improve their usability [11]. Therefore, in December 2016, the National Institute of Standards and Technology (NIST) launched the post-quantum cryptography standardization process to standardize public-key cryptographic algorithms that are quantum-resistant [10]. By the initial deadline at the end of 2017, 69 encryption and signature schemes were accepted to the first round of the standardization process, and 7 of these schemes have advanced to the third round that is currently being evaluated.

Code-based cryptosystems refer to schemes whose security relies on the hardness of problems in coding theory. In these schemes, the one-way function is often defined as adding an error to a codeword of an error-correcting code or computing a syndrome of an error using a fixed parity-check matrix of an error-correcting code [12]. These systems have gained a lot of attention, as it is believed that they can resist quantum computer attacks, and for this reason, they play an important role in the standardization process of NIST. The study of this family of schemes was initiated by the seminal paper of McEliece in the late 1970s [13]. Therein, McEliece proposed a public-key encryption scheme which uses the algebraic structure of a random binary irreducible Goppa code as private key and a scrambled generator matrix[1] of that Goppa code as public key. The ciphertext is defined as the sum of a codeword of the scrambled code and an error vector of small Hamming weight, where the error can only be efficiently removed if the private key (i.e., the algebraic structure of the Goppa code) is known. To this day, the rationale behind McEliece's proposal still has a significant impact on the design of code-based cryptosystems. For instance, two out of the three code-based

[1] McEliece proposed to scramble the generator matrix by multiplying a random full rank matrix from the left and a random permutation matrix from the right.

encryption proposals in the third round of the NIST standardization process evolved from McEliece's scheme.

The main drawback of McEliece's original scheme is the relatively large size of its public key. To overcome this issue, researchers proposed to replace Goppa codes by other algebraic codes, but most of them have been broken by structural attacks. For instance, in 1986, Niederreiter suggested the use of Generalized Reed–Solomon (GRS) codes [14], but Sidelnikov and Shestakov devised an efficient attack to generate an alternative private key given the public key [15]. Further proposed variants of McEliece schemes based on algebraic codes and efficient attacks on them are shown in [16–32]. Another approach to reduce the key size is to use errors of certain rank weight instead of certain Hamming weight, which was first proposed by Gabidulin, Paramonov, and Tretjakov in [33]. In this paper, the authors additionally replaced Goppa codes by Gabidulin codes, but the system had to be modified multiple times [34–40] due to structural attacks by Gibson [41, 42], Overbeck [43–45], and variants thereof [46–48]. To the best of our knowledge, the schemes by Berger *et al.* [49] and Loidreau [50, 51] are the only variants that have not been broken yet. Since many of the proposals based on algebraic codes have been broken by structural attacks, researchers have been investigating McEliece schemes based on codes with a very weak algebraic structure such as Moderate-Density Parity-Check Codes (MDPC) [52–54] and Low-Rank Parity-Check Codes (LRPC) [55–59]. This direction of research seems to be very promising, as the systems enable small key and ciphertext sizes, and in contrast to most McEliece schemes based on algebraic codes, there are strong arguments why probably no efficient key-recovery attack exists. Beside the idea of using codes with a weak algebraic structure, code-based schemes were proposed that do *not* rely on hiding the structure of an error-correcting code, e.g., [60–63]. While the system in [60] is not practical, and the schemes [61, 62] were broken [64, 65], the system introduced by Aguilar-Melchor *et al.* in [63] seems to be promising. The rank version of [63] reached the second round and the Hamming version of [63] even advanced to the third round of the NIST standardization process.

Outline

In this dissertation, we investigate general coding-theoretic problems that have applications in cryptography, we develop attacks on existing code-based encryption systems, and we propose a new code-based cryptographic scheme.

In **Chapter 2**, we introduce the notation that we use throughout this dissertation and review some definitions and properties of linear codes. We further define the Hamming metric, the rank metric, and the sum-rank metric, and we state the formal definitions of the well-known problems, syndrome decoding in the Hamming metric and syndrome decoding in the rank metric. We conclude this chapter by recalling the definitions of some complexity classes and some cryptographic principles.

In the first part of **Chapter 3**, we consider the problem of *syndrome decoding in the sum-rank metric*. We first derive statements about erasure-decoding in the sum-rank metric, and based on these results, we propose a non-trivial generic decoding algorithm. Our algorithm is compared to other decoding strategies and a hardness reduction to this problem is shown. In the second part, we investigate the problem of *syndrome decoding of high-order interleaved codes in the rank-metric*. We propose a new algorithm, prove conditions under which it is guaranteed to solve the problem, and compare it to other known decoding strategies. In the third part of this chapter, we develop a new strategy for *decoding Gabidulin codes beyond the unique decoding radius*. We compare this decoding strategy to other algorithms and discuss possible modifications to it. This chapter ends with a short summary and open problems related to the aforementioned problems.

Chapter 4 is devoted to attacks on two encryption schemes in the Hamming metric. The first scheme which we investigate is a variant of McEliece's system, which uses Twisted Reed–Solomon (TRS) codes instead of binary Goppa codes. After recalling the formal definition of the system, we derive a feasible key-recovery attack on it. Furthermore, we provide a detailed complexity analysis of the attack and show the average runtime of an implementation on a general purpose processor. In the second part of this chapter, we consider the system Hamming Quasi-Cyclic (HQC). In order to do that, we recall the definition of the system and review its security assumptions. We then propose a power-based side-channel chosen-ciphertext attack on the IND-CCA2-secure[2] Key-Encapsulation Mechanism (KEM) version of HQC. Furthermore, we give a detailed analysis about the success probability and the runtime of the attack. This chapter ends with remarks on both systems and open problems.

In **Chapter 5**, we propose LIGA, a code-based rank-metric encryption scheme that is based on the difficulty of list decoding and interleaved decoding of Gabidulin codes. The system constitutes a modification of the Faure–Loidreau (FL) system, which was

[2] A system is IND-CCA2 secure if it is indistinguishable under adaptive chosen-ciphertext attacks (Definition 2.33).

broken in a key-recovery attack by Gaborit, Otmani, and Talé Kalachi. We show that the public-key encryption variant of LIGA is IND-CPA secure[3] in the standard model, and the KEM variant is IND-CCA2 secure in the random oracle model, both under hardness assumptions of formally defined problems related to list decoding and interleaved decoding of Gabidulin codes. We further examine several exponential-time attacks on the aforementioned problems, state their complexity, and compare the resulting parameters to some NIST proposals. We observe that LIGA has small ciphertext and key sizes, it guarantees no decryption failures, and its security does *not* rely on hiding the structure of a code. This chapter ends with a summary and a description of a new attack by Bombar and Couvreur.

A summary of the presented results as well as an outlook is given in **Chapter 6**.

We present additional remarks in **Appendix A** to **C** and a summary of the notation and the abbreviations in **Appendix D**.

Note that Chapters 3, 4, and 5 are self contained, and the notation and the variables defined in those chapters are only valid within the scope of the respective chapter. This was done to minimize the total number of variables used in the thesis. The reader is warned not to take definitions across those chapters. Only the definitions from Chapter 2 are valid throughout the whole thesis.

[3] A system is IND-CPA secure if it indistinguishable under chosen-plaintext attacks (Definition 2.30).

2

Preliminaries

In this chapter, we introduce the notation and the concepts used throughout this dissertation. A summary of the notation and the abbreviations is given in Appendix D.

2.1 Notation

Let q be a power of a prime, and let m and u be positive integers. Then, $\mathbb{F}_q$ denotes the field of size q, $\mathbb{F}_{q^m}$ refers to the extension field of $\mathbb{F}_q$ of order q^m, and $\mathbb{F}_{q^{mu}}$ is the extension field of extension degree u of $\mathbb{F}_{q^m}$. Note that $\mathbb{F}_q \subseteq \mathbb{F}_{q^m} \subseteq \mathbb{F}_{q^{mu}}$, and $\mathbb{F}_{q^{mu}}$ is a u-dimensional vector space over $\mathbb{F}_{q^m}$ and a mu-dimensional vector space over $\mathbb{F}_q$. The respective multiplicative groups are indicated by $\mathbb{F}_q^* := \mathbb{F}_q \setminus \{0\}$, $\mathbb{F}_{q^m}^* := \mathbb{F}_{q^m} \setminus \{0\}$, and $\mathbb{F}_{q^{mu}}^* := \mathbb{F}_{q^{mu}} \setminus \{0\}$.

We denote the set of all $m \times n$ matrices over $\mathbb{F}_q$ by $\mathbb{F}_q^{m \times n}$ and the set of all length-n row vectors over $\mathbb{F}_{q^m}$ by $\mathbb{F}_{q^m}^n = \mathbb{F}_{q^m}^{1 \times n}$. Define the set of integers $[a:b] := \{i \in \mathbb{Z} : a \leq i \leq b\}$. For $i \in [1:m]$ and $j \in [1:n]$, we denote the element in the i-th row and the j-th column of the matrix $\boldsymbol{A} \in \mathbb{F}_q^{m \times n}$ by $A_{i,j}$ and use the submatrix notation

$$\boldsymbol{A}_{[a:b],[c:d]} := \begin{bmatrix} A_{a,c} & \dots & A_{a,d} \\ \vdots & \ddots & \vdots \\ A_{b,c} & \dots & A_{b,d} \end{bmatrix}.$$

The matrix $\boldsymbol{A}$ restricted to the rows indexed by $[a:b]$ is written as $\boldsymbol{A}_{[a:b],:} = \boldsymbol{A}_{[a:b],[1:n]}$, and the matrix $\boldsymbol{A}$ restricted to the columns indexed by $[c:d]$ is written as $\boldsymbol{A}_{:,[c:d]} = \boldsymbol{A}_{[1:m],[c:d]}$. By $\mathrm{rk}_q(\boldsymbol{A})$ and $\mathrm{rk}_{q^m}(\boldsymbol{B})$, we denote the rank of the matrix $\boldsymbol{A} \in \mathbb{F}_q^{m \times n}$ over $\mathbb{F}_q$ and the rank of the matrix $\boldsymbol{B} \in \mathbb{F}_{q^m}^{u \times n}$ over $\mathbb{F}_{q^m}$, respectively. The transpose of $\boldsymbol{A}$ is indicated by $\boldsymbol{A}^\top$, $\boldsymbol{A}^\perp$ denotes a matrix whose rows form a basis of the right kernel of $\boldsymbol{A}$, and $\mathrm{ref}(\boldsymbol{A})$ refers to the reduced row echelon form of $\boldsymbol{A}$. Furthermore, the number of $m \times n$ matrices over $\mathbb{F}_q$ of rank i (e.g., see [66]) is denoted by

$$\mathrm{NM}_q(m, n, i) = \prod_{j=0}^{i-1} \tfrac{(q^m - q^j)(q^n - q^j)}{q^i - q^j} \leq 4q^{i(m+n)-i^2}. \tag{2.1}$$

Let $\boldsymbol{a}$ be a vector in $\mathbb{F}_{q^m}^n$ and denote the j-th entry of $\boldsymbol{a}$ by a_j, where $j \in [1:n]$. We write the concatenation of $\boldsymbol{a} = [a_1, \dots, a_n\ ]$ and the vector $\boldsymbol{d} = [d_1, \dots, d_{n'}] \in \mathbb{F}_{q^m}^{n'}$ as $[\boldsymbol{a}, \boldsymbol{d}] := [a_1, \dots, a_n, d_1, \dots, d_{n'}] \in \mathbb{F}_{q^m}^{n+n'}$.

For $n = n'$, we define the product of $\boldsymbol{a}$ and $\boldsymbol{d}$ by

$$\boldsymbol{a}\boldsymbol{d} := \boldsymbol{a}\,\mathrm{rot}(\boldsymbol{d})^\top = \boldsymbol{d}\,\mathrm{rot}(\boldsymbol{a})^\top = \boldsymbol{d}\boldsymbol{a},$$

where the circulant matrix

$$\mathrm{rot}(\boldsymbol{a}) := \begin{bmatrix} a_1 & a_n & \dots & a_2 \\ a_2 & a_1 & \dots & a_3 \\ \vdots & \vdots & \ddots & \vdots \\ a_n & a_{n-1} & \dots & a_1 \end{bmatrix} \in \mathbb{F}_{q^m}^{n \times n}.$$

The componentwise product of the vectors $\boldsymbol{a}$ and $\boldsymbol{d}$ is equal to

$$\boldsymbol{a} \star \boldsymbol{d} := [a_1 d_1, \dots, a_n d_n] \in \mathbb{F}_{q^m}^n.$$

The Moore matrix for a vector $\boldsymbol{a} \in \mathbb{F}_{q^m}^n$ is defined by

$$\mathcal{M}_{s,q}(\boldsymbol{a}) := \begin{bmatrix} a_1 & a_2 & \dots & a_n \\ a_1^q & a_2^q & \dots & a_n^q \\ \vdots & \vdots & \ddots & \vdots \\ a_1^{q^{s-1}} & a_2^{q^{s-1}} & \dots & a_n^{q^{s-1}} \end{bmatrix} \in \mathbb{F}_{q^m}^{s \times n}.$$

If $a_1, \dots, a_n \in \mathbb{F}_{q^m}$ are linearly independent over $\mathbb{F}_q$, then $\mathrm{rk}_{q^m}(\mathcal{M}_{s,q}(\boldsymbol{a})) = \min\{s, n\}$, e.g., see [67].

The definition of a Moore matrix can be extended to matrices by

$$\mathcal{M}_{s,q}\left(\boldsymbol{B}\right) := \begin{bmatrix} B_{1,1} & B_{1,2} & \dots & B_{1,n} \\ B_{2,1} & B_{2,2} & \dots & B_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ B_{u,1} & B_{u,2} & \dots & B_{u,n} \\ B_{1,1}^{q} & B_{1,2}^{q} & \dots & B_{1,n}^{q} \\ B_{2,1}^{q} & B_{2,2}^{q} & \dots & B_{2,n}^{q} \\ \vdots & \vdots & \ddots & \vdots \\ B_{u,1}^{q^{s-1}} & B_{u,2}^{q^{s-1}} & \dots & B_{u,n}^{q^{s-1}} \end{bmatrix} \in \mathbb{F}_{q^m}^{us\times n},$$

where $\boldsymbol{B} \in \mathbb{F}_{q^m}^{u\times n}$.

The diagonal matrix with the entries of $\boldsymbol{a} \in \mathbb{F}_{q^m}^{n}$ on the main diagonal is denoted by $\mathrm{diag}(\boldsymbol{a}) \in \mathbb{F}_{q^m}^{n\times n}$.

For a fixed basis $\boldsymbol{\gamma} = [\gamma_1, \gamma_2, \dots, \gamma_m] \in \mathbb{F}_{q^m}^{m}$ of $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$, we define the mapping

$$\begin{aligned} \mathrm{ext}_{q^m/q} \, : \, \mathbb{F}_{q^m}^{n} &\to \mathbb{F}_{q}^{m\times n}, \\ \boldsymbol{a} = [a_1, a_2, \dots, a_n] &\mapsto \boldsymbol{A} = \begin{bmatrix} A_{1,1} & A_{1,2} & \dots & A_{1,n} \\ \vdots & \vdots & \ddots & \vdots \\ A_{m,1} & A_{m,2} & \dots & A_{m,n} \end{bmatrix}, \end{aligned}$$

where $a_j = \sum_{i=1}^{m} A_{i,j}\gamma_i$, for $j \in [1:n]$. Similarly, for a fixed basis of $\mathbb{F}_{q^{mu}}$ over $\mathbb{F}_{q^m}$, we can relate each vector $\boldsymbol{b} \in \mathbb{F}_{q^{mu}}^{n}$ to a matrix $\boldsymbol{B} \in \mathbb{F}_{q^m}^{u\times n}$ according to $\mathrm{ext}_{q^{mu}/q^m} : \mathbb{F}_{q^{mu}}^{n} \to \mathbb{F}_{q^m}^{u\times n}$, $\boldsymbol{b} \mapsto \boldsymbol{B}$, where the j-th column of $\boldsymbol{B}$ is the expansion of b_j in the basis of $\mathbb{F}_{q^{mu}}$ over $\mathbb{F}_{q^m}$. We apply the definition of $\mathrm{ext}_{q^m/q}$ and $\mathrm{ext}_{q^{mu}/q^m}$ also to matrices by extending each row and then vertically concatenating the resulting matrices.

We define the trace of a vector $\boldsymbol{b} \in \mathbb{F}_{q^{mu}}^{n}$ to $\mathbb{F}_{q^m}^{n}$ by

$$\begin{aligned} \mathrm{Tr} : \mathbb{F}_{q^{mu}}^{n} &\to \mathbb{F}_{q^m}^{n} \\ \boldsymbol{b} = [b_1, b_2, \dots, b_n] &\mapsto \left[\sum_{i=0}^{u-1} b_1^{q^{mi}}, \sum_{i=0}^{u-1} b_2^{q^{mi}}, \dots, \sum_{i=0}^{u-1} b_n^{q^{mi}} \right]. \end{aligned}$$

Let $[\delta_1, \delta_2, \dots, \delta_u] \in \mathbb{F}_{q^{mu}}^{u}$ be a basis of $\mathbb{F}_{q^{mu}}$ over $\mathbb{F}_{q^m}$. We call $[\delta_1^*, \delta_2^*, \dots, \delta_u^*] \in \mathbb{F}_{q^{mu}}^{u}$ a dual basis to $[\delta_1, \delta_2, \dots, \delta_u]$ if it fulfills

$$\mathrm{Tr}(\delta_i \delta_j^*) = \begin{cases} 1, & \text{if } i = j, \\ 0, & \text{else,} \end{cases}$$

for $i, j \in [1:u]$. Note that a dual basis always exists.

Let $\boldsymbol{a}_1, \dots, \boldsymbol{a}_\ell \in \mathbb{F}_{q^m}^n$. Then, the $\mathbb{F}_q$-linear vector space spanned by $\boldsymbol{a}_1, \dots, \boldsymbol{a}_\ell$ is denoted by

$$\langle \boldsymbol{a}_1, \dots, \boldsymbol{a}_\ell \rangle_q := \left\{ \sum_{i=1}^{\ell} v_i \boldsymbol{a}_i : v_i \in \mathbb{F}_q \right\}.$$

Similarly, the $\mathbb{F}_q$-linear vector space generated by the elements of a set $\mathcal{S}$ is denoted by $\langle \mathcal{S} \rangle_q$. The $\mathbb{F}_q$-dimension of a space $\mathcal{V}$ is written as $\dim_q(\mathcal{V})$, the dual space of $\mathcal{V}$ as $\mathcal{V}^\perp$, and the Grassmannian $\mathrm{Gr}_q(\mathcal{V}, k)$ of $\mathcal{V}$ is the set of all k-dimensional $\mathbb{F}_q$-linear subspaces of $\mathcal{V}$. The cardinality of $\mathrm{Gr}_q(\mathcal{V}, k)$ is equal to the Gaussian binomial coefficient

$$\begin{bmatrix} j \\ k \end{bmatrix}_q := \begin{cases} \dfrac{(1-q^j)(1-q^{j-1}) \cdots (1-q^{j-k+1})}{(1-q)(1-q^2) \cdots (1-q^k)}, & \text{for } k \leq j, \\ 0, & \text{for } k > j, \end{cases}$$

where $j = \dim_q(\mathcal{V})$. For $k \leq j$, this quantity can be lower and upper bounded [68, Lem. 4] by

$$q^{k(j-k)} \leq \begin{bmatrix} j \\ k \end{bmatrix}_q \leq 4q^{k(j-k)}. \tag{2.2}$$

The $\mathbb{F}_q$-linear row space of a matrix $\boldsymbol{A} \in \mathbb{F}_q^{m \times n}$ is denoted by $\mathcal{R}_q(\boldsymbol{A})$, and the right $\mathbb{F}_q$-kernel is defined by $\mathcal{K}_q(\boldsymbol{A}) := \{\boldsymbol{v} \in \mathbb{F}_q^n : \boldsymbol{A}\boldsymbol{v}^\top = \boldsymbol{0}\}$. For $\boldsymbol{a} \in \mathbb{F}_{q^m}^n$, we define the $\mathbb{F}_q$-rank of $\boldsymbol{a}$ by $\mathrm{rk}_q(\boldsymbol{a}) := \dim_q(\langle a_1, \dots, a_n \rangle_q) = \mathrm{rk}_q(\mathrm{ext}_{q^m/q}(\boldsymbol{a}))$. Using the definition of $\mathrm{ext}_{q^m/q}$, we define the right $\mathbb{F}_q$-kernel of $\boldsymbol{B} \in \mathbb{F}_{q^m}^{k \times n}$ by $\mathcal{K}_q(\boldsymbol{B}) := \{\boldsymbol{v} \in \mathbb{F}_q^n : \mathrm{ext}_{q^m/q}(\boldsymbol{B})\boldsymbol{v}^\top = \boldsymbol{0}\}$, and the $\mathbb{F}_q$-rank of $\boldsymbol{B}$ by $\mathrm{rk}_q(\boldsymbol{B}) := \mathrm{rk}_q(\mathrm{ext}_{q^m/q}(\boldsymbol{B}))$.

The sets of univariate polynomials over $\mathbb{F}_q$ and $\mathbb{F}_{q^m}$ are denoted by $\mathbb{F}_q[X]$ and $\mathbb{F}_{q^m}[X]$, respectively. For the degree of a polynomial $f \in \mathbb{F}_q[X]$, we write $\deg(f)$. For a vector $\boldsymbol{\alpha} = [\alpha_1, \dots, \alpha_n] \in \mathbb{F}_q^n$, we define the evaluation map by

$$\begin{aligned} \mathrm{ev}_{\boldsymbol{\alpha}} : \mathbb{F}_q[X] &\to \mathbb{F}_q^n \\ f &\mapsto [f(\alpha_1), f(\alpha_2), \dots, f(\alpha_n)]. \end{aligned}$$

Let $\mathcal{I}$ and $\mathcal{J}$ be two finite subsets of integers. We define their sumset by

$$\mathcal{I} + \mathcal{J} := \{a + b \, : \, a \in \mathcal{I}, \; b \in \mathcal{J}\}.$$

The probability of an event is denoted by Pr, and the expectation of a random variable is denoted by $\mathbb{E}$. Drawing an element i uniformly at random from a set $\mathcal{I}$ is written as $i \overset{\$}{\leftarrow} \mathcal{I}$, and assigning an element a to k is indicated by $k \leftarrow a$.

In order to classify the asymptotic running time of algorithms, we use the well-known big O notation, which we denote by O.

2.2 Linear Codes

In this section, we review some concepts of linear codes as well as the Hamming metric, the rank metric, and the sum-rank metric.

Definition 2.1 (Linear Code)**.** *An $[n, k]_{\mathbb{F}_{q^m}}$ code $\mathcal{C}$ is a k-dimensional $\mathbb{F}_{q^m}$-linear subspace of $\mathbb{F}_{q^m}^n$. A generator matrix $\boldsymbol{G} \in \mathbb{F}_{q^m}^{k \times n}$ of $\mathcal{C}$ is a matrix whose rows form a basis of $\mathcal{C}$.*

Definition 2.2 (Dual Code)**.** *Let $\mathcal{C}$ be an $[n, k]_{\mathbb{F}_{q^m}}$ code. Then, the dual code of $\mathcal{C}$ is defined by*

$$\mathcal{C}^{\perp} := \left\{ \boldsymbol{c}' \in \mathbb{F}_{q^m}^n : \sum_{i=1}^{n} c'_i c_i = 0, \forall \boldsymbol{c} \in \mathcal{C} \right\}.$$

A matrix $\boldsymbol{H} \in \mathbb{F}_{q^m}^{(n-k) \times n}$ is a parity-check matrix of $\mathcal{C}$ if and only if it is a generator matrix of $\mathcal{C}^{\perp}$.

Definition 2.3 (Interleaved Code)**.** *Let $\mathcal{C}$ be an $[n, k]_{\mathbb{F}_{q^m}}$ code and u be a positive integer. The corresponding u-interleaved $[u; n, k]_{\mathbb{F}_{q^m}}$ code is defined by*

$$\mathcal{C}^{(u)} := \left\{ \begin{bmatrix} \boldsymbol{c}_1 \\ \boldsymbol{c}_2 \\ \vdots \\ \boldsymbol{c}_u \end{bmatrix} \in \mathbb{F}_{q^m}^{u \times n} \, : \, \boldsymbol{c}_i \in \mathcal{C}, \forall i \in [1 \! : \! u] \right\}.$$

We call $\mathcal{C}$ the constituent code and u the interleaving order.

Note that any codeword $\boldsymbol{C} \in \mathbb{F}_{q^m}^{u \times n}$ of an interleaved code $\mathcal{C}^{(u)}$ can be written as $\boldsymbol{C} = \boldsymbol{M}\boldsymbol{G}$, where $\boldsymbol{M} \in \mathbb{F}_{q^m}^{u \times k}$ and $\boldsymbol{G} \in \mathbb{F}_{q^m}^{k \times n}$ is a generator matrix of the constituent code $\mathcal{C}$. Furthermore, it holds that $\boldsymbol{H}\boldsymbol{C}^\top = \boldsymbol{0} \in \mathbb{F}_{q^m}^{(n-k)\times u}$ for any codeword $\boldsymbol{C} \in \mathcal{C}^{(u)}$, where $\boldsymbol{H} \in \mathbb{F}_{q^m}^{(n-k)\times n}$ is a parity-check matrix of $\mathcal{C}$.

Definition 2.4 (Schur-Square of a Code)**.** *Let $\mathcal{C}$ be an $[n,k]_{\mathbb{F}_{q^m}}$ code. The Schur-square (or Hadamard-square) of $\mathcal{C}$ is given by*

$$\mathcal{C}^{(\star 2)} \coloneqq \langle \{\boldsymbol{a} \star \boldsymbol{b} \in \mathbb{F}_{q^m}^n : \boldsymbol{a}, \boldsymbol{b} \in \mathcal{C}\}\rangle_{q^m}.$$

2.2.1 Hamming Metric

The Hamming metric was introduced by Richard Hamming in 1950 [69]. Codes in this metric are sets of vectors over a finite field and the distance of two vectors is given by the number of positions at which the corresponding entries differ. In the following, we review definitions related to the Hamming metric.

Definition 2.5 (Hamming Support)**.** *Let $\boldsymbol{c}$ be a vector in $\mathbb{F}_q^n$. Then, the Hamming support of $\boldsymbol{c}$ is defined by*

$$\mathrm{supp}_{\mathrm{H}}(\boldsymbol{c}) \coloneqq \{i \in [1:n] : c_i \neq 0\}.$$

Definition 2.6 (Hamming Weight)**.** *Let $\boldsymbol{c}$ be a vector in $\mathbb{F}_q^n$. Then, the Hamming weight of $\boldsymbol{c}$ is denoted by*

$$\mathrm{wt}_{\mathrm{H}}(\boldsymbol{c}) \coloneqq |\mathrm{supp}_{\mathrm{H}}(\boldsymbol{c})|.$$

Definition 2.7 (Hamming Distance)**.** *Let $\boldsymbol{c}$ and $\boldsymbol{d}$ be vectors in $\mathbb{F}_q^n$. Then, the Hamming distance between $\boldsymbol{c}$ and $\boldsymbol{d}$ is equal to*

$$\mathrm{d}_{\mathrm{H}}(\boldsymbol{c}, \boldsymbol{d}) \coloneqq |\{i \in [1:n] : c_i \neq d_i\}|.$$

Note that $\mathrm{d}_{\mathrm{H}}(\boldsymbol{c}, \boldsymbol{d}) = \mathrm{wt}_{\mathrm{H}}(\boldsymbol{c} - \boldsymbol{d})$ holds.

Definition 2.8 (Minimum Hamming Distance)**.** *Let $\mathcal{C}$ be an $[n,k]_{\mathbb{F}_q}$ code. The minimum Hamming distance of $\mathcal{C}$ is given by*

$$d_{\min} = \min\{\mathrm{d}_{\mathrm{H}}(\boldsymbol{c}, \boldsymbol{d}) : \boldsymbol{c}, \boldsymbol{d} \in \mathcal{C}, \boldsymbol{c} \neq \boldsymbol{d}\},$$

and we call $\mathcal{C}$ an $[n,k,d_{\min}]_{\mathbb{F}_q}^{\mathrm{H}}$ code. If it is clear from the context, the superscript H *will be omitted. Codes that attain the Singleton bound [70] with equality, i.e., their* *minimum distance $d_{\min}$ is equal to $n-k+1$, are called Maximum Distance Separable (MDS) codes.*

Definition 2.9 (Hamming Super-Support)**.** *Let $\boldsymbol{c}$ be a vector in $\mathbb{F}_q^n$. A set $\mathcal{A} \supseteq [1:n]$ is called a Hamming super-support of $\boldsymbol{c}$ if*

$$\mathcal{A} \supseteq \mathrm{supp}_{\mathrm{H}}(\boldsymbol{c}).$$

Reed–Solomon (RS) codes are one of the best-known codes, and they are extensively used in practice to correct errors in the Hamming metric.

Definition 2.10 (Reed–Solomon Code [71])**.** *Let k and n be integers such that $1 \leq k \leq n$, and let the entries of the vector $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_n] \in \mathbb{F}_q^n$ be distinct. The RS code over $\mathbb{F}_q$ of length n, dimension k, and with locators $\boldsymbol{\alpha}$ is defined by*

$$\mathcal{RS}_k(\boldsymbol{\alpha}) \coloneqq \{\mathrm{ev}_{\boldsymbol{\alpha}}(f) : f \in \mathbb{F}_q[X], \deg(f) \leq k-1\} \subseteq \mathbb{F}_q^n.$$

Note that RS codes are MDS codes, and their decoding was studied in many publications, e.g., in [71–82].

In the next theorem, we show the implication of an algorithm that is able to determine valid locators of an RS code given any of its generator matrices.

Theorem 2.1 (Sidelnikov–Shestakov Attack [15])**.** *Let $\mathcal{RS}_k(\boldsymbol{\alpha})$ denote an RS code with code locators $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_n] \in \mathbb{F}_q^n$. Given any generator matrix of $\mathcal{RS}_k(\boldsymbol{\alpha})$, there is an algorithm which determines a vector $\boldsymbol{\alpha}' \in \mathbb{F}_q^n$ such that*

$$\mathcal{RS}_k(\boldsymbol{\alpha}) = \mathcal{RS}_k(\boldsymbol{\alpha}'),$$

where $\boldsymbol{\alpha}' = [a\alpha_1 + b, \ldots, a\alpha_n + b]$ with $a \in \mathbb{F}_q^*$ and $b \in \mathbb{F}_q$, in $O(n^4)$ operations in $\mathbb{F}_q$.

TRS codes were introduced in [83] and constitute a generalization of RS codes.

Definition 2.11 (Twisted Reed–Solomon Code [83])**.** *Let ℓ_{T}, k, and n be integers such that $\ell_{\mathrm{T}} \geq 1$ and $1 \leq k \leq n$, and let the entries of the vector $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_n] \in \mathbb{F}_q^n$ be distinct. For a vector $\boldsymbol{\tau} \in [1:n-k]^{\ell_{\mathrm{T}}}$ of distinct* twists, *a vector $\boldsymbol{\pi} \in [0:k-1]^{\ell_{\mathrm{T}}}$ of distinct increasing* hooks, *and a vector of field coefficients $\boldsymbol{\eta} \in (\mathbb{F}_q^*)^{\ell_{\mathrm{T}}}$, the set of $[\boldsymbol{\tau}, \boldsymbol{\pi}, \boldsymbol{\eta}]$-twisted polynomials is given by*

$$\mathcal{P}_k(\boldsymbol{\tau}, \boldsymbol{\pi}, \boldsymbol{\eta}) \coloneqq \left\{ \sum_{i=0}^{k-1} f_i X^i + \sum_{j=1}^{\ell_{\mathrm{T}}} \eta_j f_{\pi_j} X^{k-1+\tau_j} : f_i \in \mathbb{F}_q \right\} \subseteq \mathbb{F}_q[X].$$

Then, the $[\boldsymbol{\tau}, \boldsymbol{\pi}, \boldsymbol{\eta}]$-TRS code over $\mathbb{F}_q$ of length n, dimension k, and with locators $\boldsymbol{\alpha}$ is given by

$$\mathcal{TRS}_k(\boldsymbol{\alpha}, \boldsymbol{\tau}, \boldsymbol{\pi}, \boldsymbol{\eta}) \coloneqq \{\mathrm{ev}_{\boldsymbol{\alpha}}(f) \,:\, f \in \mathcal{P}_k(\boldsymbol{\tau}, \boldsymbol{\pi}, \boldsymbol{\eta})\}.$$

Equivalently, the code $\mathcal{TRS}_k(\boldsymbol{\alpha}, \boldsymbol{\tau}, \boldsymbol{\pi}, \boldsymbol{\eta})$ is defined by the generator matrix

$$\boldsymbol{G}_{\boldsymbol{\alpha},\boldsymbol{\tau},\boldsymbol{\pi},\boldsymbol{\eta}} \coloneqq \begin{bmatrix} 1 & 1 & \dots & 1 \\ \alpha_1^1 & \alpha_2^1 & \dots & \alpha_n^1 \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_1^{\pi_1-1} & \alpha_2^{\pi_1-1} & \dots & \alpha_n^{\pi_1-1} \\ \alpha_1^{\pi_1} + \eta_1\alpha_1^{k-1+\tau_1} & \alpha_2^{\pi_1} + \eta_1\alpha_2^{k-1+\tau_1} & \dots & \alpha_n^{\pi_1} + \eta_1\alpha_n^{k-1+\tau_1} \\ \alpha_1^{\pi_1+1} & \alpha_2^{\pi_1+1} & \dots & \alpha_n^{\pi_1+1} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_1^{\pi_{\ell_\mathrm{T}}-1} & \alpha_2^{\pi_{\ell_\mathrm{T}}-1} & \dots & \alpha_n^{\pi_{\ell_\mathrm{T}}-1} \\ \alpha_1^{\pi_{\ell_\mathrm{T}}} + \eta_{\ell_\mathrm{T}}\alpha_1^{k-1+\tau_{\ell_\mathrm{T}}} & \alpha_2^{\pi_{\ell_\mathrm{T}}} + \eta_{\ell_\mathrm{T}}\alpha_2^{k-1+\tau_{\ell_\mathrm{T}}} & \dots & \alpha_n^{\pi_{\ell_\mathrm{T}}} + \eta_{\ell_\mathrm{T}}\alpha_n^{k-1+\tau_{\ell_\mathrm{T}}} \\ \alpha_1^{\pi_{\ell_\mathrm{T}}+1} & \alpha_2^{\pi_{\ell_\mathrm{T}}+1} & \dots & \alpha_n^{\pi_{\ell_\mathrm{T}}+1} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_1^{k-1} & \alpha_2^{k-1} & \dots & \alpha_n^{k-1} \end{bmatrix}.$$

In [84], the authors derive a method to obtain a subfamily of TRS codes that are MDS.

Theorem 2.2 (Explicit MDS TRS Codes [84])**.** *Let q_0 be a power of a prime, and $1 = s_0 < \ldots < s_{\ell_\mathrm{T}}$ be non-negative integers such that $\mathbb{F}_{q_0^{s_0}} \subset \mathbb{F}_{q_0^{s_1}} \subset \ldots \subset \mathbb{F}_{q_0^{s_{\ell_\mathrm{T}}}} = \mathbb{F}_q$ is a chain of subfields. Let k and n be integers such that $k < n \leq q_0$, and let the entries of $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_n] \in \mathbb{F}_{q_0}^n$ denote distinct locators. Furthermore, let $\boldsymbol{\tau}$, $\boldsymbol{\pi}$, and $\boldsymbol{\eta}$ be chosen as in Definition 2.11, such that $\eta_i \in \mathbb{F}_{q_0^{s_i}} \setminus \mathbb{F}_{q_0^{s_i-1}}$, for $i \in [1 : \ell_\mathrm{T}]$. Then, the code $\mathcal{TRS}_{k,n}(\boldsymbol{\alpha}, \boldsymbol{\tau}, \boldsymbol{\pi}, \boldsymbol{\eta})$ is MDS.*

In [84], the authors further propose a decoding algorithm for the code construction given in Theorem 2.2. Since the complexity of this decoding strategy is in

$$O\Big(\big(q_0^{2^{\ell_\mathrm{T}}}\big)^{\ell_\mathrm{T}} n \log^2(n) \log\big(\log(n)\big)\Big),$$

it is only practical for a very small number of twists.

2.2.2 Rank Metric

Codes in the rank metric are sets of vectors over an extension field, whose elements can be interpreted as matrices over a subfield. The distance of two vectors is given by the rank of the difference of their matrix representation. The codes were independently introduced in [85–87], together with their most famous code class, Gabidulin codes. An overview of the known properties of rank-metric codes and their application in cryptography and coding theory is given in [88].

In the following, we review the definition and some concepts of the rank metric.

Definition 2.12 (Rank Weight)**.** *Let $\boldsymbol{c}$ be a vector in $\mathbb{F}_{q^m}^n$. Then, the rank weight of $\boldsymbol{c}$ is given by*

$$\mathrm{wt}_{\mathrm{R}}(\boldsymbol{c}) := \mathrm{rk}_q(\boldsymbol{c}).$$

Definition 2.13 (Rank Distance)**.** *Let $\boldsymbol{c}$ and $\boldsymbol{d}$ be vectors in $\mathbb{F}_{q^m}^n$. Then, the rank distance between $\boldsymbol{c}$ and $\boldsymbol{d}$ is defined by*

$$\mathrm{d}_{\mathrm{R}}(\boldsymbol{c}, \boldsymbol{d}) := \mathrm{wt}_{\mathrm{R}}(\boldsymbol{c} - \boldsymbol{d}).$$

Definition 2.14 (Minimum Rank Distance)**.** *Let $\mathcal{C}$ be an $[n,k]_{\mathbb{F}_{q^m}}$ code. The minimum rank distance of $\mathcal{C}$ is given by*

$$d_{\min} = \min\{\mathrm{d}_{\mathrm{R}}(\boldsymbol{c}, \boldsymbol{d}) : \boldsymbol{c}, \boldsymbol{d} \in \mathcal{C}, \boldsymbol{c} \neq \boldsymbol{d}\}.$$

The code $\mathcal{C}$ is an $[n,k,d_{\min}]_{\mathbb{F}_{q^m}}^{\mathrm{R}}$ code, and it is called a Maximum Rank Distance (MRD) code if it attains the rank-metric Singleton bound with equality, i.e., if $d_{\min} = n-k+1$.

The following lemma provides an important statement about the decomposition of vectors and matrices.

Lemma 2.3 (Matrix Decomposition [89, Theorem 1])**.** *Let $u \geq 1$ be a non-negative integer, and let $\boldsymbol{E} \in \mathbb{F}_{q^m}^{u \times n}$ be a matrix of $\mathbb{F}_q$-rank t. Then, the matrix can be decomposed into $\boldsymbol{E} = \boldsymbol{A}\boldsymbol{B}$, where both $\boldsymbol{A} \in \mathbb{F}_{q^m}^{u \times t}$ and $\boldsymbol{B} \in \mathbb{F}_q^{t \times n}$ have full $\mathbb{F}_q$-rank t. The matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ are unique up to elementary $\mathbb{F}_q$-column and $\mathbb{F}_q$-row operations, respectively.*

Using Lemma 2.3, we define the row and the column rank support as follows.

Definition 2.15 (Row and Column Rank Support)**.** *Let $u \geq 1$ be a non-negative integer, and let $\boldsymbol{E} \in \mathbb{F}_{q^m}^{u \times n}$ be a matrix of $\mathbb{F}_q$-rank t and decomposition $\boldsymbol{E} = \boldsymbol{A}\boldsymbol{B}$*

with $\boldsymbol{A} \in \mathbb{F}_{q^m}^{u \times t}$ *and* $\boldsymbol{B} \in \mathbb{F}_q^{t \times n}$. *Then, we call the* $\mathbb{F}_q$*-linear column space of* $\boldsymbol{A}$ *the column rank support* $\mathrm{supp}_{\mathrm{R}}^{(\mathrm{C})}(\boldsymbol{E})$ *and the* $\mathbb{F}_q$*-linear row space of* $\boldsymbol{B}$ *the row rank support* $\mathrm{supp}_{\mathrm{R}}^{(\mathrm{R})}(\boldsymbol{E})$.

Definition 2.16 (Row and Column Rank Super-Support)**.** *Let* $u \geq 1$ *be a non-negative integer, and let* $\boldsymbol{E} \in \mathbb{F}_{q^m}^{u \times n}$ *be a matrix of* $\mathbb{F}_q$*-rank* t *and decomposition* $\boldsymbol{E} = \boldsymbol{AB}$ *with* $\boldsymbol{A} \in \mathbb{F}_{q^m}^{u \times t}$ *and* $\boldsymbol{B} \in \mathbb{F}_q^{t \times n}$. *Then, we call the* $\mathbb{F}_q$*-linear space* $\mathcal{F}^{(\mathrm{R})}$ *a row rank super-support of* $\boldsymbol{E}$ *if* $\mathrm{supp}_{\mathrm{R}}^{(\mathrm{R})}(\boldsymbol{E}) \subseteq \mathcal{F}^{(\mathrm{R})}$, *and the* $\mathbb{F}_q$*-linear space* $\mathcal{F}^{(\mathrm{C})}$ *a column rank super-support of* $\boldsymbol{E}$ *if* $\mathrm{supp}_{\mathrm{R}}^{(\mathrm{C})}(\boldsymbol{E}) \subseteq \mathcal{F}^{(\mathrm{C})}$.

In the following, we define Gabidulin codes, which can be seen as the analogs of RS codes in the rank metric.

Definition 2.17 (Gabidulin Code [86])**.** *Let* k, n, *and* m *be integers such that* $1 \leq k \leq n \leq m$, *and let* $\boldsymbol{g} = [g_1, \ldots, g_n] \in \mathbb{F}_{q^m}^n$ *have* $\mathbb{F}_q$*-rank* n. *The* $\mathbb{F}_{q^m}$*-linear Gabidulin code* $\mathcal{G}_k(\boldsymbol{g})$ *over* $\mathbb{F}_{q^m}$ *of length* n, *dimension* k, *and with locators* $\boldsymbol{g}$ *is defined by its generator matrix*

$$\boldsymbol{G}_{\mathcal{G}} = \mathcal{M}_{k,q}(\boldsymbol{g}).$$

Gabidulin codes are MRD codes [86], and their decoding has been studied extensively, e.g., in [86, 87, 90–104].

The next lemma provides a statement about the error and erasure correction capability of Gabidulin codes.

Lemma 2.4 (Error-Erasure Decoding of Gabidulin Codes [99–104])**.** *Let* $\mathcal{G}_k(\boldsymbol{g})$ *denote an* $[n,k]_{\mathbb{F}_{q^m}}$ *Gabidulin code, and let* $\boldsymbol{r} = \boldsymbol{c} + \boldsymbol{e} \in \mathbb{F}_{q^m}^n$, *where* $\boldsymbol{c} \in \mathcal{G}_k(\boldsymbol{g})$,

$$\boldsymbol{e} = \boldsymbol{a}_{\mathrm{E}}\boldsymbol{B}_{\mathrm{E}} + \boldsymbol{a}_{\mathrm{C}}\boldsymbol{B}_{\mathrm{C}} + \boldsymbol{a}_{\mathrm{R}}\boldsymbol{B}_{\mathrm{R}}, \tag{2.3}$$

the terms $\boldsymbol{a}_{\mathrm{E}} \in \mathbb{F}_{q^m}^{t'}$, $\boldsymbol{B}_{\mathrm{E}} \in \mathbb{F}_q^{t' \times n}$, $\boldsymbol{a}_{\mathrm{C}} \in \mathbb{F}_{q^m}^{\gamma_{\mathrm{E}}}$, $\boldsymbol{B}_{\mathrm{C}} \in \mathbb{F}_q^{\gamma_{\mathrm{E}} \times n}$, $\boldsymbol{a}_{\mathrm{R}} \in \mathbb{F}_{q^m}^{\rho_{\mathrm{E}}}$, *and* $\boldsymbol{B}_{\mathrm{R}} \in \mathbb{F}_q^{\rho_{\mathrm{E}} \times n}$ *have full* $\mathbb{F}_q$*-rank, and* $\delta_{\mathrm{E}} := \rho_{\mathrm{E}} + \gamma_{\mathrm{E}}$. *If* $\boldsymbol{B}_{\mathrm{C}}$ *and* $\boldsymbol{a}_{\mathrm{R}}$ *are known to the decoder and if* $2t' + \delta_{\mathrm{E}} \leq n - k$ *is fulfilled, then there exist efficient algorithms that can decode* $\boldsymbol{r}$ *in* $\mathcal{G}_k(\boldsymbol{g})$ *uniquely.*

In this dissertation, we call the product $\boldsymbol{a}_{\mathrm{E}}\boldsymbol{B}_{\mathrm{E}}$ full rank errors, the term $\boldsymbol{a}_{\mathrm{C}}\boldsymbol{B}_{\mathrm{C}}$ column erasures, and the vector $\boldsymbol{a}_{\mathrm{R}}\boldsymbol{B}_{\mathrm{R}}$ row erasures.

Interleaved codes in the rank metric were introduced in [104, 105] and have found applications in code-based cryptography [62, 65, 106, 107], network coding [104, 108], and the construction and the decoding of space-time codes [109–114].

Definition 2.18 (Interleaved Gabidulin Codes [105])**.** *Let u, k, n, and m be positive integers such that $1 \leq k \leq n \leq m$. A linear (vertically, homogeneous) interleaved Gabidulin code $\mathcal{G}_k^{(u)}(\boldsymbol{g})$ over $\mathbb{F}_{q^m}$ of length n, dimension k, and interleaving order u is defined by*

$$\mathcal{G}_k^{(u)}(\boldsymbol{g}) := \left\{ \begin{bmatrix} \boldsymbol{c}_1 \\ \boldsymbol{c}_2 \\ \vdots \\ \boldsymbol{c}_u \end{bmatrix} \in \mathbb{F}_{q^m}^{u \times n} : \boldsymbol{c}_i \in \mathcal{G}_k(\boldsymbol{g}), \forall i \in [1:u] \right\}.$$

When considering random additive errors of rank weight t, interleaved Gabidulin codes can be decoded uniquely up to $t \leq \lfloor \frac{u}{u+1}(n-k) \rfloor$ errors with high probability [105, 115, 116]. Although the ratio of errors that can be successfully decoded is high, many error patterns exist for which all known efficient decoders fail. As shown in the next lemma, we can efficiently construct a large class of such errors.

Lemma 2.5 (Interleaved Decoding Failures [99, 105, 115])**.** *Let $\boldsymbol{G}_{\mathcal{G}} \in \mathbb{F}_{q^m}^{k \times n}$ be a generator matrix of $\mathcal{G}_k(\boldsymbol{g})$, and let $\boldsymbol{x}_i \in \mathbb{F}_{q^m}^{k}$ and $\boldsymbol{c}_i = \boldsymbol{x}_i \cdot \boldsymbol{G}_{\mathcal{G}}$, for $i \in [1:u]$. Then, the algorithms proposed in [99, 105, 115] fail to correct an additive error $\boldsymbol{Z} \in \mathbb{F}_{q^m}^{u \times n}$ of $\mathbb{F}_q$-rank t if*

$$\mathrm{rk}_{q^m} \left(\begin{bmatrix} \mathcal{M}_{n-t-1,q}\left(\boldsymbol{g}\right) \\ \mathcal{M}_{n-k-t,q}\left(\boldsymbol{c}_1 + \boldsymbol{z}_1\right) \\ \mathcal{M}_{n-k-t,q}\left(\boldsymbol{c}_2 + \boldsymbol{z}_2\right) \\ \vdots \\ \mathcal{M}_{n-k-t,q}\left(\boldsymbol{c}_u + \boldsymbol{z}_u\right) \end{bmatrix} \right) < n - 1,$$

where $\boldsymbol{z}_i$ is the i-th row of $\boldsymbol{Z}$ for all $i \in [1:u]$.

Decoding the error patterns shown in Lemma 2.5 has been subject to intensive research since the Loidreau–Overbeck decoder [105] was proposed in 2006. In the Hamming metric, the equivalent problem for interleaved RS codes has been studied since 1997 [78], and more than a dozen papers have dealt with decoding algorithms for these codes. However, no polynomial-time decoding algorithm for the case of Lemma 2.5 has been proposed so far, and it is widely conjectured that there cannot exist such a decoder.

2.2.3 Sum-Rank Metric

The sum-rank metric is a family of metrics which contains both the Hamming and the rank metric as special cases. It was introduced under the name “extended rank metric” as a suitable distance measure for multi-shot network coding in 2010 [117]. Since then, several code constructions and efficient decoders have been proposed for this metric [118–128]. These codes have been studied in the context of distributed storage [129], aspects of network coding [125], and space-time codes [130]. Furthermore, in [131], the authors derived several fundamental results on sum-rank-metric codes, including various bounds, MacWilliams identities, and new code constructions.

In the following, we review some common definitions for the sum-rank metric. Throughout this dissertation, we call $\ell_{\mathrm{SR}} \in \mathbb{N}$ the blocking parameter, where $\ell_{\mathrm{SR}} \mid n$. We refer to the integer $\eta_{\mathrm{SR}} := n/\ell_{\mathrm{SR}}$ as block size and define $\mu_{\mathrm{SR}} := \min\{\eta_{\mathrm{SR}}, m\}$.

Definition 2.19 (Sum-Rank Weight)**.** *Let $\boldsymbol{c}$ be a vector in $\mathbb{F}_{q^m}^n$. Then, the (ℓ_{SR}-)sum-rank weight of $\boldsymbol{c}$ is defined by*

$$\mathrm{wt}_{\mathrm{SR}}(\boldsymbol{c}) := \sum_{i=1}^{\ell_{\mathrm{SR}}} \mathrm{rk}_q(\boldsymbol{c}_i),$$

where $\boldsymbol{c} = [\boldsymbol{c}_1, \boldsymbol{c}_2, \dots, \boldsymbol{c}_{\ell_{\mathrm{SR}}}]$, and $\boldsymbol{c}_i \in \mathbb{F}_{q^m}^{\eta_{\mathrm{SR}}}$, for $i \in [1:\ell_{\mathrm{SR}}]$.

Note that for $\ell_{\mathrm{SR}} = 1$, the sum-rank metric coincides with the rank metric, and for $\ell_{\mathrm{SR}} = n$, the sum-rank metric is equal to the Hamming metric. Furthermore, for $\boldsymbol{c} \in \mathbb{F}_{q^m}^n$, it holds that $\mathrm{wt}_{\mathrm{R}}(\boldsymbol{c}) \leq \mathrm{wt}_{\mathrm{SR}}(\boldsymbol{c}) \leq \min\{\mu_{\mathrm{SR}}\ell_{\mathrm{SR}},\ \mathrm{wt}_{\mathrm{H}}(\boldsymbol{c})\}$.

Definition 2.20 (Sum-Rank Weight Decomposition)**.** *Let $\boldsymbol{c}$ be a vector in $\mathbb{F}_{q^m}^n$. Then, the (ℓ_{SR}-)sum-rank weight decomposition of $\boldsymbol{c}$ is given by the vector*

$$[\mathrm{rk}_q(\boldsymbol{c}_1), \dots, \mathrm{rk}_q(\boldsymbol{c}_{\ell_{\mathrm{SR}}})] \in [0:\mu_{\mathrm{SR}}]^{\ell_{\mathrm{SR}}},$$

where $\boldsymbol{c} = [\boldsymbol{c}_1, \boldsymbol{c}_2, \dots, \boldsymbol{c}_{\ell_{\mathrm{SR}}}]$, and $\boldsymbol{c}_i \in \mathbb{F}_{q^m}^{\eta_{\mathrm{SR}}}$, for $i \in [1:\ell_{\mathrm{SR}}]$. We denote the set of weight decompositions of sum-rank weight t by $\mathcal{T}_{t,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}} := \left\{\boldsymbol{t} \in [0:\mu_{\mathrm{SR}}]^{\ell_{\mathrm{SR}}} \ : \ \sum_{i=1}^{\ell_{\mathrm{SR}}} t_i = t\right\}$.

Note that it was shown in [132] that

$$|\mathcal{T}_{t,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}}| = \sum_{i=0}^{\left\lfloor \frac{t}{\mu_{\mathrm{SR}}+1} \right\rfloor} (-1)^i \binom{\ell_{\mathrm{SR}}}{i} \binom{t+\ell_{\mathrm{SR}}-1-(\mu_{\mathrm{SR}}+1)i}{\ell_{\mathrm{SR}}-1} \leq \binom{\ell_{\mathrm{SR}}+t-1}{\ell_{\mathrm{SR}}-1}. \quad (2.4)$$

Definition 2.21 (Sum-Rank Distance)**.** *Let* $\boldsymbol{c}$ *and* $\boldsymbol{d}$ *be vectors in* $\mathbb{F}_{q^m}^n$*. Then, the* (ℓ_{SR}*-)sum-rank distance between* $\boldsymbol{c}$ *and* $\boldsymbol{d}$ *is defined by*

$$\mathrm{d}_{\mathrm{SR}}(\boldsymbol{c}, \boldsymbol{d}) \coloneqq \mathrm{wt}_{\mathrm{SR}}(\boldsymbol{c} - \boldsymbol{d}).$$

Definition 2.22 (Minimum Sum-Rank Distance)**.** *Let* $\mathcal{C}$ *be an* $[n, k]_{\mathbb{F}_{q^m}}$ *code. The minimum* (ℓ_{SR}*-)sum-rank distance of* $\mathcal{C}$ *is equal to*

$$d_{\min} = \min\{\mathrm{d}_{\mathrm{SR}}(\boldsymbol{c}, \boldsymbol{d}) : \boldsymbol{c}, \boldsymbol{d} \in \mathcal{C}, \boldsymbol{c} \neq \boldsymbol{d}\}.$$

We call the code $\mathcal{C}$ *an* $[n, k, d_{\min}]_{\mathbb{F}_{q^m}}^{\mathrm{SR}}$ *code.*

2.2.4 Well-Studied Problems

In this section, we state well-known problems in coding theory and review algorithms to solve them.

Problem 2.1 (Decisional Hamming Syndrome Decoding ($\mathsf{DecSD}_{\mathrm{H}}$) Problem)**.**
Given:
- *Parity-check matrix* $\boldsymbol{H} \in \mathbb{F}_{q^m}^{(n-k)\times n}$ *of an* $[n, k]_{\mathbb{F}_{q^m}}$ *code* $\mathcal{C}$
- *Non-negative integer* t
- *Vector* $\boldsymbol{s} \in \mathbb{F}_{q^m}^{n-k}$

Question: *Is there a vector* $\boldsymbol{e} \in \mathbb{F}_{q^m}^n$ *with* $\mathrm{wt}_{\mathrm{H}}(\boldsymbol{e}) \leq t$ *such that* $\boldsymbol{s} = \boldsymbol{e}\boldsymbol{H}^\top$*?*

In 1978, Berlekamp *et al.* proved that the $\mathsf{DecSD}_{\mathrm{H}}$ problem is NP-complete [133]. At this time, the algorithm proposed by Prange [134] was the most efficient way to solve this problem by finding a solution to the associated search problem:

Problem 2.2 (Search Hamming Syndrome Decoding ($\mathsf{SeaSD}_{\mathrm{H}}$) Problem)**.**
Given:
- *Parity-check matrix* $\boldsymbol{H} \in \mathbb{F}_{q^m}^{(n-k)\times n}$ *of an* $[n, k]_{\mathbb{F}_{q^m}}$ *code* $\mathcal{C}$
- *Non-negative integer* t
- *Vector* $\boldsymbol{s} = \boldsymbol{e}\boldsymbol{H}^\top \in \mathbb{F}_{q^m}^{n-k}$*, where* $\mathrm{wt}_{\mathrm{H}}(\boldsymbol{e}) = t$

Objective: *Search for an* $\boldsymbol{e}' \in \mathbb{F}_{q^m}^n$ *such that* $\mathrm{wt}_{\mathrm{H}}(\boldsymbol{e}') \leq t$ *and* $\boldsymbol{s} = \boldsymbol{e}'\boldsymbol{H}^\top$*.*

The idea of Prange's Information-Set Decoding (ISD) algorithm [134] is to guess an error-free set $\{I_1, \ldots, I_{n-s}\} \subset [1:n]$ which contains an information set, where $t \leq s \leq n-k$. The set $\{I_1, \ldots, I_{n-s}\}$ is error-free if $e_{I_1} = \ldots = e_{I_{n-s}} = 0$, and $\{I_1, \ldots, I_{n-s}\}$ contains an information set if the columns $\bar{I}_1, \ldots, \bar{I}_s$ of $\boldsymbol{H}$ are linearly independent, where $\{\bar{I}_1, \ldots, \bar{I}_s\} = [1:n] \setminus \{I_1, \ldots, I_{n-s}\}$. Given such a set, the $\mathsf{SeaSD}_{\mathrm{H}}$

problem can be solved by a simple matrix inversion. The probability that a random set $\{I_1, \dots, I_{n-s}\}$ is error-free and contains an information set is approximately $\binom{s}{t} / \binom{n}{t}$.[1] Checking whether $\{I_1, \dots, I_{n-s}\}$ is error-free and contains an information set requires $O(n^3 m^3)$ operations in $\mathbb{F}_q$. It follows that if there is one solution to $\mathsf{SeaSD}_\mathrm{H}$, the average complexity of Prange's algorithm for solving the $\mathsf{SeaSD}_\mathrm{H}$ Problem is approximately

$$W_{\mathrm{Prange}} = n^3 m^3 \frac{\binom{n}{t}}{\binom{s}{t}} \geq n^3 m^3 \frac{\binom{n}{t}}{\binom{n-k}{t}} \tag{2.5}$$

operations in $\mathbb{F}_q$, where W_{Prange} is minimized for $s = n - k$. In case there are multiple solutions to $\mathsf{SeaSD}_\mathrm{H}$, the complexity needs to be divided by the number of solutions, which is, on average, equal to

$$N_\mathrm{H} := \frac{\sum_{i=0}^{t} \binom{n}{i} (q^m - 1)^i}{q^{m(n-k)}}.$$

From 1978 until now, several methods were proposed to accelerate the determination of an error-free information set, see [135] for a comprehensive list.

In 2016, Gaborit and Zémor probabilistically reduced the $\mathsf{DecSD}_\mathrm{H}$ problem to its rank-metric equivalent [136], which is defined as follows:

Problem 2.3 (Decisional Rank Syndrome Decoding ($\mathsf{DecSD}_\mathrm{R}$) Problem).

Given:
- *Parity-check matrix* $\boldsymbol{H} \in \mathbb{F}_{q^m}^{(n-k)\times n}$ *of an* $[n,k]_{\mathbb{F}_{q^m}}$ *code* $\mathcal{C}$
- *Non-negative integer* t
- *Vector* $\boldsymbol{s} \in \mathbb{F}_{q^m}^{n-k}$

Question: *Is there a vector* $\boldsymbol{e} \in \mathbb{F}_{q^m}^n$ *with* $\mathrm{wt}_\mathrm{R}(\boldsymbol{e}) \leq t$ *such that* $\boldsymbol{s} = \boldsymbol{e}\boldsymbol{H}^\top$*?*

The first algorithm to solve the $\mathsf{DecSD}_\mathrm{R}$ problem was proposed in 1996 [137], and since then, several improvements have been developed [138–142]. These algorithms solve the problem by finding a solution to the search variant of the problem:

Problem 2.4 (Search Rank Syndrome Decoding ($\mathsf{SeaSD}_\mathrm{R}$) Problem).

Given:
- *Parity-check matrix* $\boldsymbol{H} \in \mathbb{F}_{q^m}^{(n-k)\times n}$ *of an* $[n,k]_{\mathbb{F}_{q^m}}$ *code* $\mathcal{C}$
- *Non-negative integer* t
- *Vector* $\boldsymbol{s} = \boldsymbol{e}\boldsymbol{H}^\top \in \mathbb{F}_{q^m}^{n-k}$*, where* $\mathrm{wt}_\mathrm{R}(\boldsymbol{e}) = t$

Objective: *Search for an* $\boldsymbol{e}' \in \mathbb{F}_{q^m}^n$ *such that* $\mathrm{wt}_\mathrm{R}(\boldsymbol{e}') \leq t$ *and* $\boldsymbol{s} = \boldsymbol{e}'\boldsymbol{H}^\top$.

[1] For practical parameters, the probability that a random set $\{I_1, \dots, I_{n-s}\}$ contains an information set is much larger than the probability that this set is error free. Therefore, we neglect the former probability in our analysis.

The mentioned algorithms can be classified into two families. The algorithms belonging to the first family are known as *combinatorial* rank syndrome decoders, and they can be seen as analogs of ISD algorithms in the rank metric. In [139], Gaborit, Ruatta, and Schrek proposed a method that repeatedly samples an s-dimensional $\mathbb{F}_q$-linear subspace of either $\mathbb{F}_{q^m}$ or $\mathbb{F}_q^n$ until a subspace is found that contains the column or the row rank support of the error,[2] where $t \leq s \leq \min\{n-k, \frac{m}{n}(n-k)\}$. If such a subspace is determined, the $\mathsf{SeaSD}_{\mathrm{R}}$ problem can be solved by a matrix inversion, which requires $O(n^3m^3)$ operations in $\mathbb{F}_q$. The probability that a randomly drawn s-dimensional $\mathbb{F}_q$-linear subspace of $\mathbb{F}_{q^m}$ is a column rank super-support of the error is given by

$$\begin{bmatrix} s \\ t \end{bmatrix}_q \begin{bmatrix} m \\ t \end{bmatrix}_q^{-1} \approx q^{-t(m-s)},$$

and the probability that a randomly drawn s-dimensional $\mathbb{F}_q$-linear subspace of $\mathbb{F}_q^n$ contains the row rank support of the error is

$$\begin{bmatrix} s \\ t \end{bmatrix}_q \begin{bmatrix} n \\ t \end{bmatrix}_q^{-1} \approx q^{-t(n-s)}.$$

Furthermore, the complexity of checking whether the drawn subspace is a column or row rank super-support of the error is in $O(n^3m^3)$. Thus, in case there is only one solution to $\mathsf{SeaSD}_{\mathrm{R}}$, the Gaborit–Ruatta–Schrek decoder [139] has a complexity of approximately

$$W_{\mathrm{GRS}} = n^3m^3q^{t(\min\{n,m\}-s)} \geq n^3m^3q^{t\min\left\{k, \lceil \frac{km}{n} \rceil\right\}} \tag{2.6}$$

operations in $\mathbb{F}_q$, which is minimal for $s = \min\{n-k, \frac{m}{n}(n-k)\}$. This method has been improved, and the currently fastest variant [140] requires

$$W_{\mathrm{Comb}} = \min\left\{n^3m^3q^{t\lceil (k+1)m/n \rceil - m}, n^3m^3q^{tk}\right\}$$

operations in $\mathbb{F}_q$. If there are multiple solutions to $\mathsf{SeaSD}_{\mathrm{R}}$, the complexity of the combinatorial rank syndrome decoders has to be divided by the number of solutions,

[2] We defined the column rank support of $\boldsymbol{e} = [e_1, \ldots, e_n]$ by $\langle e_1, \ldots, e_n \rangle_q \subseteq \mathbb{F}_{q^m}$ and the row rank support of $\boldsymbol{e}$ by $\mathcal{R}_q(\mathrm{ext}_{q^m/q}(\boldsymbol{e})) \subseteq \mathbb{F}_q^n$, as this allows us a simplified notation in the following chapters. Therefore, we have to guess subspaces of $\mathbb{F}_{q^m}$ to find a column rank super-support of $\boldsymbol{e}$ and subspaces of $\mathbb{F}_q^n$ to find a row rank super-support of $\boldsymbol{e}$.

which is, on average, equal to

$$N_{\mathrm{R}} \coloneqq \frac{\sum\limits_{i=0}^{t} \left(\prod\limits_{j=0}^{i-1} \left(q^m - q^j \right) \right) \begin{bmatrix} n \\ i \end{bmatrix}_q}{q^{m(n-k)}}. \tag{2.7}$$

Therefore, we define $W_{\mathrm{CRSD}} \coloneqq W_{\mathrm{Comb}}/N_{\mathrm{R}}$.

The second family is called *algebraic* rank syndrome decoders, but a detailed description of these algorithms is outside the scope of this dissertation. The algebraic rank syndrome decoders are usually more efficient than the combinatorial rank syndrome decoders, and we denote the complexity of the algebraic methods by W_{ARSD}. Since the complexity expression is quite involved, we state the formula in Appendix A.1. Note that it is not known how to reduce the complexity of the algebraic techniques in case there are multiple solutions to $\mathsf{SeaSD}_{\mathrm{R}}$.

2.3 Complexity Classes

In the following, we review complexity classes that we make use of in this dissertation (e.g., see [143]). For that, let $\mathcal{A}$ denote an algorithm that has as input a sequence of random bits r and the input x of the considered decision problem L. Then, $\mathcal{A}$ is referred to as a Probabilistic Polynomial Time (PPT) algorithm if the length of the random sequence r is polynomial in the size of the input x and if $\mathcal{A}$ runs in time polynomial in the size of the input x. In the following definitions, the variable Δ is any constant with $0 \leq \Delta < 1$, and the stated probabilities are for a fixed input x and a random sequence r.

Definition 2.23 (Polynomial Time (P))**.** *The problem L is in the class* P *if there is a PPT algorithm $\mathcal{A}^{\mathsf{P}}$ with output* true *or* false *such that $\forall x \in L$ we have $\forall r\ \mathcal{A}^{\mathsf{P}}(x,r) =$* true*, and $\forall x \notin L$ we have $\forall r\ \mathcal{A}^{\mathsf{P}}(x,r) =$* false*.*

Definition 2.24 (Randomized Polynomial Time (RP))**.** *The problem L is in the class* RP *if there is a PPT algorithm $\mathcal{A}^{\mathsf{RP}}$ with output* true *or* false *such that $\forall x \in L$ it holds that $\Pr(\mathcal{A}^{\mathsf{RP}}(x,r) = \mathsf{true}) \geq \Delta$, and $\forall x \notin L$ we have $\forall r\ \mathcal{A}^{\mathsf{RP}}(x,r) =$* false*.*

Definition 2.25 (Co-Randomized Polynomial Time (coRP))**.** *The problem L is in the class* coRP *if there is a PPT algorithm $\mathcal{A}^{\mathsf{coRP}}$ with output* true *or* false *such that $\forall x \in L$ we have $\forall r\ \mathcal{A}^{\mathsf{coRP}}(x,r) =$* true*, and $\forall x \notin L$ it holds that $\Pr(\mathcal{A}^{\mathsf{coRP}}(x,r) = \mathsf{false}) \geq \Delta$.*

Definition 2.26 (Zero-Error Probabilistic Polynomial Time (ZPP))**.** *The problem L is in the class* ZPP *if there is a PPT algorithm* $\mathcal{A}^{\mathsf{ZPP}}$ *with output* true, false, *or* fail *such that for all x we have* $\Pr(\mathcal{A}^{\mathsf{ZPP}}(x,r) = \mathsf{fail}) \leq \Delta$*. Furthermore, for all x and r it holds that* $\mathcal{A}^{\mathsf{ZPP}}(x,r) = \mathsf{true}$ *implies* $x \in L$*, and* $\mathcal{A}^{\mathsf{ZPP}}(x,r) = \mathsf{false}$ *implies* $x \notin L$*.*

Note that $\mathsf{ZPP} = \mathsf{RP} \cap \mathsf{coRP}$.

Definition 2.27 (Non-Deterministic Polynomial Time (NP))**.** *The problem L is in the class* NP *if there is a PPT algorithm* $\mathcal{A}^{\mathsf{NP}}$ *such that* $x \in L$ *exactly when there is an r such that* $\mathcal{A}^{\mathsf{NP}}(x,r) = \mathsf{true}$*.*

Note that the chain $\mathsf{P} \subseteq \mathsf{ZPP} \subseteq \mathsf{RP} \subseteq \mathsf{NP}$ holds.

2.4 Cryptographic Schemes and Attack Models

In this dissertation, we consider two different types of public-key cryptosystems, which are known as public-key encryption schemes and KEMs.[3] The former type of systems is used for a confidential communication between parties that did not agree on any secret in advance. The systems of this type are defined as follows:

Definition 2.28 (Public-Key Encryption Scheme)**.** *A public-key encryption scheme* Π^{Enc} *consists of three PPT algorithms* $(\mathsf{KeyGen}, \mathsf{Encrypt}, \mathsf{Decrypt})$ *with the following properties:*

1. *The key-generation algorithm* KeyGen *takes as input parameters which are chosen according to the desired security level λ and returns a pair of keys* $(\mathsf{sk}, \mathsf{pk})$*. The former of these is called the private key and the latter is called the public key.*

2. *The encryption algorithm* $\mathsf{Encrypt}$ *takes as input a public key* pk *and a message* $\boldsymbol{m}$*, and it returns a ciphertext* $\boldsymbol{y}$*.*

3. *The decryption algorithm* $\mathsf{Decrypt}$ *takes as input a private key* sk *and a ciphertext* $\boldsymbol{y}$*, and it returns either a message* $\boldsymbol{m}$ *or a decryption failure.*

For any message $\boldsymbol{m}$*, the inequality* $\mathsf{Decrypt}(\mathsf{Encrypt}(\boldsymbol{m}, \mathsf{pk}), \mathsf{sk}) \neq \boldsymbol{m}$ *is only allowed to hold with negligible probability over* $(\mathsf{sk}, \mathsf{pk})$*.*

[3] A detailed discussion of the considered cryptographic schemes and attack models is given in [2, Cha. III]. Note that we use similar notations and definitions as in [2].

The security of public-key encryption schemes is commonly proven under assumptions on the capability of the adversaries. The basic requirement for these systems is IND-CPA security, which means that the adversaries can query an encryption oracle but do not have access to a decryption oracle. To formally define IND-CPA security, consider the following game between an adversary and a challenger:

Definition 2.29 (The IND-CPA Game $\mathsf{PubEnc}^{\mathsf{CPA}}_{\mathcal{A},\Pi^{\mathrm{Enc}}}(\lambda)$)**.**

1. *The challenger runs* KeyGen *to generate a key pair* $(\mathsf{sk}, \mathsf{pk})$ *depending on the security level* λ.
2. *The adversary* $\mathcal{A}$ *is given* pk *and oracle access to* $\mathsf{Encrypt}(\,\cdot\,, \mathsf{pk})$.[4] *Then, the adversary returns a pair of equal-length messages* $(\boldsymbol{m}_1, \boldsymbol{m}_2)$.
3. *The challenger draws an integer* b *uniformly at random from* $\{1, 2\}$, *computes a ciphertext* $\boldsymbol{y} = \mathsf{Encrypt}(\boldsymbol{m}_b, \mathsf{pk})$ *and gives* $\boldsymbol{y}$ *to* $\mathcal{A}$.
4. *The adversary* $\mathcal{A}$ *continues to have oracle access to* $\mathsf{Encrypt}(\,\cdot\,, \mathsf{pk})$ *and returns an integer* $b' \in \{1, 2\}$.

The output of the experiment is 1 *if* $b' = b$ *and* 0 *otherwise. If* $b' = b$, *we say that* $\mathcal{A}$ *succeeds.*

Based on this game, we define IND-CPA security as follows:

Definition 2.30 (IND-CPA Security of Public-Key Encryption Schemes)**.** *A public-key encryption scheme* $\Pi^{\mathrm{Enc}} = (\mathsf{KeyGen}, \mathsf{Encrypt}, \mathsf{Decrypt})$ *is called IND-CPA-secure if for all PPT adversaries* $\mathcal{A}$, *there exists a negligible function*[5] f_{ngl} *such that*

$$\Pr\left(\mathsf{PubEnc}^{\mathsf{CPA}}_{\mathcal{A},\Pi^{\mathrm{Enc}}}(\lambda) = 1\right) \leq \frac{1}{2} + f_{\mathrm{ngl}}(\lambda),$$

where λ *denotes the security level.*

Compared to symmetric encryption schemes, public-key encryption schemes have the disadvantage of slower encryption and decryption algorithms as well as larger key and ciphertext sizes. Therefore, hybrid schemes consisting of a symmetric and a

[4] Although the oracle access to $\mathsf{Encrypt}(\,\cdot\,, \mathsf{pk})$ is redundant, we state it for didactic reasons, as it explains why it is called a chosen-plaintext game.

[5] A function $f : \mathbb{N} \to \mathbb{R}$ is called negligible if for every positive integer p there is an integer N such that for all integers $x > N$ it holds that $|f(x)| < x^{-p}$.

public-key cryptosystem are often used to transmit large amounts of data. In these schemes, the public-key system is used to exchange a secret key, and the symmetric cryptosystem uses the exchanged secret key to encrypt the data. This type of public-key cryptosystem is known as KEM and is defined as follows:

Definition 2.31 (Key-Encapsulation Mechanism)**.** *A KEM consists of three PPT algorithms* $(\mathsf{KeyGen}, \mathsf{Encaps}, \mathsf{Decaps})$ *with the following properties:*

1. *The key-generation algorithm* KeyGen *takes as input parameters which are chosen according to the desired security level* λ *and returns a key pair* $(\mathsf{sk}, \mathsf{pk})$*.*
2. *The encapsulation algorithm* Encaps *takes as input a public key* pk*, and returns a ciphertext* $\boldsymbol{y}$ *and a shared key* K*.*
3. *The decapsulation algorithm* Decaps *takes as input a private key* sk *and a ciphertext* $\boldsymbol{y}$*, and it returns a key* K *or a decapsulation failure.*

If Encaps *outputs* $(\boldsymbol{y}, K)$*, then it is only allowed that* $\mathsf{Decaps}(\boldsymbol{y}, \mathsf{sk})$ *returns* $K' \neq K$ *with negligible probability over* $(\mathsf{sk}, \mathsf{pk})$*.*

We are interested in KEMs that provide IND-CCA2 security, which means that the adversaries have access to both an encapsulation and a decapsulation oracle. To define IND-CCA2 security, consider the following game:

Definition 2.32 (The IND-CCA2 Game $\mathsf{KEM}^{\mathsf{CCA2}}_{\mathcal{A},\Pi^{\mathrm{KEM}}}(\lambda)$)**.**

1. *The challenger runs* KeyGen *to generate a key pair* $(\mathsf{sk}, \mathsf{pk})$ *depending on the security level* λ *and runs* $\mathsf{Encaps}(\mathsf{pk})$ *to obtain* $(\boldsymbol{y}, K)$*.*
2. *The challenger draws an integer* b *uniformly at random from* $\{1, 2\}$*. If* $b = 1$*, he chooses* $\hat{K} = K$*, and if* $b = 2$*, he draws* $\hat{K}$ *uniformly at random from all possible shared keys.*
3. *The adversary* $\mathcal{A}$ *is given* $(\mathsf{pk}, \boldsymbol{y}, \hat{K})$*. Furthermore,* $\mathcal{A}$ *obtains access to the oracle* $\mathsf{Encaps}(\mathsf{pk})$ *and the oracle* $\mathsf{Decaps}(\,\cdot\,, \mathsf{sk})$*, where* $\mathcal{A}$ *is not allowed to query* $\mathsf{Decaps}(\boldsymbol{y}, \mathsf{sk})$*.*
4. *The adversary* $\mathcal{A}$ *returns an integer* $b' \in \{1, 2\}$*.*

The output of the experiment is 1 *if* $b' = b$ *and* 0 *otherwise. If* $b' = b$*, we say that* $\mathcal{A}$ *succeeds.*

Based on this game, the IND-CCA2 security of KEMs is given as follows:

Definition 2.33 (IND-CCA2 Security of KEMs)**.** *A KEM is called IND-CCA2-secure if for all PPT adversaries $\mathcal{A}$, there exists a negligible function f_{ngl} such that*

$$\Pr\left(\mathsf{KEM}^{\mathsf{CCA2}}_{\mathcal{A},\Pi^{\mathrm{KEM}}}(\lambda) = 1\right) \leq \frac{1}{2} + f_{\mathrm{ngl}}(\lambda),$$

where λ denotes the security level.

3

Coding-Theoretic Problems with Applications in Cryptography

In public-key cryptography, the security of a given system is ensured through the computational hardness of the underlying mathematical problems. Early systems like the Diffie–Hellman key exchange mechanism and the Rivest–Shamir–Adleman (RSA) encryption scheme are based on the hardness of computing discrete logarithms and factoring large integers [3, 144]. However, due to Shor's algorithm [6], the two aforementioned problems became easy-to-solve by sufficiently large quantum computers, and therefore, cryptographers started to look for other difficult problems that could serve as the core of post-quantum secure systems. Currently, there are two main branches in the field of quantum-resistant encryption schemes. One branch is referred to as lattice-based cryptography, where most systems rely on variants of the shortest vector problem, the closest vector problem, or learning with errors, see e.g., [145–147]. The other branch is known as code-based cryptography, where the problems *syndrome decoding in the Hamming metric* and *syndrome decoding in the rank metric* often serve as the starting point to build cyptographic schemes. The first Hamming-based system by McEliece partly relies on the hardness of syndrome decoding in the Hamming metric [13, 135]. Since the system by McEliece suffers from large key sizes, new schemes have been developed, whose security is based on variants of Hamming syndrome decoding, e.g., the systems Bit Flipping Key Encapsulation (BIKE) [52, 148] and HQC [63, 149]. The security of the former system depends on the hardness of finding codewords of low Hamming weight and the hardness of Hamming syndrome decoding in cyclic codes, and the security of the latter scheme only relies on the hardness of variants of Hamming syndrome decoding in cyclical codes. As syndrome decoding in the rank metric seems to be harder than in the Hamming metric, rank-based systems potentially allow one to reduce both the key and the ciphertext sizes. Two well-studied rank-based encryption schemes are ROLLO[1] [56, 150] and Rank Quasi-Cyclic (RQC) [63, 151]. While the security of ROLLO is partly based on the hardness of rank syndrome decoding in ideal codes, RQC solely relies on variants of this problem.

In this chapter, we investigate three problems that are of importance to code-based cryptography. The first problem refers to *syndrome decoding in the sum-rank metric*, where the sum-rank metric is a family of metrics that coincides with the Hamming metric and the rank metric in special cases. We formally state the decoding problem and show that it can be seen as a generalization of $\mathsf{DecSD}_{\mathrm{H}}$ and $\mathsf{DecSD}_{\mathrm{R}}$, which are formally defined in Section 2.2.4. Furthermore, we present a randomized reduction of $\mathsf{DecSD}_{\mathrm{H}}$ to syndrome decoding in the sum-rank metric, which indicates that the considered problem is hard and could be a suitable metric upon which to build cryptographic schemes. In addition, we present the first non-trivial algorithm to solve this problem.

The second problem is *syndrome decoding of high-order interleaved rank-metric codes*. This problem can be seen as a variant of $\mathsf{DecSD}_{\mathrm{R}}$, where one gets u instances of $\mathsf{DecSD}_{\mathrm{R}}$, and the solutions to these instances share the same row-support. We present an efficient algorithm to solve this variant of $\mathsf{DecSD}_{\mathrm{R}}$, we analyze the complexity of the algorithm, and we compare it to other decoding strategies. The proposed algorithm has an impact on rank-based McEliece or Niederreiter schemes, as it proves that in the case of multiple encryptions, the row-supports of the errors have to be generated independently; otherwise the systems will have a high probability of being insecure.

The third problem refers to *decoding Gabidulin codes beyond their unique decoding radius* and builds the core of rank-based systems like RQC, LIGA [106, 152], and Rank Metric Encryption Scheme with Short Keys (RAMESSES) [153]. Therefore, the hardness of this problem is important to assess the security of these schemes. Although we have no rigorous proof that the problem is actually hard, it should be noted that many scientists have tried to solve it in polynomial time and have not been successful. We propose a randomized decoding algorithm that solves the problem in exponential

[1] The system ROLLO is the compilation of Rank-Ouroboros, Low Rank Parity Check Codes Key Exchange (LAKE) and Low Rank Parity Check Codes Encryption (LOCKER).

time, and we show that it is currently the most efficient algorithm for many parameter sets. The complexity of this algorithm gives an indication and an upper bound on the hardness of the problem.

The results shown in Section 3.1 are based on [154], which is published in the proceedings of the *2020 IEEE International Symposium on Information Theory (ISIT)*, and on [155], which is currently under review for publication in the *IEEE Transactions on Information Theory*. The author of this dissertation contributed all of the content that is shown in Section 3.1. The cited papers contain further algorithms for an efficient computation of the derived bounds and for efficiently drawing vectors from a set according to a given non-uniform distribution. These algorithms are only referenced but not shown in this dissertation, as they were mainly developed by the other authors of [154, 155].

Parts of Section 3.2 are included in the proceedings of the *2021 IEEE International Symposium on Information Theory (ISIT)* [156]. The content of [156] that is described in this section was contributed by the author of this thesis. Furthermore, Section 3.2 contains proofs for some of the conjectures that are published in [156]. The cited publication contains comparisons of the proposed decoder with its Hamming metric equivalent and with Simple codes [157]. The comparison with the Hamming metric decoder is outside the scope of this dissertation, and the comparison with Simple codes is not presented, as it was mainly contributed by the other authors of [156].

The results given in Section 3.3 are published in the proceedings of the *2020 International Conference on Post-Quantum Cryptography (PQCrypto)* [158]. The author of this dissertation contributed all of the content that is presented in this section, except for the simulation result shown in the first row of Table 3.1.

3.1 Syndrome Decoding in the Sum-Rank Metric

In this section, we investigate the problem of decoding random codes in the sum-rank metric. We define the decisional version of the problem as follows:

Problem 3.1 (Decisional Sum-Rank Syndrome Decoding ($\mathsf{DecSD}_{\mathrm{SR}}$) Problem)**.**
Given: - *Parity-check matrix $\boldsymbol{H} \in \mathbb{F}_{q^m}^{(n-k)\times n}$ of an $[n,k]_{\mathbb{F}_{q^m}}$ code $\mathcal{C}$*
- *Non-negative integer t*
- *Vector $\boldsymbol{s} \in \mathbb{F}_{q^m}^{n-k}$*

Question: *Is there an $\boldsymbol{e} \in \mathbb{F}_{q^m}^{n}$ such that $\mathrm{wt}_{\mathrm{SR}}(\boldsymbol{e}) \leq t$ and $\boldsymbol{s} = \boldsymbol{e}\boldsymbol{H}^{\top}$?*

We study the hardness of the this problem and propose a non-trivial algorithm to solve it. As it is usually done for all decoding-based problems, we solve the decisional problem $\mathsf{DecSD}_{\mathrm{SR}}$ (Problem 3.1) by trying to find a solution to the associated *search* problem, which is defined as follows:

Problem 3.2 (Search Sum-Rank Syndrome Decoding ($\mathsf{SeaSD}_{\mathrm{SR}}$) Problem)**.**
Given: - *Parity-check matrix $\boldsymbol{H} \in \mathbb{F}_{q^m}^{(n-k)\times n}$ of an $[n,k]_{\mathbb{F}_{q^m}}$ code $\mathcal{C}$*
- *Non-negative integer t*
- *Vector $\boldsymbol{s} = \boldsymbol{e}\boldsymbol{H}^{\top} \in \mathbb{F}_{q^m}^{n-k}$, where $\mathrm{wt}_{\mathrm{SR}}(\boldsymbol{e}) = t$*

Objective: *Search for an $\boldsymbol{e}' \in \mathbb{F}_{q^m}^{n}$ such that $\mathrm{wt}_{\mathrm{SR}}(\boldsymbol{e}') \leq t$ and $\boldsymbol{s} = \boldsymbol{e}'\boldsymbol{H}^{\top}$.*

To solve the search problem $\mathsf{SeaSD}_{\mathrm{SR}}$ (Problem 3.2), we devise a non-trivial generic sum-rank-metric decoding algorithm. The approach of our proposed algorithm is comparable to some generic decoding algorithms in the Hamming and the rank metric. First, it determines the support of an error, and then, it obtains the full error by erasure decoding. To derive and analyze our algorithm, we need some statements about erasure decoding.

3.1.1 Erasure Decoding in the Sum-Rank Metric

We first state an upper bound on the number of vectors in $\mathbb{F}_{q^m}^{n}$ of sum-rank weight $t \leq \mu_{\mathrm{SR}}\ell_{\mathrm{SR}}$, where $\mu_{\mathrm{SR}} := \min\{\eta_{\mathrm{SR}}, m\}$.

Theorem 3.1. *Let $m, \eta_{\mathrm{SR}}, \mu_{\mathrm{SR}}, \ell_{\mathrm{SR}}$, and t be non-negative integers such that $\mu_{\mathrm{SR}} = \min\{\eta_{\mathrm{SR}}, m\}$ and $\ell_{\mathrm{SR}} \geq 1$. Then, for $t \leq \mu_{\mathrm{SR}}\ell_{\mathrm{SR}}$, the number of vectors in $\mathbb{F}_{q^m}^{\eta_{\mathrm{SR}}\ell_{\mathrm{SR}}}$ of sum-rank weight t is given by*

$$\mathcal{N}_{q,\eta_{\mathrm{SR}},m}(t, \ell_{\mathrm{SR}}) = \sum_{\boldsymbol{t} \in \mathcal{T}_{t,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}}} \prod_{i=1}^{\ell_{\mathrm{SR}}} \mathrm{NM}_q(m, \eta_{\mathrm{SR}}, t_i)$$

$$= \begin{cases} \mathrm{NM}_q(m, \eta_{\mathrm{SR}}, t), & \textit{if } \ell_{\mathrm{SR}} = 1, \\ \sum\limits_{t'=0}^{\min\{\eta_{\mathrm{SR}}, m, t\}} \mathrm{NM}_q(m, \eta_{\mathrm{SR}}, t') \cdot \mathcal{N}_{q,\eta_{\mathrm{SR}},m}(t - t', \ell_{\mathrm{SR}} - 1), & \textit{if } \ell_{\mathrm{SR}} > 1, \end{cases}$$
$$\leq \binom{\ell_{\mathrm{SR}} + t - 1}{\ell_{\mathrm{SR}} - 1} 4^{\ell_{\mathrm{SR}}} q^{t(m+\eta_{\mathrm{SR}} - \frac{t}{\ell_{\mathrm{SR}}})},$$

where the variables $\mathcal{T}_{t,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}}$ *and* $\mathrm{NM}_q(m, \eta_{\mathrm{SR}}, t)$ *denote the set of weight decompositions of sum-rank weight* t *and the number of* $m \times \eta_{\mathrm{SR}}$ *matrices over* $\mathbb{F}_q$ *of rank* t*, respectively. Furthermore, for* $t > \mu_{\mathrm{SR}}\ell_{\mathrm{SR}}$*, it holds that* $\mathcal{N}_{q,\eta_{\mathrm{SR}},m}(t, \ell_{\mathrm{SR}}) = 0$.

Proof. For $\ell_{\mathrm{SR}} = 1$, the quantity $\mathcal{N}_{q,\eta_{\mathrm{SR}},m}(t, \ell_{\mathrm{SR}})$ is the number of $m \times \eta_{\mathrm{SR}}$ matrices of rank t. For $\ell_{\mathrm{SR}} > 1$, we sum up over the number of possibilities to choose the rank weight t' of the first block multiplied with the number of sum-rank weight words in the remaining $\ell_{\mathrm{SR}} - 1$ blocks.

The upper bound can be derived as follows: Since $\mathrm{NM}_q(m, \eta_{\mathrm{SR}}, t_i) \leq 4q^{t_i(m+\eta_{\mathrm{SR}}-t_i)}$ and $|\mathcal{T}_{t,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}}| \leq \binom{\ell_{\mathrm{SR}}+t-1}{\ell_{\mathrm{SR}}-1}$, see (2.1) and (2.4), we observe that

$$\sum_{\boldsymbol{t} \in \mathcal{T}_{t,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}}} \prod_{i=1}^{\ell_{\mathrm{SR}}} \mathrm{NM}_q(m, \eta_{\mathrm{SR}}, t_i) \leq |\mathcal{T}_{t,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}}| \max_{\boldsymbol{t} \in \mathcal{T}_{t,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}}} \left\{ \prod_{i=1}^{\ell_{\mathrm{SR}}} \mathrm{NM}_q(m, \eta_{\mathrm{SR}}, t_i) \right\}$$
$$\leq \binom{\ell_{\mathrm{SR}} + t - 1}{\ell_{\mathrm{SR}} - 1} 4^{\ell_{\mathrm{SR}}} q^{\max_{\boldsymbol{t} \in \mathcal{T}_{t,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}}} \left\{ \sum_{i=1}^{\ell_{\mathrm{SR}}} t_i(m+\eta_{\mathrm{SR}}-t_i) \right\}}.$$

For $\sum_{i=1}^{\ell_{\mathrm{SR}}} t_i = t$, the term $\max_{\boldsymbol{t} \in \mathcal{T}_{t,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}}} \left\{ \sum_{i=1}^{\ell_{\mathrm{SR}}} t_i(m + \eta_{\mathrm{SR}} - t_i) \right\}$ is equal to

$$t(m + \eta_{\mathrm{SR}}) - \min_{\boldsymbol{t} \in \mathcal{T}_{t,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}}} \left\{ \sum_{i=1}^{\ell_{\mathrm{SR}}} t_i^2 \right\} \leq t(m + \eta_{\mathrm{SR}}) - \frac{t^2}{\ell_{\mathrm{SR}}},$$

where the upper bound follows from Jensen's inequality.

Since each of the ℓ_{SR} blocks has a rank weight of at most μ_{SR}, it holds that $\mathcal{N}_{q,\eta_{\mathrm{SR}},m}(t, \ell_{\mathrm{SR}}) = 0$ for $t > \mu_{\mathrm{SR}}\ell_{\mathrm{SR}}$. ■

Figure 3.1 shows example values of $\mathcal{N}_{q,\eta_{\mathrm{SR}},m}(t, \ell_{\mathrm{SR}})$ and the respective bound given in Theorem 3.1 for different values of ℓ_{SR}. It seems that the bound is quite tight for most values of ℓ_{SR} and only significantly differs for ℓ_{SR} close to n. This deviation is due to the factor $4^{\ell_{\mathrm{SR}}}$, which is large for large values of ℓ_{SR}, and which is due to a relatively loose bound on the number of matrices.

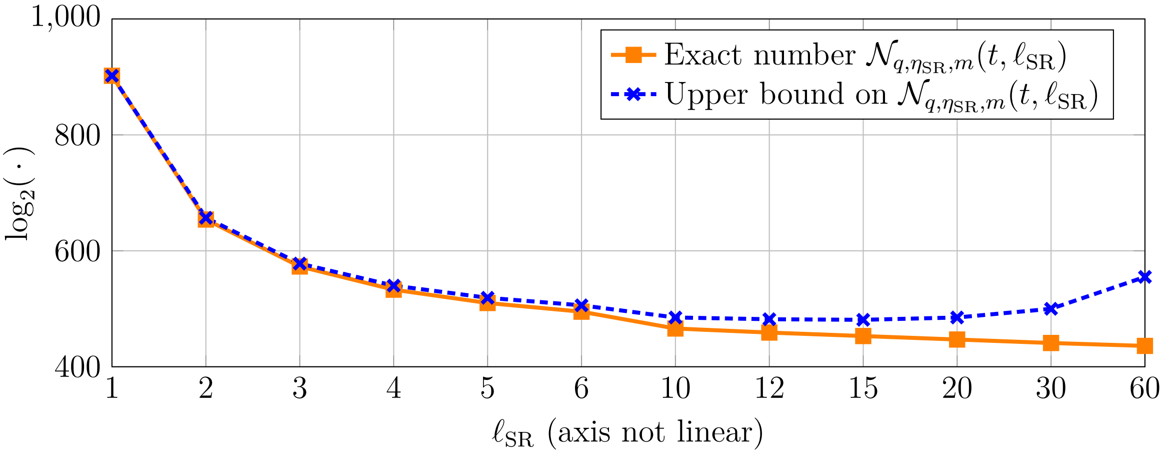


Figure 3.1: Comparison of the exact number of vectors of sum-rank weight t, i.e., $\mathcal{N}_{q,\eta_{\mathrm{SR}},m}(t,\ell_{\mathrm{SR}})$, and the bound given in Theorem 3.1 for $q = 2$, $m = 40$, $n = 60$, and $t = 10$ as a function of ℓ_{SR}.

Note that the recursion in Theorem 3.1 can be turned into an efficient algorithm to draw vectors uniformly at random from the set of vectors of sum-rank weight t, see Appendix B.1.

Supports in the Sum-Rank Metric

Similar to the rank metric, we have two types of supports in the sum-rank metric, which we call the row sum-rank support and the column sum-rank support.

Lemma 3.2. *Let $\boldsymbol{e} = [\boldsymbol{e}_1, \boldsymbol{e}_2, \dots, \boldsymbol{e}_{\ell_{\mathrm{SR}}}]$ be a vector in $\mathbb{F}_{q^m}^n$, where $\mathrm{wt}_{\mathrm{SR}}(\boldsymbol{e}) = t$. Furthermore, let the vector $\boldsymbol{t}$ denote the weight decomposition of $\boldsymbol{e}$. Then, there are vectors $\boldsymbol{a}_i \in \mathbb{F}_{q^m}^{t_i}$ and matrices $\boldsymbol{B}_i \in \mathbb{F}_q^{t_i \times \eta_{\mathrm{SR}}}$ with $\mathrm{rk}_q(\boldsymbol{a}_i) = \mathrm{rk}_q(\boldsymbol{B}_i) = t_i$, such that $\boldsymbol{e}_i = \boldsymbol{a}_i \boldsymbol{B}_i$, for $i \in [1:\ell_{\mathrm{SR}}]$. Stated differently,*

$$\boldsymbol{e} = \overbrace{\begin{bmatrix} \boldsymbol{a}_1 & \boldsymbol{a}_2 & \boldsymbol{a}_3 & \dots & \boldsymbol{a}_{\ell_{\mathrm{SR}}} \end{bmatrix}}^{=:\,\boldsymbol{a}\,\in\,\mathbb{F}_{q^m}^{t}} \cdot \overbrace{\begin{bmatrix} \boldsymbol{B}_1 & \boldsymbol{0} & \boldsymbol{0} & \dots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{B}_2 & \boldsymbol{0} & \dots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{B}_3 & \dots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \dots & \boldsymbol{B}_{\ell_{\mathrm{SR}}} \end{bmatrix}}^{=:\,\boldsymbol{B}\,\in\,\mathbb{F}_q^{t \times n}}.$$

The decomposition is unique up to elementary $\mathbb{F}_q$-linear operations on the entries of the vectors $\boldsymbol{a}_i$ and $\mathbb{F}_q$-row operations on the matrices $\boldsymbol{B}_i$.

Proof. The decomposition $\boldsymbol{e}_i = \boldsymbol{a}_i\boldsymbol{B}_i$ and its uniqueness up to $\mathbb{F}_q$-linear operations follow from the same arguments as in the rank metric [86]. ■

Definition 3.1 (Row Sum-Rank Support). *Let $\boldsymbol{e} = [\boldsymbol{e}_1, \boldsymbol{e}_2, \dots, \boldsymbol{e}_{\ell_{\mathrm{SR}}}]$ be a vector in $\mathbb{F}_{q^m}^n$, where $\mathrm{wt}_{\mathrm{SR}}(\boldsymbol{e}) = t$. Then, the row sum-rank support of the vector $\boldsymbol{e}$ is defined as the product*

$$\mathrm{supp}_{\mathrm{SR}}^{(\mathrm{R})}(\boldsymbol{e}) \coloneqq \mathrm{supp}_{\mathrm{R}}^{(\mathrm{R})}(\boldsymbol{e}_1) \times \mathrm{supp}_{\mathrm{R}}^{(\mathrm{R})}(\boldsymbol{e}_2) \times \cdots \times \mathrm{supp}_{\mathrm{R}}^{(\mathrm{R})}(\boldsymbol{e}_{\ell_{\mathrm{SR}}}),$$

where $\mathrm{supp}_{\mathrm{R}}^{(\mathrm{R})}(\boldsymbol{e}_i) \subseteq \mathbb{F}_q^{\eta_{\mathrm{SR}}}$ is equal to the $\mathbb{F}_q$-linear space spanned by the rows of $\boldsymbol{B}_i \in \mathbb{F}_q^{t_i \times \eta_{\mathrm{SR}}}$, for $i \in [1 : \ell_{\mathrm{SR}}]$, and $\boldsymbol{B}_1, \dots, \boldsymbol{B}_{\ell_{\mathrm{SR}}}$ are as in Lemma 3.2. Furthermore, a product of subspaces $\mathcal{F}^{(\mathrm{R})}$ is called a row sum-rank super-support of $\boldsymbol{e}$ if

$$\mathcal{F}^{(\mathrm{R})} \coloneqq \mathcal{F}_1^{(\mathrm{R})} \times \mathcal{F}_2^{(\mathrm{R})} \times \cdots \times \mathcal{F}_{\ell_{\mathrm{SR}}}^{(\mathrm{R})}$$

and $\mathrm{supp}_{\mathrm{R}}^{(\mathrm{R})}(\boldsymbol{e}_i) \subseteq \mathcal{F}_i^{(\mathrm{R})}$, for $i \in [1 : \ell_{\mathrm{SR}}]$.

Note that the term row support in the sum-rank metric was already defined in [159].

Definition 3.2 (Column Sum-Rank Support). *Let $\boldsymbol{e} = [\boldsymbol{e}_1, \boldsymbol{e}_2, \dots, \boldsymbol{e}_{\ell_{\mathrm{SR}}}]$ be a vector in $\mathbb{F}_{q^m}^n$, where $\mathrm{wt}_{\mathrm{SR}}(\boldsymbol{e}) = t$. Then, the column sum-rank support of the vector $\boldsymbol{e}$ is defined as the product*

$$\mathrm{supp}_{\mathrm{SR}}^{(\mathrm{C})}(\boldsymbol{e}) \coloneqq \mathrm{supp}_{\mathrm{R}}^{(\mathrm{C})}(\boldsymbol{e}_1) \times \mathrm{supp}_{\mathrm{R}}^{(\mathrm{C})}(\boldsymbol{e}_2) \times \cdots \times \mathrm{supp}_{\mathrm{R}}^{(\mathrm{C})}(\boldsymbol{e}_{\ell_{\mathrm{SR}}}),$$

where $\mathrm{supp}_{\mathrm{R}}^{(\mathrm{C})}(\boldsymbol{e}_i) \subseteq \mathbb{F}_{q^m}$ is equal to the $\mathbb{F}_q$-linear space spanned by the entries of $\boldsymbol{a}_i \in \mathbb{F}_{q^m}^{t_i}$, for $i \in [1 : \ell_{\mathrm{SR}}]$, and $\boldsymbol{a}_1, \dots, \boldsymbol{a}_{\ell_{\mathrm{SR}}}$ are as in Lemma 3.2. Furthermore, a product of subspaces $\mathcal{F}^{(\mathrm{C})}$ is called a column sum-rank super-support of $\boldsymbol{e}$ if

$$\mathcal{F}^{(\mathrm{C})} \coloneqq \mathcal{F}_1^{(\mathrm{C})} \times \mathcal{F}_2^{(\mathrm{C})} \times \cdots \times \mathcal{F}_{\ell_{\mathrm{SR}}}^{(\mathrm{C})}$$

and $\mathrm{supp}_{\mathrm{R}}^{(\mathrm{C})}(\boldsymbol{e}_i) \subseteq \mathcal{F}_i^{(\mathrm{C})}$, for $i \in [1 : \ell_{\mathrm{SR}}]$.

To simplify the notation, we will only specify whether we refer to the row or to the column support in case it is not clear from the context.

Definition 3.3. *Let μ_{SR} and s be positive integers such that $0 \leq s \leq \mu_{\mathrm{SR}}\ell_{\mathrm{SR}}$ and let $\boldsymbol{f}$ be a vector in $\mathcal{T}_{s,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}}$. Then, we denote the sets of all products of subspaces of* *weight s and weight decomposition $\boldsymbol{f}$ by*

$$\Xi^{(\mathrm{C})}_{q,\mu_{\mathrm{SR}}}(\boldsymbol{f}) := \{\mathcal{F}_1 \times \cdots \times \mathcal{F}_{\ell_{\mathrm{SR}}} : \mathcal{F}_i \textit{ is an } f_i\textit{-dimensional subspace of } \mathbb{F}_{q^{\mu_{\mathrm{SR}}}}, \forall i \in [1:\ell_{\mathrm{SR}}]\}$$

and

$$\Xi^{(\mathrm{R})}_{q,\mu_{\mathrm{SR}}}(\boldsymbol{f}) := \left\{\mathcal{F}_1 \times \cdots \times \mathcal{F}_{\ell_{\mathrm{SR}}} : \mathcal{F}_i \textit{ is an } f_i\textit{-dimensional subspace of } \mathbb{F}_q^{\mu_{\mathrm{SR}}}, \forall i \in [1:\ell_{\mathrm{SR}}]\right\}.$$

As before, we will omit the superscripts (C) and (R) if it is clear from the context.

Erasure Decoding

In the following, we prove that, given a super-support of an error of weight smaller than the minimum distance of the code, erasure decoding always leads to a unique solution. To prove this statement, we need the following lemma:

Lemma 3.3. *Let t be an integer such that $0 \leq t \leq n$, let $\boldsymbol{H} \in \mathbb{F}_{q^m}^{(n-k)\times n}$ be a parity-check matrix of an $[n,k]_{\mathbb{F}_{q^m}}$ code $\mathcal{C}$, and let*

$$\mathcal{B}_{\ell_{\mathrm{SR}},t} := \left\{ \begin{bmatrix} \boldsymbol{B}_1 & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \boldsymbol{B}_2 & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \dots & \boldsymbol{B}_{\ell_{\mathrm{SR}}} \end{bmatrix} \in \mathbb{F}_q^{t\times n} : \boldsymbol{B}_i \in \mathbb{F}_q^{t_i \times (n/\ell_{\mathrm{SR}})}, \mathrm{rk}_q(\boldsymbol{B}_i) = t_i, \sum_{i=1}^{\ell_{\mathrm{SR}}} t_i = t \right\}.$$

Then, the code $\mathcal{C}$ has minimum sum-rank distance $d^{\mathrm{SR}}_{\min}$ if and only if $\mathrm{rk}_{q^m}\left(\boldsymbol{H}\boldsymbol{B}^\top\right) = d^{\mathrm{SR}}_{\min} - 1$ for any $\boldsymbol{B} \in \mathcal{B}_{\ell_{\mathrm{SR}},d^{\mathrm{SR}}_{\min}-1}$ and $\mathrm{rk}_{q^m}\left(\boldsymbol{H}\boldsymbol{B}^\top\right) < d^{\mathrm{SR}}_{\min}$ for at least one $\boldsymbol{B} \in \mathcal{B}_{\ell_{\mathrm{SR}},d^{\mathrm{SR}}_{\min}}$.

Proof. If $\mathcal{C}$ has minimum distance $d^{\mathrm{SR}}_{\min}$, then for any vector $\boldsymbol{x}$ with $\mathrm{wt}_{\mathrm{SR}}(\boldsymbol{x}) = d^{\mathrm{SR}}_{\min} - 1$, it must hold that $\boldsymbol{H}\boldsymbol{x}^\top \neq \mathbf{0}$. Furthermore, there is a vector $\boldsymbol{y}$ with $\mathrm{wt}_{\mathrm{SR}}(\boldsymbol{y}) = d^{\mathrm{SR}}_{\min}$ such that $\boldsymbol{H}\boldsymbol{y}^\top = \mathbf{0}$. This property together with the decomposition proposed in Lemma 3.2 proves this lemma. ■

Theorem 3.4 (Row Erasure Decoding)**.** *Let $\boldsymbol{H} \in \mathbb{F}_{q^m}^{(n-k)\times n}$ be a parity-check matrix of an $[n,k,d^{\mathrm{SR}}_{\min}]^{\mathrm{SR}}_{\mathbb{F}_{q^m}}$ code $\mathcal{C}$, let $\boldsymbol{e} \in \mathbb{F}_{q^m}^n$ be an error vector with $\mathrm{wt}_{\mathrm{SR}}(\boldsymbol{e}) = t < d^{\mathrm{SR}}_{\min}$, and let $\boldsymbol{s}^\top = \boldsymbol{H}\boldsymbol{e}^\top$. Furthermore, let*

$$\mathcal{F} = \mathcal{F}^{(\mathrm{C})} = \mathcal{F}^{(\mathrm{C})}_1 \times \mathcal{F}^{(\mathrm{C})}_2 \times \cdots \times \mathcal{F}^{(\mathrm{C})}_{\ell_{\mathrm{SR}}}$$

be any column super-support of $\boldsymbol{e}$, where $\mathcal{F}^{(\mathrm{C})}_i$ has dimension f_i, for $i \in [1:\ell_{\mathrm{SR}}]$, and

$\sum_{i=1}^{\ell_{\mathrm{SR}}} f_i = s < d_{\min}^{\mathrm{SR}}$. *Then, the error* $\boldsymbol{e}$ *can be determined from* $\boldsymbol{H}$ *and* $\boldsymbol{s}$ *given the* column *super-support* $\mathcal{F} = \mathcal{F}^{(\mathrm{C})}$ *of* $\boldsymbol{e}$ *with* $O((n-k)^3 m^3)$ *operations in* $\mathbb{F}_q$*.*

Proof. Let $\boldsymbol{H} = [\boldsymbol{H}_1, \ldots, \boldsymbol{H}_{\ell_{\mathrm{SR}}}] \in \mathbb{F}_{q^m}^{(n-k)\times n}$, where $\boldsymbol{H}_i \in \mathbb{F}_{q^m}^{(n-k)\times \eta_{\mathrm{SR}}}$, for $i \in [1:\ell_{\mathrm{SR}}]$. Then, the syndrome

$$\boldsymbol{s}^\top = \boldsymbol{H}\boldsymbol{e}^\top = \boldsymbol{H}\boldsymbol{B}^\top \boldsymbol{a}^\top = \sum_{i=1}^{\ell_{\mathrm{SR}}} \boldsymbol{H}_i \boldsymbol{B}_i^\top \boldsymbol{a}_i^\top = \sum_{i=1}^{\ell_{\mathrm{SR}}} \boldsymbol{H}_i \hat{\boldsymbol{B}}_i^\top \hat{\boldsymbol{a}}_i^\top,$$

where $\hat{\boldsymbol{a}} = [\hat{\boldsymbol{a}}_1, \ldots, \hat{\boldsymbol{a}}_{\ell_{\mathrm{SR}}}] \in \mathbb{F}_{q^m}^{s}$ is a basis of the known column super-support of the error and $\hat{\boldsymbol{B}}_i \in \mathbb{F}_q^{f_i \times \eta_{\mathrm{SR}}}$ are unknown, for $i \in [1:\ell_{\mathrm{SR}}]$. To determine the unknown matrices $\hat{\boldsymbol{B}}_1, \ldots, \hat{\boldsymbol{B}}_{\ell_{\mathrm{SR}}}$, we write the aforementioned system over $\mathbb{F}_q$, which reads as $\boldsymbol{s}_{\mathrm{ext}}^\top = \hat{\boldsymbol{H}}_{\mathrm{ext}} \hat{\boldsymbol{b}}^\top$, where $\boldsymbol{s}_{\mathrm{ext}} \in \mathbb{F}_q^{(n-k)m}$ denotes the expanded syndrome, and the matrix $\hat{\boldsymbol{H}}_{\mathrm{ext}} \in \mathbb{F}_q^{m(n-k)\times \eta_{\mathrm{SR}} s}$ depends only on $\boldsymbol{H}$ and $\hat{\boldsymbol{a}}$. Furthermore, the vector $\hat{\boldsymbol{b}} := [\hat{B}_{1,1,1}, \ldots, \hat{B}_{\ell_{\mathrm{SR}}, f_{\ell_{\mathrm{SR}}}, \eta_{\mathrm{SR}}}]$, where $\hat{B}_{ijr}$ is the entry in the j-th row and the r-th column of $\hat{\boldsymbol{B}}_i$, for $i \in [1:\ell_{\mathrm{SR}}]$.

The system $\boldsymbol{s}_{\mathrm{ext}}^\top = \hat{\boldsymbol{H}}_{\mathrm{ext}} \hat{\boldsymbol{b}}^\top$ has a unique solution if and only if $\mathrm{rk}_q(\hat{\boldsymbol{H}}_{\mathrm{ext}}) = \eta_{\mathrm{SR}} s$. To see that this condition is always fulfilled, assume $\mathrm{rk}_q(\hat{\boldsymbol{H}}_{\mathrm{ext}}) < \eta_{\mathrm{SR}} s$ and $\boldsymbol{s}_{\mathrm{ext}}^\top = \hat{\boldsymbol{H}}_{\mathrm{ext}} \hat{\boldsymbol{b}}^\top = \boldsymbol{0}$. It follows that there is a vector $\hat{\boldsymbol{b}} \neq \boldsymbol{0}$ such that $\hat{\boldsymbol{H}}_{\mathrm{ext}} \hat{\boldsymbol{b}}^\top = \boldsymbol{0}$, and thus, $\boldsymbol{H}(\hat{\boldsymbol{a}}\hat{\boldsymbol{B}})^\top = \boldsymbol{0}$. The latter equality implies that $\hat{\boldsymbol{a}}\hat{\boldsymbol{B}} \in \mathcal{C} \setminus \{\boldsymbol{0}\}$, which is a contradiction due to $\mathrm{wt}_{\mathrm{SR}}(\hat{\boldsymbol{a}}\hat{\boldsymbol{B}}) = s < d_{\min}^{\mathrm{SR}}$.

We observe that the most complex part of the algorithm is to solve an $m(n-k) \times \eta_{\mathrm{SR}} s$ linear system over $\mathbb{F}_q$ which is in $O(m^3(n-k)^3)$ since $\eta_{\mathrm{SR}} s \leq m(n-k)$. ■

Theorem 3.5 (Column Erasure Decoding)**.** *Let* $\boldsymbol{H} \in \mathbb{F}_{q^m}^{(n-k)\times n}$ *be a parity-check matrix of an* $[n, k, d_{\min}^{\mathrm{SR}}]_{\mathbb{F}_{q^m}}^{\mathrm{SR}}$ *code* $\mathcal{C}$*, let* $\boldsymbol{e} \in \mathbb{F}_{q^m}^{n}$ *be an error vector with* $\mathrm{wt}_{\mathrm{SR}}(\boldsymbol{e}) = t < d_{\min}^{\mathrm{SR}}$*, and let* $\boldsymbol{s}^\top = \boldsymbol{H}\boldsymbol{e}^\top$*. Furthermore, let*

$$\mathcal{F} = \mathcal{F}^{(\mathrm{R})} = \mathcal{F}_1^{(\mathrm{R})} \times \mathcal{F}_2^{(\mathrm{R})} \times \cdots \times \mathcal{F}_{\ell_{\mathrm{SR}}}^{(\mathrm{R})}$$

be any row super-support of $\boldsymbol{e}$*, where* $\mathcal{F}_i^{(\mathrm{R})}$ *has dimension* f_i*, for* $i \in [1:\ell_{\mathrm{SR}}]$*, and* $\sum_{i=1}^{\ell_{\mathrm{SR}}} f_i = s < d_{\min}^{\mathrm{SR}}$*. Then, the error* $\boldsymbol{e}$ *can be determined from* $\boldsymbol{H}$ *and* $\boldsymbol{s}$ *given the* row *super-support* $\mathcal{F} = \mathcal{F}^{(\mathrm{R})}$ *of the error vector* $\boldsymbol{e}$ *with complexity* $O((n-k)^3 m^2)$ *operations in* $\mathbb{F}_q$*.*

Proof. The error vector $\boldsymbol{e}$ can be decomposed into $\hat{\boldsymbol{a}}\hat{\boldsymbol{B}}$, where $\hat{\boldsymbol{B}}$ is a block-diagonal matrix containing bases of the row super-support entries $\mathcal{F}_i^{(\mathrm{R})}$. Since $\mathcal{F}$ has weight

$s < d_{\min}^{\mathrm{SR}}$, we have $\mathrm{rk}_{q^m}(\boldsymbol{H}\hat{\boldsymbol{B}}^\top) = s$, see Lemma 3.3. Then, $\boldsymbol{s}^\top = \boldsymbol{H}\boldsymbol{e}^\top = (\boldsymbol{H}\hat{\boldsymbol{B}}^\top)\hat{\boldsymbol{a}}^\top$, where $\hat{\boldsymbol{a}}$ is unknown, and $\boldsymbol{s}$, $\boldsymbol{H}$, and $\hat{\boldsymbol{B}}$ are known. This linear system of equations has a unique solution, and thus, we can uniquely determine $\hat{\boldsymbol{a}}$ and $\boldsymbol{e}$ using elementary matrix multiplication, Gaussian elimination, and polynomial multiplication algorithms. The multiplication $\boldsymbol{H}\hat{\boldsymbol{B}}^\top$ requires $O((n-k)s\eta_{\mathrm{SR}}m)$ operations in $\mathbb{F}_q$ since each row of $\hat{\boldsymbol{B}}$ has at most η_{SR} non-zero entries and solving the linear system $\left(\boldsymbol{H}\hat{\boldsymbol{B}}^\top\right)\hat{\boldsymbol{a}}^\top = \boldsymbol{s}^\top$ for $\hat{\boldsymbol{a}}$ requires $O(s^2(n-k))$ operations in $\mathbb{F}_{q^m}$. Furthermore, any operation in $\mathbb{F}_{q^m}$ requires $O(m^2)$ operations in $\mathbb{F}_q$. ■

Note that for a known *row* super-support, the result can also be derived from [129, Corollary 1].

Remark 3.1. *From the previous statement it follows that we can guarantee uniqueness of erasure decoding only if $s < d_{\min}^{\mathrm{SR}}$. However, erasure decoding can return a valid solution for $s \leq \min\left\{n-k, \lfloor \frac{m}{\eta_{\mathrm{SR}}}(n-k)\rfloor\right\}$, and the probability that this happens is high for many codes. Therefore, most generic Hamming and rank-metric decoding algorithms choose s as large as possible, i.e., $s = \min\left\{n-k, \lfloor \frac{m}{\eta_{\mathrm{SR}}}(n-k)\rfloor\right\}$. This choice is also a good heuristic for our proposed algorithm.*

3.1.2 A Non-Trivial Algorithm for Solving $\mathsf{SeaSD}_{\mathrm{SR}}$

From the previous section it follows that if a super-support $\mathcal{F} \supseteq \mathrm{supp}_{\mathrm{SR}}(\boldsymbol{e})$ of weight s such that $t \leq s < d_{\min}^{\mathrm{SR}}$ is known, then the error $\boldsymbol{e}$ can be retrieved from the syndrome vector $\boldsymbol{s}$ and the parity-check matrix $\boldsymbol{H}$ in polynomial time. In this section, we propose a Las Vegas-type algorithm[2] that determines such a super-support $\mathcal{F}$ by repeatedly sampling products of subspaces according to a designed probability mass function until a sample can be used for successful erasure decoding. This routine is shown in Algorithm 1, where we omit the prefixes *row* and *column* such that we treat both cases in a unified manner.

In the following, we specify the function $\mathsf{DrawRandomSupport}(s, t, \mu_{\mathrm{SR}}, m)$ and derive Theorem 3.10, which constitutes the main statement of this section. This theorem states that Algorithm 1 always finds a solution to $\mathsf{SeaSD}_{\mathrm{SR}}$ and provides both upper and lower bounds on the complexity of the algorithm. Since the proof is quite technical, we first provide some interim results.

[2] A Las Vegas algorithm is a randomized algorithm that always outputs a correct result. However, the running time can vary from one run to another.

Algorithm 1: Generic Sum-Rank Syndrome Decoder

```
Input  : Parity-check matrix $\boldsymbol{H} \in \mathbb{F}_{q^m}^{(n-k)\times n}$
         Syndrome vector $\boldsymbol{s} \in \mathbb{F}_{q^m}^{n-k}$
         Integers $t$ and $s$
Output: Vector $\boldsymbol{e}' \in \mathbb{F}_{q^m}^{n}$
$\mu_{\mathrm{SR}} \leftarrow \min\{m, \eta_{\mathrm{SR}}\}$
$\boldsymbol{e}' \leftarrow \boldsymbol{0}$
while $\mathrm{wt}_{\mathrm{SR}}(\boldsymbol{e}') > t \vee \boldsymbol{s} \neq \boldsymbol{e}'\boldsymbol{H}^\top$ do
    $\mathcal{F} \leftarrow$ DrawRandomSupport$(s, t, \mu_{\mathrm{SR}}, m)$ (specified in Algorithm 3)
    if $\mu_{\mathrm{SR}} = m$ then
        $\boldsymbol{e}' \leftarrow$ row erasure decoding w.r.t. $\mathcal{F}$, $\boldsymbol{H}$, $\boldsymbol{s}$ (cf. Theorem 3.4)
    else
        $\boldsymbol{e}' \leftarrow$ column erasure decoding w.r.t. $\mathcal{F}$, $\boldsymbol{H}$, $\boldsymbol{s}$ (cf. Theorem 3.5)
return $\boldsymbol{e}'$
```

The proposed super-support drawing algorithm DrawRandomSupport$(s, t, \mu_{\mathrm{SR}}, m)$ is designed such that it minimizes the worst-case expected number of iterations

$$\max_{\substack{\boldsymbol{e}\in\mathbb{F}_{q^m}^n:\\ \mathrm{wt}_{\mathrm{SR}}(\boldsymbol{e})=t}} \mathbb{E}[\#\text{iterations}] = \max_{\substack{\boldsymbol{e}\in\mathbb{F}_{q^m}^n:\\ \mathrm{wt}_{\mathrm{SR}}(\boldsymbol{e})=t}} \left\{\Pr(\mathrm{supp}_{\mathrm{SR}}(\boldsymbol{e}) \subseteq \mathcal{F})^{-1}\right\}.$$

Thus, the proposed algorithm first samples $\boldsymbol{f}$ from $\mathcal{T}_{s,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}}$ according to a designed probability distribution $\tilde{p}_{\boldsymbol{f}}$, and then draws the support $\mathcal{F}$ uniformly at random from $\Xi_{q,\mu_{\mathrm{SR}}}(\boldsymbol{f})$.

Lemma 3.6. *Let* $\boldsymbol{e}$ *be a vector in* $\mathbb{F}_{q^m}^n$*, where* $\mathrm{wt}_{\mathrm{SR}}(\boldsymbol{e}) = t$*, let* $\boldsymbol{t}_{\boldsymbol{e}} = [t_1, \ldots, t_{\ell_{\mathrm{SR}}}]$ *be the weight decomposition of* $\boldsymbol{e}$*, and let* $\boldsymbol{f} \in \mathcal{T}_{s,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}}$*. If* $\mathcal{F}$ *is sampled uniformly at random from* $\Xi_{q,\mu_{\mathrm{SR}}}(\boldsymbol{f})$*, then the probability*

$$\Pr(\mathrm{supp}_{\mathrm{SR}}(\boldsymbol{e}) \subseteq \mathcal{F} \mid \boldsymbol{f}) = \varrho_{q,\mu_{\mathrm{SR}}}(\boldsymbol{f}, \boldsymbol{t}_{\boldsymbol{e}}) \coloneqq \prod_{i=1}^{\ell_{\mathrm{SR}}} \begin{bmatrix} f_i \\ t_i \end{bmatrix}_q \begin{bmatrix} \mu_{\mathrm{SR}} \\ t_i \end{bmatrix}_q^{-1}. \tag{3.1}$$

Furthermore, this probability can be bounded by

$$4^{-\ell_{\mathrm{SR}}} q^{-\sum_{i=1}^{\ell_{\mathrm{SR}}} t_i(\mu_{\mathrm{SR}}-f_i)} \leq \varrho_{q,\mu_{\mathrm{SR}}}(\boldsymbol{f}, \boldsymbol{t}_{\boldsymbol{e}}) \leq 4^{\ell_{\mathrm{SR}}} q^{-\sum_{i=1}^{\ell_{\mathrm{SR}}} t_i(\mu_{\mathrm{SR}}-f_i)}.$$

Proof. Since $\mathcal{F} = \mathcal{F}_1 \times \cdots \times \mathcal{F}_{\ell_{\mathrm{SR}}}$ is sampled uniformly, each subspace $\mathcal{F}_i$ is sampled statistically independently and uniformly from the set of all f_i-dimensional subspaces of $\mathbb{F}_q^{\mu_{\mathrm{SR}}}$ or $\mathbb{F}_{q^{\mu_{\mathrm{SR}}}}$, respectively. This implies that the probability

$$\Pr(\mathrm{supp}_{\mathrm{SR}}(\boldsymbol{e}) \subseteq \mathcal{F} \mid \boldsymbol{f}) = \prod_{i=1}^{\ell_{\mathrm{SR}}} \Pr(\mathrm{supp}_{\mathrm{R}}(\boldsymbol{e}_i) \subseteq \mathcal{F}_i \mid f_i),$$

where

$$\Pr(\mathrm{supp}_{\mathrm{R}}(\boldsymbol{e}_i) \subseteq \mathcal{F}_i \mid f_i) = \begin{bmatrix} \mu_{\mathrm{SR}} - t_i \\ f_i - t_i \end{bmatrix}_q \begin{bmatrix} \mu_{\mathrm{SR}} \\ f_i \end{bmatrix}_q^{-1} = \begin{bmatrix} f_i \\ t_i \end{bmatrix}_q \begin{bmatrix} \mu_{\mathrm{SR}} \\ t_i \end{bmatrix}_q^{-1},$$

for $i \in [1:\ell_{\mathrm{SR}}]$. Using the bounds on the Gaussian coefficients stated in (2.2) we obtain the upper and the lower bound of $\Pr(\mathrm{supp}_{\mathrm{SR}}(\boldsymbol{e}) \subseteq \mathcal{F} \mid \boldsymbol{f})$. ■

Using Lemma 3.6, the worst-case number of iterations of the proposed algorithm for a designed probability mass function $\tilde{p}_{\boldsymbol{f}}$ evaluates to

$$\max_{\substack{\boldsymbol{e}\in\mathbb{F}_{q^m}^n : \\ \mathrm{wt}_{\mathrm{SR}}(\boldsymbol{e})=t}} \mathbb{E}[\#\text{iterations}] = \max_{\boldsymbol{t}\in\mathcal{T}_{t,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}}} \left(\sum_{\boldsymbol{f}\in\mathcal{T}_{s,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}}} \tilde{p}_{\boldsymbol{f}}\, \varrho_{q,\mu_{\mathrm{SR}}}(\boldsymbol{f},\boldsymbol{t}) \right)^{-1}. \tag{3.2}$$

Minimizing (3.2) w.r.t. $\tilde{p}_{\boldsymbol{f}}$ on $\mathcal{T}_{s,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}}$ can be numerically performed using linear programming algorithms for small parameters ℓ_{SR}, μ_{SR}, and s. One should note that the number of optimization variables $|\mathcal{T}_{s,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}}|$ grows fast in ℓ_{SR}, μ_{SR}, and s and makes the optimization problem complex, see Appendix B.2 for a detailed discussion. For our analysis, the following suboptimal solution is sufficient.[3]

First, we propose a randomized[4] mapping $\mathsf{Scomp}_{\mu_{\mathrm{SR}}} : \mathcal{T}_{t,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}} \times \mathbb{Z} \to \mathcal{T}_{s,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}}$ that maximizes $\varrho_{q,\mu_{\mathrm{SR}}}(\mathsf{Scomp}_{\mu_{\mathrm{SR}}}(\boldsymbol{t},s),\boldsymbol{t})$ for a given $\boldsymbol{t} \in \mathcal{T}_{t,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}}$. Second, instead of picking $\boldsymbol{f} \in \mathcal{T}_{s,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}}$ directly, we sample $\boldsymbol{t} \in \mathcal{T}_{t,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}}$ at random according to the designed distribution $p_{\boldsymbol{t}}$ and assign $\boldsymbol{f} = \mathsf{Scomp}_{\mu_{\mathrm{SR}}}(\boldsymbol{t},s)$. It follows that for a fixed vector $\boldsymbol{e}$, the probability

$$\Pr(\mathrm{supp}_{\mathrm{SR}}(\boldsymbol{e}) \subseteq \mathcal{F}) = \sum_{\boldsymbol{f}\in\mathcal{T}_{s,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}}} \tilde{p}_{\boldsymbol{f}}\, \varrho_{q,\mu_{\mathrm{SR}}}(\boldsymbol{f},\boldsymbol{t}_{\boldsymbol{e}}) \geq p_{\boldsymbol{t}_{\boldsymbol{e}}} \cdot \varrho_{q,\mu_{\mathrm{SR}}}(\mathsf{Scomp}_{\mu_{\mathrm{SR}}}(\boldsymbol{t}_{\boldsymbol{e}},s),\boldsymbol{t}_{\boldsymbol{e}}),$$

where $\boldsymbol{t}_{\boldsymbol{e}}$ is the weight decomposition of the error. Third, we minimize the following

[3] We show that the optimal solution is close to the described suboptimal approach in Section 3.1.3.

[4] Randomized mapping means that in case there are multiple possible outputs, one output is selected uniformly at random.

upper bound

$$\max_{\substack{\boldsymbol{e}\in\mathbb{F}_{q^m}^n:\\ \mathrm{wt}_{\mathrm{SR}}(\boldsymbol{e})=t}} \mathbb{E}[\#\text{iterations}] \leq \max_{\boldsymbol{t}\in\mathcal{T}_{t,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}}} \left\{\left(p_{\boldsymbol{t}}\cdot \varrho_{q,\mu_{\mathrm{SR}}}(\mathsf{Scomp}_{\mu_{\mathrm{SR}}}(\boldsymbol{t},s),\boldsymbol{t})\right)^{-1}\right\}$$

instead of (3.2). This suboptimal solution comes at the disadvantage of a slightly smaller success probability but enables an efficient support drawing algorithm whose complexity can be lower and upper bounded.

The mapping $\mathsf{Scomp}_{\mu_{\mathrm{SR}}}$ is shown in Algorithm 2, where the randomized step in Line 4 ensures that there is no bias in preferring certain positions. This randomization seems to be especially important for instances with a large ℓ_{SR}. However, we are not able to properly incorporate the randomness in our complexity analysis of Algorithm 1. Therefore, we use the deterministic quantity $\varrho_{q,\mu_{\mathrm{SR}},s}(\boldsymbol{t})$ instead, which is given by

$$\varrho_{q,\mu_{\mathrm{SR}},s}(\boldsymbol{t}) \coloneqq \varrho_{q,\mu_{\mathrm{SR}}}(\mathsf{Scomp}_{\mu_{\mathrm{SR}}}(\boldsymbol{t},s),\boldsymbol{t})$$

for all $\boldsymbol{t}\in\mathcal{T}_{t,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}}$ and a fixed $s \geq t$.

Algorithm 2: $\mathsf{Scomp}_{\mu_{\mathrm{SR}}}$

```
Input  : Vector $\boldsymbol{t} \in \mathcal{T}_{t,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}}$
         Non-negative integer $s$
Output: Vector $\boldsymbol{f} \in \mathcal{T}_{s,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}}$
1 $\boldsymbol{f} = [f_1,\dots,f_{\ell_{\mathrm{SR}}}] \leftarrow \boldsymbol{t}$
2 $\delta \leftarrow s - t$
3 while $\delta > 0$ do
4     $h \xleftarrow{\$} \left\{h' : f_{h'} = \min_i \left\{f_i : t_i = \max_j\{t_j : f_j \neq \mu_{\mathrm{SR}}\}\right\}\right\}$
5     $f_h \leftarrow f_h + 1$
6     $\delta \leftarrow \delta - 1$
7 return $\boldsymbol{f}$
```

The following lemma proves that $\boldsymbol{f} = \mathsf{Scomp}_{\mu_{\mathrm{SR}}}(\boldsymbol{t},s)$ maximizes $\varrho_{q,\mu_{\mathrm{SR}}}(\boldsymbol{f},\boldsymbol{t})$ among all $\boldsymbol{f}\in\mathcal{T}_{s,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}}$.

Lemma 3.7. *Let t, s, ℓ_{SR}, and μ_{SR} be integers such that $t \leq s \leq \ell_{\mathrm{SR}}\mu_{\mathrm{SR}}$, and let $\boldsymbol{t}$ be a vector in $\mathcal{T}_{t,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}}$. Then, the vector $\boldsymbol{f} = \mathsf{Scomp}_{\mu_{\mathrm{SR}}}(\boldsymbol{t},s)$ maximizes $\varrho_{q,\mu_{\mathrm{SR}}}(\boldsymbol{f},\boldsymbol{t})$, where $\mathsf{Scomp}_{\mu_{\mathrm{SR}}}$ is defined in Algorithm 2.*

Proof. Since the denominator of (3.1) does not depend on $\boldsymbol{f}$, it is sufficient to prove that for a fixed vector $\boldsymbol{t}$, the vector $\boldsymbol{f} = \mathsf{Scomp}_{\mu_{\mathrm{SR}}}(\boldsymbol{t}, s)$ maximizes

$$\prod_{i=1}^{\ell_{\mathrm{SR}}} \begin{bmatrix} f_i \\ t_i \end{bmatrix}_q . \tag{3.3}$$

To see that this holds, assign $\boldsymbol{f} = \boldsymbol{t}$ and increase the entries of $\boldsymbol{f}$ until $\sum_{i=1}^{\ell_{\mathrm{SR}}} f_i = s$. If $f_i \geq t_i$ and f_i is increased by one, the quantity (3.3) is increased by a factor

$$\begin{bmatrix} f_i+1 \\ t_i \end{bmatrix}_q \begin{bmatrix} f_i \\ t_i \end{bmatrix}_q^{-1} = \prod_{\mu=1}^{t_i} \frac{\left(\frac{q^{f_i+2-\mu}-1}{q^\mu-1}\right)}{\left(\frac{q^{f_i+1-\mu}-1}{q^\mu-1}\right)} = \frac{q^{f_i+1}-1}{q^{f_i-t_i+1}-1} .$$

Since $(q^{f_i+1}-1)(q^{f_i-t_i+1}-1)^{-1}$ is monotonically decreasing in f_i for a given integer t_i, it holds that

$$q^{t_i} < \frac{q^{f_i+1}-1}{q^{f_i-t_i+1}-1} < q^{t_i+1}. \tag{3.4}$$

Furthermore, for $f_i > f_j$ and $t_i = t_j > 0$, we have

$$\begin{aligned} &\frac{q^{f_i+1}-1}{q^{f_i-t_i+1}-1} < \frac{q^{f_j+1}-1}{q^{f_j-t_j+1}-1} \\ \iff &\begin{bmatrix} f_i+1 \\ t_i \end{bmatrix}_q \begin{bmatrix} f_i \\ t_i \end{bmatrix}_q^{-1} < \begin{bmatrix} f_j+1 \\ t_j \end{bmatrix}_q \begin{bmatrix} f_j \\ t_j \end{bmatrix}_q^{-1} . \end{aligned} \tag{3.5}$$

From (3.4) and (3.5) follows that increasing any position i with smallest f_i among all positions with largest t_i leads to the largest increase of (3.3). This approach of increasing the entries is optimal since this choice also maximizes the increase of (3.3) in the following steps. ■

The Support-Drawing Algorithm

A support-drawing routine that implements the described ideas is shown in Algorithm 3, where

$$Q_{t,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}} := \sum_{\boldsymbol{t} \in \mathcal{T}_{t,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}}} \varrho_{q,\mu_{\mathrm{SR}},s}(\boldsymbol{t})^{-1}. \tag{3.6}$$

Note that an efficient implement of Line 2 in Algorithm 3 is derived in [155, Sec. V].

In the following proposition, we derive upper and lower bounds on the expected number of iterations, i.e., how often we expect to execute Algorithm 3 together with

Algorithm 3: DrawRandomSupport

```
Input  : Non-negative integers $t, s, \mu_{\mathrm{SR}}, m$
Output: Product of subspaces $\mathcal{F}$ of weight $s$
Draw $\boldsymbol{t} \in \mathcal{T}_{t,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}}$ according to the distribution
  $p_{\boldsymbol{t}} := \varrho_{q,\mu_{\mathrm{SR}},s}(\boldsymbol{t})^{-1} Q^{-1}_{t,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}} \quad \forall \boldsymbol{t} \in \mathcal{T}_{t,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}}$, where $Q_{t,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}}$ is defined as in (3.6)
$\boldsymbol{f} \leftarrow \mathsf{Scomp}_{\mu_{\mathrm{SR}}}(\boldsymbol{t}, s)$
if $\mu_{\mathrm{SR}} = m$ then
    $\mathcal{F} \xleftarrow{\$} \Xi^{(\mathrm{C})}_{q,\mu_{\mathrm{SR}}}(\boldsymbol{f})$
else
    $\mathcal{F} \xleftarrow{\$} \Xi^{(\mathrm{R})}_{q,\mu_{\mathrm{SR}}}(\boldsymbol{f})$
return $\mathcal{F}$
```

erasure decoding to solve $\mathsf{SeaSD}_{\mathrm{SR}}$.

Proposition 3.8. *Let $\boldsymbol{e}$ be a vector in $\mathbb{F}_{q^m}^n$ that has a support $\mathrm{supp}_{\mathrm{SR}}(\boldsymbol{e})$, a sum-rank weight $\mathrm{wt}_{\mathrm{SR}}(\boldsymbol{e}) = t$, and a weight decomposition $\boldsymbol{t_e}$. Let s be an integer such that $t \leq s \leq \ell_{\mathrm{SR}}\mu_{\mathrm{SR}}$ and let $\mathcal{F}$ be a super-support obtained by Algorithm 3. Then, the inverse of the probability that $\mathcal{F}$ is a super-support of $\mathrm{supp}_{\mathrm{SR}}(\boldsymbol{e})$ can be bounded by*

$$|\mathcal{T}_{t,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}}|^{-1} Q_{t,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}} \leq \Pr(\mathrm{supp}_{\mathrm{SR}}(\boldsymbol{e}) \subseteq \mathcal{F})^{-1} \leq Q_{t,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}} ,$$

where $Q_{t,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}}$ is defined as in (3.6)*.*

Proof. Let $\boldsymbol{t}$ be a random variable with the probability distribution $p_{\boldsymbol{t}}$ and let $\tilde{p}_{\boldsymbol{f}}$ denote the probability mass function of $\boldsymbol{f} = \mathsf{Scomp}_{\mu_{\mathrm{SR}}}(\boldsymbol{t}, s)$. From (3.2) follows that

$$\begin{aligned}\Pr(\mathrm{supp}_{\mathrm{SR}}(\boldsymbol{e}) \subseteq \mathcal{F}) &= \sum_{\boldsymbol{t} \in \mathcal{T}_{s,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}}} p_{\boldsymbol{t}} \varrho_{q,\mu_{\mathrm{SR}}}(\mathsf{Scomp}_{\mu_{\mathrm{SR}}}(\boldsymbol{t}, s), \boldsymbol{t_e}) \\ &\geq p_{\boldsymbol{t_e}} \varrho_{q,\mu_{\mathrm{SR}}}(\mathsf{Scomp}_{\mu_{\mathrm{SR}}}(\boldsymbol{t_e}, s), \boldsymbol{t_e}) = Q^{-1}_{t,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}},\end{aligned}$$

which means $\Pr(\mathrm{supp}_{\mathrm{SR}}(\boldsymbol{e}) \subseteq \mathcal{F})^{-1} \leq Q_{t,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}}$. Furthermore, from Lemma 3.7 follows that

$$\varrho_{q,\mu_{\mathrm{SR}}}(\mathsf{Scomp}_{\mu_{\mathrm{SR}}}(\boldsymbol{t}, s), \boldsymbol{t_e}) \leq \varrho_{q,\mu_{\mathrm{SR}}}(\mathsf{Scomp}_{\mu_{\mathrm{SR}}}(\boldsymbol{t}, s), \boldsymbol{t}) = \varrho_{q,\mu_{\mathrm{SR}},s}(\boldsymbol{t}),$$

and therefore,

$$\begin{aligned}\Pr(\mathrm{supp}_{\mathrm{SR}}(\boldsymbol{e}) \subseteq \mathcal{F}) &= \sum_{\boldsymbol{t}\in\mathcal{T}_{s,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}}} p_{\boldsymbol{t}}\varrho_{q,\mu_{\mathrm{SR}}}(\mathsf{Scomp}_{\mu_{\mathrm{SR}}}(\boldsymbol{t},s),\boldsymbol{t}_{\boldsymbol{e}})\\ &\leq \sum_{\boldsymbol{t}\in\mathcal{T}_{s,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}}} p_{\boldsymbol{t}}\varrho_{q,\mu_{\mathrm{SR}},s}(\boldsymbol{t}) = \sum_{\boldsymbol{t}\in\mathcal{T}_{s,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}}} Q^{-1}_{t,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}},\end{aligned}$$

where $\sum_{\boldsymbol{t}\in\mathcal{T}_{s,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}}} Q^{-1}_{t,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}} = |\mathcal{T}_{t,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}}|Q^{-1}_{t,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}}$. ■

Note that an efficient algorithm for computing the bounds which are derived in Proposition 3.8 is proposed in [155, Sec. V].

A Simple Upper Bound on the Success Probability

In the following proposition, we state a simple upper bound on $Q_{t,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}}$.

Proposition 3.9. *For any $t \leq s \leq \ell_{\mathrm{SR}}\mu_{\mathrm{SR}}$, we have*

$$\max_{\boldsymbol{t}\in\mathcal{T}_{t,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}}} \varrho_{q,\mu_{\mathrm{SR}},s}(\boldsymbol{t})^{-1} \leq 4^{\ell_{\mathrm{SR}}} q^{t(\mu_{\mathrm{SR}}-\frac{s}{\ell_{\mathrm{SR}}})},$$

and

$$Q_{t,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}} \leq \tbinom{\ell_{\mathrm{SR}}+t-1}{\ell_{\mathrm{SR}}-1} 4^{\ell_{\mathrm{SR}}} q^{t(\mu_{\mathrm{SR}}-\frac{s}{\ell_{\mathrm{SR}}})}.$$

Proof. By Lemma 3.6, we have

$$\begin{aligned}\max_{\boldsymbol{t}\in\mathcal{T}_{t,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}}} \varrho_{q,\mu_{\mathrm{SR}},s}(\boldsymbol{t})^{-1} &\leq 4^{\ell_{\mathrm{SR}}} \max_{\boldsymbol{t}\in\mathcal{T}_{t,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}}} \left\{ q^{\sum_{i=1}^{\ell_{\mathrm{SR}}} t_i(\mu_{\mathrm{SR}}-f_i)} \mid \boldsymbol{f} = \mathsf{Scomp}_{\mu_{\mathrm{SR}}}(\boldsymbol{t},s)\right\}\\ &= 4^{\ell_{\mathrm{SR}}} q^{t\mu_{\mathrm{SR}}} q^{-\min_{\boldsymbol{t}\in\mathcal{T}_{t,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}}}\left\{\sum_{i=1}^{\ell_{\mathrm{SR}}} t_i f_i \mid \boldsymbol{f}=\mathsf{Scomp}_{\mu_{\mathrm{SR}}}(\boldsymbol{t},s)\right\}},\end{aligned}$$

where the last exponent satisfies

$$\min_{\boldsymbol{t}\in\mathcal{T}_{t,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}}} \left\{ \sum_{i=1}^{\ell_{\mathrm{SR}}} t_i f_i \mid \boldsymbol{f} = \mathsf{Scomp}_{\mu_{\mathrm{SR}}}(\boldsymbol{t},s)\right\} \geq \frac{ts}{\ell_{\mathrm{SR}}}.$$

We prove this by relaxing the variables to real numbers and considering only the ordered vectors $\boldsymbol{t}$. Define the set

$$\mathcal{T}^{(\mathbb{R},\mathsf{ord})}_{t,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}} := \left\{ \boldsymbol{t} \in \mathbb{R}^{\ell_{\mathrm{SR}}}_{\geq 0} \;:\; \sum_{i=1}^{\ell_{\mathrm{SR}}} t_i = t,\; t_i \leq \mu_{\mathrm{SR}},\; t_1 \geq t_2 \geq \cdots \geq t_{\ell_{\mathrm{SR}}} \right\}$$

and the mapping

$$\begin{aligned}\mathsf{Scomp}^{(\mathbb{R})}_{\mu_{\mathrm{SR}}} \;:\; \mathcal{T}^{(\mathbb{R},\mathsf{ord})}_{t,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}} &\to \mathbb{R}^{\ell_{\mathrm{SR}}}_{\geq 0}, \\ \boldsymbol{t} \mapsto \Big[&\underbrace{\mu_{\mathrm{SR}},\ldots,\mu_{\mathrm{SR}}}_{h \text{ times}}, \underbrace{t_{h+1}+\xi+1,\ldots,t_{h+g}+\xi+1}_{g \text{ times}}, \\ &\underbrace{t_{h+g+1}+\xi+\delta,\ldots,t_{h+f}+\xi+\delta}_{f-g \text{ times}}, t_{h+f+1},\ldots,t_{\ell_{\mathrm{SR}}}\Big],\end{aligned}$$

where

$$h := \max\left\{h' \in [0\,{:}\,\ell_{\mathrm{SR}}] : \sum_{i=1}^{h'}(\mu_{\mathrm{SR}} - t_i) \leq s - t, t_{h'} > t_{h'+1}, t_0 := \mu_{\mathrm{SR}}, t_{\ell_{\mathrm{SR}}+1} := -1\right\},$$
$$f := \max\{f' \in \{1,\ldots,\ell_{\mathrm{SR}}\} \;:\; t_{f'} = t_{h+1}\} - h,$$

$s_{\mathrm{rem}} := s - t - \sum_{i=1}^{h}(\mu_{\mathrm{SR}} - t_i)$, $\xi := \left\lfloor\frac{s_{\mathrm{rem}}}{f}\right\rfloor$, $g := \lfloor s_{\mathrm{rem}}\rfloor - \xi f$, and $\delta := \frac{s_{\mathrm{rem}} - \lfloor s_{\mathrm{rem}}\rfloor}{f-g}$.

Note that $\mathsf{Scomp}^{(\mathbb{R})}_{\mu_{\mathrm{SR}}}$ agrees with a deterministic variant[5] of $\mathsf{Scomp}_{\mu_{\mathrm{SR}}}$ on $\mathcal{T}^{(\mathbb{R},\mathsf{ord})}_{t,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}} \cap \mathbb{Z}^{\ell_{\mathrm{SR}}}$. Since $\sum_{i=1}^{\ell_{\mathrm{SR}}} t_i f_i|_{\boldsymbol{f}=\mathsf{Scomp}_{\mu_{\mathrm{SR}}}(\boldsymbol{t},s)}$ is independent of the ordering of the entries of $\boldsymbol{t}$ and the set of sorted elements of $\mathcal{T}_{t,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}}$ are subset of $\mathcal{T}^{(\mathbb{R},\mathsf{ord})}_{t,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}}$, we have

$$\min_{\boldsymbol{t}\in\mathcal{T}_{t,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}}} \left\{\sum_{i=1}^{\ell_{\mathrm{SR}}} t_i f_i \mid \boldsymbol{f} = \mathsf{Scomp}_{\mu_{\mathrm{SR}}}(\boldsymbol{t},s)\right\} \geq \min_{\boldsymbol{t}\in\mathcal{T}^{(\mathbb{R},\mathsf{ord})}_{t,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}}} \left\{\sum_{i=1}^{\ell_{\mathrm{SR}}} t_i f_i \mid \boldsymbol{f} = \mathsf{Scomp}^{(\mathbb{R})}_{\mu_{\mathrm{SR}}}(\boldsymbol{t},s)\right\}.$$

For $\boldsymbol{t} \in \mathcal{T}^{(\mathbb{R},\mathsf{ord})}_{t,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}}$ and $\boldsymbol{f} = \mathsf{Scomp}^{(\mathbb{R})}_{\mu_{\mathrm{SR}}}(\boldsymbol{t},s)$, we have

$$\sum_{i=1}^{\ell_{\mathrm{SR}}} t_i f_i = \mu_{\mathrm{SR}}\sum_{i=1}^{h} t_i + \sum_{i=h+1}^{h+g}(t_i+\xi+1)t_i + \sum_{i=h+g+1}^{h+f}(t_i+\xi+\delta)t_i + \sum_{i=h+f+1}^{\ell_{\mathrm{SR}}} t_i^2. \tag{3.7}$$

Since $t_{i+1} \leq t_i + \xi + \delta \leq t_i + \xi + 1 \leq \mu_{\mathrm{SR}}$, it follows that (3.7) is minimized by a sequence in $\mathcal{T}^{(\mathbb{R},\mathsf{ord})}_{t,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}}$ with smallest-possible h. Among these sequences with minimal h, it is minimized by sequence with largest f. Since t_i are non-increasing, these requirements directly imply that (3.7) is minimized for

$$\boldsymbol{t} = \left[\tfrac{t}{\ell_{\mathrm{SR}}},\ldots,\tfrac{t}{\ell_{\mathrm{SR}}}\right],$$

[5] The outputs are equal if we choose $j \leftarrow \min\big\{j \;:\; f_j = \max\{f_i : f_i < \mu_{\mathrm{SR}}, i \in [1\,{:}\,\ell_{\mathrm{SR}}]\}\big\}$ instead of a random choice in Line 4 of Algorithm 2.

for which we have

$$\sum_{i=1}^{\ell_{\mathrm{SR}}} t_i f_i = \frac{t}{\ell_{\mathrm{SR}}} \sum_{i=1}^{\ell_{\mathrm{SR}}} f_i = \frac{ts}{\ell_{\mathrm{SR}}}.$$

This proves the first claim. We get the bound on $Q_{t,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}}$ by

$$\begin{aligned} Q_{t,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}} =& \sum_{\boldsymbol{t} \in \mathcal{T}_{t,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}}} \frac{1}{\varrho_{q,\mu_{\mathrm{SR}},s}(\boldsymbol{t})} \\ \leq& |\mathcal{T}_{t,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}}| \max_{\boldsymbol{t} \in \mathcal{T}_{t,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}}} \frac{1}{\varrho_{q,\mu_{\mathrm{SR}},s}(\boldsymbol{t})} \\ \leq& \binom{\ell_{\mathrm{SR}}+t-1}{\ell_{\mathrm{SR}}-1} 4^{\ell_{\mathrm{SR}}} q^{t(\mu_{\mathrm{SR}} - \frac{s}{\ell_{\mathrm{SR}}})}. \end{aligned}$$ ■

In the following theorem, we derive bounds on the average complexity of Algorithm 1.

Theorem 3.10. *Let $\boldsymbol{H} \in \mathbb{F}_{q^m}^{(n-k)\times n}$ be a parity-check matrix of an $[n,k,d_{\min}^{\mathrm{SR}}]_{\mathbb{F}_{q^m}}^{\mathrm{SR}}$ code $\mathcal{C}$, let $\boldsymbol{e}$ be a vector in $\mathbb{F}_{q^m}^n$ with $\mathrm{wt}_{\mathrm{SR}}(\boldsymbol{e}) = t$, let $\boldsymbol{s} = \boldsymbol{e}\boldsymbol{H}^\top$, and let s be an integer such that $t \leq s < d_{\min}^{\mathrm{SR}}$. Then, Algorithm 1 returns a vector $\boldsymbol{e}'$ such that $\mathrm{wt}_{\mathrm{SR}}(\boldsymbol{e}') \leq t$ and $\boldsymbol{s} = \boldsymbol{e}'\boldsymbol{H}^\top$ with an average complexity of W_{SR} operations in $\mathbb{F}_q$. The average complexity is bounded by $W_{\mathrm{SR}}^{(\mathrm{LB})} \leq W_{\mathrm{SR}} \leq W_{\mathrm{SR}}^{(\mathrm{UB})} \leq W_{\mathrm{SR}}^{(\mathrm{UB,simple})}$, where*

$$\begin{aligned} W_{\mathrm{SR}}^{(\mathrm{LB})} &:= |\mathcal{T}_{t,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}}|^{-1} Q_{t,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}}, \\ W_{\mathrm{SR}}^{(\mathrm{UB})} &:= n^3 m^3 Q_{t,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}}, \text{ and} \\ W_{\mathrm{SR}}^{(\mathrm{UB,simple})} &:= n^3 m^3 \binom{\ell_{\mathrm{SR}}+t-1}{\ell_{\mathrm{SR}}-1} 4^{\ell_{\mathrm{SR}}} q^{t(\mu_{\mathrm{SR}} - \frac{s}{\ell_{\mathrm{SR}}})}, \end{aligned}$$

for $\mu_{\mathrm{SR}} = \min\{\eta_{\mathrm{SR}}, m\}$.

Proof. Because the probability that a super-support of $\boldsymbol{e}$ is drawn is larger than 0 and erasure decoding has a unique result for a super-support of weight $s < d_{\min}^{\mathrm{SR}}$, see Theorem 3.4 and Theorem 3.5, it follows that Algorithm 1 always outputs a vector $\boldsymbol{e}'$ such that $\mathrm{wt}_{\mathrm{SR}}(\boldsymbol{e}') \leq t$ and $\boldsymbol{s} = \boldsymbol{e}\boldsymbol{H}^\top$.

The average complexity W_{SR} is the complexity of one iteration W_{Iter} times the average number of iterations. From Theorem 3.4, Theorem 3.5, and [155, Prop. 23] follows that the quantity W_{Iter} is bounded by $1 \leq W_{\mathrm{Iter}} \leq O\left(n^3 m^3\right)$ operations in $\mathbb{F}_q$. Furthermore, the applied bounds on the expected number of iterations are derived in Proposition 3.8 and Proposition 3.9. ■

3.1.3 Comparison to Other Decoding Algorithms

To benchmark the performance of Algorithm 1, we compare it to other decoding strategies, as well as to known generic decoders in the cases $\ell_{\mathrm{SR}} = 1$ and $\ell_{\mathrm{SR}} = n$, respectively.

A naïve decoding strategy is to brute-force all vectors $\boldsymbol{e}$ with $\mathrm{wt}_{\mathrm{SR}}(\boldsymbol{e}) = t$. Since the number of such vectors is $\mathcal{N}_{q,\eta_{\mathrm{SR}},m}(t, \ell_{\mathrm{SR}}) \leq \binom{\ell_{\mathrm{SR}}+t-1}{\ell_{\mathrm{SR}}-1} 4^{\ell_{\mathrm{SR}}} q^{t(m+\eta_{\mathrm{SR}}-\frac{t}{\ell_{\mathrm{SR}}})}$ and checking if $\boldsymbol{e}$ fulfills the parity-check equations requires at most $O(n^3m^3)$ operations in $\mathbb{F}_q$, the average complexity of this approach is at most

$$W_{\mathrm{Error}} = n^3m^3\mathcal{N}_{q,\eta_{\mathrm{SR}},m}(t, \ell_{\mathrm{SR}}) \leq n^3m^3\binom{\ell_{\mathrm{SR}}+t-1}{\ell_{\mathrm{SR}}-1} 4^{\ell_{\mathrm{SR}}} q^{t(m+\eta_{\mathrm{SR}}-\frac{t}{\ell_{\mathrm{SR}}})}$$

operation is $\mathbb{F}_q$.

Our proposed decoding algorithm uses an efficient but suboptimal support-drawing algorithm. For small parameters, one can replace this suboptimal algorithm by an optimal support-drawing algorithm based on linear programming, see Appendix B.2. The complexity of this approach is denoted by $W_{\mathrm{SR}}^{(\mathrm{optimal})}$.

In the case $\ell_{\mathrm{SR}} = n$, the sum-rank metric and the Hamming metric coincide, and Prange's ISD algorithm [134] can be applied.[6] Assuming that there is only one solution to the problem, the algorithm has an average complexity of approximately W_{Prange} operations in $\mathbb{F}_q$, where $t \leq s \leq n-k$, see (2.5) in Section 2.2.4. Note that for $\ell_{\mathrm{SR}} = n$, the set $\mathcal{T}_{t,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}}$ contains all binary vectors of Hamming weight t, and therefore, it holds that $|\mathcal{T}_{t,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}}| = \binom{n}{t}$. Furthermore, for $\boldsymbol{t} \in \mathcal{T}_{t,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}}$ and $t \leq s \leq n-k$, Algorithm 2 outputs a random binary vector $\boldsymbol{f}$ of Hamming weight s whose Hamming support is a super-support of $\boldsymbol{t}$. It follows that Algorithm 1 repeatedly samples a binary vector of length n and of Hamming weight s. It succeeds if and only if the Hamming support of the sampled vector is a Hamming super-support of an error $\boldsymbol{e}'$ such that $\mathrm{wt}(\boldsymbol{e}') \leq t$ and $\boldsymbol{s} = \boldsymbol{e}'\boldsymbol{H}^\top$. This implies that for $\ell_{\mathrm{SR}} = n$, the average complexity of Algorithm 1 is equal to the average complexity of Prange's algorithm.

In the case $\ell_{\mathrm{SR}} = 1$, the sum-rank metric is equal to the rank metric, and the problem $\mathsf{SeaSD}_{\mathrm{SR}}$ can be solved by the basic version of the combinatorial rank syndrome decoder by Gaborit, Ruatta, and Schrek [139]. The average complexity of

[6] ISD algorithms like Lee and Brickell's algorithm [160] or Stern's algorithm [161] can also be applied to solve instances with $\ell_{\mathrm{SR}} = n$. However, for the example parameters that we consider, the field size is so large so that the complexity of both algorithms is minimized if no errors are allowed in the information set. In this case, both algorithms are equal to Prange's algorithm, and thus, we do not show them separately.

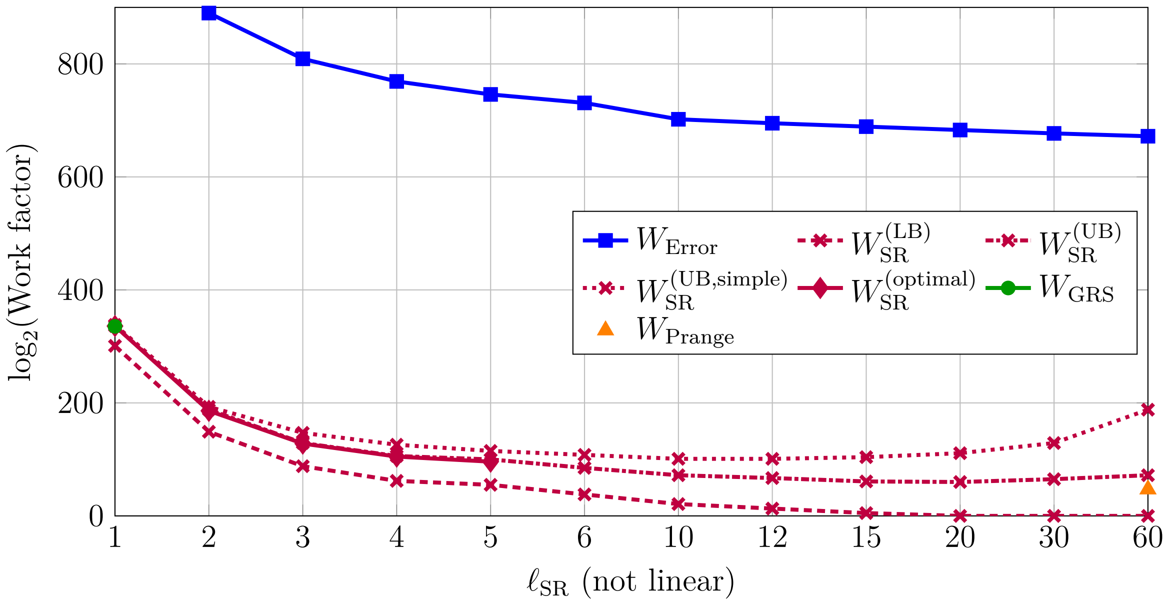


Figure 3.2: Comparison of different generic decoding strategies for $q = 2$, $m = 60$, $n = 60$, $k = 30$, $t = 10$, $s = 30$, where we choose the row support for all values of ℓ_{SR} in the proposed algorithm. The work factor W_{Error} is equal to 2^{1138} for $\ell_{\mathrm{SR}} = 1$.

the Gaborit–Ruatta–Schrek decoder is approximately W_{GRS} operations in $\mathbb{F}_q$, where $t \leq s \leq \min\left\{n-k, \left\lfloor \frac{m}{n}(n-k) \right\rfloor\right\}$, see (2.6) in Section 2.2.4. Note that for $\ell_{\mathrm{SR}} = 1$, the set $\mathcal{T}_{t,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}} = \{[t]\}$, and thus, our proposed algorithm coincides with the basic combinatorial decoding algorithm in [139].

In Figures 3.2, 3.3, and 3.4, we show the bounds on the complexity of Algorithm 1 stated in Theorem 3.10 as well as the expected complexities of the other decoding strategies as a function of ℓ_{SR} for different values of q, m, n, k, t, and s. We observe that our algorithm is more efficient than the approach of brute-forcing all vectors of sum-rank weight t and that our suboptimal support-drawing method is close to the optimal support-drawing method. Furthermore, for $\ell_{\mathrm{SR}} = 1$ and $\ell_{\mathrm{SR}} = n$, the values of the upper bound on the complexity of the proposed algorithm are close to the true complexities W_{GRS} and W_{Prange}, respectively.

For arbitrary ℓ_{SR} and $t \leq s \leq \min\left\{n-k, \lfloor \frac{m}{\eta_{\mathrm{SR}}}(n-k) \rfloor\right\}$, the simple upper bound on the complexity given in Theorem 3.10 is

$$W_{\mathrm{SR}}^{(\mathrm{UB,simple})} = n^3 m^3 \binom{\ell_{\mathrm{SR}}+t-1}{\ell_{\mathrm{SR}}-1} 4^{\ell_{\mathrm{SR}}} q^{t(\mu_{\mathrm{SR}} - \frac{s}{\ell_{\mathrm{SR}}})}$$

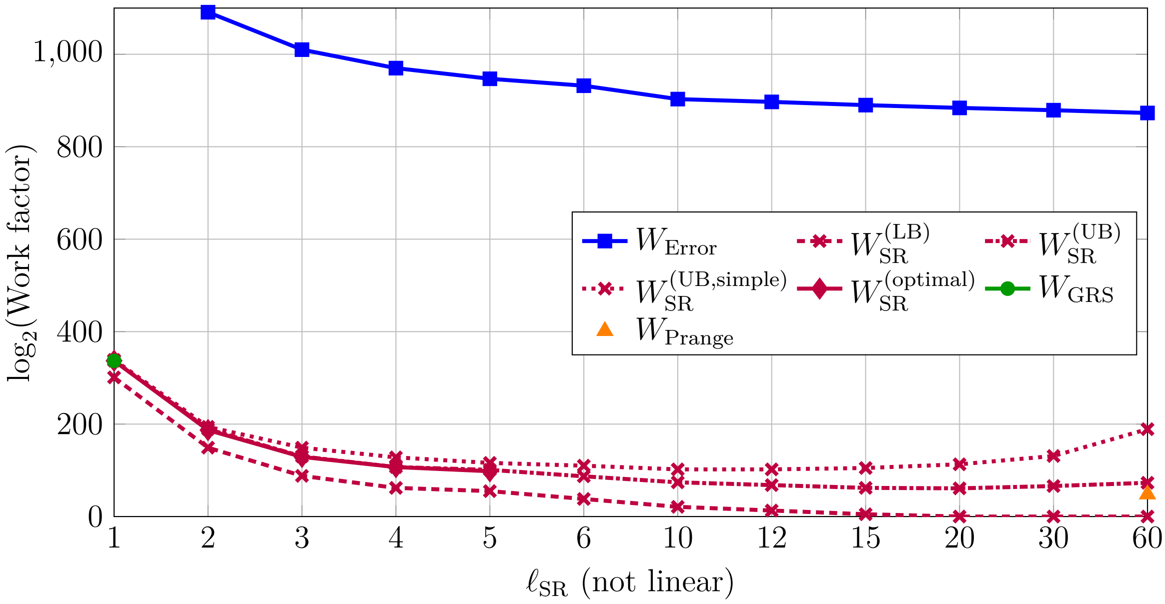


Figure 3.3: Comparison of different generic decoding strategies for $q = 2$, $m = 80$, $n = 60$, $k = 30$, $t = 10$, $s = 30$, where we choose the row support for all values of ℓ_{SR} in the proposed algorithm. The work factor W_{Error} is equal to 2^{1339} for $\ell_{\mathrm{SR}} = 1$.

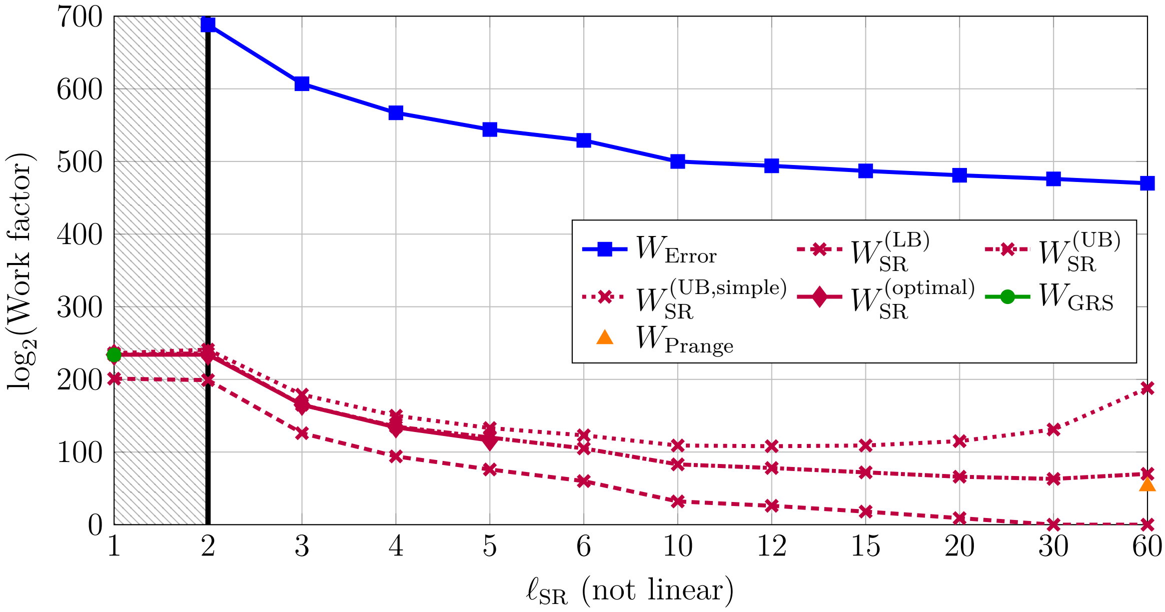


Figure 3.4: Comparison of different generic decoding strategies for $q = 2$, $m = 40$, $n = 60$, $k = 30$, $t = 10$, $s = 20$, where we choose the column support for $\ell_{\mathrm{SR}} = 1$ (gray pattern) and the row support for $\ell_{\mathrm{SR}} > 1$ in the proposed algorithm. The work factor W_{Error} is equal to 2^{936} for $\ell_{\mathrm{SR}} = 1$.

$$\leq n^3 m^3 \binom{\ell_{\mathrm{SR}}+t-1}{\ell_{\mathrm{SR}}-1} 4^{\ell_{\mathrm{SR}}} q^{t\frac{\max\{n,\ell_{\mathrm{SR}}m\}-s}{\ell_{\mathrm{SR}}}}.$$

For constant ℓ_{SR}, the factor $\binom{\ell_{\mathrm{SR}}+t-1}{\ell_{\mathrm{SR}}-1}4^{\ell_{\mathrm{SR}}}$ is polynomial in the code length, and can be neglected compared to the exponential term. Hence, the exponent of the sum-rank-metric generic decoder is roughly a factor ℓ_{SR} smaller than in the rank-metric case ($\ell_{\mathrm{SR}} = 1$). Note that the bound $W_{\mathrm{SR}}^{(\mathrm{UB,simple})}$ appears to be a loose approximation of the actual work factor for large ℓ_{SR} (cf. Figure 3.2, 3.3, and 3.4). Therefore, we refrain from a discussion of $W_{\mathrm{SR}}^{(\mathrm{UB,simple})}$ for these values of ℓ_{SR}, as this does not necessarily give a good intuition about the work factor.

3.1.4 A Hardness Reduction to $\mathsf{DecSD}_{\mathrm{SR}}$

In [136], Gaborit and Zémor probabilistically reduced $\mathsf{DecSD}_{\mathrm{H}}$ to $\mathsf{DecSD}_{\mathrm{R}}$, where $\mathsf{DecSD}_{\mathrm{H}}$ and $\mathsf{DecSD}_{\mathrm{R}}$ are defined in Problem 2.1 and in Problem 2.3, respectively. In this section, we propose a similar reduction of $\mathsf{DecSD}_{\mathrm{H}}$ to $\mathsf{DecSD}_{\mathrm{SR}}$ as defined in Problem 3.1. More precisely, we first present a reduction algorithm that proves that if a coRP-algorithm for solving $\mathsf{DecSD}_{\mathrm{SR}}$ exists, then a coRP-algorithm for $\mathsf{DecSD}_{\mathrm{H}}$ also exists. Then, we propose another reduction routine which shows that if a RP-algorithm for solving $\mathsf{DecSD}_{\mathrm{SR}}$ exists, then a RP-algorithm for $\mathsf{DecSD}_{\mathrm{H}}$ also exists. Since $\mathsf{ZPP} = \mathsf{coRP} \cap \mathsf{RP} \supset \mathsf{P}$, it follows that if $\mathsf{DecSD}_{\mathrm{SR}}$ would be in ZPP, then $\mathsf{DecSD}_{\mathrm{H}}$ would also be in ZPP. And since $\mathsf{DecSD}_{\mathrm{H}}$ is known to be NP-complete, this would imply that $\mathsf{ZPP} = \mathsf{NP}$, which is believed to be wrong. In other words, if the widely believed conjecture $\mathsf{ZPP} \neq \mathsf{NP}$ is true, then $\mathsf{DecSD}_{\mathrm{SR}}$ is in $\mathsf{NP} \setminus \mathsf{ZPP} \subset \mathsf{NP} \setminus \mathsf{P}$.

We require the following lemma to derive and prove the correctness of the proposed reduction algorithms.

Lemma 3.11. *Let ε be a constant positive real number, let m, n, and ℓ_{SR} be positive integers such that $m \geq n^2{\ell_{\mathrm{SR}}}^{-1} + n\log_q(8n) + \log_q(2\varepsilon^{-1})$, let $\boldsymbol{H} \in \mathbb{F}_q^{(n-k)\times n}$ be a parity-check matrix, and let $\boldsymbol{s} \in \mathbb{F}_q^{n-k}$. Furthermore, let the vector $\boldsymbol{x} \in \mathbb{F}_q^n$ be of minimum Hamming weight t_{H} such that $\boldsymbol{x}\boldsymbol{H}^\top = \boldsymbol{s}$, let the vector $\boldsymbol{\beta} \xleftarrow{\$} (\mathbb{F}_{q^m}^*)^n$, and let the vector $\boldsymbol{x}' \in \mathbb{F}_{q^m}^n$ be of minimum sum-rank weight such that $\boldsymbol{x}'\Big(\boldsymbol{H}\operatorname{diag}(\boldsymbol{\beta})\Big)^\top = \boldsymbol{s}$. Then, the probability*

$$\Pr\Big(\mathrm{wt}_{\mathrm{SR}}(\boldsymbol{x}') < t_{\mathrm{H}}\Big) \leq \varepsilon.$$

Proof. Let $\boldsymbol{H}$, $\boldsymbol{s}$, t_{H} be fixed, let $\mathcal{E}_{\boldsymbol{a}}$ denote the event that for a fixed vector $\boldsymbol{a} \in \mathbb{F}_{q^m}^n$ and a random vector $\boldsymbol{\beta} \xleftarrow{\$} (\mathbb{F}_{q^m}^*)^n$, the equality $\boldsymbol{a}\Big(\boldsymbol{H}\operatorname{diag}(\boldsymbol{\beta})\Big)^\top = \boldsymbol{s}$ holds, and let the set $\mathcal{X}(t_{\mathrm{H}}-1) \coloneqq \{\boldsymbol{a} \in \mathbb{F}_{q^m}^n : \mathrm{wt}_{\mathrm{SR}}(\boldsymbol{a}) < t_{\mathrm{H}}\}$. Then, the probability

$$\begin{aligned} P &\coloneqq \Pr\left(\exists \boldsymbol{x}' \in \mathbb{F}_{q^m}^n \ : \ \boldsymbol{x}'\Big(\boldsymbol{H}\,\mathrm{diag}(\boldsymbol{\beta})\Big)^\top = \boldsymbol{s} \wedge \mathrm{wt}_{\mathrm{SR}}(\boldsymbol{x}') < t_{\mathrm{H}}\right) \\ &= \Pr\left(\bigcup_{\boldsymbol{x}' \in \mathcal{X}(t_{\mathrm{H}}-1)} \mathcal{E}_{\boldsymbol{x}'}\right) \leq \sum_{\boldsymbol{x}' \in \mathcal{X}(t_{\mathrm{H}}-1)} \Pr(\mathcal{E}_{\boldsymbol{x}'}), \end{aligned}$$

where the randomness is in $\boldsymbol{\beta}$.

If there is no $\boldsymbol{\beta} \in (\mathbb{F}_{q^m}^*)^n$ such that $\boldsymbol{x}'\Big(\boldsymbol{H}\,\mathrm{diag}(\boldsymbol{\beta})\Big)^\top = \boldsymbol{s}$, then $\Pr(\mathcal{E}_{\boldsymbol{x}'}) = 0$. If a vector $\boldsymbol{\beta} \in (\mathbb{F}_{q^m}^*)^n$ such that $\boldsymbol{x}'\Big(\boldsymbol{H}\,\mathrm{diag}(\boldsymbol{\beta})\Big)^\top = \boldsymbol{s}$ exists, then there must be a set $\mathcal{W} \subseteq \mathrm{supp}_{\mathrm{H}}(\boldsymbol{x}')$ with $|\mathcal{W}| = t_{\mathrm{H}}$ such that that the matrix $\boldsymbol{H}_{\mathcal{W}} \in \mathbb{F}_q^{(n-k)\times t_{\mathrm{H}}}$ has full rank, where $\boldsymbol{H}_{\mathcal{W}}$ is a submatrix of $\boldsymbol{H}$ and consists of the columns of $\boldsymbol{H}$ indexed by $\mathcal{W}$ [136, Lem. 4]. Thus, for a fixed $\boldsymbol{x}'$, the cardinality

$$\left|\left\{\boldsymbol{x}'\Big(\boldsymbol{H}\,\mathrm{diag}(\boldsymbol{\beta})\Big)^\top : \beta_i \in \mathbb{F}_{q^m}^* \forall i \in \mathcal{W} \wedge \beta_j \text{ are fixed } \forall j \notin \mathcal{W}, \right\}\right| = \Big(q^m - 1\Big)^{t_{\mathrm{H}}},$$

and we can bound $\Pr(\mathcal{E}_{\boldsymbol{x}'}) \leq (q^m-1)^{-t_{\mathrm{H}}}$ for $\boldsymbol{\beta} \xleftarrow{\$} (\mathbb{F}_{q^m}^*)^n$. Since $m \geq t_{\mathrm{H}}$, it holds that

$$\Gamma(q,m,t_{\mathrm{H}}) \coloneqq \frac{q^{mt_{\mathrm{H}}}}{(q^m-1)^{t_{\mathrm{H}}}} \leq \frac{1}{(1-q^{-m})^m} = \frac{1}{\sum_{i=0}^m \binom{m}{i}(-q^{-m})^i} \overset{\text{(i)}}{\leq} \frac{1}{1-mq^{-m}} \overset{\text{(ii)}}{\leq} 2,$$

where inequality (i) holds, since for increasing i, the quantity $\left|\binom{m}{i}(-q^{-m})^i\right|$ is strictly monotonically decreasing and the sign of $\left|\binom{m}{i}(-q^{-m})^i\right|$ is alternating. Furthermore, inequality (ii) holds due to $mq^{-m} \leq \frac{1}{2}$. Then, combining the aforementioned results gives

$$\begin{aligned} P &\leq \frac{1}{(q^m-1)^{t_{\mathrm{H}}}} |\mathcal{X}(t_{\mathrm{H}}-1)| \\ &= \Gamma(q,m,t_{\mathrm{H}}) \frac{1}{q^{mt_{\mathrm{H}}}} \sum_{i=1}^{t_{\mathrm{H}}-1} \mathcal{N}_{q,\eta_{\mathrm{SR}},m}(i,\ell_{\mathrm{SR}}) \\ &\leq 2\frac{1}{q^{mt_{\mathrm{H}}}}(t_{\mathrm{H}}-1) \max_{i \in [1:t_{\mathrm{H}}-1]} \mathcal{N}_{q,\eta_{\mathrm{SR}},m}(i,\ell_{\mathrm{SR}}) \\ &\leq 2\frac{1}{q^{mt_{\mathrm{H}}}}(t_{\mathrm{H}}-1)\binom{\ell_{\mathrm{SR}}+t_{\mathrm{H}}-2}{\ell_{\mathrm{SR}}-1} 4^{\ell_{\mathrm{SR}}} q^{(t_{\mathrm{H}}-1)(m+\eta_{\mathrm{SR}}-\frac{t_{\mathrm{H}}-1}{\ell_{\mathrm{SR}}})} \\ &= 2(t_{\mathrm{H}}-1)\binom{\ell_{\mathrm{SR}}+t_{\mathrm{H}}-2}{\ell_{\mathrm{SR}}-1} 4^{\ell_{\mathrm{SR}}} q^{-m+(t_{\mathrm{H}}-1)\eta_{\mathrm{SR}}-\frac{(t_{\mathrm{H}}-1)^2}{\ell_{\mathrm{SR}}}} \end{aligned}$$

$$
\begin{aligned}
&\leq 2(t_{\mathrm{H}}-1)\binom{\ell_{\mathrm{SR}}+t_{\mathrm{H}}-2}{\ell_{\mathrm{SR}}-1}4^{\ell_{\mathrm{SR}}}q^{-m+\frac{n^2}{\ell_{\mathrm{SR}}}-\frac{(t_{\mathrm{H}}-1)^2}{\ell_{\mathrm{SR}}}}\\
&\leq 2\underbrace{(t_{\mathrm{H}}-1)}_{\leq \ell_{\mathrm{SR}}+t_{\mathrm{H}}-2}(\ell_{\mathrm{SR}}+t_{\mathrm{H}}-2)^{\ell_{\mathrm{SR}}-1}4^{\ell_{\mathrm{SR}}}q^{-m+\frac{n^2}{\ell_{\mathrm{SR}}}-\frac{(t_{\mathrm{H}}-1)^2}{\ell_{\mathrm{SR}}}}\\
&\leq 2[4(\ell_{\mathrm{SR}}+t_{\mathrm{H}}-2)]^{\ell_{\mathrm{SR}}}q^{-m+\frac{n^2}{\ell_{\mathrm{SR}}}-\frac{(t_{\mathrm{H}}-1)^2}{\ell_{\mathrm{SR}}}}\\
&\leq 2q^{-m+\frac{n^2}{\ell_{\mathrm{SR}}}-\frac{(t_{\mathrm{H}}-1)^2}{\ell_{\mathrm{SR}}}+\ell_{\mathrm{SR}}\log_q[4(\ell_{\mathrm{SR}}+t_{\mathrm{H}}-2)]}\\
&\leq 2q^{-m+\frac{n^2}{\ell_{\mathrm{SR}}}+\ell_{\mathrm{SR}}\log_q[4(\ell_{\mathrm{SR}}+t_{\mathrm{H}}-2)]}\\
&\leq 2q^{-m+\frac{n^2}{\ell_{\mathrm{SR}}}+n\log_q(8n)}\\
&\leq \varepsilon. \qquad\qquad \blacksquare
\end{aligned}
$$

In the following lemma, we consider the coRP reduction.

Lemma 3.12. *Let $\ell_{\mathrm{SR}} < n$, let $m > n^2{\ell_{\mathrm{SR}}}^{-1} + n\log_q(8n) + \log_q(2)$ and suppose that* $\mathsf{DecSD}_{\mathrm{SR}}$ *is in* coRP. *Then, the problem* $\mathsf{DecSD}_{\mathrm{H}}$ *is also in* coRP.

Proof. Let $\boldsymbol{H} \in \mathbb{F}_q^{(n-k)\times n}$, $\boldsymbol{s} \in \mathbb{F}_q^{n-k}$, $t \in \mathbb{Z}_{>0}$, and let t_{H} be the minimum Hamming weight of the vectors $\boldsymbol{x} \in \mathbb{F}_q^n$ such that $\boldsymbol{x}\boldsymbol{H}^\top = \boldsymbol{s}$. Let $\boldsymbol{H}' \in \mathbb{F}_{q^m}^{(n-k)\times n}$ and let t_{SR} denote the minimum sum-rank weight of the vectors $\boldsymbol{x}' \in \mathbb{F}_{q^m}^n$ such that $\boldsymbol{x}'\boldsymbol{H}'^\top = \boldsymbol{s}$. Let $\tilde{\varepsilon}$ be a fixed non-negative constant and let $\mathcal{A}_{\mathrm{H}}^{\mathsf{coRP}}$ be defined as in Algorithm 4. Furthermore, let $\mathcal{A}_{\mathrm{SR}}^{\mathsf{coRP}}$ denote a coRP-algorithm for $\mathsf{DecSD}_{\mathrm{SR}}$, i.e., $\mathcal{A}_{\mathrm{SR}}^{\mathsf{coRP}}$ takes as inputs $(\boldsymbol{H}', \boldsymbol{s}, t)$, it always returns true if $t_{\mathrm{SR}} \leq t$, and it outputs false with probability at least $1-\tilde{\varepsilon}$ if $t_{\mathrm{SR}} > t$.

To show that $\mathcal{A}_{\mathrm{H}}^{\mathsf{coRP}}$ is a coRP-algorithm for $\mathsf{DecSD}_{\mathrm{H}}$, we prove that (i) the algorithm always outputs true if $t_{\mathrm{H}} \leq t$, and (ii) returns false with at least some non-zero constant probability if $t_{\mathrm{H}} > t$. The first requirement on $\mathcal{A}_{\mathrm{H}}^{\mathsf{coRP}}$ is fulfilled since if $t_{\mathrm{H}} \leq t$, then $t_{\mathrm{SR}} \leq t$, and thus, both $\mathcal{A}_{\mathrm{SR}}^{\mathsf{coRP}}$ and $\mathcal{A}_{\mathrm{H}}^{\mathsf{coRP}}$ output true. For the second requirement, we observe that if $t_{\mathrm{H}} > t$, the equality $t_{\mathrm{SR}} = t_{\mathrm{H}} > t$ holds with probability at least $1-\varepsilon$, see Lemma 3.11. By choosing a non-negative constant $\varepsilon < 1$ such that $m \geq n^2{\ell_{\mathrm{SR}}}^{-1} + n\log_q(8n) + \log_q(2\varepsilon^{-1})$, the algorithm $\mathcal{A}_{\mathrm{H}}^{\mathsf{coRP}}$ outputs false with probability at least $(1-\varepsilon)(1-\tilde{\varepsilon})$, which is a constant. ■

The RP reduction is shown in the following.

Lemma 3.13. *Let $\ell_{\mathrm{SR}} < n$, let $m \geq n^2{\ell_{\mathrm{SR}}}^{-1} + n\log_q(8n) + \log_q(4n)$ and suppose that* $\mathsf{DecSD}_{\mathrm{SR}}$ *is in* RP. *Then, the problem* $\mathsf{DecSD}_{\mathrm{H}}$ *is also in* RP.

Algorithm 4: $\mathcal{A}_{\mathrm{H}}^{\mathsf{coRP}}$

```
Input : Parity-check matrix $\boldsymbol{H} \in \mathbb{F}_q^{(n-k)\times n}$
        Syndrome vector $\boldsymbol{s} \in \mathbb{F}_q^{n-k}$
        Non-negative integer $t$
Output: Boolean true or false
$\boldsymbol{\beta} \xleftarrow{\$} (\mathbb{F}_{q^m}^*)^n$
$\boldsymbol{H}' \leftarrow \boldsymbol{H}\,\mathrm{diag}(\boldsymbol{\beta}) \in \mathbb{F}_{q^m}^{(n-k)\times n}$
return $\mathcal{A}_{\mathrm{SR}}^{\mathsf{coRP}}(\boldsymbol{H}', \boldsymbol{s})$
```

Proof. Let $\boldsymbol{H} \in \mathbb{F}_q^{(n-k)\times n}$, $\boldsymbol{s} \in \mathbb{F}_q^{n-k}$, $t \in \mathbb{Z}_{>0}$, and let t_{H} be the minimum Hamming weight of the vectors $\boldsymbol{x} \in \mathbb{F}_q^n$ such that $\boldsymbol{x}\boldsymbol{H}^\top = \boldsymbol{s}$. Let $\boldsymbol{H}' \in \mathbb{F}_{q^m}^{(n-k)\times n}$ and let t_{SR} denote the minimum sum-rank weight of the vectors $\boldsymbol{x}' \in \mathbb{F}_{q^m}^n$ such that $\boldsymbol{x}'\boldsymbol{H}'^\top = \boldsymbol{s}$. Let $\tilde{\varepsilon}$ be a fixed non-negative constant, and let $\mathcal{A}_{\mathrm{H}}^{\mathsf{RP}}$ be defined as in Algorithm 5, where the function $\mathsf{Cols}(\boldsymbol{H}, \mathcal{T})$ returns a sub-matrix of $\boldsymbol{H}$ consisting of the columns indexed by the set $\mathcal{T}$. Furthermore, let $\mathcal{A}_{\mathrm{SR}}^{\mathsf{RP}}$ denote a RP-algorithm for $\mathsf{DecSD}_{\mathrm{SR}}$, i.e., $\mathcal{A}_{\mathrm{SR}}^{\mathsf{RP}}$ takes as inputs $(\boldsymbol{H}', \boldsymbol{s}, t)$, it always returns false if $t_{\mathrm{SR}} > t$, and it outputs true with a probability of at least $1 - \tilde{\varepsilon}$ if $t_{\mathrm{SR}} \leq t$. Note, we can assume that $\tilde{\varepsilon} < \frac{1}{2n}$, if $\mathcal{A}_{\mathrm{SR}}^{\mathsf{RP}}$ is called at most $O(\log n)$ times. To show that $\mathcal{A}_{\mathrm{H}}^{\mathsf{RP}}$ is an RP-algorithm for $\mathsf{DecSD}_{\mathrm{H}}$, we prove that (i) the algorithm always outputs false if $t_{\mathrm{H}} > t$, and (ii) it returns true with at least some constant non-zero probability if $t_{\mathrm{H}} \leq t$.

The goal of Algorithm 5 is to find a Hamming super-support $\mathcal{S}$ of a vector $\boldsymbol{x} \in \mathbb{F}_q^n$ such that $|\mathcal{S}| \leq t$ and $\boldsymbol{x}\boldsymbol{H}^\top = \boldsymbol{s}$. From Lines 9–12 follows that only if such a super-support is determined, then $\mathcal{A}_{\mathrm{H}}^{\mathsf{RP}}$ returns true. This implies that if $t_{\mathrm{H}} > t$, the algorithm always returns false, which means (i) is fulfilled.

To see that (ii) is fulfilled, suppose $t_{\mathrm{H}} \leq t$. In the Lines 3–7, the algorithm *tries* to determine whether the set $\mathcal{S}\setminus\{i\}$ is a super-support of a vector $\boldsymbol{x} \in \mathbb{F}_q$ with $\mathrm{wt}_{\mathrm{H}}(\boldsymbol{x}) \leq t$ and syndrome $\boldsymbol{s} = \boldsymbol{x}\boldsymbol{H}^\top$. If the algorithm determines this in all iterations of the loop correctly, then the set $\mathcal{S}$ is the support of a vector $\boldsymbol{x}$ with $\mathrm{wt}_{\mathrm{H}}(\boldsymbol{x}) \leq t$ and syndrome $\boldsymbol{s} = \boldsymbol{x}\boldsymbol{H}^\top$. In case the algorithm determines a wrong result one or more times, then it is not guaranteed that $\mathcal{S}$ has this property. In case $\mathcal{S}$ does not have this required property, it is detected by the algorithm in Lines 9–12.

In the following, we prove that in Lines 3–7, the algorithm $\mathcal{A}_{\mathrm{H}}^{\mathsf{RP}}$ determines whether $\mathcal{S}$ has the required property in *all* iterations of the loop correctly with a probability of at least a constant. Given $\boldsymbol{s} \in \mathbb{F}_q^{n-k}$, $\mathcal{S}$, i, $\boldsymbol{\beta} \in \mathbb{F}_{q^m}^{|\mathcal{S}|-1}$, let $\tilde{t}_{\mathrm{H}}$ be the smallest Hamming weight of a vector $\tilde{\boldsymbol{x}} \in \mathbb{F}_q^{|\mathcal{S}|-1}$ such that $\tilde{\boldsymbol{x}}\bar{\boldsymbol{H}}^\top = \boldsymbol{s}$, and let $\tilde{t}_{\mathrm{SR}}$ be the smallest sum-rank

```
Algorithm 5: $\mathcal{A}_{\mathrm{H}}^{\mathsf{RP}}$
Input  : Parity-check matrix $\boldsymbol{H} \in \mathbb{F}_q^{(n-k)\times n}$
         Syndrome vector $\boldsymbol{s} \in \mathbb{F}_q^{n-k}$
         Non-negative integer $t$
Output: Boolean true or false
1  $\mathcal{S} = \{1, \dots, n\}$
2  for $i = 1, \dots, n$ do
3      $\bar{\boldsymbol{H}} \leftarrow \mathsf{Cols}(\boldsymbol{H}, \mathcal{S} \setminus \{i\}) \in \mathbb{F}_q^{(n-k)\times(|\mathcal{S}|-1)}$
4      $\boldsymbol{\beta} \xleftarrow{\$} (\mathbb{F}_{q^m}^*)^{|\mathcal{S}|-1}$
5      $\bar{\boldsymbol{H}}' \leftarrow \bar{\boldsymbol{H}}\,\mathrm{diag}(\boldsymbol{\beta}) \in \mathbb{F}_{q^m}^{(n-k)\times(|\mathcal{S}|-1)}$
6      if $\mathcal{A}_{\mathrm{SR}}^{\mathsf{RP}}(\bar{\boldsymbol{H}}', \boldsymbol{s}, t) = \mathsf{true}$ then
7          $\mathcal{S} \leftarrow \mathcal{S} \setminus \{i\}$
8  $\bar{\boldsymbol{H}} \leftarrow \mathsf{Cols}(\boldsymbol{H}, \mathcal{S}) \in \mathbb{F}_q^{(n-k)\times|\mathcal{S}|}$
9  if $1 \leq |\mathcal{S}| \leq t \wedge \exists \boldsymbol{x} \in \mathbb{F}_q^{|\mathcal{S}|}$ s.t. $\boldsymbol{x}\bar{\boldsymbol{H}}^\top = \boldsymbol{s}$ then
10     return true
11 else
12     return false
```

weight of a vector $\tilde{\boldsymbol{x}}' \in \mathbb{F}_{q^m}^{|\mathcal{S}|-1}$ such that $\tilde{\boldsymbol{x}}'\bar{\boldsymbol{H}}'^\top = \boldsymbol{s}$. First, consider the case $\tilde{t}_{\mathrm{H}} \leq t$. Since $\tilde{t}_{\mathrm{SR}} \leq \tilde{t}_{\mathrm{H}} \leq t$, the algorithm $\mathcal{A}_{\mathrm{SR}}^{\mathsf{RP}}(\bar{\boldsymbol{H}}', \boldsymbol{s}, t)$ outputs the correct answer true with a probability of at least $1 - \tilde{\varepsilon} > 1 - \frac{1}{2n}$, where the randomness is in $\mathcal{A}_{\mathrm{SR}}^{\mathsf{RP}}$. Second, consider the case $\tilde{t}_{\mathrm{H}} > t$. From Lemma 3.11 follows that the vector $\boldsymbol{\beta}$ is sampled such that $\tilde{t}_{\mathrm{SR}} = \tilde{t}_{\mathrm{H}}$ with probability greater than $1 - \frac{1}{2n}$, where we chose $\varepsilon = \frac{1}{2n}$. Since the algorithm $\mathcal{A}_{\mathrm{SR}}^{\mathsf{RP}}(\bar{\boldsymbol{H}}', \boldsymbol{s}, t)$ always outputs the correct answer false for $\tilde{t}_{\mathrm{SR}} = \tilde{t}_{\mathrm{H}} > t$, the probability that $\mathcal{A}_{\mathrm{SR}}^{\mathsf{RP}}(\bar{\boldsymbol{H}}', \boldsymbol{s}, t)$ outputs the correct answer false for $\tilde{t}_{\mathrm{H}} > t$ is greater than $1 - \frac{1}{2n} = 1 - \varepsilon$, where the randomness is in the sampling of $\boldsymbol{\beta}$.

This proves that in Lines 3–7, the algorithm $\mathcal{A}_{\mathrm{H}}^{\mathsf{RP}}$ determines whether $\mathcal{S}$ has the required properties correctly with probability greater than $1 - \frac{1}{2n}$. Since $\mathcal{A}_{\mathrm{H}}^{\mathsf{RP}}$ has to determine this at most n times, the overall success probability is at least $1 - \frac{n}{2n} = \frac{1}{2}$. ■

Using the derived lemmata, we can now prove our main statement about the hardness of $\mathsf{DecSD}_{\mathrm{SR}}$.

Theorem 3.14. *Let $\ell_{\mathrm{SR}} < n$, let $m \geq n^2{\ell_{\mathrm{SR}}}^{-1} + n\log_q(8n) + \log_q(4n)$, and suppose that $\mathsf{DecSD}_{\mathrm{SR}}$ is in $\mathsf{ZPP} = \mathsf{RP} \cap \mathsf{coRP}$. Then, it holds that $\mathsf{NP} = \mathsf{ZPP}$.*

Proof. Since $\mathsf{DecSD}_{\mathrm{H}}$ is NP-complete, Lemma 3.12 and Lemma 3.13 imply that $\mathsf{ZPP} \supseteq \mathsf{NP}$ and since it is well-known that $\mathsf{ZPP} \subseteq \mathsf{NP}$, the equality $\mathsf{NP} = \mathsf{ZPP}$ holds. ■

3.2 Syndrome Decoding of High-Order Interleaved Rank-Metric Codes

In this section, we investigate the problem of decoding *any* high-order interleaved code in the rank metric. The decisional version of this problem reads as follows:

Problem 3.3 (Decisional Interleaved Rank Syndrome Decoding ($\mathsf{DecISD}_{\mathrm{R}}$) Problem)**.**

Given:
- *Parity-check matrix* $\boldsymbol{H} \in \mathbb{F}_{q^m}^{(n-k)\times n}$ *of an* $[n, k, d_{\min}^{\mathrm{R}}]_{\mathbb{F}_{q^m}}^{\mathrm{R}}$ *code* $\mathcal{C}$
- *Non-negative integer* t *with* $0 \leq t \leq \min\{u, d_{\min}^{\mathrm{R}} - 2\}$
- *Matrix* $\boldsymbol{S} \in \mathbb{F}_{q^m}^{(n-k)\times u}$

Question: *Is there an* $\boldsymbol{E} \in \mathbb{F}_{q^m}^{u\times n}$ *such that* $\mathrm{rk}_q(\boldsymbol{E}) = \mathrm{rk}_{q^m}(\boldsymbol{E}) \leq t$ *and* $\boldsymbol{S} = \boldsymbol{H}\boldsymbol{E}^\top$*?*

In [136], it is proven that the non-interleaved syndrome decoding problem $\mathsf{DecSD}_{\mathrm{R}}$ is a difficult problem. We show that the interleaved variant of $\mathsf{DecSD}_{\mathrm{R}}$ as defined in Problem 3.3 can be solved in polynomial time. For this purpose, we propose an efficient algorithm that solves the associated *search* problem.

Problem 3.4 (Search Interleaved Rank Syndrome Decoding ($\mathsf{SeaISD}_{\mathrm{R}}$) Problem)**.**

Given:
- *Parity-check matrix* $\boldsymbol{H} \in \mathbb{F}_{q^m}^{(n-k)\times n}$ *of an* $[n, k, d_{\min}^{\mathrm{R}}]_{\mathbb{F}_{q^m}}^{\mathrm{R}}$ *code* $\mathcal{C}$
- *Non-negative integer* t *with* $0 \leq t \leq \min\{u, d_{\min}^{\mathrm{R}} - 2\}$
- *Matrix* $\boldsymbol{S} = \boldsymbol{H}\boldsymbol{E}^\top \in \mathbb{F}_{q^m}^{(n-k)\times u}$*, where* $\mathrm{rk}_q(\boldsymbol{E}) = \mathrm{rk}_{q^m}(\boldsymbol{E}) = t$

Objective: *Search for an* $\boldsymbol{E}' \in \mathbb{F}_{q^m}^{u\times n}$ *such that* $\mathrm{rk}_q(\boldsymbol{E}') = \mathrm{rk}_{q^m}(\boldsymbol{E}') \leq t$ *and* $\boldsymbol{S} = \boldsymbol{H}\boldsymbol{E}'^\top$.

The proposed algorithm can be seen as an adaption of Metzner and Kaputrowski's Hamming metric decoder [162] (Metzner and Kaputrowski's algorithm was also derived in [163] and generalized to dependent errors in [164, 165]). It is similar to many other decoding algorithms in the sense that it first determines the row rank support of the error, i.e., the subspace $\mathrm{supp}_{\mathrm{R}}^{(\mathrm{R})}(\boldsymbol{E})$, and then recovers the entire error matrix $\boldsymbol{E}$ using an erasure decoding algorithm.

3.2.1 A New Algorithm for Solving $\mathsf{SeaISD}_{\mathrm{R}}$

In the following, we construct an efficient algorithm for solving the $\mathsf{SeaISD}_{\mathrm{R}}$ problem. The decoding routine is presented in Algorithm 6, and the correctness and the complexity of the algorithm are proven in Theorem 3.20. Since the proof is quite technical, we first derive some intermediate results.

Algorithm 6: Generic High-Order Interleaved Rank Syndrome Decoder

```
Input  : Parity-check matrix $\boldsymbol{H} \in \mathbb{F}_{q^m}^{(n-k)\times n}$
         Non-negative integer $t$
         Syndrome matrix $\boldsymbol{S} \in \mathbb{F}_{q^m}^{(n-k)\times u}$
Output: Matrix $\boldsymbol{E}' \in \mathbb{F}_{q^m}^{u\times n}$
Determine $\boldsymbol{P} \in \mathbb{F}_{q^m}^{(n-k)\times(n-k)}$ s.t. $\boldsymbol{PS} = \mathrm{ref}(\boldsymbol{S})$
$\boldsymbol{H}_{\mathrm{sub}} \leftarrow (\boldsymbol{PH})_{[t+1:n-k],:} \in \mathbb{F}_{q^m}^{(n-k-t)\times n}$
Determine $\boldsymbol{B}' \in \mathbb{F}_q^{t\times n}$ s.t. $\mathrm{ext}_{q^m/q}(\boldsymbol{H}_{\mathrm{sub}})\boldsymbol{B}'^{\top} = \boldsymbol{0}$ and $\mathrm{rk}_q(\boldsymbol{B}') = t$
Determine $\boldsymbol{A}' \in \mathbb{F}_{q^m}^{u\times t}$ s.t. $\boldsymbol{HB}'^{\top}\boldsymbol{A}'^{\top} = \boldsymbol{S}$
$\boldsymbol{E}' \leftarrow \boldsymbol{A}'\boldsymbol{B}' \in \mathbb{F}_{q^m}^{u\times n}$
return $\boldsymbol{E}'$
```

In the following lemma, we show a well-known statement about erasure decoding of interleaved codes in the rank metric.

Lemma 3.15. *Let $\boldsymbol{H} \in \mathbb{F}_{q^m}^{(n-k)\times n}$ be a parity-check matrix of an $[n, k, d_{\mathrm{min}}^{\mathrm{R}}]_{\mathbb{F}_{q^m}}^{\mathrm{R}}$ code $\mathcal{C}$, let $\boldsymbol{E} \in \mathbb{F}_{q^m}^{u\times n}$ be an error matrix with $\mathrm{wt}_{\mathrm{R}}(\boldsymbol{E}) = t \leq d_{\mathrm{min}}^{\mathrm{R}}-1$, let $\boldsymbol{S} = \boldsymbol{HE}^{\top} \in \mathbb{F}_{q^m}^{(n-k)\times u}$, and let the rows of the matrix $\boldsymbol{B} \in \mathbb{F}_q^{t\times n}$ be a basis of the row rank support of $\boldsymbol{E} \in \mathbb{F}_{q^m}^{u\times n}$. Then, the error can be written as $\boldsymbol{E} = \boldsymbol{AB}$, where $\boldsymbol{A} \in \mathbb{F}_{q^m}^{u\times t}$ is the unique solution of $\boldsymbol{S} = (\boldsymbol{HB}^{\top})\boldsymbol{A}^{\top}$, and $\boldsymbol{E}$ can be determined from $\boldsymbol{B}$, $\boldsymbol{H}$, and $\boldsymbol{S}$ in $O(\max\{un^2, n^3\})$ operations in $\mathbb{F}_{q^m}$.*

Proof. See, e.g., [86]. ■

The previous lemma proves that if we know the row rank support of the error, then we can compute the error in polynomial time. To determine the row rank support of the error, we first transform $\boldsymbol{S}$ into reduced row echelon form and apply the same row operations to the parity-check matrix $\boldsymbol{H}$. Then, we use the matrix $\boldsymbol{H}_{\mathrm{sub}}$ to determine $\mathrm{supp}_{\mathrm{R}}^{(\mathrm{R})}(\boldsymbol{E})$, where $\boldsymbol{H}_{\mathrm{sub}}$ consists of the rows of the transformed matrix $\boldsymbol{H}$ that correspond to the zero rows of the echelon form of $\boldsymbol{S}$. The mentioned steps are illustrated in Figure 3.5.

In the following lemma, we state a property of $\boldsymbol{H}_{\mathrm{sub}}$ that we need for proving the correctness of the proposed algorithm.

Lemma 3.16. *Let $\boldsymbol{H} \in \mathbb{F}_{q^m}^{(n-k)\times n}$ be a parity-check matrix of an $[n, k, d_{\mathrm{min}}^{\mathrm{R}}]_{\mathbb{F}_{q^m}}^{\mathrm{R}}$ code, let $\boldsymbol{E} \in \mathbb{F}_{q^m}^{u\times n}$ be an error matrix with $\mathrm{rk}_q(\boldsymbol{E}) = t < n-k$ and let $\boldsymbol{S} = \boldsymbol{HE}^{\top} \in \mathbb{F}_{q^m}^{(n-k)\times u}$ be the corresponding syndrome matrix. Furthermore, let $\boldsymbol{P} \in \mathbb{F}_{q^m}^{(n-k)\times(n-k)}$ be a full*

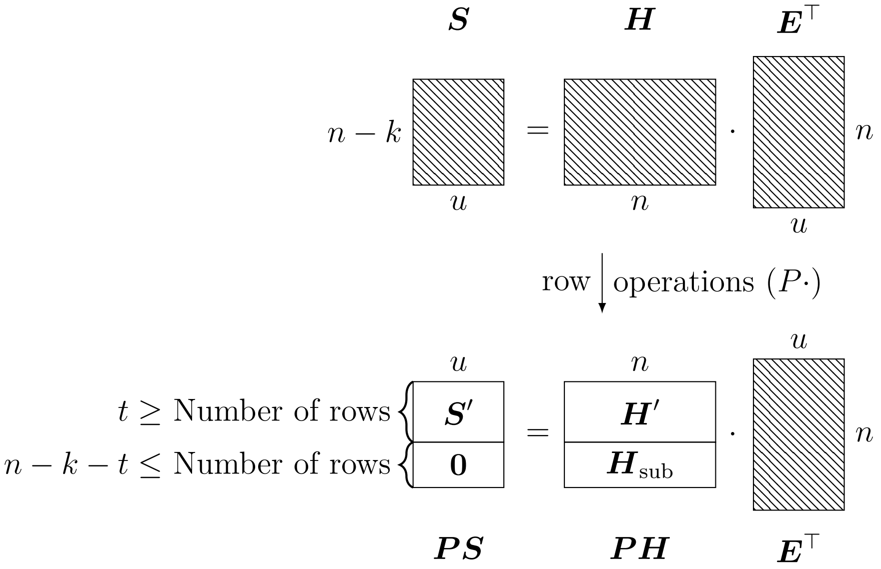


Figure 3.5: Illustration of Lemma 3.16.

$\mathbb{F}_{q^m}$-rank matrix such that $\boldsymbol{PS}$ is in row echelon form, and let $\boldsymbol{H}_{\text{sub}}$ be the rows of $\boldsymbol{PH}$ corresponding to the zero rows in $\boldsymbol{PS}$. Then, at least $n-k-t$ rows of $\boldsymbol{PS}$ are zero, and the rows of $\boldsymbol{H}_{\text{sub}}$ form a basis of $\mathcal{K}_{q^m}(\boldsymbol{E}) \cap \mathcal{C}^\perp$.

Proof. Since $\text{rk}_q(\boldsymbol{E}) = t$, it follows that the $\mathbb{F}_{q^m}$-rank of $\boldsymbol{E}$ and of $\boldsymbol{S}$ is at most t, and thus, at least $n-k-t$ rows of $\boldsymbol{PS}$ are zero. The rows of $\boldsymbol{PH}$ are in the row space of $\boldsymbol{H}$, and since $\boldsymbol{H}_{\text{sub}}\boldsymbol{E}^\top = \boldsymbol{0}$, it follows that the row space of $\boldsymbol{H}_{\text{sub}}$ is in the kernel of $\boldsymbol{E}$, i.e., $\mathcal{R}_{q^m}(\boldsymbol{H}_{\text{sub}}) \subseteq \mathcal{K}_{q^m}(\boldsymbol{E}) \cap \mathcal{C}^\perp$.

To show that the rows of $\boldsymbol{H}_{\text{sub}}$ span the entire intersection $\mathcal{K}_{q^m}(\boldsymbol{E}) \cap \mathcal{C}^\perp$, we observe that

$$\boldsymbol{PS} = \begin{bmatrix} \boldsymbol{S}' \\ \boldsymbol{0} \end{bmatrix} \quad \text{and} \quad \boldsymbol{PH} = \begin{bmatrix} \boldsymbol{H}' \\ \boldsymbol{H}_{\text{sub}} \end{bmatrix},$$

where the matrix $\boldsymbol{S}' = \boldsymbol{H}'\boldsymbol{E}^\top$ has full $\mathbb{F}_{q^m}$-rank and has the same number of rows as $\boldsymbol{H}'$. Let the vector $\boldsymbol{h} := [\boldsymbol{v}_1, \boldsymbol{v}_2][\boldsymbol{H}'^\top, \boldsymbol{H}_{\text{sub}}^\top]^\top$ be in the kernel of $\boldsymbol{E}$, i.e., $\boldsymbol{h} \in \mathcal{K}_{q^m}(\boldsymbol{E}) \cap \mathcal{C}^\perp$. Since $\boldsymbol{H}_{\text{sub}}\boldsymbol{E}^\top = \boldsymbol{0}$, it holds that

$$\boldsymbol{0} = \boldsymbol{h}\boldsymbol{E}^\top = [\boldsymbol{v}_1, \boldsymbol{v}_2] \begin{bmatrix} \boldsymbol{H}' \\ \boldsymbol{H}_{\text{sub}} \end{bmatrix} \boldsymbol{E}^\top = \boldsymbol{v}_1 \boldsymbol{H}' \boldsymbol{E}^\top = \boldsymbol{v}_1 \boldsymbol{S}',$$

and because the rows of $\boldsymbol{S}'$ are linearly independent, the vector $\boldsymbol{v}_1 = \boldsymbol{0}$. Therefore, the vector $\boldsymbol{h}$ must be in the row space of $\boldsymbol{H}_{\text{sub}}$, i.e., $\mathcal{K}_{q^m}(\boldsymbol{E}) \cap \mathcal{C}^\perp \subseteq \mathcal{R}_{q^m}(\boldsymbol{H}_{\text{sub}})$. ■

Lemma 3.16 indicates that the matrix $\boldsymbol{H}_{\text{sub}}$ is connected to the right kernel of $\boldsymbol{E}$. We show in the next lemma that if $\text{rk}_{q^m}(\boldsymbol{E}) = t$, then the right kernel of $\boldsymbol{E}$ allows us to determine the right kernel of the row rank support of the error.

Lemma 3.17. *Let $\boldsymbol{E} \in \mathbb{F}_{q^m}^{u\times n}$ be an error with $\text{rk}_q(\boldsymbol{E}) = \text{rk}_{q^m}(\boldsymbol{E}) = t$, and let the integer $u \geq t$. Then, we have $\mathcal{K}_{q^m}(\boldsymbol{E}) = \mathcal{K}_{q^m}(\boldsymbol{B})$, where $\boldsymbol{B}$ is a basis of the row rank support of $\boldsymbol{E}$.*

Proof. Let $\boldsymbol{A} \in \mathbb{F}_{q^m}^{u\times t}$ with $\text{rk}_q(\boldsymbol{A}) = t$ and $\boldsymbol{B} \in \mathbb{F}_{q}^{t\times n}$ with $\text{rk}_q(\boldsymbol{B}) = t$ such that $\boldsymbol{E} = \boldsymbol{A}\boldsymbol{B}$. Since the $\mathbb{F}_{q^m}$-rank of $\boldsymbol{E}$ is equal to t, it follows that $\boldsymbol{A}$ has full $\mathbb{F}_{q^m}$-rank and since $u \geq t$, the kernel $\mathcal{K}_{q^m}(\boldsymbol{A}) = \boldsymbol{0}$. Therefore, for all vectors $\boldsymbol{v} \in \mathbb{F}_{q^m}^{n}$, it holds that $\boldsymbol{B}\boldsymbol{v}^\top = \boldsymbol{0}$ if and only if $(\boldsymbol{A}\boldsymbol{B})\boldsymbol{v}^\top = \boldsymbol{0}$, which implies that $\mathcal{K}_{q^m}(\boldsymbol{E}) = \mathcal{K}_{q^m}(\boldsymbol{B})$. ■

To prove that the row rank support of the error can be determined from $\boldsymbol{H}_{\text{sub}}$, we need the following property of the matrix $\text{ext}_{q^m/q}(\boldsymbol{H}_{\text{sub}})$.

Lemma 3.18. *Let $\boldsymbol{H}_{\text{sub}}$ be a matrix in $\mathbb{F}_{q^m}^{(n-k-t)\times n}$, and let $\boldsymbol{h}$ be a vector in $\mathcal{R}_{q^m}(\boldsymbol{H}_{\text{sub}})$. Then, each row of the matrix $\text{ext}_{q^m/q}(\boldsymbol{h}) \in \mathbb{F}_{q}^{m\times n}$ is in $\mathcal{R}_q(\text{ext}_{q^m/q}(\boldsymbol{H}_{\text{sub}}))$.*

Proof. Since $\boldsymbol{h}$ is in the $\mathbb{F}_{q^m}$-rowspace of $\boldsymbol{H}_{\text{sub}}$, the vector $\boldsymbol{h}$ can be written as a $\mathbb{F}_{q^m}$-linear combination of the rows of $\boldsymbol{H}_{\text{sub}}$, i.e., $\boldsymbol{h} = \sum_{i=1}^{n-k-t} a_i \boldsymbol{H}_{\text{sub},i}$, where $a_1, \dots, a_{n-k-t} \in \mathbb{F}_{q^m}$ and $\boldsymbol{H}_{\text{sub},i}$ is the i-th row of the matrix $\boldsymbol{H}_{\text{sub}}$. This expression can be mapped to $\text{ext}_{q^m/q}(\boldsymbol{h}) = \sum_{i=1}^{m} \boldsymbol{M}_{a_i} \text{ext}_{q^m/q}(\boldsymbol{H}_{\text{sub},i})$, where $\boldsymbol{M}_{a_i} \in \mathbb{F}_{q}^{m\times m}$ is the matrix representation of a_i over $\mathbb{F}_q$ for a given basis $\boldsymbol{\gamma}$ [166]. Since the entries of $\boldsymbol{M}_{a_1}, \dots, \boldsymbol{M}_{a_m}$ are in $\mathbb{F}_q$, each row of $\text{ext}_{q^m/q}(\boldsymbol{h})$ must be in the $\mathbb{F}_q$-rowspace of $\text{ext}_{q^m/q}(\boldsymbol{H}_{\text{sub}})$. ■

The previous lemmata enable us to prove the following theorem, which states that the row rank support of $\boldsymbol{E}$ can be determined from $\boldsymbol{H}_{\text{sub}}$ if $u \geq t$ and $\text{rk}_{q^m}(\boldsymbol{E}) = t$. An illustration of the theorem is given in Figure 3.6.

Theorem 3.19. *Let $\boldsymbol{H} \in \mathbb{F}_{q^m}^{(n-k)\times n}$ be a parity-check matrix of an $[n, k, d_{\min}^{\text{R}}]_{\mathbb{F}_{q^m}}^{\text{R}}$ code, let $\boldsymbol{E} \in \mathbb{F}_{q^m}^{u\times n}$ be an error with $\text{rk}_q(\boldsymbol{E}) = \text{rk}_{q^m}(\boldsymbol{E}) = t \leq d_{\min}^{\text{R}} - 2$, and let $u \geq t$. Then, the row rank support $\text{supp}_{\text{R}}^{(\text{R})}(\boldsymbol{E}) = \mathcal{K}_q(\text{ext}_{q^m/q}(\boldsymbol{H}_{\text{sub}}))$, where $\boldsymbol{H}_{\text{sub}}$ is defined as in Lemma 3.16.*

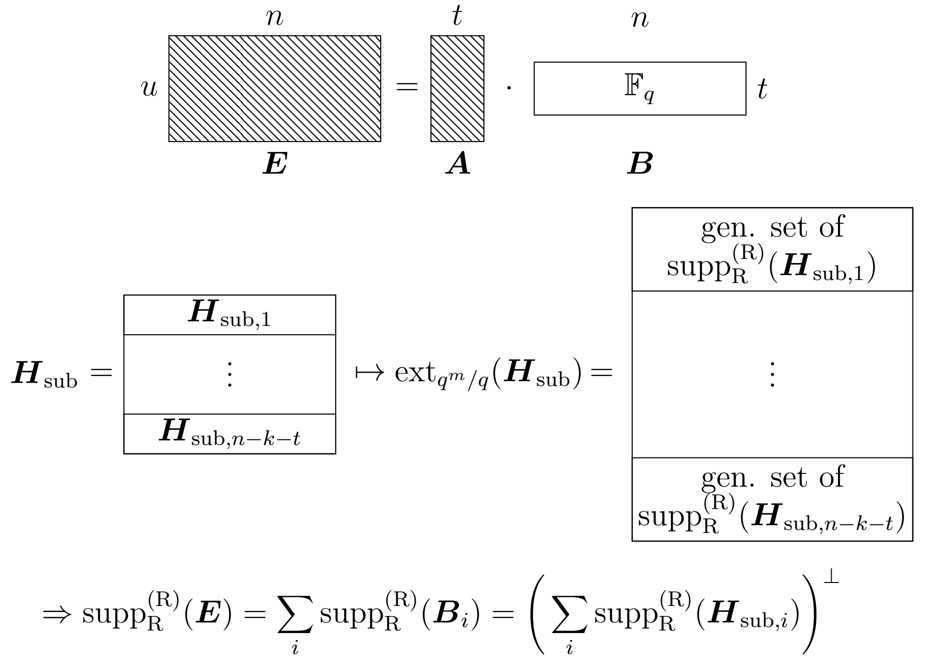

n
t
n
u
$=$
$\cdot$
$\mathbb{F}_q$
t
$\boldsymbol{E}$
$\boldsymbol{A}$
$\boldsymbol{B}$
$\boldsymbol{H}_{\mathrm{sub}} =$
$\boldsymbol{H}_{\mathrm{sub},1}$
$\vdots$
$\boldsymbol{H}_{\mathrm{sub},n-k-t}$
$\mapsto \mathrm{ext}_{q^m/q}(\boldsymbol{H}_{\mathrm{sub}}) =$
gen. set of $\mathrm{supp}_{\mathrm{R}}^{(\mathrm{R})}(\boldsymbol{H}_{\mathrm{sub},1})$
$\vdots$
gen. set of $\mathrm{supp}_{\mathrm{R}}^{(\mathrm{R})}(\boldsymbol{H}_{\mathrm{sub},n-k-t})$
$\Rightarrow \mathrm{supp}_{\mathrm{R}}^{(\mathrm{R})}(\boldsymbol{E}) = \sum_i \mathrm{supp}_{\mathrm{R}}^{(\mathrm{R})}(\boldsymbol{B}_i) = \left(\sum_i \mathrm{supp}_{\mathrm{R}}^{(\mathrm{R})}(\boldsymbol{H}_{\mathrm{sub},i})\right)^{\perp}$

Figure 3.6: Illustration of Theorem 3.19.

Proof. From Lemma 3.16 and Lemma 3.17 follows that

$$\mathcal{R}_{q^m}(\boldsymbol{H}_{\mathrm{sub}}) = \mathcal{K}_{q^m}(\boldsymbol{B}) \cap \mathcal{C}^{\perp}. \tag{3.8}$$

In the following, we show that $\mathcal{R}_q(\mathrm{ext}_{q^m/q}(\boldsymbol{H}_{\mathrm{sub}})) = \mathcal{K}_q(\boldsymbol{B})$. First, we prove that the $\mathbb{F}_q$-linear row space of $\mathrm{ext}_{q^m/q}(\boldsymbol{H}_{\mathrm{sub}})$ is a subspace of $\mathcal{K}_q(\boldsymbol{B})$, where it is sufficient to show that any row of $\mathrm{ext}_{q^m/q}(\boldsymbol{H}_{\mathrm{sub}})$ is in $\mathcal{K}_q(\boldsymbol{B})$. Denote such a row of $\mathrm{ext}_{q^m/q}(\boldsymbol{H}_{\mathrm{sub}})$ by $\boldsymbol{v}_i$, and let $\boldsymbol{v}_i$ also be a row of $\mathrm{ext}_{q^m/q}(\boldsymbol{v})$, where $\boldsymbol{v} \in \mathcal{R}_{q^m}(\boldsymbol{H}_{\mathrm{sub}})$. Since $\boldsymbol{v}_i \in \mathbb{F}_q^n$ and $\boldsymbol{v} \in \mathcal{K}_{q^m}(\boldsymbol{B})$ (see (3.8)) it follows that $\mathrm{ext}_{q^m/q}(\boldsymbol{B}\boldsymbol{v}^\top) = \boldsymbol{B}\,\mathrm{ext}_{q^m/q}(\boldsymbol{v})^\top = \boldsymbol{0}$, and thus, $\boldsymbol{B}\boldsymbol{v}_i^\top = 0$.

Second, we prove that $\mathcal{K}_q(\boldsymbol{B}) \subseteq \mathcal{R}_q(\mathrm{ext}_{q^m/q}(\boldsymbol{H}_{\mathrm{sub}}))$ by showing that

$$r := \dim\left(\mathcal{R}_q(\mathrm{ext}_{q^m/q}(\boldsymbol{H}_{\mathrm{sub}}))\right) = \dim(\mathcal{K}_q(\boldsymbol{B})) = n - t.$$

Because $\mathcal{R}_q(\mathrm{ext}_{q^m/q}(\boldsymbol{H}_{\mathrm{sub}})) \subseteq \mathcal{K}_q(\boldsymbol{B})$, we observe that $r > n-t$ is not possible, and it is left to show that $r < n-t$ is not possible either. Let $\boldsymbol{h}_1, \dots, \boldsymbol{h}_r \in \mathbb{F}_q^n$ form a basis of $\mathcal{R}_q(\mathrm{ext}_{q^m/q}(\boldsymbol{H}_{\mathrm{sub}}))$ and assume $r < n-t$. From the basis extension theorem follows that there must be a matrix $\boldsymbol{B}'' \in \mathbb{F}_q^{(n-t)\times n}$ such that $\boldsymbol{B}' := \left[\boldsymbol{B}^\top, \boldsymbol{B}''^\top\right] \in \mathbb{F}_q^{n\times n}$ and

$\mathrm{rk}_q(\boldsymbol{B}') = n$. Since $\mathrm{rk}_q(\boldsymbol{B}') = n$, it follows that the matrix

$$\boldsymbol{H}' = \begin{bmatrix} \boldsymbol{h}'_1 \\ \vdots \\ \boldsymbol{h}'_r \end{bmatrix} := \begin{bmatrix} \boldsymbol{h}_1 \\ \vdots \\ \boldsymbol{h}_r \end{bmatrix} \boldsymbol{B}' = \begin{bmatrix} \boldsymbol{h}_1\boldsymbol{B}^\top & \boldsymbol{h}_1\boldsymbol{B}''^\top \\ \vdots & \vdots \\ \boldsymbol{h}_r\boldsymbol{B}^\top & \boldsymbol{h}_r\boldsymbol{B}''^\top \end{bmatrix} \in \mathbb{F}_q^{r\times n}$$

has $\mathbb{F}_q$-rank r. Due to $\boldsymbol{h}_1, \dots, \boldsymbol{h}_r \in \mathcal{K}_q(\boldsymbol{B})$, we have

$$\boldsymbol{H}' = \begin{bmatrix} 0 & \dots & 0 & h'_{1,t+1} & h'_{1,t+2} & \dots & h'_{1,n} \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & h'_{r,t+1} & h'_{1,t+2} & \dots & h'_{r,n} \end{bmatrix} \in \mathbb{F}_q^{r\times n}.$$

By assumption, we have $r < n-t$, and thus, there is a full $\mathbb{F}_q$-rank matrix $\left[\boldsymbol{I}^\top \boldsymbol{J}^\top\right]^\top \in \mathbb{F}_q^{n\times n}$ such that

$$\tilde{\boldsymbol{H}} := \boldsymbol{H}' \begin{bmatrix} \boldsymbol{I} \\ \boldsymbol{J} \end{bmatrix} = \begin{bmatrix} 0 & \dots & 0 & 0 & h'_{1,t+2} & \dots & h'_{1,n} \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & 0 & h'_{r,t+2} & \dots & h'_{r,n} \end{bmatrix} \in \mathbb{F}_q^{r\times n}, \tag{3.9}$$

where $\boldsymbol{I} \in \mathbb{F}_q^{t\times n}$ denotes a matrix that consists of an identity matrix in the first t columns and a zero matrix in the last $n-t$ columns, and $\boldsymbol{J} \in \mathbb{F}_q^{(n-t)\times n}$ has full $\mathbb{F}_q$-rank. Furthermore, the matrix

$$\boldsymbol{D} := \left[\boldsymbol{B}^\top \boldsymbol{B}''^\top\right] \begin{bmatrix} \boldsymbol{I} \\ \boldsymbol{J} \end{bmatrix} \in \mathbb{F}_q^{n\times n}$$

has $\mathbb{F}_q$-rank n, which implies that

$$\tilde{\boldsymbol{H}} = \begin{bmatrix} \boldsymbol{h}_1 \\ \vdots \\ \boldsymbol{h}_r \end{bmatrix} \boldsymbol{D} = \begin{bmatrix} \boldsymbol{h}_1\boldsymbol{D} \\ \vdots \\ \boldsymbol{h}_r\boldsymbol{D} \end{bmatrix} \in \mathbb{F}_q^{r\times n}$$

has $\mathbb{F}_q$-rank r. It follows from (3.9) that

$$\boldsymbol{h}_i\boldsymbol{D}' = [0\ \dots\ 0\ 0] \in \mathbb{F}_q^{t+1}, \tag{3.10}$$

for $i \in [1\!:\!r]$, where the matrix $\boldsymbol{D}' := \boldsymbol{D}_{:,[1:t+1]} \in \mathbb{F}_q^{n\times(t+1)}$ has $\mathbb{F}_q$-rank $t+1$. Thus, for all $\tilde{\boldsymbol{h}} \in \mathcal{R}_q(\mathrm{ext}_{q^m/q}(\boldsymbol{H}_{\mathrm{sub}}))$, it holds that $\tilde{\boldsymbol{h}}\boldsymbol{D}' = \boldsymbol{0}$.

However, since $\boldsymbol{H} \in \mathbb{F}_{q^m}^{(n-k)\times n}$ is a parity-check matrix of an $[n,k,d_{\min}^{\mathrm{R}}]_{\mathbb{F}_{q^m}}^{\mathrm{R}}$ code and $\mathrm{rk}_q(\boldsymbol{D}')$ is equal to $t+1 \leq d_{\min}^{\mathrm{R}} - 1$, we observe that $\mathrm{rk}_{q^m}(\boldsymbol{H}\boldsymbol{D}') = t+1$ [86, Theorem 1], and there must be a vector $\boldsymbol{g} \in \mathcal{R}_{q^m}(\boldsymbol{H})$ such that

$$\boldsymbol{g}\boldsymbol{D}' = [0 \ \dots \ 0 \ g'_{t+1}] \in \mathbb{F}_{q^m}^{t+1}, \tag{3.11}$$

where $g'_{t+1} \in \mathbb{F}_{q^m} \setminus \{0\}$. Since, $\boldsymbol{B}' = \left[\boldsymbol{B}^\top, \boldsymbol{B}''^\top\right]$ has full $\mathbb{F}_q$-rank, the columns of $\boldsymbol{B}^\top$ and $\boldsymbol{B}''^\top$ are $\mathbb{F}_q$-linearly independent. Thus, $\boldsymbol{g}\boldsymbol{D}'_{:,[1:t]} = \boldsymbol{g}(\boldsymbol{B}^\top + \boldsymbol{B}''^\top \boldsymbol{J}_{:,[1:t]}) = \boldsymbol{0}$ implies that $\boldsymbol{g} \in \mathcal{K}_{q^m}(\boldsymbol{B})$ and $\boldsymbol{g} \in \mathcal{K}_{q^m}(\boldsymbol{B}'')$. Thus, according to (3.8),

$$\boldsymbol{g} \in \mathcal{K}_{q^m}(\boldsymbol{B}) \cap \mathcal{R}_{q^m}(\boldsymbol{H}) = \mathcal{K}_{q^m}(\boldsymbol{B}) \cap \mathcal{C}^\perp = \mathcal{R}_{q^m}(\boldsymbol{H}_{\mathrm{sub}}).$$

Then, (3.11) implies that

$$\mathrm{ext}_{q^m/q}(\boldsymbol{g})\boldsymbol{D}' = \mathrm{ext}_{q^m/q}(\boldsymbol{g}\boldsymbol{D}') = \begin{bmatrix} 0 & \dots & 0 & g'_{1,t+1} \\ 0 & \dots & 0 & g'_{2,t+1} \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & 0 & g'_{m,t+1} \end{bmatrix} \in \mathbb{F}_q^{m\times(t+1)}, \tag{3.12}$$

where $\mathrm{ext}_{q^m/q}(g'_{t+1}) = [g'_{1,t+1}, \dots, g'_{m,t+1}]^\top \in \mathbb{F}_q^{m\times 1}$. If $\boldsymbol{g}_i$ is the i-th row of the matrix $\mathrm{ext}_{q^m/q}(\boldsymbol{g})$ with $g'_{i,t+1} \neq 0$, then $\boldsymbol{g}_i\boldsymbol{D}' = [0,\dots,0,g'_{i,t+1}] \neq \boldsymbol{0}$, see (3.12). This constitutes a contradiction since we have on the one hand that $\boldsymbol{g}_i$ is in the $\mathbb{F}_q$-linear row space of $\mathrm{ext}_{q^m/q}(\boldsymbol{H}_{\mathrm{sub}})$ according to Lemma 3.18, and we have on the other hand that for all $\boldsymbol{g}_i \in \mathcal{R}_q(\mathrm{ext}_{q^m/q}(\boldsymbol{H}_{\mathrm{sub}}))$, it must hold that $\boldsymbol{g}_i\boldsymbol{D}' = \boldsymbol{0}$ according to (3.10). It follows that $r < n-t$ is not possible, and thus, $\mathcal{R}_q(\mathrm{ext}_{q^m/q}(\boldsymbol{H}_{\mathrm{sub}})) = \mathcal{K}_q(\boldsymbol{B})$, which is equivalent to $\mathrm{supp}_{\mathrm{R}}^{(\mathrm{R})}(\boldsymbol{E}) = \mathcal{R}_q(\boldsymbol{B}) = \mathcal{K}_q(\mathrm{ext}_{q^m/q}(\boldsymbol{H}_{\mathrm{sub}}))$. ■

The previous statements allow us to prove the correctness and the complexity of Algorithm 6.

Theorem 3.20. *Let $\boldsymbol{H} \in \mathbb{F}_{q^m}^{(n-k)\times n}$ be a parity-check matrix of an $[n,k,d_{\min}^{\mathrm{R}}]_{\mathbb{F}_{q^m}}^{\mathrm{R}}$ code, let t and u be integers such that $0 \leq t \leq \min\{u, d_{\min}^{\mathrm{R}}-2\}$, let $\boldsymbol{E} \in \mathbb{F}_{q^m}^{u\times n}$ be an error matrix with $\mathrm{rk}_q(\boldsymbol{E}) = \mathrm{rk}_{q^m}(\boldsymbol{E}) = t$, and let the syndrome matrix $\boldsymbol{S} = \boldsymbol{H}\boldsymbol{E}^\top \in \mathbb{F}_{q^m}^{(n-k)\times u}$. Then, given $\boldsymbol{H}$, $\boldsymbol{S}$, and t, Algorithm 6 returns a matrix $\boldsymbol{E}'$ such that $\mathrm{rk}_q(\boldsymbol{E}') = \mathrm{rk}_{q^m}(\boldsymbol{E}') \leq t$ and $\boldsymbol{S} = \boldsymbol{H}\boldsymbol{E}'^\top$ in $O(\max\{n^3, n^2u\}m^2)$ operations in $\mathbb{F}_q$.*

Proof. Algorithm 6 determines $\boldsymbol{E}' = \boldsymbol{A}'\boldsymbol{B}'$ in two steps. First, it computes a basis of the row rank support of the error in Lines 1–3, and then, it applies erasure decoding to obtain the matrix $\boldsymbol{A}'$ in Line 4.

To obtain the row space of $\boldsymbol{B}'$, the algorithm determines a transformation matrix $\boldsymbol{P}$ such that $\boldsymbol{PS}$ is in reduced row echelon form in Line 1 and chooses the matrix $\boldsymbol{H}_{\mathrm{sub}}$ as the last $n-k-t$ rows of $\boldsymbol{PH}$ in Line 2. Then, the algorithm computes $\boldsymbol{B}'$ by finding a basis of $\mathcal{K}_q(\mathrm{ext}_{q^m/q}(\boldsymbol{H}_{\mathrm{sub}}))$, which is possible due to Theorem 3.19. To determine $\boldsymbol{A}'$, the algorithm solves the linear system of equations $\boldsymbol{S} = (\boldsymbol{H}\boldsymbol{B}'^{\top})\boldsymbol{A}'^{\top}$ for $\boldsymbol{A}'$ in Line 4, which has a unique solution according to Lemma 3.15. Finally, Algorithm 6 returns $\boldsymbol{E}' = \boldsymbol{A}'\boldsymbol{B}'$ in Line 6.

Algorithm 6 is only based on linear operations, where Line 4 requires

$$O(\max\{n^3, n^2u\}m^2)$$

operations in $\mathbb{F}_q$ and determines the complexity of the algorithm, as it is the most expensive step. ■

3.2.2 Further Results and Remarks

In the following, we discuss some properties of the proposed decoding algorithm and compare it to other known decoding algorithms.

Generalization of Problem 3.4

Let $\mathcal{C}^{(u)}$ be a $[u; n, k]_{\mathbb{F}_{q^m}}$ interleaved code with minimum rank distance $d_{\min}^{\mathrm{R}}$, and let $\boldsymbol{R} = \boldsymbol{C} + \boldsymbol{E} \in \mathbb{F}_{q^m}^{u \times n}$, where $\boldsymbol{C} \in \mathcal{C}^{(u)}$ and $\mathrm{rk}_q(\boldsymbol{E}) = t$ with $t \leq \min\{u, d_{\min}^{\mathrm{R}} - 2\}$. If $\mathrm{rk}_{q^m}(\boldsymbol{E}) = t$, then it is well-known that decoding $\boldsymbol{R}$ in $\mathcal{C}^{(u)}$ can be reduced to the problem $\mathsf{SeaISD}_{\mathrm{R}}$. However, in many channel models, the error $\boldsymbol{E}$ is sampled uniformly from the set $\{\boldsymbol{X} \in \mathbb{F}_{q^m}^{u \times n} : \mathrm{rk}_q(\boldsymbol{X}) = t\}$, which implies that $\mathrm{rk}_{q^m}(\boldsymbol{E})$ could be smaller than t. In the following theorem, we derive a lower bound on the probability that our proposed algorithm also succeeds for this error model.

Theorem 3.21. *Let $\mathcal{C}^{(u)}$ be a $[u; n, k]_{\mathbb{F}_{q^m}}$ code with minimum rank distance $d_{\min}^{\mathrm{R}}$, let the integer $t \leq \min\{u, d_{\min}^{\mathrm{R}} - 2\}$, and let $\boldsymbol{R} = \boldsymbol{C} + \boldsymbol{E}$, where $\boldsymbol{C} \in \mathcal{C}^{(u)}$ and $\boldsymbol{E} \xleftarrow{\$} \{\boldsymbol{X} \in \mathbb{F}_{q^m}^{u \times n} : \mathrm{rk}_q(\boldsymbol{X}) = t\}$. Then, the codeword $\boldsymbol{C}$ can be uniquely reconstructed* *from* $\boldsymbol{R}$ *using Algorithm 6 with a probability of at least*

$$\prod_{i=0}^{t-1}(1-q^{m(i-u)}) \geq 1 - tq^{m(t-1-u)}.$$

Proof. From Theorem 3.20 follows that if $\mathrm{rk}_{q^m}(\boldsymbol{E}) = t$, then $\boldsymbol{C}$ can be uniquely reconstructed from $\boldsymbol{R}$ using Algorithm 6. Let $\boldsymbol{A} \in \mathbb{F}_{q^m}^{u\times t}$ and $\boldsymbol{B} \in \mathbb{F}_q^{t\times n}$ such that $\boldsymbol{E} = \boldsymbol{AB}$. Then, the matrix $\boldsymbol{E}$ has $\mathbb{F}_{q^m}$-rank t if and only if $\mathrm{rk}_{q^m}(\boldsymbol{A}) = t$. Furthermore, let $\boldsymbol{B}$ be fixed and let the rows of $\boldsymbol{B}$ be a basis of the subspace $\mathcal{V} \subseteq \mathbb{F}_q^n$. Then, the function $\boldsymbol{A} \mapsto \boldsymbol{AB}$ bijectively maps from $\{\boldsymbol{A} \in \mathbb{F}_{q^m}^{u\times t} : \mathrm{rk}_{q^m}(\boldsymbol{A}) = t\}$ to the set of errors $\boldsymbol{E} \in \mathbb{F}_{q^m}^{u\times n}$ with row rank support $\mathcal{V}$. It follows that sampling $\boldsymbol{E}$ uniformly from $\{\boldsymbol{X} \in \mathbb{F}_{q^m}^{u\times n} : \mathrm{rk}_q(\boldsymbol{X}) = t\}$ is equivalent to choosing $\boldsymbol{E} = \boldsymbol{AB}$, where $\boldsymbol{B} \xleftarrow{\$} \{\boldsymbol{X} \in \mathbb{F}_q^{t\times n} : \mathrm{rk}_q(\boldsymbol{X}) = t\}$ and $\boldsymbol{A} \xleftarrow{\$} \{\boldsymbol{X} \in \mathbb{F}_{q^m}^{u\times t} : \mathrm{rk}_q(\boldsymbol{X}) = t\}$. Thus, we have

$$\Pr(\mathrm{rk}_{q^m}(\boldsymbol{E}) = t) = \Pr(\mathrm{rk}_{q^m}(\boldsymbol{A}) = t) \geq \prod_{i=0}^{t-1}(1-q^{m(i-u)}) \geq 1 - tq^{m(t-1-u)},$$

see [167, Lemma 3.13]. ■

Decoding Beyond $d_{\min}^{\mathrm{R}} - 2$

It is shown [168] that certain types of locally repairable codes are able to correct most errors of Hamming weight up to $n-k-1$ by high-order interleaving and using the Hamming-metric variant of the proposed algorithm. The key observation in [168] is that the upper bound on the Hamming weight of the error can be relaxed into a weaker condition on the error positions. For the codes in [168], most error patterns of weight at most $n-k-1$ fulfill this condition. Likewise, in the rank metric, we can replace the condition $t \leq d_{\min}^{\mathrm{R}} - 2$ by

$$\mathrm{rk}_{q^m}\left(\boldsymbol{H}\left[\boldsymbol{B}^\top, \boldsymbol{b}^\top\right]\right) = t+1 \quad \forall\, \boldsymbol{b} \in \mathbb{F}_q^n \setminus \mathcal{R}_q(\boldsymbol{B}), \tag{3.13}$$

see the proof of Theorem 3.19. Therefore, the proposed decoder is able to correct errors of rank weight $t > d_{\min}^{\mathrm{R}} - 2$ if (3.13) holds and if $u \geq t$ and $\mathrm{rk}_{q^m}(\boldsymbol{E}) = t$ is fulfilled.

Generic Decoding of $\mathbb{F}_{q^{mu}}$-Linear Codes in the Rank Metric

It is shown in [169] that an interleaved RS code can be viewed as an RS code over an extension field. Similarly, a u-interleaved code with an $\mathbb{F}_{q^m}$-linear constituent code, can be viewed as an $\mathbb{F}_{q^{mu}}$-linear code in $\mathbb{F}_{q^{mu}}^n$ with the same minimum rank distance as the constituent code. Therefore, for large u, the results shown in this section imply that the proposed decoder can be used to efficiently correct up to $d_{\min}^{\mathrm{R}} - 2$ errors in any $\mathbb{F}_{q^{mu}}$-linear code with high probability.

Decoding High-Order Interleaved Gabidulin Codes

Let $\mathcal{G}_k^{(u)}(\boldsymbol{g})$ be a $[u; n, k]_{\mathbb{F}_{q^m}}$ Gabidulin code with minimum rank distance $d_{\min}^{\mathrm{R}}$, and let $\boldsymbol{R} = \boldsymbol{C} + \boldsymbol{E} \in \mathbb{F}_{q^m}^{u \times n}$, where $\boldsymbol{C} \in \mathcal{G}_k^{(u)}(\boldsymbol{g})$ and $\mathrm{rk}_q(\boldsymbol{E}) = \mathrm{rk}_{q^m}(\boldsymbol{E}) = t$ with $t \leq \min\{u, d_{\min}^{\mathrm{R}} - 2\}$. Then, it is well-known that decoding $\boldsymbol{R}$ in $\mathcal{G}_k^{(u)}(\boldsymbol{g})$ is equivalent to solving $\mathsf{SeaISD}_{\mathrm{R}}$, where the input matrix $\boldsymbol{H}$ is chosen to be a parity-check matrix of $\mathcal{G}_k^{(u)}(\boldsymbol{g})$ and the input matrix $\boldsymbol{S} = \boldsymbol{H}\boldsymbol{R}^\top$. We show in the next theorem that there are multiple algorithms known to solve this particular instance of $\mathsf{SeaISD}_{\mathrm{R}}$.

Theorem 3.22. *Let t and u be integers such that $0 \leq t \leq u$, let $\boldsymbol{C}$ be a codeword of an $[u; n, k]_{\mathbb{F}_{q^m}}$ Gabidulin code $\mathcal{G}_k^{(u)}(\boldsymbol{g})$ with minimum rank distance $d_{\min}^{\mathrm{R}}$, and let $\boldsymbol{E} \in \mathbb{F}_{q^m}^{u \times n}$ be an error matrix with $\mathrm{rk}_q(\boldsymbol{E}) = \mathrm{rk}_{q^m}(\boldsymbol{E}) = t \leq d_{\min}^{\mathrm{R}} - 2$. Then, $\boldsymbol{C}$ can be uniquely retrieved from $\boldsymbol{R} = \boldsymbol{C} + \boldsymbol{E}$ using the Loidreau–Overbeck [105], Sidorenko–Bossert[7] [108], or Wachter-Zeh–Zeh [116] decoder.*

Proof. To prove that all three algorithms succeed under the given conditions, it is sufficient to show that the Loidreau–Overbeck decoder succeeds, as this implies that the other two algorithms also succeed [99, Lemma 4.1], [99, Lemma 4.8].

As proven in [105], the Loidreau–Overbeck decoder retrieves $\boldsymbol{C}$ from $\boldsymbol{R}$ uniquely if

[7] In order to work with the error model assumed here, we must use the variant described in [99, Section 4.1].

and only if

$$\mathrm{rk}_{q^m}\left(\begin{bmatrix} \boldsymbol{g}^{[0]} \\ \boldsymbol{g}^{[1]} \\ \vdots \\ \boldsymbol{g}^{[n-t-2]} \\ \boldsymbol{E}^{[0]} \\ \boldsymbol{E}^{[1]} \\ \vdots \\ \boldsymbol{E}^{[n-k-t-1]} \end{bmatrix}\right) = n-1,$$

where the superscript $[i]$ indicates that each entry of this vector or of this matrix is raised to the power of q^i.

To show that this holds, we prove that the submatrix $\tilde{\boldsymbol{G}} := \left[\boldsymbol{g}^{[0]\top} \boldsymbol{g}^{[1]\top} \dots \boldsymbol{g}^{[n-t-2]\top}\right]^\top$ has $\mathbb{F}_{q^m}$-rank $n-t-1$ and that the submatrix $\boldsymbol{Z} := \left[\boldsymbol{E}^{[0]\top} \boldsymbol{E}^{[1]\top} \dots \boldsymbol{E}^{[n-k-t-1]\top}\right]^\top$ has $\mathbb{F}_{q^m}$-rank t. Then, it is left to show that the rows of $\tilde{\boldsymbol{G}}$ and the rows of $\boldsymbol{Z}$ are linearly independent.

First, we observe that $\tilde{\boldsymbol{G}}$ is a generator matrix of an $[n, n-t-1, t+2]^{\mathrm{R}}_{\mathbb{F}_{q^m}}$ Gabidulin code which implies that $\mathrm{rk}_{q^m}(\tilde{\boldsymbol{G}}) = n-t-1$ and each $\mathbb{F}_{q^m}$-linear combination of $\boldsymbol{g}, \dots, \boldsymbol{g}^{[n-t-2]}$ has a rank weight of at least $t+2$.

Second, since $\mathrm{rk}_{q^m}(\boldsymbol{E}) = t$, it follows that $\mathrm{rk}_{q^m}(\boldsymbol{Z}) \geq t$. Furthermore, from $\mathrm{rk}_q(\boldsymbol{E}) = t$ follows that there must be a full-rank matrix $\boldsymbol{P} \in \mathbb{F}_q^{n \times n}$ such that

$$\boldsymbol{EP} = \begin{bmatrix} \tilde{\boldsymbol{E}} & \boldsymbol{0}_{u,n-t} \end{bmatrix},$$

where $\tilde{\boldsymbol{E}} \in \mathbb{F}_{q^m}^{u \times t}$, and $\boldsymbol{0}_{u,n-t}$ denotes the $u \times (n-t)$ zero matrix. Since $\tilde{\boldsymbol{E}}$ has t columns,

$$\begin{aligned} \mathrm{rk}_{q^m}(\boldsymbol{Z}) &= \mathrm{rk}_{q^m}(\boldsymbol{ZP}) \\ &= \mathrm{rk}_{q^m}\left(\begin{bmatrix} \tilde{\boldsymbol{E}} & \boldsymbol{0}_{u,n-t} \\ \tilde{\boldsymbol{E}}^{[1]} & \boldsymbol{0}_{u,n-t} \\ \vdots & \vdots \\ \tilde{\boldsymbol{E}}^{[n-k-t-1]} & \boldsymbol{0}_{u,n-t} \end{bmatrix}\right) \\ &\leq t. \end{aligned}$$

This implies that $\mathrm{rk}_{q^m}(\boldsymbol{Z}) = t$, and each $\mathbb{F}_{q^m}$-linear combination has a rank weight of at most t. We conclude that each $\mathbb{F}_{q^m}$-linear combination of the rows of $\tilde{\boldsymbol{G}}$ has a rank weight of at least $t+2$, and each linear $\mathbb{F}_{q^m}$-linear combination of the rows of $\boldsymbol{Z}$ has a rank weight of at most t. Thus, the rows of $\tilde{\boldsymbol{G}}$ and the rows of $\boldsymbol{Z}$ must be linearly independent. ■

Theorem 3.22 shows that for high-order interleaved Gabidulin codes, the proposed generic decoding algorithm has the same error correction capability as all currently known decoding algorithms which are tailored to interleaved Gabidulin codes.

3.3 Decoding of Gabidulin Codes Beyond the Unique Decoding Radius

In this section, we investigate the problem of decoding Gabidulin codes, where the rank weight of the additive errors is larger than half the minimum distance of the respective Gabidulin codes. We define the decisional version of this problem as follows:

Problem 3.5 (Decisional Gabidulin Decoding (DecGab) Problem).

Given:
- *Parity-check matrix* $\boldsymbol{H} \in \mathbb{F}_{q^m}^{(n-k)\times n}$ *of a Gabidulin code* $\mathcal{G}_k(\boldsymbol{g}) \subset \mathbb{F}_{q^m}^n$
- *Non-negative integer* t *with* $\frac{n-k}{2} < t < n-k$
- *Vector* $\boldsymbol{s} \in \mathbb{F}_{q^m}^{n-k}$

Question: *Is there an* $\boldsymbol{e} \in \mathbb{F}_{q^m}^n$ *such that* $\mathrm{wt}_\mathrm{R}(\boldsymbol{e}) \leq t$ *and* $\boldsymbol{s} = \boldsymbol{e}\boldsymbol{H}^\top$*?*

As for the previous problems, we solve the decisional problem DecGab by trying to find a solution to the associated *search* problem.[8]

Problem 3.6 (Search Gabidulin Decoding (SeaGab) Problem).

Given:
- *Parity-check matrix* $\boldsymbol{H} \in \mathbb{F}_{q^m}^{(n-k)\times n}$ *of a Gabidulin code* $\mathcal{G}_k(\boldsymbol{g}) \subset \mathbb{F}_{q^m}^n$
- *Non-negative integer* t *with* $\frac{n-k}{2} < t < n-k$
- *Vector* $\boldsymbol{r} = \boldsymbol{c} + \boldsymbol{e} \in \mathbb{F}_{q^m}^n$*, where* $\boldsymbol{c}\boldsymbol{H}^\top = \boldsymbol{0}$ *and* $\mathrm{wt}_\mathrm{R}(\boldsymbol{e}) = t$

Objective: *Search for an* $\boldsymbol{e}' \in \mathbb{F}_{q^m}^n$ *such that* $\mathrm{wt}_\mathrm{R}(\boldsymbol{e}') \leq t$ *and* $(\boldsymbol{r} - \boldsymbol{e}')\boldsymbol{H}^\top = \boldsymbol{0}$.

In the following, we provide new details about the complexity of the latter problem by proposing a Las Vegas-type algorithm and analyzing the work factor of the algorithm.

3.3.1 A New Algorithm for Solving SeaGab

In Problem 3.6, the vector $\boldsymbol{r}$ can be interpreted as the sum of a codeword $\boldsymbol{c}$ of the Gabidulin code $\mathcal{G}_k(\boldsymbol{g})$ and an error $\boldsymbol{e}$ with $\mathrm{wt}_\mathrm{R}(\boldsymbol{e}) = t =: \xi + \frac{n-k}{2} > \frac{n-k}{2}$. Furthermore, we do not have any knowledge about the row or the column rank support of the error, which implies that the known decoders are not able to decode $\boldsymbol{r}$ in $\mathcal{G}_k(\boldsymbol{g})$ efficiently. The idea of our proposed algorithm is thus as follows: We repeatedly guess parts

[8] Note that Problem 3.5 has a vector $\boldsymbol{s}$ as input, which can be interpreted as a syndrome, whereas Problem 3.6 has a vector $\boldsymbol{r}$ as input, which can be seen as received vector, i.e., a codeword that is corrupted by an error of rank weight t. Thus, Problem 3.5 can be solved by trying to find a solution to Problem 3.6, where the input $\boldsymbol{r}$ of Problem 3.6 is chosen to be a vector from the set $\{\boldsymbol{x} \in \mathbb{F}_{q^m}^n : \boldsymbol{x}\boldsymbol{H}^\top = \boldsymbol{s}\}$. We use this alternative definition of Problem 3.6, as it simplifies the derivation and the analysis of our proposed algorithm.

of the row and/or the column rank support of the vector $\boldsymbol{e}$ and try to correct the corresponding error and column/row erasures using a basis for the guessed spaces, see Lemma 2.4. In case the dimension of the intersection of the guessed spaces and the rank supports of the error is large enough, the algorithm outputs a solution to the problem.[9]

In the following analysis, we use the notation and the well-known statements about error-erasure decoding of Gabidulin codes which are given in Section 2.2.2. Furthermore, we focus on guessing only the row rank support of the error, which means $\delta_{\mathrm{E}} = \gamma_{\mathrm{E}}$ and $\rho_{\mathrm{E}} = 0$ in (2.3). Later we show that this approach minimizes the expected work factor of our algorithm.

A formal description of our decoding approach is shown in Algorithm 7, where the function $\mathsf{ErrEraDec}(\boldsymbol{r}, \hat{\boldsymbol{a}}_{\mathrm{R}}, \hat{\boldsymbol{B}}_{\mathrm{C}})$ refers to an error-erasure decoder for the Gabidulin code $\mathcal{G}_k(\boldsymbol{g})$ that has the error correction capability stated in Lemma 2.4. Depending on $\hat{\boldsymbol{a}}_{\mathrm{R}}$ and $\hat{\boldsymbol{B}}_{\mathrm{C}}$, the function $\mathsf{ErrEraDec}$ outputs either a codeword in rank distance of at most t from $\boldsymbol{r}$ or $\boldsymbol{0}$. Furthermore, the integer δ_{E} specifies the sum of the dimensions of the guessed row and column rank support.

Algorithm 7: Randomized Gabidulin Decoder

```
Input  : Parity-check matrix $\boldsymbol{H} \in \mathbb{F}_{q^m}^{(n-k)\times n}$ of the Gabidulin code $\mathcal{G}_k(\boldsymbol{g})$
         Vector $\boldsymbol{r} \in \mathbb{F}_{q^m}^{n}$
         Non-negative integer $\delta_{\mathrm{E}}$
         Non-negative integer $t$
Output: Vector $\hat{\boldsymbol{e}} \in \mathbb{F}_{q^m}^{n}$
$\hat{\boldsymbol{e}} \leftarrow \boldsymbol{0}$
while $\mathrm{wt}_{\mathrm{R}}(\hat{\boldsymbol{e}}) > t \vee (\boldsymbol{r} - \hat{\boldsymbol{e}})\boldsymbol{H}^{\top} \neq \boldsymbol{0}$ do
    $\mathcal{U} \xleftarrow{\$} \mathrm{Gr}_q(\mathbb{F}_q^n, \delta_{\mathrm{E}})$
    $\hat{\boldsymbol{B}}_{\mathrm{C}} \leftarrow$ full-rank matrix whose row space equals $\mathcal{U}$
    $\hat{\boldsymbol{c}} \leftarrow \mathsf{ErrEraDec}(\boldsymbol{r}, \boldsymbol{0}, \hat{\boldsymbol{B}}_{\mathrm{C}})$
    $\hat{\boldsymbol{e}} \leftarrow \boldsymbol{r} - \hat{\boldsymbol{c}}$
return $\hat{\boldsymbol{e}}$
```

We denote the dimension of the intersection of our guess and the true rank supports

[9] Our approach is a generalization of the algorithm presented in [170]. The algorithm in [170] applies criss-cross erasures, and therefore, it is only capable of decoding a tiny fraction of error patterns with a rank weight larger than the unique decoding radius.

of the error by ϵ_{Int}. It follows that if

$$2(t-\epsilon_{\mathrm{Int}})+\delta_{\mathrm{E}} \leq n-k, \tag{3.14}$$

a Gabidulin error-erasure decoder is able to correct the error, see Lemma 2.4. To derive the work factor of the proposed algorithm, we require the following lemma:

Lemma 3.23. *Let $\mathcal{V}$ be a subspace chosen uniformly at random from the Grassmannian $\mathrm{Gr}_q(\mathbb{F}_{q^\ell}, v)$ and fix $\mathcal{U}$ as a u-dimensional $\mathbb{F}_q$-linear subspace of $\mathbb{F}_{q^\ell}$. Then, the probability*

$$\begin{aligned}\Pr(\dim(\mathcal{U}\cap\mathcal{V})\geq\omega) &= \frac{\sum_{i=\omega}^{\min\{u,v\}} \begin{bmatrix}\ell-u\\ v-i\end{bmatrix}_q \begin{bmatrix}u\\ i\end{bmatrix}_q q^{(u-i)(v-i)}}{\begin{bmatrix}\ell\\ v\end{bmatrix}_q}\\ &\leq 16(\min\{u,v\}+1-\omega)q^{(j^*-v)(\ell-u-j^*)},\end{aligned}$$

where $j^* := \min\{v-\omega, \frac{1}{2}(\ell+v-u)\}$.

Proof. For a fixed subspace $\mathcal{U}$ of dimension u, the probability

$$\begin{aligned}\Pr(\dim(\mathcal{U}\cap\mathcal{V})\geq\omega) &= \frac{\left|\left\{\mathcal{V}\in\mathrm{Gr}_q(\mathbb{F}_{q^\ell},v) : \dim(\mathcal{U}\cap\mathcal{V})\geq\omega\right\}\right|}{|\mathrm{Gr}_q(\mathbb{F}_{q^\ell},v)|}\\ &= \frac{\sum_{i=\omega}^{\min\{u,v\}} \begin{bmatrix}\ell-u\\ v-i\end{bmatrix}_q \begin{bmatrix}u\\ i\end{bmatrix}_q q^{(u-i)(v-i)}}{\begin{bmatrix}\ell\\ v\end{bmatrix}_q}\\ &= \frac{\sum_{j=\max\{0,v-u\}}^{v-\omega} \begin{bmatrix}\ell-u\\ j\end{bmatrix}_q \begin{bmatrix}u\\ v-j\end{bmatrix}_q q^{j(u-v+j)}}{\begin{bmatrix}\ell\\ v\end{bmatrix}_q},\end{aligned}$$

where the cardinality $|\{\mathcal{V}\in\mathrm{Gr}_q(\mathbb{F}_{q^\ell},v) : \dim(\mathcal{U}\cap\mathcal{V})\geq\omega\}|$ is derived in [171]. This expression can be upper bounded by applying the bounds on the Gaussian coefficient given in [68, Lemma 4], i.e.,

$$\begin{aligned}\Pr(\dim(\mathcal{U}\cap\mathcal{V})\geq\omega) &\leq 16\sum_{j=\max\{0,v-u\}}^{v-\omega} q^{j(\ell-u-j)+v(u-v+j)-v(\ell-v)}\\ &= 16\sum_{j=\max\{0,v-u\}}^{v-\omega} q^{(j-v)(\ell-u-j)}\\ &\leq 16\ (\min\{u,v\}+1-\omega)q^{(j^*-v)(\ell-u-j^*)},\end{aligned}$$

where $j^* \coloneqq \min\{v-\omega, \frac{1}{2}(\ell+v-u)\}$. ■

In the next lemma, we derive the probability that an error-erasure decoder which uses a single random guess of the row rank support outputs *exactly* the transmitted codeword.

Lemma 3.24. *Let* $\boldsymbol{r}' = \boldsymbol{c}' + \boldsymbol{e}' \in \mathbb{F}_{q^m}^n$*, where* $\boldsymbol{c}' \in \mathcal{G}_k(\boldsymbol{g}) \subset \mathbb{F}_{q^m}^n$*,* $\mathrm{wt}_\mathrm{R}(\boldsymbol{e}') = j$ *and* $\boldsymbol{e}' = \boldsymbol{a}'\boldsymbol{B}'$ *with* $\boldsymbol{a}' \in \mathbb{F}_{q^m}^j$*,* $\boldsymbol{B}' \in \mathbb{F}_q^{j\times n}$*. Furthermore, let* $\delta_\mathrm{E} \in [2j-(n-k):n-k]$*, let* $\mathcal{R}_q(\hat{\boldsymbol{B}}_\mathrm{C})$ *be a random* δ_E*-dimensional subspace of* $\mathbb{F}_q^n$*, and suppose that no part of the row or the column rank support of* $\boldsymbol{e}'$ *is known. Then, the probability that an error-erasure decoder that uses* $\hat{\boldsymbol{B}}_\mathrm{C}$ *outputs* $\boldsymbol{c}'$ *is*

$$\begin{aligned}P_{n,k,\delta_\mathrm{E},j} &\coloneqq \frac{\displaystyle\sum_{i=\left\lceil j-\frac{n-k}{2}+\frac{\delta_\mathrm{E}}{2}\right\rceil}^{\min\{\delta_\mathrm{E},j\}} \begin{bmatrix} n-j\\ \delta_\mathrm{E}-i\end{bmatrix}_q \begin{bmatrix} j\\ i\end{bmatrix}_q q^{(j-i)(\delta_\mathrm{E}-i)}}{\begin{bmatrix} n\\ \delta_\mathrm{E}\end{bmatrix}_q}\\ &\leq 16nq^{-\left(\left\lceil\frac{\delta_\mathrm{E}}{2}+j-\frac{n-k}{2}\right\rceil\right)\left(\frac{n+k}{2}-\left\lceil\frac{\delta_\mathrm{E}}{2}\right\rceil\right)},\end{aligned}$$

if $2j+\delta_\mathrm{E} > n-k$*, and* $P_{n,k,\delta_\mathrm{E},j} \coloneqq 1$ *otherwise.*

Proof. To derive the success probability for the case $2j+\delta_\mathrm{E} > n-k$, we define $\xi' \coloneqq j-\frac{n-k}{2}$, and thus, it follows from (3.14) that error-erasure decoding is successful if

$$2j-2\epsilon_\mathrm{Int}+\delta_\mathrm{E} = n-k+2\xi'-2\epsilon_\mathrm{Int}+\delta_\mathrm{E} \leq n-k, \tag{3.15}$$

where $\epsilon_\mathrm{Int} \coloneqq \mathcal{R}_q(\hat{\boldsymbol{B}}_\mathrm{C})\cap\mathcal{R}_q(\boldsymbol{B}')$. Since $\epsilon_\mathrm{Int} \leq \delta_\mathrm{E}$, the integer δ_E must satisfy

$$2\xi' \leq 2\epsilon_\mathrm{Int}-\delta_\mathrm{E} \leq \delta_\mathrm{E} \leq n-k,$$

see (3.15). The same equation implies further that the space $\mathcal{R}_q(\hat{\boldsymbol{B}}_\mathrm{C})$ does not need to be a subspace of $\mathcal{R}_q(\boldsymbol{B}')$, but it is sufficient that their intersection is large enough, i.e., $\epsilon_\mathrm{Int} \geq \xi' + \frac{\delta_\mathrm{E}}{2}$. Thus, the probability that the dimension of the intersection of a randomly chosen space and the row rank support of $\boldsymbol{e}'$ is large enough such that an error-erasure decoder outputs $\boldsymbol{c}'$ is

$$\begin{aligned}
&\frac{\sum\limits_{i=\left\lceil \xi'+\frac{\delta_\mathrm{E}}{2}\right\rceil}^{\min\{\delta_\mathrm{E},j\}} \begin{bmatrix} n-j \\ \delta_\mathrm{E}-i \end{bmatrix}_q \begin{bmatrix} j \\ i \end{bmatrix}_q q^{(j-i)(\delta_\mathrm{E}-i)}}{\begin{bmatrix} n \\ \delta_\mathrm{E} \end{bmatrix}_q} \\
\leq& 16\Big(\min\{j,\delta_\mathrm{E}\}+1-\Big(\xi'+\frac{\delta_\mathrm{E}}{2}\Big)\Big)\, q^{-\left(\left\lceil \frac{\delta_\mathrm{E}}{2}+\xi'\right\rceil\right)\left(\frac{n+k}{2}-\left\lceil\frac{\delta_\mathrm{E}}{2}\right\rceil\right)} \\
\leq& 16nq^{-\left(\left\lceil \frac{\delta_\mathrm{E}}{2}+\xi'\right\rceil\right)\left(\frac{n+k}{2}-\left\lceil\frac{\delta_\mathrm{E}}{2}\right\rceil\right)},
\end{aligned}$$

see Lemma 3.23. The probability $P_{n,k,\delta_\mathrm{E},j} = 1$, for $2j + \delta_\mathrm{E} \leq n-k$, follows directly from Lemma 2.4. ■

We observe that Lemma 3.24 states the probability that an error-erasure decoder outputs exactly the transmitted codeword $\boldsymbol{c}'$. However, in Problem 3.6, it is not necessary to exactly determine $\boldsymbol{c}'$, but it is sufficient to find any codeword of $\mathcal{G}_k(\boldsymbol{g})$ in rank distance at most t to $\boldsymbol{r}$. To derive a lower bound on the work factor of the proposed algorithm, we derive an upper bound on its success probability in the following lemma.

Lemma 3.25. *Let the vector $\boldsymbol{r}$ be chosen uniformly at random from $\mathbb{F}_{q^m}^n$, let the integer $\delta_\mathrm{E} \in [2\xi : n-k]$, and let $\mathcal{R}_q(\hat{\boldsymbol{B}}_\mathrm{C})$ be a random δ_E-dimensional subspace of $\mathbb{F}_q^n$. Then, an error-erasure decoder that uses $\hat{\boldsymbol{B}}_\mathrm{C}$ decodes $\boldsymbol{r}$ to $\boldsymbol{c} \in \mathcal{G}_k(\boldsymbol{g})$ such that $\mathrm{d}_\mathrm{R}(\boldsymbol{c},\boldsymbol{r}) \leq t$ with a probability of at most*

$$\sum_{j=0}^{t} \bar{A}_j P_{n,k,\delta_\mathrm{E},j} \leq 64nq^{m(k-n)+t(n+m)-t^2-\left(\left\lceil \frac{\delta_\mathrm{E}}{2}+t-\frac{n-k}{2}\right\rceil\right)\left(\frac{n+k}{2}-\left\lceil\frac{\delta_\mathrm{E}}{2}\right\rceil\right)},$$

where $\bar{A}_j \coloneqq q^{m(k-n)} \prod_{i=0}^{j-1} \frac{(q^m-q^i)(q^n-q^i)}{q^j-q^i}$.

Proof. Define $\hat{\mathcal{C}}$ as the set

$$\hat{\mathcal{C}} \coloneqq \{\boldsymbol{c} \in \mathcal{G}_k(\boldsymbol{g}) : \mathrm{d_R}(\boldsymbol{c}, \boldsymbol{r}) \leq t\} = \{\hat{\boldsymbol{c}}_1, \ldots, \hat{\boldsymbol{c}}_{N_\mathrm{R}}\},$$

X_i as the event that an error-erasure decoder decodes $\boldsymbol{r}$ to $\hat{\boldsymbol{c}}_i$ where $i \in [1 : N_\mathrm{R}]$, and N_R is defined in (2.7). Furthermore, let $\mathcal{A}_j$ denote the set of codeword indices that refer to codewords of rank distance j, i.e., $\mathcal{A}_j \coloneqq \{i : \mathrm{d_R}(\hat{\boldsymbol{c}}_i, \boldsymbol{r}) = j\}$. By recalling the definition of $P_{n,k,\delta_\mathrm{E},j}$ given in Lemma 3.24, we observe that $\Pr(X_i) = P_{n,k,\delta_\mathrm{E},j}$ for $i \in \mathcal{A}_j$. Then, we can apply a union bound argument to upper bound the success probability

$$\Pr(\mathit{success}) = \Pr\left(\bigcup_{i=1}^{N_\mathrm{R}} X_i\right) \leq \sum_{i=1}^{N_\mathrm{R}} \Pr(X_i) = \sum_{j=0}^{t} |\mathcal{A}_j| P_{n,k,\delta_\mathrm{E},j}.$$

Furthermore, we denote the average size of the set $\mathcal{A}_j$ by $\bar{A}_j$, which can be bounded by

$$\bar{A}_j = q^{m(k-n)} \prod_{i=0}^{j-1} \frac{(q^m - q^i)(q^n - q^i)}{q^j - q^i} \leq 4q^{m(k-n)+j(n+m)-j^2}.$$

Finally, we can approximate the success probability by

$$\begin{aligned}\Pr(\mathit{success}) &= \bar{A}_t P_{n,k,\delta_\mathrm{E},t} \\ &\leq 64nq^{m(k-n)+t(n+m)-t^2-\left(\left\lceil \frac{\delta_\mathrm{E}}{2}+t-\frac{n-k}{2}\right\rceil\right)\left(\frac{n+k}{2}-\left\lceil\frac{\delta_\mathrm{E}}{2}\right\rceil\right)},\end{aligned}$$

using the fact that $\bar{A}_t$ is exponentially larger than $\bar{A}_{t-i}$ for any positive integer i. ■

By combining the derived lemmata, we can lower bound the average work factor of the proposed algorithm as follows.

Theorem 3.26. *Let $\boldsymbol{H} \in \mathbb{F}_{q^m}^{(n-k)\times n}$ be a parity-check matrix of an $[n,k]_{\mathbb{F}_{q^m}}$ Gabidulin code $\mathcal{G}_k(\boldsymbol{g}) \subset \mathbb{F}_{q^m}^n$, let $\boldsymbol{r}$ be drawn uniformly at random from $\mathbb{F}_{q^m}^n$, and let t and δ_E be integers such that $\frac{n-k}{2} < t < n-k$ and $2\xi \leq \delta_\mathrm{E} \leq n-k$. Then, Algorithm 7 outputs $\boldsymbol{e}' \in \mathbb{F}_{q^m}^n$ such that $\mathrm{wt_R}(\boldsymbol{e}') \leq t$ and $(\boldsymbol{r} - \boldsymbol{e}')\boldsymbol{H}^\top = \boldsymbol{0}$ with, on average, at least*

$$W_\mathrm{RD} = \min_{\delta_\mathrm{E} \in [2\xi : n-k]} \left\{ \frac{n^2}{\sum_{j=0}^{t} \bar{A}_j P_{n,k,\delta_\mathrm{E},j}} \right\}$$

$$= \min_{\delta_{\mathrm{E}} \in [2\xi : n-k]} \left\{ \frac{n^2 \begin{bmatrix} n \\ \delta_{\mathrm{E}} \end{bmatrix}_q}{\sum\limits_{j=0}^{\left\lfloor \frac{n-k-\delta_{\mathrm{E}}}{2} \right\rfloor} q^{m(k-n)} \prod\limits_{\ell=0}^{j-1} \frac{(q^m - q^\ell)(q^n - q^\ell)}{q^j - q^\ell} + \sum\limits_{j=\left\lfloor \frac{n-k-\delta_{\mathrm{E}}}{2} \right\rfloor + 1}^{t} q^{m(k-n)} \cdots} \right.$$

$$\left. \overline{\cdots \left(\prod_{\ell=0}^{j-1} \frac{(q^m - q^\ell)(q^n - q^\ell)}{q^j - q^\ell} \right) \left(\sum_{i=\left\lceil j - \frac{n-k}{2} + \frac{\delta_{\mathrm{E}}}{2} \right\rceil}^{\min\{\delta_{\mathrm{E}}, j\}} \begin{bmatrix} n-j \\ \delta_{\mathrm{E}} - i \end{bmatrix}_q \begin{bmatrix} j \\ i \end{bmatrix}_q q^{(j-i)(\delta_{\mathrm{E}}-i)} \right)} \right\}$$

operations in $\mathbb{F}_{q^m}$, where $\bar{A}_j$ and $P_{n,k,\delta_{\mathrm{E}},j}$ are defined as in Lemma 3.25.

Proof. We derived in Lemma 3.25 that an error-erasure decoder that uses a random δ_{E}-dimensional subspace of $\mathbb{F}_q^n$ outputs $\boldsymbol{c} \in \mathcal{G}_k(\boldsymbol{g})$ such that $\mathrm{d}_{\mathrm{R}}(\boldsymbol{c}, \boldsymbol{r}) \leq t$ only with a certain probability. This implies that we have to sample, on average, at least

$$\min_{\delta_{\mathrm{E}} \in [2\xi : n-k]} \left\{ \frac{1}{\sum_{j=0}^{t} \bar{A}_j P_{n,k,\delta_{\mathrm{E}},j}} \right\}$$

δ_{E}-dimensional subspaces of $\mathbb{F}_q^n$ in order to determine a codeword in rank distance of at most t to $\boldsymbol{r}$. Because error-erasure decoding requires $O(n^2)$ operations in $\mathbb{F}_{q^m}$, it follows that the work factor of the proposed algorithm is equal to

$$W_{\mathrm{RD}} = \min_{\delta_{\mathrm{E}} \in [2\xi : n-k]} \left\{ \frac{n^2}{\sum_{j=0}^{t} \bar{A}_j P_{n,k,\delta_{\mathrm{E}},j}} \right\}. \qquad \blacksquare$$

A closed form expression of a lower bound on the work factor is given in the following corollary.

Corollary 3.27. *Let $\boldsymbol{H} \in \mathbb{F}_{q^m}^{(n-k) \times n}$ be a parity-check matrix of an $[n, k]_{\mathbb{F}_{q^m}}$ Gabidulin code $\mathcal{G}_k(\boldsymbol{g}) \subset \mathbb{F}_{q^m}^n$, let $\boldsymbol{r}$ be drawn uniformly at random from $\mathbb{F}_{q^m}^n$, and let t and δ_{E} be an integers such that $\frac{n-k}{2} < t < n-k$ and $2\xi \leq \delta_{\mathrm{E}} \leq n-k$. Then, Algorithm 7* *outputs* $\boldsymbol{e}' \in \mathbb{F}_{q^m}^n$ *such that* $\mathrm{wt}_\mathrm{R}(\boldsymbol{e}') \leq t$ *and* $(\boldsymbol{r} - \boldsymbol{e}')\boldsymbol{H}^\top = \boldsymbol{0}$ *with, on average, at least*

$$W_\mathrm{RD} \geq \frac{n}{64} \cdot q^{m(n-k)-t(n+m)+t^2+\min\{2\xi(\frac{n+k}{2}-\xi),tk\}}$$

operations in $\mathbb{F}_{q^m}$*.*

Proof. This follows directly from Theorem 3.26 and the fact that the upper bound on the probability given in Lemma 3.25 is convex in δ_E, and thus, it is maximized for either 2ξ or $n-k$. ■

Similarly to the previous corollary, we can also give a closed form expression of an *upper* bound on the complexity of the proposed algorithm.

Corollary 3.28. *Let* $\boldsymbol{H} \in \mathbb{F}_{q^m}^{(n-k)\times n}$ *be a parity-check matrix of an* $[n,k]_{\mathbb{F}_{q^m}}$ *Gabidulin code* $\mathcal{G}_k(\boldsymbol{g}) \subset \mathbb{F}_{q^m}^n$*, let* $\boldsymbol{r}$ *be drawn uniformly at random from* $\mathbb{F}_{q^m}^n$*, and let* t *and* δ_E *be an integers such that* $\frac{n-k}{2} < t < n-k$ *and* $2\xi \leq \delta_\mathrm{E} \leq n-k$*. Then, Algorithm 7 outputs* $\boldsymbol{e}' \in \mathbb{F}_{q^m}^n$ *such that* $\mathrm{wt}_\mathrm{R}(\boldsymbol{e}') \leq t$ *and* $(\boldsymbol{r} - \boldsymbol{e}')\boldsymbol{H}^\top = \boldsymbol{0}$ *with, on average, at most*

$$W_\mathrm{RD} \leq n^2 q^{m(n-k)-t(n+m)+t^2+\min\{2\xi(\frac{n+k}{2}-\xi),tk\}},$$

operations in $\mathbb{F}_{q^m}$*.*

Proof. The expression can be derived by combining the arguments of Lemma 3.24, Lemma 3.25, and Theorem 3.26. However, we apply lower bounds instead of upper bounds on the Gaussian binomial coefficient (see [68, Lemma 4]), we use the maximal instead of the minimal terms in the sums, and we consider the maximal probability of events instead of union-bounds. ■

In the definition of Problem 3.6, the vector $\boldsymbol{r}$ is the sum of a Gabidulin codeword and an error of rank t, where neither parts of the row nor the column rank support of the error are known. In this case, the vector $\boldsymbol{r} \in \mathbb{F}_{q^m}^n$ can be interpreted as a vector drawn uniformly at random from the set $\mathbb{F}_{q^m}^n$. It follows that the complexity that is derived in Theorem 3.26 can be used as an estimation of the work factor to solve Problem 3.6. The simulation results given in Section 3.3.2 verify that this assumption is sound and the resulting estimation is accurate.

3.3.2 Comparison to Other Algorithms

In the following, we compare the proposed Algorithm 7 with known approaches to solve Problem 3.6. Besides comparing our algorithm to generic rank syndrome decoders, see Section 2.2.4, we also take an algorithm based on solving the *key equation* into consideration.

Key Equation Based Decoding

There exists a family of decoding algorithms of Gabidulin codes that are based on solving the key equation [86, Lemma 4]. This equation refers to a linear system of $n-k-t$ equations with t unknowns. If $t \leq \lfloor \frac{n-k}{2} \rfloor$, the system is overdetermined and has exactly one solution. However, in the setting described by Problem 3.6, we have $t > \lfloor \frac{n-k}{2} \rfloor$ unknowns meaning that the linear system has a solution space of dimension $t-(n-k-t) = 2\xi$ and all codewords $\boldsymbol{c} \in \mathcal{G}_k(\boldsymbol{g})$ such that $\mathrm{d_R}(\boldsymbol{c}, \boldsymbol{r}) \leq t$ are in this space [172]. Thus, solving Problem 3.6 requires to iterate through all elements of the solution space to find such a codeword. This leads to a complexity of

$$W_{\mathrm{Key}} = \frac{n^2 q^{m2\xi}}{N_{\mathrm{R}}},$$

where each iteration requires $O(n^2)$ operations in $\mathbb{F}_{q^m}$ and N_{R} is defined in (2.7).

Numerical Results

In order to validate the accuracy of our complexity estimation of Algorithm 7, we conducted simulations using the error-erasure decoder proposed in [99]. In the first row of Table 3.1, we observe that the true complexity of Algorithm 7, which is denoted by W_{Sim}, is very close to our theoretical estimation W_{RD}. The remaining rows of Table 3.1 refer to parameter sets which are proposed in [152, 153]. The results indicate that Algorithm 7 requires, on average, a significantly smaller number of operations compared to key equation based decoding W_{Key}, combinatorial generic decoding W_{CRSD} and algebraic generic decoding W_{ARSD}, see Section 2.2.4.

3.3.3 Possible Modifications to the Algorithm

Since error-erasure decoding algorithms are capable of decoding row and column erasures at the same time, Algorithm 7 can be modified such that parts of the row and

Table 3.1: Operations in $\mathbb{F}_q$ required by different algorithms for solving Problem 3.6. The value of W_{Sim} was obtained by simulating 6844700 iterations that led to 4488 successes. The parameter $q = 2$ is used for all sets.

m	n	k	t	ξ	δ_{E}	W_{Sim}	W_{RD}	W_{CRSD}	W_{ARSD}	W_{Key}
24	24	16	6	2	4	$2^{28.91}$	$2^{28.82}$	$2^{43.74}$	$2^{108.94}$	$2^{52.57}$
64	64	32	19	3	6	-	$2^{269.21}$	$2^{574.21}$	$2^{460.01}$	$2^{383.21}$
80	80	40	23	3	6	-	$2^{414.49}$	$2^{900.93}$	$2^{576.15}$	$2^{505.29}$
96	96	48	27	3	6	-	$2^{591.55}$	$2^{1266.51}$	$2^{694.93}$	$2^{602.34}$
82	82	48	20	3	6	-	$2^{303.64}$	$2^{842.35}$	$2^{504.70}$	$2^{423.64}$

the column rank support of the error are guessed jointly. In the following, we analyze the success probability of that approach.

Lemma 3.29. *Let $\boldsymbol{r}' = \boldsymbol{c}' + \boldsymbol{e}' \in \mathbb{F}_{q^m}^n$, where $\boldsymbol{c}' \in \mathcal{G}_k(\boldsymbol{g}) \subset \mathbb{F}_{q^m}^n$, $\text{wt}_{\text{R}}(\boldsymbol{e}') = j$ and $\boldsymbol{e}' = \boldsymbol{a}'\boldsymbol{B}'$ with $\boldsymbol{a}' \in \mathbb{F}_{q^m}^j$, $\boldsymbol{B}' \in \mathbb{F}_q^{j \times n}$. Furthermore, let $\delta_{\text{E}} \in [2j - (n-k) : n-k]$, let $\mathcal{R}_q(\hat{\boldsymbol{B}}_{\text{C}})$ be a random δ_{E}^r-dimensional subspace of $\mathbb{F}_q^n$, let $\mathcal{R}_q(\text{ext}_{q^m/q}(\hat{\boldsymbol{a}}_{\text{R}})^\top)$ be a random δ_{E}^c-dimensional subspace of $\mathbb{F}_q^m$ such that $\delta_{\text{E}}^r + \delta_{\text{E}}^c = \delta_{\text{E}}$, and suppose that no part of the row or column rank support of $\boldsymbol{e}'$ is known. Then, the probability that an error-erasure decoder that uses $\hat{\boldsymbol{B}}_{\text{C}}$ and $\hat{\boldsymbol{a}}_{\text{R}}$ outputs $\boldsymbol{c}'$ is at most*

$$\frac{\displaystyle\sum_{i=\left\lceil \xi + \frac{\delta_{\text{E}}}{2} \right\rceil}^{\min\{\delta_{\text{E}}, j\}} \sum_{\substack{0 \le t_r, t_c \le i \\ t_r + t_c = i}} \begin{bmatrix} n-j \\ \delta_{\text{E}}^r - t_r \end{bmatrix}_q \begin{bmatrix} j \\ t_r \end{bmatrix}_q q^{(j-t_r)(\delta_{\text{E}}^r - t_r)} \begin{bmatrix} m-j \\ \delta_{\text{E}}^c - t_c \end{bmatrix}_q \begin{bmatrix} j \\ t_c \end{bmatrix}_q q^{(j-t_c)(\delta_{\text{E}}^c - t_c)}}{\begin{bmatrix} n \\ \delta_{\text{E}}^r \end{bmatrix}_q \begin{bmatrix} m \\ \delta_{\text{E}}^c \end{bmatrix}_q}.$$

Proof. The lemma can be proven similarly to Lemma 3.24. The probability that two randomly drawn vector spaces of dimension δ_{E}^r and δ_{E}^c intersect with the row and column rank support of the error in exactly t_r and t_c dimensions is given by

$$\frac{\begin{bmatrix} n-j \\ \delta_{\mathrm{E}}^{r} - t_r \end{bmatrix}_q \begin{bmatrix} j \\ t_r \end{bmatrix}_q q^{(j-t_r)(\delta_{\mathrm{E}}^{r}-t_r)} \begin{bmatrix} m-j \\ \delta_{\mathrm{E}}^{c} - t_c \end{bmatrix}_q \begin{bmatrix} j \\ t_c \end{bmatrix}_q q^{(j-t_c)(\delta_{\mathrm{E}}^{c}-t_c)}}{\begin{bmatrix} n \\ \delta_{\mathrm{E}}^{r} \end{bmatrix}_q \begin{bmatrix} m \\ \delta_{\mathrm{E}}^{c} \end{bmatrix}_q}.$$

Summing up over all positive integers t_r and t_c such that $t_r + t_c \geq \left\lceil \xi + \frac{\delta_{\mathrm{E}}}{2} \right\rceil$ gives then an upper bound on the success probability. ■

Note that the proof of Lemma 3.29 uses the optimistic argument that correctly guessing t_r dimensions of the row and t_c dimensions of the column rank support of the error reduces the rank of the error by $t_r + t_c$, which is not always the case.

We do not know how to prove that guessing the column and row rank support of the error is never advantageous over guessing only the row rank support of the error. However, this was the case for all parameter sets we investigated. For instance, for the following set of parameters.

Example 3.1. *Let* $q = 2$, $m = n = 24$, $k = 16$ *and* $t = 6$. *Then, an error-erasure decoder succeeds with probability* $1.66 \cdot 10^{-22}$ *if we draw* $\delta_{\mathrm{E}} = 4$ *dimensional subspaces* $\mathcal{R}_q(\hat{\boldsymbol{B}}_{\mathrm{C}})$ *uniformly at random. In case we use* $\delta_{\mathrm{E}}^{r} = \delta_{\mathrm{E}}^{c} = 2$, *the error-erasure algorithms only decodes correctly with probability* $1.93 \cdot 10^{-22}$.

A further approach is to only guess the column rank support of the error. To analyze the work factor, we need to replace n by m in Lemma 3.24 and in the probability P_j in the proof of Theorem 3.26. Since $n \leq m$ for Gabidulin codes, this approach cannot decrease the resulting work factor, and it is clear that this approach is never advantageous over only guessing the row space.

3.4 Concluding Remarks

In Chapter 3, we have investigated three different problems in coding theory with potential applications in cryptography. The first problem that we have considered was generic decoding in the sum-rank metric, where we proposed a non-trivial algorithm that solves this problem in exponential time. This algorithm can be seen as a combination of Prange's generic decoding algorithm in the Hamming metric [134] and Gaborit, Ruatta, and Schrek's basic generic decoding algorithm in the rank metric [139]. Furthermore, we have presented a randomized reduction of the decisional Hamming syndrome decoding problem to the decisional sum-rank syndrome decoding problem, which implies that if the widely believed conjecture $\mathsf{ZPP} \neq \mathsf{NP}$ is true, then the latter problem is in $\mathsf{NP} \setminus \mathsf{ZPP} \subset \mathsf{NP} \setminus \mathsf{P}$. Although it is an open problem whether there is a deterministic reduction from an NP-hard problem to the decisional sum-rank syndrome decoding problem, our result strongly motivates future studies of cryptographic schemes based on the sum-rank metric. A further open problem is the incorporation of the improvements of the algorithms presented in [134] and [139] into our proposed algorithm, and the adaption of the algebraic decoding algorithm presented in [142] to the sum-rank metric.

In the second part of this chapter, we have addressed the problem of generic decoding of high-order interleaved rank-metric codes. We have devised a polynomial-time decoding algorithm that enables us to correct errors of rank weight up to $d_{\mathrm{min}}^{\mathrm{R}} - 2$ in any interleaved code of minimum rank distance $d_{\mathrm{min}}^{\mathrm{R}}$ if the error fulfills two conditions: First, the rank weight of the error is at most the interleaving order; and second, the rank of the error over the large field is equal to the rank weight of the error. Furthermore, we have proved that for a random error of rank weight of at most the interleaving order, it holds with high probability that the rank of this error over the large field is equal to its rank weight. This finding implies that our decoding algorithm can even correct errors of this kind with high probability. The presented analysis has an impact on rank-based McEliece or Niederreiter schemes, as it shows that in case of multiple encryptions, the row rank supports of the errors have to be generated independently to ensure secure encryptions; otherwise these systems will be unsecure with high probability. An open problem is the adaption of our algorithm to errors whose rank over the large field is smaller than their rank weight.[10] Furthermore, the adaption to instances with an interleaving order smaller than the rank weight of the error is left

[10] This problem was solved for the Hamming metric in [164].

for future work. The latter problem is of interest, as there exist cryptosystems based on the hardness of generic decoding of rank metric codes with a small interleaving order, e.g., [107]. In addition, there are cryptographic schemes whose security relies on the hardness of syndrome decoding of interleaved codes in the rank metric, where the errors share the same column rank-support [157, 173]. Therefore, it would be of interest to modify our proposed algorithm such that it is capable of decoding such errors.

In the third part of Chapter 3, we have considered the problem of decoding Gabidulin codes beyond their unique decoding radius. The complexity of this problem is of importance to rank-based cryptography, as cryptosystems exist that rely on the aforementioned problem, e.g., [106, 152, 153]. To obtain an upper bound on the complexity of decoding Gabidulin codes beyond their unique error-correction radius, we have developed a new algorithm that introduces random row or column erasures to decrease the rank of the error in order to enable polynomial-time Gabidulin code error-erasure decoding. The proposed algorithm improves upon generic rank-metric decoders and other known approaches by an exponential factor. It should be noted that there is a list decoding algorithm for Gabidulin codes based on Gröbner bases that can also correct errors beyond the unique decoding radius [174]. However, the authors provide no upper bound on the list size, and therefore, the work factor of this algorithm cannot be evaluated. In future work, the work factor of the algorithm proposed in [174] could be determined which would help to asses the hardness of the problem of decoding Gabidulin codes beyond their unique decoding radius. Furthermore, there are only some complexity results known for this problem [175–177] but no proper reduction to it. Therefore, a formal hardness proof should be developed in future work. In addition, the idea of our decoding algorithm can easily be adapted to all code classes that feature error-erasure decoders. For instance, one can adapt the idea to decode RS codes in the Hamming metric. For these codes, it is well known that there are error weights in the interval $[n - \sqrt{n(k-1)} : n-k]$ where the problem of decoding RS codes cannot be solved in polynomial time unless the polynomial hierarchy collapses [77, 178, 179]. The adaption of the proposed algorithm to RS codes would allow us to solve these instances in exponential time.

4

Attacks on Hamming-Based Encryption Schemes

Virtually all Hamming-based encryption schemes deploy the syndrome decoding problem as the trapdoor function. In these systems, the problem of syndrome decoding is instantiated in such a way that given a parity-check matrix of a publicly known code and a syndrome, it is hard to determine a valid error vector[1] that fulfills the parity-check equations unless some secret is known. Currently, there are two main families of Hamming-based encryption schemes. The first family developed from McEliece's original proposal [13], where the aforementioned secret is the structure of the publicly known code which is required for efficient decoding. For the second family, the structure of the public code is common knowledge, but the secret is the structure of the error.

The security of McEliece's original proposal is based on the hardness of retrieving the code locators of a Goppa code from its scrambled parity-check matrix [13]. Although this system is still considered to be secure and is a promising candidate in the third round of the NIST competition [135], it suffers from large key sizes. To overcome this issue, researchers have investigated most of the known code classes to replace Goppa codes. In the first part of this chapter, we consider the system introduced by Beelen *et al.* in [84]. The authors propose to replace Goppa codes by TRS codes

[1] The problem can also be stated as given a generator matrix and the sum of a codeword and an error, the task is to retrieve the codeword.

to reduce the key sizes. They show that the applied TRS codes are different from GRS codes, and therefore, the attack by Sidelnikov and Shestakov [15] cannot be applied to their system. Furthermore, the authors prove that shortenings of these codes up to two positions have maximal Schur square dimension [180], meaning that the proposed system is invulnerable to the attack presented by Couvreur *et al.* in [181]. In addition, Beelen *et al.* give evidence that the proposed system is unassailable to the methods introduced by Wieschebrink in [17, 182]. Since the known structural attacks on variants of the McEliece scheme cannot be mounted on [84], we analyze the security of this scheme. More precisely, we develop a new efficient key-recovery attack on the cryptosystem based on TRS codes, which recovers the structure of a well-chosen *subfield subcode* of the public TRS code. We show that this subfield subcode is a subspace of low codimension contained in an RS code, and we prove that the Wieschebrink squaring method can *always* be applied to this subfield subcode to recover an algebraic description of the RS code. By an analysis of equivalent representations of TRS codes, we finally determine an algebraic description of the public TRS code that is sufficient to break the proposed system.

The development of the second family of Hamming-based encryption schemes started with a proposal by Augot and Finiasz [61] in 2003. The authors propose an RS code as public code and choose the ciphertext as the sum of a codeword of this RS code and an error of large Hamming weight but with a certain structure. At a first glance, the system seemed to be promising, as the authors proposed parameters that led to small key sizes, but the system was broken only one year later [64]. In the same year when Augot and Finiasz published their paper, Alekhnovich proposed a framework that only relies on the difficulty of decoding random codes [60]. Although this system is not practical, it is the starting point of the HQC scheme [63]. HQC is a promising candidate in the NIST competition, as it features small key sizes and allows precise estimations of its decryption failure rate. Since so far no mathematical weaknesses have been observed, we investigate the security of the implementation of HQC which was proposed in [183]. Recent attacks on the implementation of HQC use a timing side-channel of the applied decoder to gather information about the decryption [184, 185]. Utilizing this information, both attacks are able to successfully retrieve the private key. However, this attack vector has been removed, as the authors of [185] provide a constant-time implementation of the decoder, which has been merged into the HQC reference implementation [183]. In our attack,[2] we utilize the method presented

[2] Please note that the presented attack specifically targets the IND-CCA2-secure KEM version of

in [187], which exploits a power side-channel to construct an oracle that takes as input a ciphertext and returns information about the error that is corrected in the decryption algorithm. Based on this oracle, we derive a chosen-ciphertext attack that allows us to determine the private key in case of error-free side-channel information. However, in case of noisy side-channel information, the proposed attack potentially retrieves only a part of the private key. For these cases, we present modifications of Prange's [134], Lee and Brickell's [160], and Stern's [161] ISD algorithms that are capable of utilizing the determined part of the private key, and thus, achieve a complexity below the claimed security level. Furthermore, theoretical thoughts regarding a potential countermeasure against the proposed attacks are stated at end.

The results shown in Section 4.1 are published in the journal *Designs, Codes and Cryptography* [188]. The author of this dissertation contributed both the presented attack and the respective analysis. The cited publication contains an additional security analysis of an attempt to repair the considered system, where the repair was proposed by the authors of [84] after they had been notified about the attack. This discussion is not presented in this thesis, as it was mainly conducted by the author Julien Lavauzelle. Furthermore, the paper [188] comprises a discussion about the feasibility of adapting the proposed attack to a variant of the McEliece scheme based on twisted Gabidulin codes. This discussion is outside the scope of this thesis, and thus, it is also not contained in this thesis.

Parts of Section 4.2 are published in the proceedings of the *2020 International Conference on Smart Card Research and Advanced Applications (CARDIS)* [189]. The attack that is proposed in the cited paper is based on side-channel information which is gained from a power analysis. As this power analysis was performed by the other authors of [189], no details about this analysis are presented in this thesis. In order to still work with the acquired side-channel information, we model it as an oracle in Section 4.2. The attacks given in [189] are then described using this oracle. The attack which requires error-free side-channel information was jointly developed by Thomas Schamberger and the author of this thesis. The other attack, which is also applicable in case of noisy side-channel information, was solely derived by the author of this thesis. In addition to the attacks that are given in [189], two even more powerful attacks are presented in Section 4.2. These attacks are partly based on [190, Sec. 4], which has been accepted to the *2021 International Workshop on Code-Based Cryptography*

HQC, as it has already been shown that the IND-CPA-secure public-key encryption version is assailable to chosen-ciphertext attacks [186].

(CBCrypto), and were solely developed by the author of this thesis. The remaining content of [190] is not shown, as it is outside the scope of this dissertation. The theoretical thoughts about potential countermeasures at the end of Section 4.2 are not included in [189] and were solely devised by the author of this thesis.

4.1 Cryptanalysis of a McEliece System Based on TRS Codes

In this section, we present a feasible attack on the variant of the McEliece encryption scheme based on TRS codes [84]. We first recall the aforementioned variant, and then, we derive the attack and analyze its complexity.

4.1.1 The Variant of the McEliece System Based on TRS Codes

We denote the TRS-based McEliece encryption scheme by $\Pi_{\mathrm{TRS}}^{\mathrm{Enc}}$. The setup of $\Pi_{\mathrm{TRS}}^{\mathrm{Enc}}$ is as follows: Let q_0 be a prime power, and let k and n be integers that fulfill $n \leq q_0 - 1$ and $2\sqrt{n} + 6 < k \leq \frac{n}{2} - 2$. Furthermore, let ℓ_{T} be a positive integer such that

$$\frac{n+1}{k-\sqrt{n}} < \ell_{\mathrm{T}} + 2 < \min\left\{k+3, \frac{2n}{k}, \sqrt{n} - 2\right\} .$$

Choose $q_i := q_{i-1}^2 = q_0^{2^i}$, for $i \in [1:\ell_{\mathrm{T}}]$, such that $\mathbb{F}_{q_0} \subset \mathbb{F}_{q_1} \subset \ldots \subset \mathbb{F}_{q_{\ell_{\mathrm{T}}}} = \mathbb{F}_q$ is a chain of subfields, and choose $\tau_i = (i+1)(r_{\mathrm{T}} - 2) - k + 2$ and $\pi_i = r_{\mathrm{T}} - 1 + i$, for $i \in [1:\ell_{\mathrm{T}}]$, where $r_{\mathrm{T}} := \left\lceil \frac{n+1}{\ell_{\mathrm{T}}+2} \right\rceil + 2$. Integers q_0, n, k, and ℓ_{T} and vectors $\boldsymbol{\tau} = [\tau_1, \ldots, \tau_{\ell_{\mathrm{T}}}]$ and $\boldsymbol{\pi} = [\pi_1, \ldots, \pi_{\ell_{\mathrm{T}}}]$ that fulfill the above restrictions are considered as valid parameters of the cryptosystem and are public knowledge [84].

The encryption scheme is defined as

$$\Pi_{\mathrm{TRS}}^{\mathrm{Enc}} := (\mathsf{KeyGen}_{\mathrm{TRS}}, \mathsf{Encrypt}_{\mathrm{TRS}}, \mathsf{Decrypt}_{\mathrm{TRS}}),$$

where the key-generation, encryption, and decryption algorithms are stated in Algorithm 8, Algorithm 9, and Algorithm 10, respectively. Note that the function DecodeTRS refers to the decoding algorithm of $\mathcal{TRS}_k(\boldsymbol{\alpha}, \boldsymbol{\tau}, \boldsymbol{\pi}, \boldsymbol{\eta})$ proposed in [84].

In [84], the designers of $\Pi_{\mathrm{TRS}}^{\mathrm{Enc}}$ propose parameters which are shown in Table 4.1. There are two crucial reasons for a small number of twists ℓ_{T}. First, the decoding algorithm presented in [84] has a complexity in

$$O\Big(q_0^{\ell_{\mathrm{T}} 2^{\ell_{\mathrm{T}}}} n \log^2(n) \log\Big(\log(n)\Big)\Big),$$

and therefore, the complexity of Algorithm 10 increases doubly exponentially in the number of twists ℓ_{T}. Second, the cardinality of the largest field $\mathbb{F}_q$ scales exponentially

Algorithm 8: $\mathsf{KeyGen}_{\mathrm{TRS}}$

```
Input  : Integers $q_0$, $n$, $k$
         Vector $\boldsymbol{\tau} \in [1\!:\!n-k]^{\ell_{\mathrm{T}}}$
         Vector $\boldsymbol{\pi} \in [0\!:\!k-1]^{\ell_{\mathrm{T}}}$
Output: Private key $(\boldsymbol{S}, \boldsymbol{\alpha}, \boldsymbol{\eta}) \in \mathbb{F}_q^{k\times k} \times \mathbb{F}_{q_0}^{n} \times \mathbb{F}_q^{\ell_{\mathrm{T}}}$
         Public key $\boldsymbol{G}_{\mathrm{pub}} \in \mathbb{F}_q^{k\times n}$
$\boldsymbol{S} \xleftarrow{\$} \left\{\boldsymbol{X} \in \mathbb{F}_q^{k\times k} : \mathrm{rk}_q(\boldsymbol{X}) = k\right\}$
$\boldsymbol{\alpha} \xleftarrow{\$} \left\{\boldsymbol{x} \in \mathbb{F}_{q_0}^{n} : x_i \neq x_j, \text{ for } i,j \in [1\!:\!n] \text{ and } i \neq j\right\}$
$\boldsymbol{\eta} \xleftarrow{\$} \left\{\boldsymbol{x} \in \mathbb{F}_q^{\ell_{\mathrm{T}}} : x_i \in \mathbb{F}_{q_i} \setminus \mathbb{F}_{q_{i-1}}, \text{ for } i \in [1\!:\!\ell_{\mathrm{T}}]\right\}$
$\boldsymbol{G}_{\mathrm{pub}} \leftarrow \boldsymbol{S}\boldsymbol{G}_{\boldsymbol{\alpha},\boldsymbol{\tau},\boldsymbol{\pi},\boldsymbol{\eta}} \in \mathbb{F}_q^{k\times n}$, where $\boldsymbol{G}_{\boldsymbol{\alpha},\boldsymbol{\tau},\boldsymbol{\pi},\boldsymbol{\eta}}$ is defined in Definition 2.11
return Private key $(\boldsymbol{S}, \boldsymbol{\alpha}, \boldsymbol{\eta})$, Public key $\boldsymbol{G}_{\mathrm{pub}}$
```

Algorithm 9: $\mathsf{Encrypt}_{\mathrm{TRS}}$

```
Input  : Plaintext vector $\boldsymbol{m} \in \mathbb{F}_q^{k}$
         Public key $\boldsymbol{G}_{\mathrm{pub}} \in \mathbb{F}_q^{k\times n}$
Output: Ciphertext vector $\boldsymbol{y} \in \mathbb{F}_q^{n}$
$\boldsymbol{e}' \xleftarrow{\$} \left\{\boldsymbol{x} \in \mathbb{F}_q^{n} : \mathrm{wt}_{\mathrm{H}}(\boldsymbol{x}) = \left\lfloor \frac{n-k}{2} \right\rfloor\right\}$
$\boldsymbol{y} \leftarrow \boldsymbol{m}\boldsymbol{G}_{\mathrm{pub}} + \boldsymbol{e}' \in \mathbb{F}_q^{n}$
return Ciphertext $\boldsymbol{y}$
```

Algorithm 10: $\mathsf{Decrypt}_{\mathrm{TRS}}$

```
Input  : Ciphertext vector $\boldsymbol{y} \in \mathbb{F}_q^{n}$
         Private key $(\boldsymbol{S}, \boldsymbol{\alpha}, \boldsymbol{\eta}) \in \mathbb{F}_q^{k\times k} \times \mathbb{F}_{q_0}^{n} \times \mathbb{F}_q^{\ell_{\mathrm{T}}}$
         Vector $\boldsymbol{\tau} \in [1\!:\!n-k]^{\ell_{\mathrm{T}}}$
         Vector $\boldsymbol{\pi} \in [0\!:\!k-1]^{\ell_{\mathrm{T}}}$
Output: Plaintext vector $\boldsymbol{m} \in \mathbb{F}_q^{k}$
$\tilde{\boldsymbol{m}} \leftarrow \mathsf{DecodeTRS}(\boldsymbol{y}, \boldsymbol{\alpha}, \boldsymbol{\tau}, \boldsymbol{\pi}, \boldsymbol{\eta}) \in \mathbb{F}_q^{k}$
$\boldsymbol{m} \leftarrow \tilde{\boldsymbol{m}}\boldsymbol{S}^{-1} \in \mathbb{F}_q^{k}$
return Plaintext $\boldsymbol{m}$
```

with the number of twists, which has a significant impact on the key sizes.

4.1.2 A Feasible Key-Recovery Attack on TRS-Based McEliece

In this section, we derive a key-recovery algorithm on $\Pi_{\mathrm{TRS}}^{\mathrm{Enc}}$ for the parameters proposed in [84]. For that, we first prove that *multiple* private keys exist that can be used in

Table 4.1: Parameters for $\Pi_{\mathrm{TRS}}^{\mathrm{Enc}}$ proposed in [84]. The designers claim a security ≥ 100 bits.

q_0	n	k	ℓ_{T}	$\boldsymbol{\tau}$	$\boldsymbol{\pi}$
256	255	117	1	[57]	[88]

$\mathsf{Decrypt}_{\mathrm{TRS}}$, and it is sufficient to determine one of them.

Lemma 4.1. *Let the vectors $\boldsymbol{\alpha}$, $\boldsymbol{\tau}$, $\boldsymbol{\pi}$, and $\boldsymbol{\eta}$ be defined as in Definition 2.11. Then, for any $a \in \mathbb{F}_q^*$, it holds that*

$$\mathcal{TRS}_k(\boldsymbol{\alpha}, \boldsymbol{\tau}, \boldsymbol{\pi}, \boldsymbol{\eta}) = \mathcal{TRS}_k(\hat{\boldsymbol{\alpha}}, \boldsymbol{\tau}, \boldsymbol{\pi}, \hat{\boldsymbol{\eta}}),$$

where $\hat{\boldsymbol{\alpha}} = a\boldsymbol{\alpha}$ and $\hat{\boldsymbol{\eta}} = [\hat{\eta}_1, \ldots, \hat{\eta}_{\ell_{\mathrm{T}}}]$ with $\hat{\eta}_j = \eta_j a^{-(k-1+\tau_j-\pi_j)}$, for $j \in [1:\ell_{\mathrm{T}}]$.

Proof. Let $\mathrm{ev}_{\hat{\boldsymbol{\alpha}}}(f) \in \mathcal{TRS}_k(\hat{\boldsymbol{\alpha}}, \boldsymbol{\tau}, \boldsymbol{\pi}, \hat{\boldsymbol{\eta}})$, where $f(X) = \sum_{i=0}^{k-1} f_i X^i + \sum_{j=1}^{\ell_{\mathrm{T}}} \hat{\eta}_j f_{\pi_j} X^{k-1+\tau_j}$. We have

$$f(aX) = \sum_{i=0}^{k-1} f_i (aX)^i + \sum_{j=1}^{\ell_{\mathrm{T}}} \hat{\eta}_j f_{\pi_j} (aX)^{k-1+\tau_j} = \sum_{i=0}^{k-1} g_i X^i + \sum_{j=1}^{\ell_{\mathrm{T}}} \eta_j g_{\pi_j} X^{k-1+\tau_j} = g(X)\,,$$

where $g_i = f_i a^i$, for $i \in [0:k-1]$, $\eta_j = \hat{\eta}_j a^{k-1+\tau_j-\pi_j}$, for $j \in [1:\ell_{\mathrm{T}}]$, and $g(X) \in \mathcal{P}_k(\boldsymbol{\tau}, \boldsymbol{\pi}, \boldsymbol{\eta})$. We have by definition that $\mathrm{ev}_{\hat{\boldsymbol{\alpha}}}(f) \in \mathcal{TRS}_k(\boldsymbol{\alpha}, \boldsymbol{\tau}, \boldsymbol{\pi}, \boldsymbol{\eta})$, and hence, it follows that $\mathcal{TRS}_k(\hat{\boldsymbol{\alpha}}, \boldsymbol{\tau}, \boldsymbol{\pi}, \hat{\boldsymbol{\eta}}) \subseteq \mathcal{TRS}_k(\boldsymbol{\alpha}, \boldsymbol{\tau}, \boldsymbol{\pi}, \boldsymbol{\eta})$. The proof of the converse inclusion is similar since a is non-zero. ■

The key-recovery algorithm works as follows: In the first step, the algorithm computes a linear transformation of the secret code locators $\boldsymbol{\alpha}$ by exploiting structural properties of the *subfield subcode* of the public code. In the second step, the presented routine determines the coefficients of the twist monomials by Lagrange interpolation. In the final step, the algorithm outputs $(\hat{\boldsymbol{S}}, \hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\eta}})$ such that $\hat{\boldsymbol{S}}\boldsymbol{G}_{\hat{\boldsymbol{\alpha}},\boldsymbol{\tau},\boldsymbol{\pi},\hat{\boldsymbol{\eta}}} = \boldsymbol{G}_{\mathrm{pub}}$. Note, from Lemma 4.1 follows that $(\hat{\boldsymbol{S}}, \hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\eta}})$ is a valid private key and can be used as input of $\mathsf{Decrypt}_{\mathrm{TRS}}$ to decrypt any ciphertext.

First Step: Recovery of an Affine Transformation of the Secret Locators

To determine a linear transformation of the secret code locators $\boldsymbol{\alpha}$ from $\boldsymbol{G}_{\mathrm{pub}}$, we need the following lemma:

Lemma 4.2. *Let $\mathbb{F}_q$ be an extension field of $\mathbb{F}_{q_0}$, let the entries $\boldsymbol{\alpha} \in \mathbb{F}_{q_0}^n$ be distinct, and let $P \in \mathbb{F}_q[X]$ have degree smaller than n. Then, the vector $\mathrm{ev}_{\boldsymbol{\alpha}}(P)$ is in $\mathbb{F}_{q_0}^n$ if and only if $P \in \mathbb{F}_{q_0}[X]$.*

Proof. It is easy to see that if $P \in \mathbb{F}_{q_0}[X]$, then $\mathrm{ev}_{\boldsymbol{\alpha}}(P)$ is in $\mathbb{F}_{q_0}^n$. To prove the converse, choose $\boldsymbol{c} = \mathrm{ev}_{\boldsymbol{\alpha}}(P)$ and suppose that $\boldsymbol{c}$ is in $\mathbb{F}_{q_0}^n$. Since $\boldsymbol{\alpha} \in \mathbb{F}_{q_0}^n$ and $n \leq q_0$, there is a polynomial $Q \in \mathbb{F}_{q_0}[X]$ of a degree of at most n such that $\boldsymbol{c} = \mathrm{ev}_{\boldsymbol{\alpha}}(Q)$. Furthermore, because $\mathrm{ev}_{\boldsymbol{\alpha}}$ is injective over the $\mathbb{F}_q$-subspace of polynomials of a degree smaller than q_0, it must hold that $P = Q$. ■

We denote the set of exponents of monomials that do not support any twist by

$$\bar{\mathcal{I}} \coloneqq [0 \colon k-1] \setminus \{\pi_1, \ldots, \pi_{\ell_\mathrm{T}}\}.$$

Since k and $\pi_1, \ldots, \pi_{\ell_\mathrm{T}}$ are public knowledge, an attacker can compute the set $\bar{\mathcal{I}}$. Because $\pi_i = r_\mathrm{T} - 1 + i$, for $i \in [1 : \ell_\mathrm{T}]$, it holds for valid parameters that $\bar{\mathcal{I}} = [0 : r_\mathrm{T} - 1] \cup [r_\mathrm{T} + \ell_\mathrm{T} \colon k-1]$.

Proposition 4.3. *Let q_0, n, k, ℓ_T, $\boldsymbol{\alpha}$, $\boldsymbol{\tau}$, $\boldsymbol{\pi}$, and $\boldsymbol{\eta}$ be chosen according to the parameter restrictions shown in Section 4.1.1, and choose $\bar{\mathcal{I}} = [0 \colon k-1] \setminus \{\pi_1, \ldots, \pi_{\ell_\mathrm{T}}\}$. Then, the intersection*

$$\mathcal{TRS}_k(\boldsymbol{\alpha}, \boldsymbol{\tau}, \boldsymbol{\pi}, \boldsymbol{\eta}) \cap \mathbb{F}_{q_0}^n = \langle \{\mathrm{ev}_{\boldsymbol{\alpha}}(X^i), i \in \bar{\mathcal{I}}\} \rangle_{q_0}.$$

Proof. Since $\boldsymbol{\alpha}$ is in $\mathbb{F}_{q_0}^n$ and $\mathrm{ev}_{\boldsymbol{\alpha}}(X^i) \in \mathcal{TRS}_k(\boldsymbol{\alpha}, \boldsymbol{\tau}, \boldsymbol{\pi}, \boldsymbol{\eta})$, for $i \in \bar{\mathcal{I}}$, it holds that $\mathrm{ev}_{\boldsymbol{\alpha}}(X^i) \in \mathbb{F}_{q_0}^n$, and thus, $\langle \{\mathrm{ev}_{\boldsymbol{\alpha}}(X^i), i \in \bar{\mathcal{I}}\} \rangle_{q_0} \subseteq \mathcal{TRS}_k(\boldsymbol{\alpha}, \boldsymbol{\tau}, \boldsymbol{\pi}, \boldsymbol{\eta}) \cap \mathbb{F}_{q_0}^n$. To prove the converse, let $f \in \mathcal{P}_k(\boldsymbol{\tau}, \boldsymbol{\pi}, \boldsymbol{\eta})$, and let $\boldsymbol{c} = \mathrm{ev}_{\boldsymbol{\alpha}}(f) \in \mathcal{TRS}_k(\boldsymbol{\alpha}, \boldsymbol{\tau}, \boldsymbol{\pi}, \boldsymbol{\eta}) \cap \mathbb{F}_{q_0}^n$. Since $\deg(f) < n$ for valid parameters, it follows that the polynomial f is in $\mathbb{F}_{q_0}[X]$, see Lemma 4.2, and $\mathbb{F}_{q_0}[X] \cap \mathcal{P}_k(\boldsymbol{\tau}, \boldsymbol{\pi}, \boldsymbol{\eta}) = \langle \{X^i, i \in \bar{\mathcal{I}}\} \rangle_{q_0}$. ■

The previous proposition implies that the subfield subcode

$$\mathcal{C}_{\mathrm{sub}} \coloneqq \mathcal{TRS}_k(\boldsymbol{\alpha}, \boldsymbol{\tau}, \boldsymbol{\pi}, \boldsymbol{\eta}) \cap \mathbb{F}_{q_0}^n$$

is a subcode of the RS code $\mathcal{RS}_k(\boldsymbol{\alpha})$. Let $\mathcal{C}'$ denote a random subcode of $\mathcal{RS}_k(\boldsymbol{\alpha})$. In [17], Wieschebrink showed that the square code $\mathcal{C}'^{(\star 2)}$ is an RS code with high probability and the Sidelnikov–Shestakov attack can be applied to $\mathcal{C}'^{(\star 2)}$ to recover the code locators [15]. In the following proposition, we show that for *all* practical parameters, the square code $\mathcal{C}_{\mathrm{sub}}^{(\star 2)}$ is an RS code and the Sidelnikov–Shestakov attack can be mounted.

Proposition 4.4. *Let q_0, n, k, ℓ_{T}, $\boldsymbol{\alpha}$, $\boldsymbol{\tau}$, $\boldsymbol{\pi}$, and $\boldsymbol{\eta}$ be chosen according to the parameter restrictions shown in Section 4.1.1, let $\bar{\mathcal{I}} = [0:k-1] \setminus \{\pi_1, \dots, \pi_{\ell_{\mathrm{T}}}\}$, and let $\mathcal{C}_{\mathrm{sub}} = \mathcal{TRS}_k(\boldsymbol{\alpha}, \boldsymbol{\tau}, \boldsymbol{\pi}, \boldsymbol{\eta}) \cap \mathbb{F}_{q_0}^n$. If $\ell_{\mathrm{T}} \leq \frac{1}{2}(\sqrt{n}-3)$, then it holds that*

$$(\mathcal{C}_{\mathrm{sub}})^{(\star 2)} = \mathcal{RS}_{2k-1}(\boldsymbol{\alpha}).$$

Proof. By definition,

$$(\mathcal{C}_{\mathrm{sub}})^{(\star 2)} = \langle \{\mathrm{ev}_{\boldsymbol{\alpha}}(X^i) \star \mathrm{ev}_{\boldsymbol{\alpha}}(X^j) \,:\, i,j \in \bar{\mathcal{I}}\} \rangle_{q_0} = \langle \{\mathrm{ev}_{\boldsymbol{\alpha}}(X^i) \,:\, i \in \bar{\mathcal{I}} + \bar{\mathcal{I}}\} \rangle_{q_0}.$$

It is left to show that $\bar{\mathcal{I}} + \bar{\mathcal{I}} = [0:2k-2]$. For valid parameters, it holds that $2k-1 \leq n-3$, $\bar{\mathcal{I}} = \bar{\mathcal{I}}_1 \cup \bar{\mathcal{I}}_2$, where $\bar{\mathcal{I}}_1 \coloneqq [0:r_{\mathrm{T}}-1]$, $\bar{\mathcal{I}}_2 \coloneqq [r_{\mathrm{T}}+\ell_{\mathrm{T}}:k-1]$, and $r_{\mathrm{T}} = \lceil \frac{n+1}{\ell_{\mathrm{T}}+2} \rceil + 2$. Since $\{0\} + \bar{\mathcal{I}}_1 = [0:r_{\mathrm{T}}-1]$, we have $\bar{\mathcal{I}}_1 + \bar{\mathcal{I}}_2 = [r_{\mathrm{T}}+\ell_{\mathrm{T}}:k+r_{\mathrm{T}}-2]$, $\{k-1\} + \bar{\mathcal{I}}_2 = [k+r_{\mathrm{T}}+\ell_{\mathrm{T}}-1:2k-2]$, and

$$\begin{aligned} &[0:r_{\mathrm{T}}-1] \cup [r_{\mathrm{T}}+\ell_{\mathrm{T}}:k+r_{\mathrm{T}}-2] \cup [k+r_{\mathrm{T}}+\ell_{\mathrm{T}}-1:2k-2] \\ &= [0:2k-2] \setminus \Big([r_{\mathrm{T}}:r_{\mathrm{T}}+\ell_{\mathrm{T}}-1] \cup [k+r_{\mathrm{T}}-1:k+r_{\mathrm{T}}+\ell_{\mathrm{T}}-2]\Big) \end{aligned}$$

is a subset of $\bar{\mathcal{I}}+\bar{\mathcal{I}}$. Furthermore, if $\ell_{\mathrm{T}} \leq r_{\mathrm{T}}-1$, then $[r_{\mathrm{T}}:r_{\mathrm{T}}+\ell_{\mathrm{T}}-1] \subseteq \bar{\mathcal{I}}_1+\bar{\mathcal{I}}_1$, where $\ell_{\mathrm{T}} \leq r_{\mathrm{T}}-1$ is always fulfilled for valid parameters since $\ell_{\mathrm{T}} < \sqrt{n}-3$ and $r_{\mathrm{T}} > \sqrt{n}$. From the assumption $\ell_{\mathrm{T}} \leq \frac{1}{2}(\sqrt{n}-3)$ follows that $\ell_{\mathrm{T}} \leq \frac{k-r_{\mathrm{T}}-1}{2}$ due to the constraints on valid parameters. The latter inequality implies that $2(r_{\mathrm{T}}+\ell_{\mathrm{T}}) \leq 2r_{\mathrm{T}}+k-r_{\mathrm{T}}-1 = k+r_{\mathrm{T}}-1$, and therefore, we have $[k+r_{\mathrm{T}}-1:k+r_{\mathrm{T}}+\ell_{\mathrm{T}}-2] \subseteq [2r_{\mathrm{T}}+2\ell_{\mathrm{T}}:2k-2] = \bar{\mathcal{I}}_2+\bar{\mathcal{I}}_2$. ■

Note that the assumption $\ell_{\mathrm{T}} \leq \frac{1}{2}(\sqrt{n}-3)$ is not restrictive in practice, as the decryption algorithm is only feasible for $\ell_{\mathrm{T}} \ll \log n$.

For valid parameters, we have $2k-1 \leq n-3$, and thus, the Sidelnikov–Shestakov attack can be mounted on $\mathcal{C}_{\mathrm{sub}}^{(\star 2)} \subseteq \mathbb{F}_{q_0}^n$. The attack returns a vector of code locators $\boldsymbol{\alpha}' \in \mathbb{F}_{q_0}$ which is an affine transformation of the secret locators $\boldsymbol{\alpha}$, see Theorem 2.1. Stated differently, it holds that $\boldsymbol{\alpha}' = a\boldsymbol{\alpha} + b\mathbf{1}$ for some $a \in \mathbb{F}_{q_0}^*$ and some $b \in \mathbb{F}_{q_0}$, where $\mathbf{1} \coloneqq [1, \dots, 1] \in \mathbb{F}_{q_0}^n$.

Second Step: From an Affine to a Linear Transformation of the Secret Locators

According to Lemma 4.1, if $\hat{\boldsymbol{\alpha}} = a\boldsymbol{\alpha}$ for a non-zero $a \in \mathbb{F}_{q_0}$, then a vector $\hat{\boldsymbol{\eta}}$ exists such that $\mathcal{TRS}_k(\boldsymbol{\alpha}, \boldsymbol{\tau}, \boldsymbol{\pi}, \boldsymbol{\eta}) = \mathcal{TRS}_k(\hat{\boldsymbol{\alpha}}, \boldsymbol{\tau}, \boldsymbol{\pi}, \hat{\boldsymbol{\eta}})$. Thus, given $\boldsymbol{\alpha}' = a\boldsymbol{\alpha} + b\mathbf{1}$, we need to search for a $b \in \mathbb{F}_{q_0}$ such that $\boldsymbol{\alpha}' - b\mathbf{1} = a\boldsymbol{\alpha}$. As q_0 is small, this search can be conducted as follows: For a given vector $\boldsymbol{\alpha}'$, repeatedly sample b from $\mathbb{F}_{q_0}$ and compute the code

$$\mathcal{A}_b \coloneqq \langle \{\mathrm{ev}_{\boldsymbol{\alpha}'-b\mathbf{1}}(X^i) : i \in \bar{\mathcal{I}}\} \rangle_{q_0}$$

until $\mathcal{A}_b \subseteq \mathcal{C}_{\mathrm{pub}}$ is fulfilled. If $\mathcal{A}_b \subseteq \mathcal{C}_{\mathrm{pub}}$ holds, then $\boldsymbol{\alpha}' - b\mathbf{1} = \hat{\boldsymbol{\alpha}}$.

Third Step: Recovery of a Valid Pair $(\hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\eta}})$

Using the previous steps, we can determine a vector $\hat{\boldsymbol{\alpha}} \in \mathbb{F}_{q_0}^n$ whose entries can be used as code locators for the public TRS code. With the following result, we can obtain a vector $\hat{\boldsymbol{\eta}} \in \mathbb{F}_q^n$ such that $\mathcal{TRS}_k(\boldsymbol{\alpha}, \boldsymbol{\tau}, \boldsymbol{\pi}, \boldsymbol{\eta}) = \mathcal{TRS}_k(\hat{\boldsymbol{\alpha}}, \boldsymbol{\tau}, \boldsymbol{\pi}, \hat{\boldsymbol{\eta}})$.

Lemma 4.5. *Let $1 \leq \ell_{\mathrm{T}}$, and let $P(X) = \sum_{i=0}^{k-1} u_i X^i + \sum_{j=1}^{\ell_{\mathrm{T}}} \eta_j u_{\pi_j} X^{k-1+\tau_j} \in \mathcal{P}_k(\boldsymbol{\tau}, \boldsymbol{\pi}, \boldsymbol{\eta})$ such that $u_{\pi_j} \neq 0$. Furthermore, let $\hat{p}_{\pi_j}$ and $\hat{p}_{k-1+\tau_j}$ be the coefficients of the monomials X^{π_j} and $X^{k-1+\tau_j}$ in the polynomial $\hat{P}(X) = P(a^{-1}X)$, respectively. Then, it follows that*

$$\hat{\eta}_j = \eta_j a^{-(k-1+\tau_j-\pi_j)} = \frac{\hat{p}_{k-1+\tau_j}}{\hat{p}_{\pi_j}} \,.$$

Proof. The statement holds since

$$\begin{aligned}
\hat{P}(X) &= P(a^{-1}X) \\
&= \sum_{i=0}^{k-1} u_i a^{-i} X^i + \sum_{j=1}^{\ell_{\mathrm{T}}} \eta_j u_{\pi_j} a^{-(k-1+\tau_j)} X^{k-1+\tau_j} \\
&= \sum_{i=0}^{k-1} \hat{u}_i X^i + \sum_{j=1}^{\ell_{\mathrm{T}}} \hat{\eta}_j \hat{u}_{\pi_j} X^{k-1+\tau_j} \,,
\end{aligned}$$

where $\hat{u}_i = u_i a^{-i}$ for $i \in [0:k-1]$ and $\hat{\eta}_j = \eta_j a^{-(k-1+\tau_j-\pi_j)}$ for $j \in [1:\ell_{\mathrm{T}}]$. ∎

Lemma 4.5 implies that we can determine a vector of coefficients $\hat{\boldsymbol{\eta}}$ such that

$$\mathcal{TRS}_k(\boldsymbol{\alpha}, \boldsymbol{\tau}, \boldsymbol{\pi}, \boldsymbol{\eta}) = \mathcal{TRS}_k(\hat{\boldsymbol{\alpha}}, \boldsymbol{\tau}, \boldsymbol{\pi}, \hat{\boldsymbol{\eta}})$$

as follows: We randomly sample a codeword $\boldsymbol{c} = \mathrm{ev}_{\boldsymbol{\alpha}}(P)$ from $\mathcal{C}_{\mathrm{pub}}$, and we interpolate this codeword $\boldsymbol{c} = \mathrm{ev}_{\hat{\boldsymbol{\alpha}}}(\hat{P})$ as a polynomial evaluated over the code locators $\hat{\boldsymbol{\alpha}}$. Since $\hat{P}(X) = P(a^{-1}X)$, we obtain the coefficient $\hat{\eta}_j$ for each non-zero coefficient u_{π_j} of P.

Note that if $\boldsymbol{c}$ is sampled uniformly from $\mathcal{C}_{\mathrm{pub}}$, then the probability that $u_j = 0$ is approximately $1/q$. Since $\ell_{\mathrm{T}} \ll q$, a random codeword $\boldsymbol{c}$ leads to the recovery of the whole vector $\hat{\boldsymbol{\eta}}$ with high probability.

Final Step: Recovery of an Alternative Private Key $(\hat{\boldsymbol{S}}, \hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\eta}})$

After we have obtained $\hat{\boldsymbol{\alpha}}$ and $\hat{\boldsymbol{\eta}}$, we choose the matrix $\hat{\boldsymbol{S}}$ such that $\hat{\boldsymbol{S}}\boldsymbol{G}_{\hat{\boldsymbol{\alpha}},\boldsymbol{\tau},\boldsymbol{\pi},\hat{\boldsymbol{\eta}}} = \boldsymbol{G}_{\mathrm{pub}}$. It follows that $(\hat{\boldsymbol{S}}, \hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\eta}})$ can be used as a valid private key in $\mathsf{Decrypt}_{\mathrm{TRS}}$ to retrieve any secret plaintext $\boldsymbol{m}$ from its ciphertext $\boldsymbol{y}$.

4.1.3 Analysis of the Key-Recovery Attack

The described attack is summarized in Algorithm 11, where the definitions of the applied functions are given in Table 4.2.

Theorem 4.6. *Let $\boldsymbol{\alpha}$, $\boldsymbol{\tau}$, $\boldsymbol{\pi}$, and $\boldsymbol{\eta}$ be defined as in Section 4.1.1, and let $\boldsymbol{G}_{\mathrm{pub}} \in \mathbb{F}_q^{k\times n}$ be a generator matrix of a TRS code $\mathcal{C}_{\mathrm{pub}} = \mathcal{TRS}_k(\boldsymbol{\alpha}, \boldsymbol{\tau}, \boldsymbol{\pi}, \boldsymbol{\eta}) \subseteq \mathbb{F}_q^n$. Then, Algorithm 11 determines a tuple $(\hat{\boldsymbol{S}}, \hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\eta}})$ from $\boldsymbol{G}_{\mathrm{pub}}$ such that $\hat{\boldsymbol{S}}\boldsymbol{G}_{\hat{\boldsymbol{\alpha}},\boldsymbol{\tau},\boldsymbol{\pi},\hat{\boldsymbol{\eta}}}$ is a generator matrix of $\mathcal{C}_{\mathrm{pub}}$ in $O(\max\{q_0, 2^{\ell_{\mathrm{T}}}, n\}n^3)$ operations in $\mathbb{F}_q$.*

Proof. The correctness of Algorithm 11 was already proven in Section 4.1.2, and therefore, it is left to prove its complexity. For that, we analyze the complexity of each line.

- Line 1 requires $O(n^2(k+n)) \subseteq O(n^3)$ operations in $\mathbb{F}_q$ and $O(n^2(2^{\ell_{\mathrm{T}}}(n-k)+n)) \subseteq O(2^{\ell_{\mathrm{T}}}n^3)$ operations in $\mathbb{F}_{q_0}$.
- Line 2 requires to find a basis of $\langle\{(\boldsymbol{G}_{\mathrm{sub}})_{\{i\},:} \star (\boldsymbol{G}_{\mathrm{sub}})_{\{j\},:}, i,j \in [1:\dim\mathcal{C}_{\mathrm{sub}}]\}\rangle_{q_0}$. Such a basis can be determined in an iterative fashion, where updating the basis with a new element is in $O(n^3)$ operations in $\mathbb{F}_{q_0}$ and needs to be executed $O(n)$ times. Furthermore, rejecting candidates is in $O(n^2)$ operations in $\mathbb{F}_{q_0}$ and needs to be executed $O(n^2)$ times.
- Line 3 requires $O((2k-2)^4 + (2k-2)n) \subseteq O(n^4)$ operations in $\mathbb{F}_{q_0}$ [15].
- Line 4 to Line 8 are in $O(q_0n^3)$ operations in $\mathbb{F}_{q_0}$, as the computation of $\hat{\boldsymbol{\alpha}} \in \mathbb{F}_{q_0}^n$ needs $O(n)$ operations in $\mathbb{F}_{q_0}$, building $\boldsymbol{G}' \in \mathbb{F}_{q_0}^{(k-\ell_{\mathrm{T}})\times n}$ requires $O((k-\ell_{\mathrm{T}})n)$ operations in

Algorithm 11: Key-Recovery Attack on TRS-based McEliece

```
   Input   : Public key $\boldsymbol{G}_{\mathrm{pub}} \in \mathbb{F}_q^{k \times n}$
   Output: Private key $(\hat{\boldsymbol{S}}, \hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\eta}}) \in \mathbb{F}_q^{k \times k} \times \mathbb{F}_{q_0}^{n} \times \mathbb{F}_q^{\ell_{\mathrm{T}}}$

   Step 1: Determine an affine transformation of the secret locators
 1 $\boldsymbol{G}_{\mathrm{sub}} \leftarrow \mathsf{SubfieldSubcode}(\boldsymbol{G}_{\mathrm{pub}}) \in \mathbb{F}_{q_0}^{(k-\ell_{\mathrm{T}}) \times n}$
 2 $\boldsymbol{G}_{\mathrm{sq}} \leftarrow \mathsf{Square}(\boldsymbol{G}_{\mathrm{sub}}) \in \mathbb{F}_{q_0}^{(2k-1) \times n}$
 3 $\boldsymbol{\alpha}' \leftarrow \mathsf{SidelShest}(\boldsymbol{G}_{\mathrm{sq}}) \in \mathbb{F}_{q_0}^{n}$

   Step 2: Determine a linear transformation of the secret locators
 4 for $b \in \mathbb{F}_{q_0}$ do
 5     $\hat{\boldsymbol{\alpha}} \leftarrow (\alpha'_1 - b, \dots, \alpha'_n - b) \in \mathbb{F}_{q_0}^{n}$
 6     $\boldsymbol{G}' \leftarrow \mathsf{GenSub}(\hat{\boldsymbol{\alpha}}) \in \mathbb{F}_{q_0}^{(k-\ell_{\mathrm{T}}) \times n}$
 7     if $\boldsymbol{G}'(\boldsymbol{G}_{\mathrm{sub}}^{\perp})^{\top} = \boldsymbol{0}$ then
 8         break

   Step 3: Determine $\hat{\boldsymbol{\eta}}$
 9 $J \leftarrow [1:\ell_{\mathrm{T}}]$
10 for $i \in [1:k]$ do
11     $\boldsymbol{r}_i \leftarrow$ row $i$ of $\boldsymbol{G}_{\mathrm{pub}}$
12     $P(X) \leftarrow \mathsf{Interpolate}(\hat{\boldsymbol{\alpha}}, \boldsymbol{r}_i) \in \mathbb{F}_q^{n}$
13     for $j \in J$ do
14         if $p_{\pi_j} \neq 0$ then
15             $\hat{\eta}_j \leftarrow \frac{p_{k-1+\tau_j}}{p_{\pi_j}} \in \mathbb{F}_q$
16             $J \leftarrow J \setminus \{j\}$
17     if $J = \varnothing$ then
18         break

   Step 4: Determine $\hat{\boldsymbol{S}}$
19 $\hat{\boldsymbol{G}}_{\mathrm{TRS}} \leftarrow \mathsf{GTRS}(\hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\eta}}) \in \mathbb{F}_q^{k \times n}$
20 $\hat{\boldsymbol{S}} \leftarrow \mathsf{SolveLeft}(\hat{\boldsymbol{G}}_{\mathrm{TRS}}, \boldsymbol{G}_{\mathrm{pub}}) \in \mathbb{F}_q^{k \times k}$
21 return Private key $(\hat{\boldsymbol{S}}, \hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\eta}})$
```

$\mathbb{F}_{q_0}$, the matrix multiplication $\boldsymbol{G}'(\boldsymbol{G}_{\mathrm{sub}}^{\perp})^{\top}$ is in $O((k-\ell_{\mathrm{T}})(n-k+\ell_{\mathrm{T}})n) \subseteq O(n^3)$ operations in $\mathbb{F}_{q_0}$, and in the worst case, these computations have to be performed q_0 times.

- Line 10 to Line 18 are in $O(\ell_{\mathrm{T}} n^3)$ operations in $\mathbb{F}_q$ since a single interpolation is in

Table 4.2: List of functions used in Algorithm 11.

Function	Description
SubfieldSubcode	For a generator matrix of $\mathcal{C}_{\text{pub}}$, it outputs a generator matrix of the subfield subcode of $\mathcal{C}_{\text{pub}}$.
Square	For a generator matrix of $\mathcal{C}_{\text{sub}}$, it outputs a generator matrix of the code $\mathcal{C}_{\text{sub}}^{(\star 2)}$.
SidelShest	For a generator matrix $\boldsymbol{G}$ of an RS code, it outputs a vector of locators $\boldsymbol{\alpha}'$ describing the code.
GenSub	For a vector $\boldsymbol{a} = [a_1, \dots, a_n] \in \mathbb{F}_{q_0}^n$, it outputs a matrix $\boldsymbol{A} \in \mathbb{F}_{q_0}^{(k-\ell_{\text{T}}) \times n}$ whose rows are $[a_1^j, \dots, a_n^j]$, for $j \in \mathcal{I} = [0 : k-1] \setminus \{\pi_1, \dots, \pi_{\ell_{\text{T}}}\}$.
Interpolate	For vectors $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{F}_q^n$, it outputs $P(X)$ such that $\deg(P) < n$ and $P(a_i) = b_i$, for $i \in [1 : n]$.
GTRS	For vectors $\hat{\boldsymbol{\alpha}}$ and $\hat{\boldsymbol{\eta}}$, it outputs the generator matrix $\boldsymbol{G}_{\hat{\boldsymbol{\alpha}}, \boldsymbol{\tau}, \boldsymbol{\pi}, \hat{\boldsymbol{\eta}}}$ of the corresponding TRS code.
SolveLeft	For matrices $\boldsymbol{A}$ and $\boldsymbol{B}$, where $\boldsymbol{A}$ and $\boldsymbol{B}$ have the same rowspace, it outputs a matrix $\boldsymbol{D}$ such that $\boldsymbol{DA} = \boldsymbol{B}$.

$O(n^2)$, and in the worst case, $\ell_{\text{T}} k \leq \ell_{\text{T}} n$ interpolations have to be performed.

- Line 19 is in $O(kn) \subseteq O(n^2)$ operations in $\mathbb{F}_q$.
- Line 20 can be performed by reducing the matrix $\left[\hat{\boldsymbol{G}}_{\text{TRS}}^{\top} \boldsymbol{G}_{\text{pub}}^{\top}\right] \in \mathbb{F}_q^{n \times 2k}$ to row echelon form and is therefore in $O(n^2(2k)) \subseteq O(n^3)$ operations in $\mathbb{F}_q$. ■

Note that in order to obtain moderate key sizes and to ensure an acceptable decryption time, the parameters ℓ_{T} and $q_0 = q^{1/2^{\ell_{\text{T}}}}$ need to be small, e.g., $\ell_{\text{T}} = 1$ and $q_0 = n + 1 = 2^8$ as proposed in [84]. Therefore, Algorithm 11 has a complexity in $O(n^4)$ for practical parameters.

We have implemented our attack in the computer algebra system SageMath [191]. Although the implementation is not optimized, it is capable of recovering a valid private key within a few minutes for the proposed parameters, see Table 4.3.

Table 4.3: Average runtime of Algorithm 11 required on an Intel(R) Core(TM) i7-7600U CPU @ 2.80GHz. The first row refers to parameters suggested by the designers of the system [84]. The remaining security levels were determined according to the formulae given in [84].

q_0	n	k	ℓ_{T}	$\mathrm{wt}_{\mathrm{H}}(\boldsymbol{e})$	Claimed security level	Runtime of Algorithm 11
2^8	255	117	1	83	128 bits	133 seconds
2^8	255	117	2	83	128 bits	141 seconds
2^9	511	200	3	192	192 bits	2260 seconds
2^9	511	170	3	217	256 bits	1532 seconds

4.2 A Power Side-Channel Attack on the HQC KEM

In this section, we derive a power-based side-channel chosen-chiphertext attack on the HQC encryption scheme and on the HQC KEM [183]. For this purpose, we first review the definition of the systems and the underlying security assumptions.

4.2.1 The HQC Encryption Scheme

The HQC scheme relies on two distinct codes. It is based on a public code $\mathcal{C}_{\mathrm{HQC}} \subseteq \mathbb{F}_2^n$ of dimension k and length n, where an efficient encoding algorithm EncHQC and an efficient decoding algorithm DecHQC are public knowledge. The second code has dimension n, length $2n$, and a parity-check matrix $[\boldsymbol{I}, \mathrm{rot}(\boldsymbol{h})] \in \mathbb{F}_2^{n \times 2n}$, where $\boldsymbol{I} \in \mathbb{F}_2^{n \times n}$ denotes the identity matrix and $\boldsymbol{h}$ is uniformly sampled from $\mathbb{F}_2^n$. Contrary to $\mathcal{C}_{\mathrm{HQC}}$, one assumes that *no party* knows an efficient decoding algorithm for the latter code. In particular, an adversary must not possess an efficient algorithm since otherwise the HQC system would be insecure. Note that an efficient decoding of the code with parity-check matrix $[\boldsymbol{I}, \mathrm{rot}(\boldsymbol{h})] \in \mathbb{F}_2^{n \times 2n}$ is neither needed in the encryption nor in the decryption algorithm of HQC.

In the following, we state the IND-CPA-secure variant of the HQC public-key encryption scheme $\Pi_{\mathrm{HQC}}^{\mathrm{Enc}}$ as it was submitted to the second round of the NIST post-quantum cryptography standardization [183]. The proposal is defined by

$$\Pi_{\mathrm{HQC}}^{\mathrm{Enc}} := (\mathsf{KeyGen}_{\mathrm{HQC}}, \mathsf{Encrypt}_{\mathrm{HQC}}, \mathsf{Decrypt}_{\mathrm{HQC}}),$$

where the algorithms $\mathsf{KeyGen}_{\mathrm{HQC}}$, $\mathsf{Encrypt}_{\mathrm{HQC}}$, and $\mathsf{Decrypt}_{\mathrm{HQC}}$ are given in Algorithms 12 to 14. The algorithms encode in and decode in $\mathcal{C}_{\mathrm{HQC}}$ using the functions EncHQC and DecHQC, which are formally defined in Section 4.2.2. The parameter sets for the security levels 128 bit, 192 bit, and 256 bit are shown in Table 4.4. In [192], Hofheinz *et al.* present a generic transformation of IND-CPA-secure encryption schemes into IND-CCA2-secure KEMs. This transformation is utilized in the HQC proposal and leads to the encapsulation and decapsulation algorithms of the HQC KEM described in [183]. Note that our attack exploits side-channel information that is leaked in the algorithm $\mathsf{Decrypt}_{\mathrm{HQC}}$. Since the execution of $\mathsf{Decrypt}_{\mathrm{HQC}}$ is the first step in the decapsulation algorithm of the KEM variant of HQC, it suffices to derive the attack based on the IND-CPA-secure encryption scheme of HQC. The proposed attack can then be mounted in the same way on the IND-CCA2-secure KEM variant of HQC.

Table 4.4: Parameter sets for HQC presented in [183].

Instance	n_1	n_2	n	k	m	w_{y}	w_{r}	w_{e}	δ
HQC-128	766	31	23869	256	10	67	77	77	57
HQC-192	766	59	45197	256	10	101	117	117	57
HQC-256	796	87	69259	256	10	133	153	153	60

Algorithm 12: $\mathsf{KeyGen}_{\mathrm{HQC}}$

```
Input  : Non-negative integers $n_1, n_2, n, k, \delta, w_{\mathrm{y}}, w_{\mathrm{r}}, w_{\mathrm{e}}$
Output: Private key $(\boldsymbol{x}, \boldsymbol{y}) \in \mathbb{F}_2^n \times \mathbb{F}_2^n$
        Public key $(\boldsymbol{h}, \boldsymbol{s}) \in \mathbb{F}_2^n \times \mathbb{F}_2^n$
Choose the $[n,k]_{\mathbb{F}_2}$ code $\mathcal{C}_{\mathrm{HQC}}$ and make the code public
$\boldsymbol{h} \xleftarrow{\$} \mathbb{F}_2^n$
$\boldsymbol{x} \xleftarrow{\$} \{\boldsymbol{a} \in \mathbb{F}_2^n : \mathrm{wt}_{\mathrm{H}}(\boldsymbol{a}) = w_{\mathrm{y}}\}$
$\boldsymbol{y} \xleftarrow{\$} \{\boldsymbol{a} \in \mathbb{F}_2^n : \mathrm{wt}_{\mathrm{H}}(\boldsymbol{a}) = w_{\mathrm{y}}\}$
$\boldsymbol{s} \leftarrow \boldsymbol{x} + \boldsymbol{h}\boldsymbol{y} \in \mathbb{F}_2^n$
return Private key $(\boldsymbol{x}, \boldsymbol{y})$, Public key $(\boldsymbol{h}, \boldsymbol{s})$
```

Algorithm 13: $\mathsf{Encrypt}_{\mathrm{HQC}}$

```
Input  : Non-negative integers $n, k, \delta, w_{\mathrm{y}}, w_{\mathrm{r}}, w_{\mathrm{e}}$
         Plaintext vector $\boldsymbol{m} \in \mathbb{F}_2^k$
         Public key $(\boldsymbol{h}, \boldsymbol{s}) \in \mathbb{F}_2^n \times \mathbb{F}_2^n$
Output: Ciphertext $(\boldsymbol{u}, \boldsymbol{v}) \in \mathbb{F}_2^n \times \mathbb{F}_2^n$
$\boldsymbol{e}' \xleftarrow{\$} \{\boldsymbol{a} \in \mathbb{F}_2^n : \mathrm{wt}_{\mathrm{H}}(\boldsymbol{a}) = w_{\mathrm{e}}\}$
$\boldsymbol{r}_1 \xleftarrow{\$} \{\boldsymbol{a} \in \mathbb{F}_2^n : \mathrm{wt}_{\mathrm{H}}(\boldsymbol{a}) = w_{\mathrm{r}}\}$
$\boldsymbol{r}_2 \xleftarrow{\$} \{\boldsymbol{a} \in \mathbb{F}_2^n : \mathrm{wt}_{\mathrm{H}}(\boldsymbol{a}) = w_{\mathrm{r}}\}$
$\boldsymbol{u} \leftarrow \boldsymbol{r}_1 + \boldsymbol{h}\boldsymbol{r}_2 \in \mathbb{F}_2^n$
$\boldsymbol{v} \leftarrow \mathsf{EncHQC}(\mathcal{C}_{\mathrm{HQC}}, \boldsymbol{m}) + \boldsymbol{s}\boldsymbol{r}_2 + \boldsymbol{e}' \in \mathbb{F}_2^n$
return Ciphertext $(\boldsymbol{u}, \boldsymbol{v})$
```

```
Algorithm 14: $\mathsf{Decrypt}_{\mathrm{HQC}}$
  Input  : Ciphertext $(\boldsymbol{u}, \boldsymbol{v}) \in \mathbb{F}_2^n \times \mathbb{F}_2^n$
           Private key $(\boldsymbol{x}, \boldsymbol{y}) \in \mathbb{F}_2^n \times \mathbb{F}_2^n$
  Output: Plaintext vector $\boldsymbol{m} \in \mathbb{F}_2^k$
  $\boldsymbol{v}' \leftarrow \boldsymbol{v} - \boldsymbol{u}\boldsymbol{y} \in \mathbb{F}_2^n$
  $\boldsymbol{m} \leftarrow \mathsf{DecHQC}(\mathcal{C}_{\mathrm{HQC}}, \boldsymbol{v}') \in \mathbb{F}_2^k$
  return Plaintext vector $\boldsymbol{m}$
```

4.2.2 The Error-Correcting Code $\mathcal{C}_{\mathrm{HQC}}$

In the original proposal, the code $\mathcal{C}_{\mathrm{HQC}}$ is chosen as a product code of a shortened Bose–Chaudhuri–Hocquenghem (BCH) code $\mathcal{C}_1$, which has a generator matrix $\boldsymbol{G}_1 \in \mathbb{F}_2^{k \times n_1}$ and an error correction capability δ, and a repetition code $\mathcal{C}_2$ of length n_2. Note that the HQC proposal was recently modified and now comprises of an additional variant called HQC-RMRS. The aforementioned modification deploys a code concatenation of a Reed–Muller code and an RS code for the error-correcting code $\mathcal{C}_{\mathrm{HQC}}$. This modification does not result from security concerns regarding the original HQC scheme but allows one to reduce the parameter sizes due to an improved error correction capability of $\mathcal{C}_{\mathrm{HQC}}$. Attacking the new variant is beyond the scope of this dissertation, and therefore, we consider only the original proposal in the following.

Encoding algorithm

To encode information vectors into codewords of $\mathcal{C}_{\mathrm{HQC}}$, the mapping

$$\begin{aligned} \mathsf{EncHQC} : \mathbb{F}_2^k &\to \mathbb{F}_2^n, \\ \boldsymbol{m} &\mapsto [\underbrace{m'_1, \dots, m'_1}_{n_2 \text{ times}}, \underbrace{m'_2, \dots, m'_2}_{n_2 \text{ times}}, m'_3, \dots, m'_{n_1}, \underbrace{0, 0, \dots, 0}_{n - n_1 n_2 \text{ times}}] \end{aligned}$$

is applied, where $\boldsymbol{m}' = [m'_1, \dots, m'_{n_1}] = \boldsymbol{m}\boldsymbol{G}_1$, and $\boldsymbol{G}_1 \in \mathbb{F}_2^{k \times n_1}$ is a generator matrix of the shortened BCH code $\mathcal{C}_1$.

Decoding algorithm

Given the vector $\boldsymbol{v}' = [\boldsymbol{v}'_1, \dots, \boldsymbol{v}'_{n_1}, \boldsymbol{v}'_{n_1+1}] \in \mathbb{F}_2^n$, where $\boldsymbol{v}'_1, \dots, \boldsymbol{v}'_{n_1} \in \mathbb{F}_2^{n_2}$ and $\boldsymbol{v}'_{n_1+1} \in \mathbb{F}_2^{n-n_1 n_2}$, the applied decoding routine $\mathsf{DecHQC} : \mathbb{F}_2^n \to \mathbb{F}_2^k$ consists of two consecutive algorithms. In the first algorithm, the vectors $\boldsymbol{v}'_1, \dots, \boldsymbol{v}'_{n_1}$ are decoded separately in the repetition code $\mathcal{C}_2$ to a vector $\tilde{\boldsymbol{v}} = [\tilde{v}_1, \dots, \tilde{v}_{n_1}] \in \mathbb{F}_2^{n_1}$, where $\tilde{v}_i$ is 1 if $\mathrm{wt}_{\mathrm{H}}(\boldsymbol{v}'_i) \geq \left\lceil \frac{n_2}{2} \right\rceil$ and 0 otherwise. In the second algorithm, the vector $\tilde{\boldsymbol{v}}$ is decoded in the BCH code $\mathcal{C}_1$ to the vector $\boldsymbol{m} \in \mathbb{F}_2^k$. In the proposal, a key equation based approach is applied for the decoding of $\mathcal{C}_1$. In this algorithm, the syndromes are computed by the transpose of the additive fast Fourier transformation [193], the error locator polynomial is obtained by a modification of Berlekamp's algorithm, and the error values are determined with an additive fast Fourier transformation.

4.2.3 The Security of HQC

In [183, Thm. 4.1], it is shown that the public-key encryption version of HQC is IND-CPA secure under the assumption that both the problem Decisional 2-Quasi-Cyclic Hamming Syndrome Decoding ($\mathsf{Dec2QCSD}_{\mathrm{H}}$) [183, Def. 2.1.15] and the problem Decisional 3-Quasi-Cyclic Hamming Syndrome Decoding [183, Def. 2.1.17] are hard. For our attack, it is crucial to observe that retrieving the private key $(\boldsymbol{x}, \boldsymbol{y})$ from the public key $(\boldsymbol{h}, \boldsymbol{s})$ is equal to solving an instance of the search version of the former problem, where we denote the search version by $\mathsf{Sea2QCSD}_{\mathrm{H}}$. This can be seen by

$$\boldsymbol{s} = \boldsymbol{x} + \boldsymbol{h}\boldsymbol{y} = [\boldsymbol{x}, \boldsymbol{y}] \begin{bmatrix} \boldsymbol{I} \\ \mathrm{rot}(\boldsymbol{h})^\top \end{bmatrix} = \boldsymbol{e}\boldsymbol{H}^\top,$$

where $\boldsymbol{e} := [\boldsymbol{x}, \boldsymbol{y}] \in \mathbb{F}_2^{2n}$ with $\mathrm{wt}_{\mathrm{H}}(\boldsymbol{x}) = \mathrm{wt}_{\mathrm{H}}(\boldsymbol{y}) = w_{\mathrm{y}}$ and $\boldsymbol{H} := [\boldsymbol{I}, \mathrm{rot}(\boldsymbol{h})] \in \mathbb{F}_2^{n \times 2n}$. It follows that the vector $\boldsymbol{s}$ can be interpreted as the syndrome of the error vector $\boldsymbol{e}$ and the parity-check matrix $\boldsymbol{H}$.

The $\mathsf{Sea2QCSD}_{\mathrm{H}}$ problem can be solved by ISD algorithms, e.g., Prange's ISD algorithm [134] or one of its improvements (for instance [160, 161, 194–196]). The idea of Prange's ISD algorithm is to guess an error-free information set, see Section 2.2.4. For an error weight $\mathrm{wt}_{\mathrm{H}}(\boldsymbol{e}) = 2w_{\mathrm{y}}$, the probability that a set of cardinality n which is drawn uniformly at random from $[1:2n]$ is an error-free information set is approximately $\binom{n}{2w_{\mathrm{y}}} \big/ \binom{2n}{2w_{\mathrm{y}}}$. To check whether a set of indices is an error-free information set has a complexity in $O((2n)^3)$. It follows that the average complexity of Prange's algorithm for solving the $\mathsf{Sea2QCSD}_{\mathrm{H}}$ problem is approximately $(2n)^3 \binom{2n}{2w_{\mathrm{y}}} \big/ \binom{n}{2w_{\mathrm{y}}}$, cf. Section 2.2.4.

The considered power side-channel attack determines information about the support of $\boldsymbol{y}$, which can be incorporated in ISD algorithms to reduce their complexity. We later state the exact information that we retrieve by the side-channel attack, but for now, it suffices to consider the following generalized version of the $\mathsf{Sea2QCSD}_{\mathrm{H}}$ problem.

Problem 4.1 (Search 2 Hamming Syndrome Decoding ($\mathsf{Sea2SD}_{\mathrm{H}}$) Problem)**.**

Given:
- *Parity-check matrix* $\boldsymbol{H} \in \mathbb{F}_2^{(n+n'-k')\times(n+n')}$ *of an* $[n+n', k']_{\mathbb{F}_2}$ *code* $\mathcal{C}$
- *Non-negative integers* n, n', w_{y}, w'
- *Syndrome vector* $\boldsymbol{s} = [\boldsymbol{x}, \boldsymbol{y}]\boldsymbol{H}^\top \in \mathbb{F}_2^{n+n'-k'}$, *where* $\boldsymbol{x} \in \mathbb{F}_2^n$, $\mathrm{wt_H}(\boldsymbol{x}) = w_{\mathrm{y}}$, $\boldsymbol{y} \in \mathbb{F}_2^{n'}$, *and* $\mathrm{wt_H}(\boldsymbol{y}) = w'$

Objective: *Search for an* $\boldsymbol{e}' \in \mathbb{F}_2^{n+n'}$ *such that* $\mathrm{wt_H}(\boldsymbol{e}') \leq w_{\mathrm{y}} + w'$ *and* $\boldsymbol{s} = \boldsymbol{e}'\boldsymbol{H}^\top$.

To solve the $\mathsf{Sea2SD}_{\mathrm{H}}$ problem, we propose modifications of Prange's algorithm [134], Lee and Brickell's algorithm [160], and Stern's algorithm [161] in Algorithm 15, 16, and 17, respectively.

As in the algorithm proposed by Prange, the goal of Algorithm 15 is to obtain an error-free information set. We keep the part of the algorithm by Prange which tests whether a chosen set is an error-free information set, but we modify the method of choosing the indices. In the following theorem, we prove that Algorithm 15 finds a solution to the $\mathsf{Sea2SD}_{\mathrm{H}}$ problem and derive the complexity of the algorithm.

Algorithm 15: Modified Prange Algorithm

```
Input  : Parity-check matrix $\boldsymbol{H} \in \mathbb{F}_2^{(n+n'-k')\times(n+n')}$
         Non-negative integers $n$, $n'$, $w_{\mathrm{y}}$, $w'$
         Syndrome vector $\boldsymbol{s} \in \mathbb{F}_2^{n+n'-k'}$
         Non-negative integer $k_1$
Output: Vector $\boldsymbol{e}' \in \mathbb{F}_2^{n+n'}$
$\boldsymbol{e}' \leftarrow \boldsymbol{0} \in \mathbb{F}_2^{n+n'}$
while $\mathrm{wt_H}(\boldsymbol{e}') > w_{\mathrm{y}} + w' \vee \boldsymbol{s} \neq \boldsymbol{e}'\boldsymbol{H}^\top$ do
    $\mathcal{X}_1 \xleftarrow{\$} \{\mathcal{S} \subseteq [1:n] : |\mathcal{S}| = k_1\}$
    $\mathcal{X}_2 \xleftarrow{\$} \{\mathcal{S} \subseteq [n+1:n+n'] : |\mathcal{S}| = k' - k_1\}$
    $\boldsymbol{e}' \leftarrow$ Iteration of the original Prange algorithm w.r.t. the set $\mathcal{X}_1 \cup \mathcal{X}_2$
return $\boldsymbol{e}'$
```

Theorem 4.7. *Let* k_1 *be a non-negative integer such that* $k' - n' \leq k_1 \leq \min\{k', n\}$, *and suppose that there is only one solution to the considered instance of the* $\mathsf{Sea2SD}_{\mathrm{H}}$ *problem. Then, Algorithm 15 solves the* $\mathsf{Sea2SD}_{\mathrm{H}}$ *problem with, on average, approxi-*

mately

$$W_{\mathrm{ModPr}} = (n+n')^3 \frac{\binom{n}{w_{\mathrm{y}}}\binom{n'}{w'}}{\binom{n-k_1}{w_{\mathrm{y}}}\binom{n'-k'+k_1}{w'}}$$

operations in $\mathbb{F}_2$*.*

Proof. Instead of drawing k' indices uniformly at random from $[1:n+n']$ as in the original algorithm by Prange, Algorithm 15 draws k_1 indices uniformly at random from $[1:n]$ and k_2 indices uniformly at random from $[n+1:n+n']$, where $k_1+k_2=k'$. Then, the probability of drawing an error-free information set is approximately $\binom{n-k_1}{w_{\mathrm{y}}}\big/\binom{n}{w_{\mathrm{y}}}\cdot\binom{n'-k_2}{w'}\big/\binom{n'}{w'}$, and it follows that the complexity of this modified algorithm is given by

$$W_{\mathrm{ModPr}} = (n+n')^3 \frac{\binom{n}{w_{\mathrm{y}}}\binom{n'}{w'}}{\binom{n-k_1}{w_{\mathrm{y}}}\binom{n'-k_2}{w'}} = (n+n')^3 \frac{\binom{n}{w_{\mathrm{y}}}\binom{n'}{w'}}{\binom{n-k_1}{w_{\mathrm{y}}}\binom{n'-k'+k_1}{w'}}. \qquad \blacksquare$$

Algorithm 16: Modified Lee–Brickell Algorithm

```
Input  : Parity-check matrix $\boldsymbol{H} \in \mathbb{F}_2^{(n+n'-k')\times(n+n')}$
         Non-negative integers $n$, $n'$, $w_{\mathrm{y}}$, $w'$
         Syndrome vector $\boldsymbol{s} \in \mathbb{F}_2^{n+n'-k'}$
         Non-negative integers $k_1$, $p_{\mathrm{LB}}$
Output: Vector $\boldsymbol{e}' \in \mathbb{F}_2^{n+n'}$
$\boldsymbol{e}' \leftarrow \boldsymbol{0} \in \mathbb{F}_2^{n+n'}$
while $\mathrm{wt}_{\mathrm{H}}(\boldsymbol{e}') > w_{\mathrm{y}} + w' \lor \boldsymbol{s} \neq \boldsymbol{e}'\boldsymbol{H}^\top$ do
    $\mathcal{X}_1 \xleftarrow{\$} \{\mathcal{S} \subseteq [1:n] : |\mathcal{S}| = k_1\}$
    $\mathcal{X}_2 \xleftarrow{\$} \{\mathcal{S} \subseteq [n+1:n+n'] : |\mathcal{S}| = k'-k_1\}$
    $\boldsymbol{e}' \leftarrow$ Iteration of the original Lee–Brickell algorithm w.r.t. the set $\mathcal{X}_1 \cup \mathcal{X}_2$
      and the parameter $p_{\mathrm{LB}}$
return $\boldsymbol{e}'$
```

Furthermore, the Lee–Brickell algorithm [160] can also be used to solve the aforementioned $\mathsf{Sea2SD}_{\mathrm{H}}$ problem. This algorithm is similar to Prange's algorithm but allows p_{LB} errors in the drawn information set. This relaxation implies that the probability of drawing such an information set is increased compared to Prange's algorithm, but the complexity of one iteration is also slightly higher [160]. We modify the Lee–Brickell algorithm in the same way as Prange's algorithm, which means that we keep the part of the algorithm by Lee–Brickell which checks whether a chosen set is of the desired form, but we change the method of sampling the indices. In the next theorem, we prove the correctness of the modified Lee–Brickell algorithm and state its complexity.

Theorem 4.8. *Let k_1 and p_{LB} be non-negative integers such that $k' - n' \leq k_1 \leq \min\{k', n\}$ and $p_{\mathrm{LB}} \leq w' + w_{\mathrm{y}}$, and suppose that there is only one solution to the considered instance of the* $\mathsf{Sea2SD_H}$ *problem. Then, Algorithm 16 solves the* $\mathsf{Sea2SD_H}$ *problem with, on average, approximately*

$$W_{\mathrm{ModLB}} \coloneqq \frac{W_{\mathrm{LB,Iter}}}{P_{\mathrm{LB}}}$$

operations in $\mathbb{F}_2$, where

$$W_{\mathrm{LB,Iter}} \coloneqq (n+n')^3 + (n+n'-k')(p_{\mathrm{LB}}+1)\binom{k'}{p_{\mathrm{LB}}}$$

and

$$P_{\mathrm{LB}} \coloneqq \sum_{\substack{\boldsymbol{a} \in \mathbb{N}_0^2 \\ a_1 \leq w_{\mathrm{y}} \\ a_2 \leq w' \\ a_1+a_2=p_{\mathrm{LB}}}} \frac{\binom{k_1}{a_1}\binom{n-k_1}{w_{\mathrm{y}}-a_1}}{\binom{n}{w_{\mathrm{y}}}} \frac{\binom{k'-k_1}{a_2}\binom{n'-k'+k_1}{w'-a_2}}{\binom{n'}{w'}}.$$

Proof. Instead of drawing k' indices uniformly at random from $[1:n+n']$ as in the original algorithm by Lee and Brickell, Algorithm 16 draws k_1 indices uniformly at random from $[1:n]$ and k_2 indices uniformly at random from $[n+1:n+n']$. As we allow exactly p_{LB} errors among these $k_1+k_2=k'$ positions, the success probability of drawing a set in the desired form is approximately

$$P_{\mathrm{LB}} \coloneqq \sum_{\substack{\boldsymbol{a} \in \mathbb{N}_0^2 \\ a_1 \leq w_{\mathrm{y}} \\ a_2 \leq w' \\ a_1+a_2=p_{\mathrm{LB}}}} \frac{\binom{k_1}{a_1}\binom{n-k_1}{w_{\mathrm{y}}-a_1}}{\binom{n}{w_{\mathrm{y}}}} \frac{\binom{k_2}{a_2}\binom{n'-k_2}{w'-a_2}}{\binom{n'}{w'}} = \sum_{\substack{\boldsymbol{a} \in \mathbb{N}_0^2 \\ a_1 \leq w_{\mathrm{y}} \\ a_2 \leq w' \\ a_1+a_2=p_{\mathrm{LB}}}} \frac{\binom{k_1}{a_1}\binom{n-k_1}{w_{\mathrm{y}}-a_1}}{\binom{n}{w_{\mathrm{y}}}} \frac{\binom{k'-k_1}{a_2}\binom{n'-k'+k_1}{w'-a_2}}{\binom{n'}{w'}}.$$

As we adapt the original Lee–Brickell algorithm only by changing the selection of the information set, the cost per iteration

$$W_{\mathrm{LB,Iter}} \coloneqq (n+n')^3 + (n+n'-k')(p_{\mathrm{LB}}+1)\binom{k'}{p_{\mathrm{LB}}}$$

stays the same [197, Thm. 2.7], and we get a complexity of

$$W_{\mathrm{ModLB}} \coloneqq \frac{W_{\mathrm{LB,Iter}}}{P_{\mathrm{LB}}}$$

operations in $\mathbb{F}_2$. ■

Note that P_{LB} can be computed in polynomial time [155].

Algorithm 17: Modified Stern Algorithm

```
Input  : Parity-check matrix $\boldsymbol{H} \in \mathbb{F}_2^{(n+n'-k')\times(n+n')}$
         Non-negative integers $n$, $n'$, $w_{\mathrm{y}}$, $w'$
         Syndrome vector $\boldsymbol{s} \in \mathbb{F}_2^{n+n'-k'}$
         Non-negative integers $k_1$, $p_{\mathrm{LB}}$, $\nu_{\mathrm{St},1}$, $\nu_{\mathrm{St},2}$
Output: Vector $\boldsymbol{e}' \in \mathbb{F}_2^{n+n'}$
$\boldsymbol{e}' \leftarrow \boldsymbol{0} \in \mathbb{F}_2^{n+n'}$
while $\mathrm{wt_H}(\boldsymbol{e}') > w_{\mathrm{y}} + w' \vee \boldsymbol{s} \neq \boldsymbol{e}'\boldsymbol{H}^\top$ do
    $\mathcal{X}_1 \xleftarrow{\$} \{\mathcal{S} \subseteq [1:n] : |\mathcal{S}| = \lfloor k_1/2 \rfloor\}$
    $\mathcal{Y}_1 \xleftarrow{\$} \{\mathcal{S} \subseteq [1:n] \setminus \mathcal{X}_1 : |\mathcal{S}| = \lceil k_1/2 \rceil\}$
    $\mathcal{Z}_1 \xleftarrow{\$} \{\mathcal{S} \subseteq [1:n] \setminus (\mathcal{X}_1 \cup \mathcal{Y}_1) : |\mathcal{S}| = \nu_{\mathrm{St},1}\}$
    $\mathcal{X}_2 \xleftarrow{\$} \{\mathcal{S} \subseteq [n+1:n+n'] : |\mathcal{S}| = \lfloor (k'-k_1)/2 \rfloor\}$
    $\mathcal{Y}_2 \xleftarrow{\$} \{\mathcal{S} \subseteq [n+1:n+n'] \setminus \mathcal{X}_2 : |\mathcal{S}| = \lceil (k'-k_1)/2 \rceil\}$
    $\mathcal{Z}_2 \xleftarrow{\$} \{\mathcal{S} \subseteq [n+1:n+n'] \setminus (\mathcal{X}_2 \cup \mathcal{Y}_2) : |\mathcal{S}| = \nu_{\mathrm{St},2}\}$
    $\boldsymbol{e}' \leftarrow$ Iteration of the original Stern algorithm w.r.t. the sets $\mathcal{X}_1 \cup \mathcal{X}_2$,
      $\mathcal{Y}_1 \cup \mathcal{Y}_2$, $\mathcal{Z}_1 \cup \mathcal{Z}_2$ and the parameters $p_{\mathrm{LB}}$ and $\nu_{\mathrm{St}} = \nu_{\mathrm{St},1} + \nu_{\mathrm{St},2}$
return $\boldsymbol{e}'$
```

Another algorithm to solve the $\mathsf{Sea2SD_H}$ problem is Stern's algorithm [161]. This algorithm uses two parameters p_{St} and ν_{St} to draw an information set. Again, there can be some errors in the information set, but now, the error positions outside the information set are restricted. Stern's algorithm divides the information set into two equal-size subsets $\mathcal{X}$ and $\mathcal{Y}$ and looks for vectors of Hamming weight p_{St} at the indices in $\mathcal{X}$, of Hamming weight p_{St} at the indices in $\mathcal{Y}$, and of Hamming weight 0 on a fixed uniform random set $\mathcal{Z}$ of ν_{St} positions outside the information set. As before, we keep the part of the algorithm by Stern which examines whether a chosen set is of the desired form, but we change the way of sampling the indices of the set.

Theorem 4.9. *Let k_1, p_{LB}, $\nu_{\mathrm{St},1}$, and $\nu_{\mathrm{St},2}$ be non-negative integers such that $k' - n' \leq k_1 \leq \min\{k', n\}$, $p_{\mathrm{LB}} \leq w_{\mathrm{y}} + w'$, $\nu_{\mathrm{St},1} \leq n - k_1$, and $\nu_{\mathrm{St},2} \leq n - k' + k_1$. Furthermore, suppose that there is only one solution to the considered instance of the* $\mathsf{Sea2SD_H}$ *problem. Then, Algorithm 17 solves the* $\mathsf{Sea2SD_H}$ *problem with, on average, approximately*

$$W_{\mathrm{ModSt}} := \frac{W_{\mathrm{St,Iter}}}{P_{\mathrm{St}}}$$

operations in $\mathbb{F}_2$, where

$$\begin{aligned} W_{\mathrm{St,Iter}} := (n+n')^3 + (\nu_{\mathrm{St},1} + \nu_{\mathrm{St},2}) \left(\sum_{i=1}^{p_{\mathrm{St}}} \binom{M_1}{i} + \sum_{i=1}^{p_{\mathrm{St}}} \binom{M_2}{i} - k' + \binom{M_2}{p_{\mathrm{St}}} \right) \\ + 2^{1-\nu_{\mathrm{St},1}-\nu_{\mathrm{St},2}} \binom{M_1}{p_{\mathrm{St}}} \binom{M_2}{p_{\mathrm{St}}} (w_{\mathrm{y}} + w' - 2p_{\mathrm{St}} + 1)(2p_{\mathrm{St}} + 1), \end{aligned}$$

the quantities $M_1 = \lfloor k_1/2 \rfloor + \lfloor (k'-k_1)/2 \rfloor$ and $M_2 = \lceil k_1/2 \rceil + \lceil (k'-k_1)/2 \rceil$, and

$$P_{\mathrm{St}} := \sum_{\substack{\boldsymbol{a} \in \mathbb{N}_0^2 \\ a_1 \leq w_{\mathrm{y}} \\ a_2 \leq w' \\ a_1 + a_2 = p_{\mathrm{St}}}} \sum_{\substack{\boldsymbol{b} \in \mathbb{N}_0^2 \\ b_1 \leq w_{\mathrm{y}} - a_1 \\ b_2 \leq w' - a_2 \\ b_1 + b_2 = p_{\mathrm{St}}}} \frac{\binom{\lfloor k_1/2 \rfloor}{a_1} \binom{\lceil k_1/2 \rceil}{b_1} \binom{n-k_1-\nu_{\mathrm{St},1}}{w_{\mathrm{y}}-a_1-b_1}}{\binom{n}{w_{\mathrm{y}}}} \frac{\binom{\lfloor (k'-k_1)/2 \rfloor}{a_2} \binom{\lceil (k'-k_1)/2 \rceil}{b_2} \binom{n'-k'+k_1-\nu_{\mathrm{St},2}}{w'-a_2-b_2}}{\binom{n'}{w'}}.$$

Proof. Instead of drawing k' indices uniformly at random from $[1 : n+n']$ as in the original algorithm by Stern, Algorithm 17 draws

$$\begin{aligned} \mathcal{X}_1 &\xleftarrow{\$} \{\mathcal{S} \subseteq [1 \!:\! n] : |\mathcal{S}| = \lfloor k_1/2 \rfloor\} \\ \mathcal{Y}_1 &\xleftarrow{\$} \{\mathcal{S} \subseteq [1 \!:\! n] \setminus \mathcal{X}_1 : |\mathcal{S}| = \lceil k_1/2 \rceil\} \\ \mathcal{Z}_1 &\xleftarrow{\$} \{\mathcal{S} \subseteq [1 \!:\! n] \setminus (\mathcal{X}_1 \cup \mathcal{Y}_1) : |\mathcal{S}| = \nu_{\mathrm{St},1}\} \\ \mathcal{X}_2 &\xleftarrow{\$} \{\mathcal{S} \subseteq [n+1 \!:\! n+n'] : |\mathcal{S}| = \lfloor k_2/2 \rfloor\} \\ \mathcal{Y}_2 &\xleftarrow{\$} \{\mathcal{S} \subseteq [n+1 \!:\! n+n'] \setminus \mathcal{X}_2 : |\mathcal{S}| = \lceil k_2/2 \rceil\} \\ \mathcal{Z}_2 &\xleftarrow{\$} \{\mathcal{S} \subseteq [n+1 \!:\! n+n'] \setminus (\mathcal{X}_2 \cup \mathcal{Y}_2) : |\mathcal{S}| = \nu_{\mathrm{St},2}\}, \end{aligned}$$

where $k_1 + k_2 = k'$, until there are exactly p_{St} error positions in $\mathcal{X}_1 \cup \mathcal{X}_2$ and in $\mathcal{Y}_1 \cup \mathcal{Y}_2$ and no error positions in $\mathcal{Z}_1$ and $\mathcal{Z}_2$. The probability that an iteration fulfills this condition is approximately given by

$$
\begin{aligned}
P_{\mathrm{St}} &= \sum_{\substack{\boldsymbol{a}\in\mathbb{N}_0^2\\ a_1\le w_{\mathrm{y}}\\ a_2\le w'\\ a_1+a_2=p_{\mathrm{St}}}} \sum_{\substack{\boldsymbol{b}\in\mathbb{N}_0^2\\ b_1\le w_{\mathrm{y}}-a_1\\ b_2\le w'-a_2\\ b_1+b_2=p_{\mathrm{St}}}} \frac{\binom{\lfloor k_1/2\rfloor}{a_1}\binom{\lceil k_1/2\rceil}{b_1}\binom{n-k_1-\nu_{\mathrm{St},1}}{w_{\mathrm{y}}-a_1-b_1}}{\binom{n}{w_{\mathrm{y}}}} \frac{\binom{\lfloor k_2/2\rfloor}{a_2}\binom{\lceil k_2/2\rceil}{b_2}\binom{n'-k_2-\nu_{\mathrm{St},2}}{w'-a_2-b_2}}{\binom{n'}{w'}} \\
&= \sum_{\substack{\boldsymbol{a}\in\mathbb{N}_0^2\\ a_1\le w_{\mathrm{y}}\\ a_2\le w'\\ a_1+a_2=p_{\mathrm{St}}}} \sum_{\substack{\boldsymbol{b}\in\mathbb{N}_0^2\\ b_1\le w_{\mathrm{y}}-a_1\\ b_2\le w'-a_2\\ b_1+b_2=p_{\mathrm{St}}}} \frac{\binom{\lfloor k_1/2\rfloor}{a_1}\binom{\lceil k_1/2\rceil}{b_1}\binom{n-k_1-\nu_{\mathrm{St},1}}{w_{\mathrm{y}}-a_1-b_1}}{\binom{n}{w_{\mathrm{y}}}} \frac{\binom{\lfloor (k'-k_1)/2\rfloor}{a_2}\binom{\lceil (k'-k_1)/2\rceil}{b_2}\binom{n'-k'+k_1-\nu_{\mathrm{St},2}}{w'-a_2-b_2}}{\binom{n'}{w'}}.
\end{aligned}
$$

As we modified Stern's original algorithm only by adapting the sampling of the information set, the complexity per iteration [197, Thm. 2.8] remains equal to

$$
\begin{aligned}
W_{\mathrm{St,Iter}} &:= (n+n')^3 + (\nu_{\mathrm{St},1}+\nu_{\mathrm{St},2})\left(\sum_{i=1}^{p_{\mathrm{St}}}\binom{M_1}{i} + \sum_{i=1}^{p_{\mathrm{St}}}\binom{M_2}{i} - k' + \binom{M_2}{p_{\mathrm{St}}}\right) \\
&\quad + 2^{1-\nu_{\mathrm{St},1}-\nu_{\mathrm{St},2}}\binom{M_1}{p_{\mathrm{St}}}\binom{M_2}{p_{\mathrm{St}}}(w_{\mathrm{y}}+w'-2p_{\mathrm{St}}+1)(2p_{\mathrm{St}}+1),
\end{aligned}
$$

where $M_1 = \lfloor k_1/2\rfloor + \lfloor (k'-k_1)/2\rfloor$ and $M_2 = \lceil k_1/2\rceil + \lceil (k'-k_1)/2\rceil$, which gives an overall complexity of

$$
W_{\mathrm{ModSt}} := \frac{W_{\mathrm{St,Iter}}}{P_{\mathrm{St}}}
$$

operations in $\mathbb{F}_2$. ■

In [190], the proposed modifications to Prange's, Lee and Brickell's, and Stern's algorithms are generalized to cases in which the Hamming weight of more than two error blocks is known.

4.2.4 A Side-Channel Attack on HQC

In this section, we derive Algorithm 18, which is an attack that exploits side-channel information to retrieve the private key $(\boldsymbol{x}, \boldsymbol{y})$. In this attack, the vector $\boldsymbol{y}$ is decomposed into $\boldsymbol{y} = \left[\boldsymbol{y}^{(1)}, \boldsymbol{y}^{(2)}\right] \in \mathbb{F}_2^n$, where $\boldsymbol{y}^{(1)} \in \mathbb{F}_2^{n_1 n_2}$ and $\boldsymbol{y}^{(2)} \in \mathbb{F}_2^{n-n_1 n_2}$. We first determine the vector $\boldsymbol{y}^{(1)}$ with high probability given a decoding oracle $\mathcal{O}^{\mathrm{Dec}}_{(\boldsymbol{x},\boldsymbol{y})}$. This oracle is defined in Definition 4.1, and a technique of how to build $\mathcal{O}^{\mathrm{Dec}}_{(\boldsymbol{x},\boldsymbol{y})}$ through a power side-channel is shown in [187] and [189, Sec. 4]. In case $\boldsymbol{y}^{(1)}$ was successfully recovered, we retrieve $\boldsymbol{y}^{(2)}$ using linear algebra. If the support of $\boldsymbol{y}^{(1)}$ was only partly recovered, we run a modified ISD algorithm.

Algorithm 18: Power Side-Channel Attack on HQC

```
Input  : Non-negative integers $n_1$, $n_2$, $n$, $w_{\mathrm{y}}$
         Public key $(\boldsymbol{h}, \boldsymbol{s}) \in \mathbb{F}_2^n \times \mathbb{F}_2^n$
         Oracle access $\mathcal{O}^{\mathrm{Dec}}_{(\boldsymbol{x},\boldsymbol{y})}$
Output: Private key $(\boldsymbol{x}, \boldsymbol{y}) \in \mathbb{F}_2^n \times \mathbb{F}_2^n$ or failure $\perp$
$\hat{\mathcal{S}}_{\boldsymbol{y}^{(1)}} \leftarrow \{\}$
for $i \in [1:n_1]$ do
    $\tilde{\mathcal{S}}_{\boldsymbol{y}_i} \leftarrow$ FindSuperSupport$\left(i, (n_1, n_2, n), \mathcal{O}^{\mathrm{Dec}}_{(\boldsymbol{x},\boldsymbol{y})}\right)$ (see Alg. 19)
    if $|\tilde{\mathcal{S}}_{\boldsymbol{y}_i}| > 1$ then
        $\hat{\mathcal{S}}_{\boldsymbol{y}_i} \leftarrow$ FindSupport$\left(i, \tilde{\mathcal{S}}_{\boldsymbol{y}_i}, (n_1, n_2, n), \mathcal{O}^{\mathrm{Dec}}_{(\boldsymbol{x},\boldsymbol{y})}\right)$ (see Alg. 20)
    $\hat{\mathcal{S}}_{\boldsymbol{y}^{(1)}} \leftarrow \hat{\mathcal{S}}_{\boldsymbol{y}^{(1)}} \cup \left(\{(i-1)n_2\} + \hat{\mathcal{S}}_{\boldsymbol{y}_i}\right)$
if $|\hat{\mathcal{S}}_{\boldsymbol{y}^{(1)}}| < w_{\mathrm{y}}$ then
    $\hat{\mathcal{S}}_{\boldsymbol{y}} \leftarrow \hat{\mathcal{S}}_{\boldsymbol{y}^{(1)}} \cup$ FindRemainingSupport$\left(\hat{\mathcal{S}}_{\boldsymbol{y}^{(1)}}, (n_1, n_2, n, w_{\mathrm{y}}), (\boldsymbol{h}, \boldsymbol{s})\right)$ (see Alg. 21)
else
    $\hat{\mathcal{S}}_{\boldsymbol{y}} \leftarrow \hat{\mathcal{S}}_{\boldsymbol{y}^{(1)}}$
$\hat{\boldsymbol{y}} \leftarrow$ Vector in $\mathbb{F}_2^n$ with support $\hat{\mathcal{S}}_{\boldsymbol{y}}$
$\hat{\boldsymbol{x}} \leftarrow \boldsymbol{s} - \hat{\boldsymbol{y}}\boldsymbol{h}$
if $\mathrm{wt_H}(\hat{\boldsymbol{x}}) = w_{\mathrm{y}} \wedge \mathrm{wt_H}(\hat{\boldsymbol{y}}) = w_{\mathrm{y}} \wedge \hat{\boldsymbol{x}} + \hat{\boldsymbol{y}}\boldsymbol{h} = \boldsymbol{s}$ then
    return $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}})$
else
    return $\perp$
```

The main statement of this section is Theorem 4.13, which proves the success probability of Algorithm 18. Since the proof of this theorem is rather long, we provide some intermediate statements before.

Distribution of the Support of $\boldsymbol{y}$

The success of the attack depends on the positions of the non-zero entries in the private vector $\boldsymbol{y}$. To analyze the support of $\boldsymbol{y}$, we decompose the vector into

$$\boldsymbol{y} = \left[\boldsymbol{y}_1^{(1)}, \ldots, \boldsymbol{y}_{n_1}^{(1)}, \boldsymbol{y}^{(2)}\right] \in \mathbb{F}_2^n,$$

Table 4.5: Estimated probabilities that $\boldsymbol{y}_1^{(1)}, \ldots, \boldsymbol{y}_{n_1}^{(1)}$ have a Hamming weight of at most 1, 2, or 3 for the parameter sets given in Table 4.4.

$\max\limits_{j' \in [1:n_1]} \left\{ \mathrm{wt_H}\left(\boldsymbol{y}_{j'}^{(1)}\right)\right\}$	HQC-128	HQC-192	HQC-256
1	5.59%	0.11%	0.00%
2	93.20%	77.98%	58.99%
3	99.86%	99.25%	97.99%

where $\boldsymbol{y}_1^{(1)}, \ldots, \boldsymbol{y}_{n_1}^{(1)} \in \mathbb{F}_2^{n_2}$, and $\boldsymbol{y}^{(2)} \in \mathbb{F}_2^{n-n_1 n_2}$. From the parameters shown in Table 4.4, we conclude that $\boldsymbol{y}$ is a sparse vector, and the vectors $\boldsymbol{y}_1^{(1)}, \ldots, \boldsymbol{y}_{n_1}^{(1)}, \boldsymbol{y}^{(2)}$ have a Hamming weight close to zero with high probability. We performed Monte-Carlo simulations by generating 10^7 private keys to estimate the weight distribution of $\boldsymbol{y}_1^{(1)}, \ldots, \boldsymbol{y}_{n_1}^{(1)}$, where we observed that less than 3% of the private keys have a vector $\boldsymbol{y}_1^{(1)}, \ldots, \boldsymbol{y}_{n_1}^{(1)}$ of Hamming weight of more than 3, see Table 4.5, and the probability that $\mathrm{wt_H}\left(\boldsymbol{y}^{(2)}\right) > 0$ is approximately 29.23%, 0.69%, and 1.52% for HQC-128, HQC-192, and HQC-256, respectively.

Retrieving a Super-Support of $\boldsymbol{y}^{(1)}$ Using a Decoding Oracle

To retrieve $\boldsymbol{y}^{(1)} = \left[\boldsymbol{y}_1^{(1)}, \ldots, \boldsymbol{y}_{n_1}^{(1)}\right]$, we conduct a chosen-ciphertext attack that is based on the decoding oracle $\mathcal{O}^{\mathrm{Dec}}_{(\boldsymbol{x},\boldsymbol{y})}$.

Definition 4.1 (HQC Decoding Oracle $\mathcal{O}^{\mathrm{Dec}}_{(\boldsymbol{x},\boldsymbol{y})}$[187] [189, Sec. 4])**.** *Let $(\boldsymbol{x}, \boldsymbol{y})$ and $(\boldsymbol{h}, \boldsymbol{s})$ be a private and public key pair generated by* $\mathsf{KeyGen}_{\mathrm{HQC}}$*, and let $(\boldsymbol{u}, \boldsymbol{v})$ be a ciphertext computed by* $\mathsf{Encrypt}_{\mathrm{HQC}}$ *using $(\boldsymbol{h}, \boldsymbol{s})$. If $(\boldsymbol{u}, \boldsymbol{v})$ is queried to $\mathcal{O}^{\mathrm{Dec}}_{(\boldsymbol{x},\boldsymbol{y})}$, then $\mathcal{O}^{\mathrm{Dec}}_{(\boldsymbol{x},\boldsymbol{y})}$ outputs* 1 *if the BCH decoder in* $\mathsf{Decrypt}_{\mathrm{HQC}}$ *corrects an error for $(\boldsymbol{u}, \boldsymbol{v})$, $(\boldsymbol{x}, \boldsymbol{y})$, and $(\boldsymbol{h}, \boldsymbol{s})$, and* 0 *otherwise.*

Our proposed attack first determines a Hamming super-support of $\boldsymbol{y}_i^{(1)}$, for $i \in [1:n_1]$, using Algorithm 19.

Lemma 4.10. *Let i be an element of $[1:n_1]$ and n_1, n_2, n, k, w_{y}, w_{r}, w_{e}, and δ be parameters chosen according to Table 4.4. Let $(\boldsymbol{x}, \boldsymbol{y})$ and $(\boldsymbol{h}, \boldsymbol{s})$ be a private and public key pair generated by* $\mathsf{KeyGen}_{\mathrm{HQC}}$ *for the chosen parameters. Furthermore, let $\mathcal{O}^{\mathrm{Dec}}_{(\boldsymbol{x},\boldsymbol{y})}$ be defined as in Definition 4.1 and $\max_{j' \in [1:n_1]} \left\{ \mathrm{wt_H}\left(\boldsymbol{y}_{j'}^{(1)}\right)\right\} \leq 2$. Then, given i, n_1, n_2, n, and oracle access $\mathcal{O}^{\mathrm{Dec}}_{(\boldsymbol{x},\boldsymbol{y})}$, Algorithm 19 requires $O(n)$ operations in $\mathbb{F}_2$ and six queries to $\mathcal{O}^{\mathrm{Dec}}_{(\boldsymbol{x},\boldsymbol{y})}$ in order to output a super-support of $\boldsymbol{y}_i^{(1)}$ of size at most $\frac{n_2+1}{2}$.*

Algorithm 19: FindSuperSupport

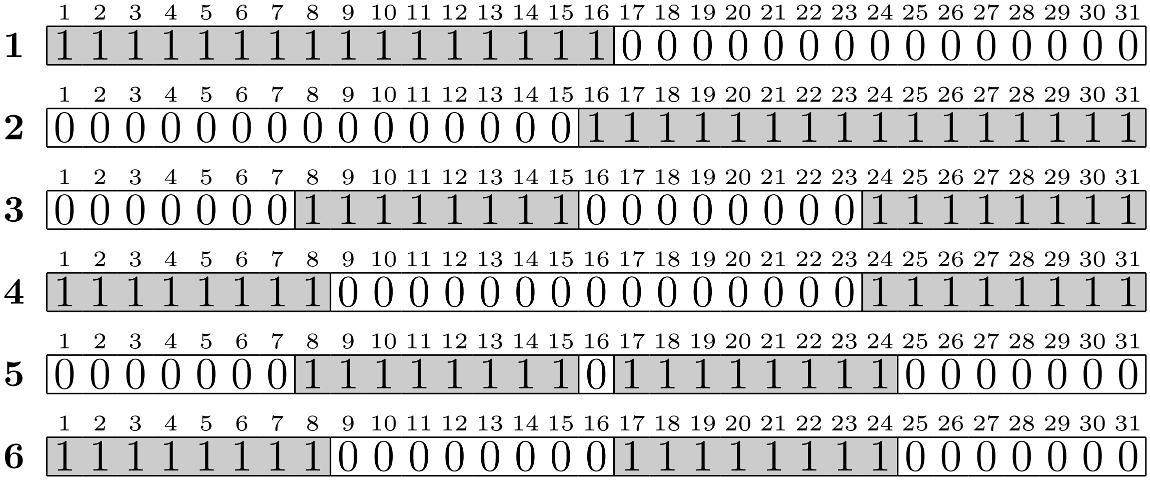
```
Input  : Non-negative integer $i$
         Non-negative integers $n_1$, $n_2$, $n$
         Oracle access $\mathcal{O}^{\mathrm{Dec}}_{(\boldsymbol{x},\boldsymbol{y})}$
Output: Super-support $\tilde{\mathcal{S}}_{\boldsymbol{y}_i}$
$\boldsymbol{u} \leftarrow [1, 0, \dots, 0] \in \mathbb{F}_2^n$
$\boldsymbol{v} = [\boldsymbol{v}_1, \dots, \boldsymbol{v}_{n_1}, \boldsymbol{v}_{n_1+1}] \leftarrow \mathbf{0}_n \in \mathbb{F}_2^n$, where $\boldsymbol{v}_1, \dots, \boldsymbol{v}_{n_1} \in \mathbb{F}_2^{n_2}$ and
  $\boldsymbol{v}_{n_1+1} \in \mathbb{F}_2^{n-n_1 n_2}$
$\boldsymbol{p} \leftarrow \mathbf{0}_6 \in \mathbb{F}_2^6$
for $\ell \in [1:6]$ do
    $\boldsymbol{v}_i \leftarrow$ Vector in $\mathbb{F}_2^{n_2}$ with support according to the pattern $\ell$ given in
      Table 4.6
    $p_\ell \leftarrow$ Output of $\mathcal{O}^{\mathrm{Dec}}_{(\boldsymbol{x},\boldsymbol{y})}$ on the query $(\boldsymbol{u}, \boldsymbol{v})$
$\tilde{\mathcal{S}}_{\boldsymbol{y}_i} \leftarrow$ Super-support according to the row $\boldsymbol{p}$ in Table 4.7
return $\tilde{\mathcal{S}}_{\boldsymbol{y}_i}$
```

Table 4.6: Patterns of $\boldsymbol{v}_i$ used in Algorithm 19 to determine a super-support of $\boldsymbol{y}_i^{(1)}$, for $i \in [1:n_1]$. An illustration of the patterns is shown Figure 4.1 for $n_2 = 31$.

Pattern	$\mathrm{supp}_{\mathrm{H}}(\boldsymbol{v}_i)$
1	$\left[1 : \left\lceil \frac{n_2}{2} \right\rceil\right]$
2	$\left[\left\lceil \frac{n_2}{2} \right\rceil : n_2\right]$
3	$\left[\left\lceil \frac{n_2}{4} \right\rceil : \left\lceil \frac{n_2}{2} \right\rceil - 1\right] \cup \left[\left\lceil \frac{3n_2}{4} \right\rceil : n_2\right]$
4	$\left[1 : \left\lceil \frac{n_2}{4} \right\rceil\right] \cup \left[\left\lceil \frac{3n_2}{4} \right\rceil : n_2\right]$
5	$\left[\left\lceil \frac{n_2}{4} \right\rceil : \left\lceil \frac{n_2}{2} \right\rceil - 1\right] \cup \left[\left\lceil \frac{n_2}{2} \right\rceil + 1 : \left\lceil \frac{3n_2}{4} \right\rceil\right]$
6	$\left[1 : \left\lceil \frac{n_2}{4} \right\rceil\right] \cup \left[\left\lceil \frac{n_2}{2} \right\rceil + 1 : \left\lceil \frac{3n_2}{4} \right\rceil\right]$

Table 4.7: Mapping of $\boldsymbol{p} \in \mathbb{F}_2^6$ to a super-support of $\boldsymbol{y}_i^{(1)}$, where $\boldsymbol{p}$ results from the outputs of the oracle $\mathcal{O}^{\mathrm{Dec}}_{(\boldsymbol{x},\boldsymbol{y})}$ queried on the six patterns of $\boldsymbol{v}_i$ (see Table 4.6 and Figure 4.1). An asterisk indicates that the entry in $\boldsymbol{p}$ can be either 0 and 1.

p_1	p_2	p_3	p_4	p_5	p_6	Super-support $\tilde{\mathcal{S}}_{\boldsymbol{y}_i}$
1	1	1	1	1	1	$\{\ \}$
0	0	$*$	$*$	$*$	$*$	$\left\{\frac{n_2+1}{2}\right\}$
0	1	$*$	$*$	$*$	$*$	$\left[1 : \frac{n_2+1}{2}\right]$
1	0	$*$	$*$	$*$	$*$	$\left[\frac{n_2+1}{2} : n_2\right]$
1	1	0	1	1	1	$\left[\frac{n_2+1}{4} + 1 : \frac{n_2+1}{2} - 1\right] \cup \left[\frac{3(n_2+1)}{4} + 1 : n_2\right]$
1	1	1	0	1	1	$\left[1 : \frac{n_2+1}{4} - 1\right] \cup \left[\frac{3(n_2+1)}{4} + 1 : n_2\right]$
1	1	1	1	0	1	$\left[\frac{n_2+1}{4} + 1 : \frac{n_2+1}{2} - 1\right] \cup \left[\frac{n_2+1}{2} + 1 : \frac{3(n_2+1)}{4} - 1\right]$
1	1	1	1	1	0	$\left[1 : \frac{n_2+1}{4} - 1\right] \cup \left[\frac{n_2+1}{2} + 1 : \frac{3(n_2+1)}{4} - 1\right]$
1	1	0	0	1	1	$\left[\frac{n_2+1}{4}\right] \cup \left[\frac{3(n_2+1)}{4} + 1 : n_2\right]$
1	1	0	1	0	1	$\left[\frac{n_2+1}{4} + 1 : \frac{n_2+1}{2} - 1\right] \cup \left[\frac{3(n_2+1)}{4}\right]$
1	1	1	0	1	0	$\left[1 : \frac{n_2+1}{4} - 1\right] \cup \left[\frac{3(n_2+1)}{4}\right]$
1	1	1	1	0	0	$\left[\frac{n_2+1}{4}\right] \cup \left[\frac{n_2+1}{2} + 1 : \frac{3(n_2+1)}{4} - 1\right]$
1	1	0	0	0	0	$\left[\frac{n_2+1}{4}\right] \cup \left[\frac{3(n_2+1)}{4}\right]$



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
3	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
4	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
5	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0
6	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0

Figure 4.1: Illustration of the patterns of $\boldsymbol{v}_i \in \mathbb{F}_2^{n_2}$ that are used to obtain a super-support of $\boldsymbol{y}_i^{(1)}$ for $n_2 = 31$. The gray parts refer to non-zero entries, and the white parts indicate zero entries.

Proof. In Algorithm 19, the vector $\boldsymbol{u} \in \mathbb{F}_2^n$ is chosen to $[1, 0, \dots, 0] \in \mathbb{F}_2^n$ such that the input of the decoder of $\mathcal{C}_{\mathrm{HQC}}$ is equal to the first $n_1 n_2$ entries of

$$\boldsymbol{v}' = [\boldsymbol{v}'_1, \dots, \boldsymbol{v}'_{n_1}, \boldsymbol{v}'_{n_1+1}] = \left[\boldsymbol{v}_1 - \boldsymbol{y}_1^{(1)}, \dots, \boldsymbol{v}_{n_1} - \boldsymbol{y}_{n_1}^{(1)}, \boldsymbol{v}_{n_1+1} - \boldsymbol{y}^{(2)}\right] = \boldsymbol{v} - \boldsymbol{y} \in \mathbb{F}_2^n,$$

where $\boldsymbol{v}'_{j'}, \boldsymbol{v}_{j'}, \boldsymbol{y}_{j'}^{(1)} \in \mathbb{F}_2^{n_2}$, for $j' \in [1:n_1]$, see Algorithm 14. Furthermore, the vectors $\boldsymbol{v}_j = \boldsymbol{0}_{n_2} \in \mathbb{F}_2^{n_2}$, for $j \in [1:n_1] \setminus i$. Then, the oracle $\mathcal{O}_{(\boldsymbol{x},\boldsymbol{y})}^{\mathrm{Dec}}$ is queried on the six different patters of $\boldsymbol{v}_i$ given in Table 4.6 (these patterns are illustrated in Figure 4.1). The outputs of the oracle are then mapped to a set according to Table 4.7.

To see that the obtained set is a super-support of $\boldsymbol{y}_i^{(1)}$, recall that $\mathcal{C}_{\mathrm{HQC}}$ is chosen as a product code of a length-n_2 repetition code $\mathcal{C}_2$ and a length-n_1 BCH code $\mathcal{C}_1$. The algorithm used for decoding $\mathcal{C}_{\mathrm{HQC}}$ first decodes the vectors $\boldsymbol{v}'_1, \dots, \boldsymbol{v}'_{n_1} \in \mathbb{F}_2^{n_2}$ separately in the repetition codes to $\tilde{v}_1, \dots, \tilde{v}_{n_1} \in \mathbb{F}_2$. By assumption, we have

$$\max_{j \in [1:n_1] \setminus \{i\}} \left\{ \mathrm{wt_H}\left(\boldsymbol{v}'_j\right) \right\} = \max_{j \in [1:n_1] \setminus \{i\}} \left\{ \mathrm{wt_H}\left(\boldsymbol{v}_j - \boldsymbol{y}_j^{(1)}\right) \right\} = \max_{j \in [1:n_1] \setminus \{i\}} \left\{ \mathrm{wt_H}\left(\boldsymbol{y}_j^{(1)}\right) \right\} \leq 2,$$

and therefore, the output of the repetition decoder $\tilde{v}_j = 0$, for $j \in [1:n_1] \setminus i$. The outputs $\tilde{v}_1, \dots, \tilde{v}_{n_1}$ are then fed into the decoder of the BCH code $\mathcal{C}_1$. Since the vector $\boldsymbol{0} \in \mathbb{F}_2^{n_1}$ is a codeword of $\mathcal{C}_1$, and vectors of Hamming weight one[3] are not in $\mathcal{C}_1$, we conclude the following: The oracle $\mathcal{O}_{(\boldsymbol{x},\boldsymbol{y})}^{\mathrm{Dec}}$ outputs 0 (meaning no error is corrected in the BCH code) if and only if $\tilde{v}_i = 0$, and $\tilde{v}_i = 0$ holds if and only if $\mathrm{wt_H}(\boldsymbol{v}'_i) = \mathrm{wt_H}(\boldsymbol{v}_i - \boldsymbol{y}_i) < \lceil \frac{n_2}{2} \rceil$. This in turn implies that $\left|\mathrm{supp_H}\left(\boldsymbol{y}_i^{(1)}\right) \cap \mathrm{supp_H}(\boldsymbol{v}_i)\right| > \mathrm{wt_H}\left(\boldsymbol{y}_i^{(1)}\right) / 2$, since $\mathrm{wt_H}(\boldsymbol{v}_i) = \lceil \frac{n_2}{2} \rceil$. Furthermore, by the same arguments as before, the oracle $\mathcal{O}_{(\boldsymbol{x},\boldsymbol{y})}^{\mathrm{Dec}}$ outputs 1 if and only if $\mathrm{wt_H}(\boldsymbol{v}'_i) \geq \lceil \frac{n_2}{2} \rceil$, which means $\left|\mathrm{supp_H}\left(\boldsymbol{y}_i^{(1)}\right) \cap \mathrm{supp_H}(\boldsymbol{v}_i)\right| \leq \mathrm{wt_H}\left(\boldsymbol{y}_i^{(1)}\right) / 2$. This observation directly implies that the obtained set is a super-support of $\boldsymbol{y}_i^{(1)}$.

Since Line 5 and 7 can be performed in $O(n)$ operations and $\ell \in [1:6]$, the algorithm requires $O(n)$ operations in $\mathbb{F}_2$ and 6 queries to the oracle. ■

Retrieving the Support of $\boldsymbol{y}^{(1)}$ Using a Decoding Oracle

From Lemma 4.10 follows that for $\max_{j' \in [1:n_1]} \left\{ \mathrm{wt_H}\left(\boldsymbol{y}_{j'}^{(1)}\right) \right\} \leq 2$, Algorithm 19 returns a Hamming super-support of $\boldsymbol{y}_i^{(1)}$. The next lemma shows that Algorithm 20 retrieves the support of $\boldsymbol{y}_i^{(1)}$ given the super-support from Algorithm 19.

[3] This follows from the fact that the used BCH code has a minimum distance larger than 1.

Algorithm 20: FindSupport

```
Input  : Non-negative integer $i$
         Super-support $\tilde{\mathcal{S}}_{\boldsymbol{y}_i}$
         Non-negative integers $n_1$, $n_2$, $n$
         Oracle access $\mathcal{O}^{\mathrm{Dec}}_{(\boldsymbol{x},\boldsymbol{y})}$
Output: Support $\hat{\mathcal{S}}_{\boldsymbol{y}_i}$
$\boldsymbol{u} \leftarrow [1, 0, \dots, 0] \in \mathbb{F}_2^n$
$\boldsymbol{v} = [\boldsymbol{v}_1, \dots, \boldsymbol{v}_{n_1}, \boldsymbol{v}_{n_1+1}] \leftarrow \boldsymbol{0}_n \in \mathbb{F}_2^n$, where $\boldsymbol{v}_1, \dots, \boldsymbol{v}_{n_1} \in \mathbb{F}_2^{n_2}$ and
  $\boldsymbol{v}_{n_1+1} \in \mathbb{F}_2^{n-n_1n_2}$
$\bar{\boldsymbol{v}}_i \leftarrow$ Vector in $\mathbb{F}_2^{n_2}$ with weight $\lceil \frac{n_2}{2} \rceil - 2$ and support contained in $[1\!:\!n_2] \setminus \tilde{\mathcal{S}}_{\boldsymbol{y}_i}$
for $\{j_1, j_2\} \in \{\mathcal{J}' \subseteq \tilde{\mathcal{S}}_{\boldsymbol{y}_i} : |\mathcal{J}'| = 2\}$ do
    $\boldsymbol{v}_i \leftarrow \bar{\boldsymbol{v}}_i$
    $j_1$-th and $j_2$-th element of $\boldsymbol{v}_i \leftarrow 1$
    $p \leftarrow$ Output of $\mathcal{O}^{\mathrm{Dec}}_{(\boldsymbol{x},\boldsymbol{y})}$ on the query $(\boldsymbol{u}, \boldsymbol{v})$
    if $p = 0$ then
        $j_1$-th element of $\boldsymbol{v}_i \leftarrow 0$
        $p' \leftarrow$ Output of $\mathcal{O}^{\mathrm{Dec}}_{(\boldsymbol{x},\boldsymbol{y})}$ on the query $(\boldsymbol{u}, \boldsymbol{v})$
        if $p' = 1$ then
            $\hat{\mathcal{S}}_{\boldsymbol{y}_i} \leftarrow \{j_1\}$
        else
            $j_1$-th element of $\boldsymbol{v}_i \leftarrow 1$
            $j_2$-th element of $\boldsymbol{v}_i \leftarrow 0$
            $p'' \leftarrow$ Output of $\mathcal{O}^{\mathrm{Dec}}_{(\boldsymbol{x},\boldsymbol{y})}$ on the query $(\boldsymbol{u}, \boldsymbol{v})$
            if $p'' = 1$ then
                $\hat{\mathcal{S}}_{\boldsymbol{y}_i} \leftarrow \{j_2\}$
            else
                $\hat{\mathcal{S}}_{\boldsymbol{y}_i} \leftarrow \{j_1, j_2\}$
        return $\hat{\mathcal{S}}_{\boldsymbol{y}_i}$
return $\hat{\mathcal{S}}_{\boldsymbol{y}_i} \leftarrow \{\}$
```

Lemma 4.11. *Let i be an element of $[1 : n_1]$ and n_1, n_2, n, k, w_{y}, w_{r}, w_{e}, and δ be parameters chosen according to Table 4.4. Let $(\boldsymbol{x}, \boldsymbol{y})$ and $(\boldsymbol{h}, \boldsymbol{s})$ be a private and public key pair generated by* $\mathsf{KeyGen}_{\mathrm{HQC}}$ *for the chosen parameters. Furthermore, let $\mathcal{O}^{\mathrm{Dec}}_{(\boldsymbol{x},\boldsymbol{y})}$ be defined as in Definition 4.1, $\max_{j' \in [1:n_1]} \left\{ \mathrm{wt}_{\mathrm{H}} \left(\boldsymbol{y}^{(1)}_{j'} \right) \right\} \leq 2$ and $\tilde{\mathcal{S}}_{\boldsymbol{y}_i}$ be a Hamming super-support of $\boldsymbol{y}^{(1)}_i$ obtained by Algorithm 19. Then, given i, $\tilde{\mathcal{S}}_{\boldsymbol{y}_i}$, n_1, n_2, n, and oracle access $\mathcal{O}^{\mathrm{Dec}}_{(\boldsymbol{x},\boldsymbol{y})}$, Algorithm 20 requires $O\left(nn_2^2\right)$ operations in $\mathbb{F}_2$ and at* *most $2n_2^2+2$ queries to $\mathcal{O}^{\mathrm{Dec}}_{(\boldsymbol{x},\boldsymbol{y})}$ in order to output the support of $\boldsymbol{y}_i^{(1)}$.*

Proof. As in Algorithm 19, the vector $\boldsymbol{u} \in \mathbb{F}_2^n$ is chosen to be $[1,0,\ldots,0] \in \mathbb{F}_2^n$, and the vector $\boldsymbol{v}_j = \boldsymbol{0}_{n_2} \in \mathbb{F}_2^{n_2}$, for $j \in [1\!:\!n_1] \setminus i$. Then, the oracle $\mathcal{O}^{\mathrm{Dec}}_{(\boldsymbol{x},\boldsymbol{y})}$ is queried on different patterns of $\boldsymbol{v}_i$. The patterns have weight $\lceil\frac{n_2}{2}\rceil$ and are designed such that $\lceil\frac{n_2}{2}\rceil - 2$ entries in $\boldsymbol{v}_i$ whose indices are not in $\tilde{\mathcal{S}}_{\boldsymbol{y}_i}$ are fixed to 1 and $|\mathrm{supp}_{\mathrm{H}}(\boldsymbol{v}_i) \cap \tilde{\mathcal{S}}_{\boldsymbol{y}_i}| = 2$ (see Figure 4.2 for $n_2 = 31$ and $\tilde{\mathcal{S}}_{\boldsymbol{y}_i} = [1:16]$). If $\mathcal{O}^{\mathrm{Dec}}_{(\boldsymbol{x},\boldsymbol{y})}$ outputs 1 (meaning an error is corrected in the BCH code) for a query, it follows that $\tilde{v}_i = 1$, and $\tilde{v}_i = 1$ holds if and only if $\mathrm{wt}_{\mathrm{H}}(\boldsymbol{v}_i') = \mathrm{wt}_{\mathrm{H}}(\boldsymbol{v}_i - \boldsymbol{y}_i) \geq \lceil\frac{n_2}{2}\rceil$. This implies that $\left|\mathrm{supp}_{\mathrm{H}}\left(\boldsymbol{y}_i^{(1)}\right) \cap \mathrm{supp}_{\mathrm{H}}(\boldsymbol{v}_i)\right|$ is at most $\mathrm{wt}_{\mathrm{H}}\left(\boldsymbol{y}_i^{(1)}\right)/2$, since $\mathrm{wt}_{\mathrm{H}}(\boldsymbol{v}_i) = \lceil\frac{n_2}{2}\rceil$. Therefore, it holds that $\mathrm{supp}_{\mathrm{H}}(\boldsymbol{v}_i) \cap \tilde{\mathcal{S}}_{\boldsymbol{y}_i} \not\supseteq \mathrm{supp}_{\mathrm{H}}\left(\boldsymbol{y}_i^{(1)}\right)$. Whereas if $\mathcal{O}^{\mathrm{Dec}}_{(\boldsymbol{x},\boldsymbol{y})}$ outputs 0, it holds that $\left|\mathrm{supp}_{\mathrm{H}}\left(\boldsymbol{y}_i^{(1)}\right) \cap \mathrm{supp}_{\mathrm{H}}(\boldsymbol{v}_i)\right|$ is greater than $\mathrm{wt}_{\mathrm{H}}\left(\boldsymbol{y}_i^{(1)}\right)/2$, and thus, it can be deduced that $\mathrm{supp}_{\mathrm{H}}(\boldsymbol{v}_i) \cap \tilde{\mathcal{S}}_{\boldsymbol{y}_i} \supseteq \mathrm{supp}_{\mathrm{H}}\left(\boldsymbol{y}_i^{(1)}\right)$.

Let j_1 and j_2 be two distinct indices in $\mathrm{supp}_{\mathrm{H}}(\boldsymbol{v}_i)$ such that $|\{j_1, j_2\} \cap \tilde{\mathcal{S}}_{\boldsymbol{y}_i}| = 2$ and $\mathcal{O}^{\mathrm{Dec}}_{(\boldsymbol{x},\boldsymbol{y})}$ outputs 0. To determine $\mathrm{supp}_{\mathrm{H}}\left(\boldsymbol{y}_i^{(1)}\right)$, the algorithm chooses the j_1-th element of $\boldsymbol{v}_i$ to 0 and queries the oracle on $(\boldsymbol{u}, \boldsymbol{v})$. If $\mathcal{O}^{\mathrm{Dec}}_{(\boldsymbol{x},\boldsymbol{y})}$ outputs 1 for this query, it implies that the BCH decoder corrected an error, and therefore, $\{j_2\} \not\subset \mathrm{supp}_{\mathrm{H}}\left(\boldsymbol{y}_i^{(1)}\right)$ and $\{j_1\} = \mathrm{supp}_{\mathrm{H}}\left(\boldsymbol{y}_i^{(1)}\right)$. If $\mathcal{O}^{\mathrm{Dec}}_{(\boldsymbol{x},\boldsymbol{y})}$ outputs 0, it follows that $\{j_2\} \in \mathrm{supp}_{\mathrm{H}}\left(\boldsymbol{y}_i^{(1)}\right)$. Then, the algorithm chooses the j_1-th element of $\boldsymbol{v}_i$ to be 1, the j_2-th element of $\boldsymbol{v}_i$ to be 0 and queries the oracle on $(\boldsymbol{u}, \boldsymbol{v})$. If $\mathcal{O}^{\mathrm{Dec}}_{(\boldsymbol{x},\boldsymbol{y})}$ outputs 1, it must hold that $\{j_1\} \not\subset \mathrm{supp}_{\mathrm{H}}\left(\boldsymbol{y}_i^{(1)}\right)$ and $\{j_2\} = \mathrm{supp}_{\mathrm{H}}\left(\boldsymbol{y}_i^{(1)}\right)$. If $\mathcal{O}^{\mathrm{Dec}}_{(\boldsymbol{x},\boldsymbol{y})}$ outputs 0, it means $\{j_1, j_2\} = \mathrm{supp}_{\mathrm{H}}\left(\boldsymbol{y}_i^{(1)}\right)$.

Since $|\{\mathcal{J}' \subseteq \tilde{\mathcal{S}}_{\boldsymbol{y}_i} : |\mathcal{J}'| = 2\}| \leq \binom{n_2}{2} \leq 2n_2^2$, the algorithm requires $O\left(nn_2^2\right)$ operations in $\mathbb{F}_2$ and at most $2n_2^2 + 2$ queries to the oracle. ■

Retrieving $\boldsymbol{y}^{(2)}$ Using Linear Algebra

Since only the first $n_1 n_2$ positions of $\boldsymbol{v}' \in \mathbb{F}_2^n$ are decoded in the code $\mathcal{C}_{\mathrm{HQC}}$, the last $n - n_1 n_2$ positions of $\boldsymbol{y}$ cannot be determined with the previously described strategy. Therefore, we present Algorithm 21, which retrieves $\mathrm{supp}_{\mathrm{H}}\left(\boldsymbol{y}^{(2)}\right)$ under the assumption that $\mathrm{supp}_{\mathrm{H}}\left(\boldsymbol{y}^{(1)}\right)$ is known. In this algorithm, $\boldsymbol{h}_\ell \in \mathbb{F}_2^n$ denotes the ℓ-th column of the matrix $\boldsymbol{H} \in \mathbb{F}_2^{n \times 2n}$.

Lemma 4.12. *Let n_1, n_2, n, k, w_{y}, w_{r}, w_{e}, and δ be parameters chosen according to Table 4.4. Furthermore, let $(\boldsymbol{x}, \boldsymbol{y})$ and $(\boldsymbol{h}, \boldsymbol{s})$ be a private and a public key pair generated by* $\mathsf{KeyGen}_{\mathrm{HQC}}$ *for the chosen parameters, and let $\hat{\mathcal{S}}_{\boldsymbol{y}^{(1)}}$ be the Hamming support*

Algorithm 21: FindRemainingSupport

```
Input  : Support $\hat{\mathcal{S}}_{\boldsymbol{y}^{(1)}}$
         Parameters $n_1$, $n_2$, $n$, $w_{\mathrm{y}}$
         Public key $(\boldsymbol{h}, \boldsymbol{s}) \in \mathbb{F}_2^n \times \mathbb{F}_2^n$
Output: Support $\hat{\mathcal{S}}_{\boldsymbol{y}}$
$\tau \leftarrow |\hat{\mathcal{S}}_{\boldsymbol{y}^{(1)}}|$
$\{j_1, \ldots, j_\tau\} \leftarrow \hat{\mathcal{S}}_{\boldsymbol{y}^{(1)}}$
$\boldsymbol{H} \leftarrow [\boldsymbol{I}, \mathrm{rot}(\boldsymbol{h})] \in \mathbb{F}_2^{n \times 2n}$
$\tilde{\boldsymbol{s}} \leftarrow \boldsymbol{s} + \boldsymbol{h}_{n+j_1}^\top + \ldots + \boldsymbol{h}_{n+j_\tau}^\top \in \mathbb{F}_2^n$, where $\boldsymbol{h}_\ell$ is the $\ell$-th column of $\boldsymbol{H}$
for $\{\hat{l}_1, \ldots, \hat{l}_{w_{\mathrm{y}}-\tau}\} \in \{\mathcal{L}' \subseteq [n_1 n_2 + 1 : n] : |\mathcal{L}'| = w_{\mathrm{y}} - \tau\}$ do
    $\hat{\boldsymbol{x}} \leftarrow \tilde{\boldsymbol{s}} + \boldsymbol{h}_{n+\hat{l}_1}^\top + \ldots + \boldsymbol{h}_{n+\hat{l}_{w_{\mathrm{y}}-\tau}}^\top \in \mathbb{F}_2^n$, where $\boldsymbol{h}_\ell$ is the $\ell$-th column of $\boldsymbol{H}$
    $\hat{\boldsymbol{y}} \leftarrow$ Vector in $\mathbb{F}_2^n$ with support $\hat{\mathcal{S}}_{\boldsymbol{y}^{(1)}} \cup \{\hat{l}_1, \ldots, \hat{l}_{w_{\mathrm{y}}-\tau}\}$
    if $\mathrm{wt}_{\mathrm{H}}(\hat{\boldsymbol{x}}) = w_{\mathrm{y}} \wedge \hat{\boldsymbol{x}} + \hat{\boldsymbol{y}}\boldsymbol{h} = \boldsymbol{s}$ then
        $\hat{\mathcal{S}}_{\boldsymbol{y}} \leftarrow \hat{\mathcal{S}}_{\boldsymbol{y}^{(1)}} \cup \{\hat{l}_1, \ldots, \hat{l}_{w_{\mathrm{y}}-\tau}\}$
        return $\hat{\mathcal{S}}_{\boldsymbol{y}}$
return $\hat{\mathcal{S}}_{\boldsymbol{y}} \leftarrow \{\}$
```

of $\boldsymbol{y}^{(1)}$ with $|\hat{\mathcal{S}}_{\boldsymbol{y}^{(1)}}| = \tau$. Then, given $\hat{\mathcal{S}}_{\boldsymbol{y}^{(1)}}$, n_1, n_2, n, w_{y}, and $(\boldsymbol{h}, \boldsymbol{s})$, Algorithm 21 requires

$$O\left(n(w_{\mathrm{y}} - \tau)\binom{n - n_1 n_2}{w_{\mathrm{y}} - \tau}\right)$$

operations in $\mathbb{F}_2$ in order to output the support of $\boldsymbol{y}^{(2)}$.

Proof. Let $\{j_1, \ldots, j_\tau\} = \hat{\mathcal{S}}_{\boldsymbol{y}^{(1)}} \subseteq [1 : n_1 n_2]$ be the known support of $\boldsymbol{y}^{(1)}$, and let the unknown support of $\boldsymbol{y}^{(2)}$ be denoted by $\mathcal{L} = \{l_1, \ldots, l_{w_{\mathrm{y}}-\tau}\} \subseteq [n_1 n_2 + 1 : n]$. It holds that

$$\boldsymbol{s} = \boldsymbol{x} + \boldsymbol{h}\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{h}_{n+j_1}^\top + \ldots + \boldsymbol{h}_{n+j_\tau}^\top + \boldsymbol{h}_{n+l_1}^\top + \ldots + \boldsymbol{h}_{n+l_{w_{\mathrm{y}}-\tau}}^\top,$$

where $\boldsymbol{h}_\ell \in \mathbb{F}_2^n$ is the ℓ-th column of the matrix $\boldsymbol{H} = [\boldsymbol{I}, \mathrm{rot}(\boldsymbol{h})] \in \mathbb{F}_2^{n \times 2n}$, for $\ell \in [1 : 2n]$. By assumption, the vector $\boldsymbol{s}$, the vector $\boldsymbol{h}$, and the set $\hat{\mathcal{S}}_{\boldsymbol{y}^{(1)}}$ are known, and therefore, the vector

$$\tilde{\boldsymbol{s}} = \boldsymbol{s} + \boldsymbol{h}_{n+j_1}^\top + \ldots + \boldsymbol{h}_{n+j_\tau}^\top = \boldsymbol{x} + \boldsymbol{h}_{n+l_1}^\top + \ldots + \boldsymbol{h}_{n+l_{w_{\mathrm{y}}-\tau}}^\top$$

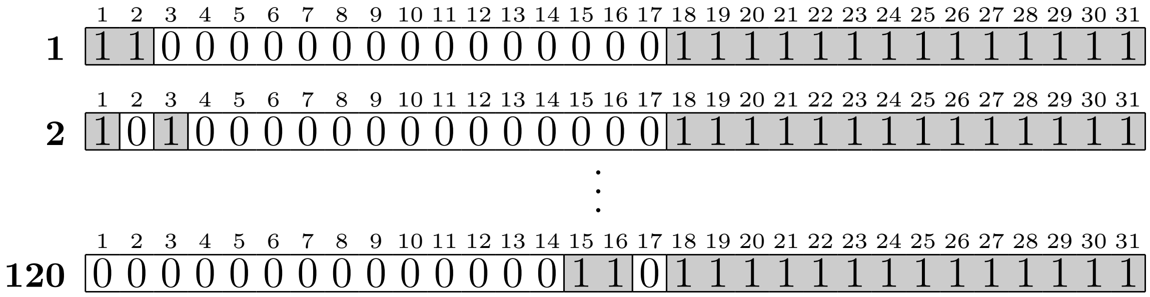


Figure 4.2: Illustration of the patterns used to obtain $\mathrm{supp}_\mathrm{H}\Big(\boldsymbol{y}_i^{(1)}\Big)$ from $\tilde{\mathcal{S}}_{\boldsymbol{y}_i}$ for $n_2 = 31$ and $\tilde{\mathcal{S}}_{\boldsymbol{y}_i} = [1:16]$. The gray parts refer to non-zero entries, and the white parts indicate zero entries.

can be computed by the algorithm. Then, the algorithm computes

$$\hat{\boldsymbol{x}} := \tilde{\boldsymbol{s}} + \boldsymbol{h}_{n+\hat{l}_1}^\top + \ldots + \boldsymbol{h}_{n+\hat{l}_{w_\mathrm{y}-\tau}}^\top$$

for $\{\hat{l}_1, \ldots, \hat{l}_{w_\mathrm{y}-\tau}\} \in \{\mathcal{L}' \subseteq [n_1 n_2 + 1 : n] : |\mathcal{L}'| = w_\mathrm{y} - \tau\}$ until both $\mathrm{wt}_\mathrm{H}(\hat{\boldsymbol{x}}) = w_\mathrm{y}$ and $\hat{\boldsymbol{x}} + \hat{\boldsymbol{y}}\boldsymbol{h} = \boldsymbol{s}$ are fulfilled, where $\hat{\boldsymbol{y}} \in \mathbb{F}_2^n$ has support $\hat{\mathcal{S}}_{\boldsymbol{y}^{(1)}} \cup \{\hat{l}_1, \ldots, \hat{l}_{w_\mathrm{y}-\tau}\}$. Since $\binom{2n}{2w_\mathrm{y}} \ll 2^n$ for the parameters in Table 4.4, it holds that $\hat{\boldsymbol{x}} = \boldsymbol{x}$ and $\hat{\boldsymbol{y}} = \boldsymbol{y}$ with high probability. Furthermore, the set $\{\hat{l}_1, \ldots, \hat{l}_{w_\mathrm{y}-\tau}\}$ is equal to $\mathcal{L}$ and $\hat{\mathcal{S}}_{\boldsymbol{y}^{(1)}} \cup \{\hat{l}_1, \ldots, \hat{l}_{w_\mathrm{y}-\tau}\} = \mathrm{supp}_\mathrm{H}(\boldsymbol{y})$.

The probability that a randomly drawn set $\{\hat{l}_1, \ldots, \hat{l}_{w_\mathrm{y}-\tau}\}$ is equal to $\mathcal{L}$ is given by $\binom{n-n_1n_2}{w_\mathrm{y}-\tau}^{-1}$ and examining whether $\{\hat{l}_1, \ldots, \hat{l}_{w_\mathrm{y}-\tau}\}$ corresponds to $\mathcal{L}$ needs $w_\mathrm{y} - \tau$ column additions, which is in $O(n(w_\mathrm{y} - \tau))$. Therefore, the complexity of this method is given by

$$n(w_\mathrm{y} - \tau)\binom{n - n_1 n_2}{w_\mathrm{y} - \tau}. \qquad \blacksquare$$

Remark 4.1. *Although Algorithm 21 has an exponential complexity, it is feasible since $n - n_1n_2$ is a small number for all parameter sets,[4] and therefore, $w_\mathrm{y} - \tau$ has a value close to zero with high probability. Assuming $w_\mathrm{y} - \tau \leq 2$, the complexity of this approach is $2^{28.42}$, $2^{18.05}$, and $2^{21.47}$ for the parameter sets of HQC-128, HQC-192, and HQC-256, respectively.*

[4] For HQC-128, HQC-192, and HQC-256, the variable $n - n_1n_2$ is equal to 123, 3, and 7, respectively.

Success Probability and Complexity of Algorithm 18

Using the intermediate results from above, we are able to prove the main statement of this section.

Theorem 4.13. *Let n_1, n_2, n, k, w_y, w_r, w_e, and δ be parameters chosen according to Table 4.4. Let $(\boldsymbol{x}, \boldsymbol{y})$ and $(\boldsymbol{h}, \boldsymbol{s})$ be a private and public key pair generated by* $\mathsf{KeyGen}_\mathrm{HQC}$ *for the chosen parameters. Furthermore, let $\mathcal{O}^\mathrm{Dec}_{(\boldsymbol{x},\boldsymbol{y})}$ be defined as in Definition 4.1. Then, given n_1, n_2, n, w_y, $(\boldsymbol{h}, \boldsymbol{s})$, and oracle access $\mathcal{O}^\mathrm{Dec}_{(\boldsymbol{x},\boldsymbol{y})}$, Algorithm 18 requires at most $2n_1n_2^2 + 8n_1$ queries to $\mathcal{O}^\mathrm{Dec}_{(\boldsymbol{x},\boldsymbol{y})}$ to retrieve the private key $(\boldsymbol{x}, \boldsymbol{y})$ with a probability of at least 93.20%, 77.98%, and 58.99% for security levels of 128 bit, 192 bit, and 256 bit, respectively.*

Proof. First, from Lines 13–16 of Algorithm 18 follows that Algorithm 18 outputs $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}})$ only if $\mathrm{wt_H}(\hat{\boldsymbol{x}}) = w_\mathrm{y}$, $\mathrm{wt_H}(\hat{\boldsymbol{y}}) = w_\mathrm{y}$, and $\hat{\boldsymbol{x}} + \hat{\boldsymbol{y}}\boldsymbol{h} = \boldsymbol{s}$. Since $\binom{2n}{2w_\mathrm{y}} \ll 2^n$ for the parameters of HQC-128, HQC-192, and HQC-256, it holds that $\hat{\boldsymbol{x}} = \boldsymbol{x}$ and $\hat{\boldsymbol{y}} = \boldsymbol{y}$ with high probability. This means that with high probability, Algorithm 18 either outputs the private key or a failure.

From Table 4.5 follows that the probability that $\max_{j' \in [1:n_1]} \left\{ \mathrm{wt_H}\left(\boldsymbol{y}^{(1)}_{j'}\right)\right\} \leq 2$ is equal to 93.20%, 77.98%, and 58.99% for the parameter sets HQC-128, HQC-192 and HQC-256, respectively. Then, Lemma 4.10, 4.11, and 4.12 directly imply that Algorithm 18 determines $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}})$ such that $\mathrm{wt_H}(\hat{\boldsymbol{x}}) = w_\mathrm{y}$, $\mathrm{wt_H}(\hat{\boldsymbol{y}}) = w_\mathrm{y}$, and $\hat{\boldsymbol{x}} + \hat{\boldsymbol{y}}\boldsymbol{h} = \boldsymbol{s}$ with probability at least 93.20%, 77.98%, and 58.99% for security levels of 128 bit, 192 bit, and 256 bit using at most $2n_1n_2^2 + 8n_1$ queries to $\mathcal{O}^\mathrm{Dec}_{(\boldsymbol{x},\boldsymbol{y})}$. ■

Remark 4.2. *Note that we conjecture that Algorithm 18 can easily be extended to cases with $\max_{j' \in [1:n_1]} \left\{ \mathrm{wt_H}\left(\boldsymbol{y}^{(1)}_{j'}\right)\right\} > 2$. To do so, the number of patterns shown in Table 4.6 must be increased and refined. Furthermore, the mapping given in Table 4.7 must be adapted to these patterns. However, this comes at the disadvantage of a higher complexity and more queries to the oracle $\mathcal{O}^\mathrm{Dec}_{(\boldsymbol{x},\boldsymbol{y})}$.*

ISD Attacks on HQC

There are scenarios in which we are only able to obtain a subset $\mathcal{P} = \{p_1, \dots, p_\tau\} \subsetneq \mathrm{supp_H}(\boldsymbol{y})$. This may be due to errors during the power measurements[5] or due to private

[5] Errors during the power measurements imply that the oracle $\mathcal{O}^\mathrm{Dec}_{(\boldsymbol{x},\boldsymbol{y})}$ does not always return the correct answer.

keys with vectors $\boldsymbol{y}_i^{(1)}$ of rather large Hamming weight. In these cases, we can exploit the knowledge of $\mathcal{P}$ to reduce the complexity of ISD algorithms in order to recover the support of $\boldsymbol{y}$. For that, we compute $\boldsymbol{s}' = \boldsymbol{s} + \boldsymbol{h}_{n+p_1}^\top + \ldots + \boldsymbol{h}_{n+p_\tau}^\top$, where $\boldsymbol{h}_\ell \in \mathbb{F}_2^n$ is the ℓ-th column of the matrix $[\boldsymbol{I}, \mathrm{rot}(\boldsymbol{h})] \in \mathbb{F}_2^{n \times 2n}$, for $\ell \in [1:2n]$. It holds that $\boldsymbol{s}'$ is the syndrome of the parity-check matrix $\boldsymbol{H} \in \mathbb{F}_2^{n \times 2n}$ and the error $[\boldsymbol{e}_1', \boldsymbol{e}_2']$, where $\boldsymbol{e}_1' \in \mathbb{F}_2^n$ has Hamming weight w_y and $\boldsymbol{e}_2' \in \mathbb{F}_2^n$ has Hamming weight $w_\mathrm{y} - \tau$. Then, we can use the proposed modifications of Prange's, Lee and Brickell's, and Stern's algorithms (see Algorithm 15, 16, and 17 in Section 4.2.3), which have a complexity of

$$W_{\mathrm{HQC,Pr}} \coloneqq \min_{k_1} (2n)^3 \frac{\binom{n}{w_\mathrm{y}}}{\binom{n-k_1}{w_\mathrm{y}}} \frac{\binom{n}{w_\mathrm{y}-\tau}}{\binom{k_1}{w_\mathrm{y}-\tau}},$$

$$W_{\mathrm{HQC,LB}} \coloneqq \min_{k_1, p_\mathrm{LB}} \frac{(2n)^3 + n(p_\mathrm{LB}+1)\binom{n}{p_\mathrm{LB}}}{\sum\limits_{\substack{\boldsymbol{a} \in \mathbb{N}_0^2 \\ a_1 \leq w_\mathrm{y} \\ a_2 \leq w_\mathrm{y}-\tau \\ a_1+a_2=p_\mathrm{LB}}} \frac{\binom{k_1}{a_1}\binom{n-k_1}{w_\mathrm{y}-a_1}}{\binom{n}{w_\mathrm{y}}} \frac{\binom{n-k_1}{a_2}\binom{k_1}{w_\mathrm{y}-\tau-a_2}}{\binom{n}{w_\mathrm{y}-\tau}}},$$

and

$$W_{\mathrm{HQC,St}} \coloneqq \min_{k_1, p_\mathrm{St}, \nu_{\mathrm{St},1}, \nu_{\mathrm{St},2}} \frac{(2n)^3 + (\nu_{\mathrm{St},1}+\nu_{\mathrm{St},2}) \left(\sum\limits_{i=1}^{p_\mathrm{St}} \binom{M_1}{i} + \sum\limits_{i=1}^{p_\mathrm{St}} \binom{M_2}{i} - n + \binom{M_2}{p_\mathrm{St}} \right) \cdots}{\sum\limits_{\substack{\boldsymbol{a} \in \mathbb{N}_0^2 \\ a_1 \leq w_\mathrm{y} \\ a_2 \leq w_\mathrm{y}-\tau \\ a_1+a_2=p_\mathrm{St}}} \sum\limits_{\substack{\boldsymbol{b} \in \mathbb{N}_0^2 \\ b_1 \leq w_\mathrm{y}-a_1 \\ b_2 \leq w_\mathrm{y}-\tau-a_2 \\ b_1+b_2=p_\mathrm{St}}} \frac{\binom{\lfloor k_1/2 \rfloor}{a_1}\binom{\lceil k_1/2 \rceil}{b_1}\binom{n-k_1-\nu_{\mathrm{St},1}}{w_\mathrm{y}-a_1-b_1}}{\binom{n}{w_\mathrm{y}}} \cdots}$$

$$\frac{\cdots + 2^{1-\nu_{\mathrm{St},1}-\nu_{\mathrm{St},2}} \binom{M_1}{p_\mathrm{St}}\binom{M_2}{p_\mathrm{St}} (2w_\mathrm{y} - \tau - 2p_\mathrm{St} + 1)(2p_\mathrm{St}+1)}{\cdots \times \frac{\binom{\lfloor (n-k_1)/2 \rfloor}{a_2}\binom{\lceil (n-k_1)/2 \rceil}{b_2}\binom{k_1-\nu_{\mathrm{St},2}}{w_\mathrm{y}-\tau-a_2-b_2}}{\binom{n}{w_\mathrm{y}-\tau}}},$$

where $M_1 = \lfloor k_1/2 \rfloor + \lfloor (n-k_1)/2 \rfloor$ and $M_2 = \lceil k_1/2 \rceil + \lceil (n-k_1)/2 \rceil$.

We show the complexities of the modified ISD algorithms for the parameters of HQC-128, HQC-192, and HQC-256 as a function of τ in the Figures 4.3, 4.4, and 4.5, respectively. We observe that the modification of Stern's algorithm requires considerably fewer operations than the modifications of Prange's algorithm and Lee and Brickell's algorithm. Furthermore, the complexity of the modification of Stern's al-

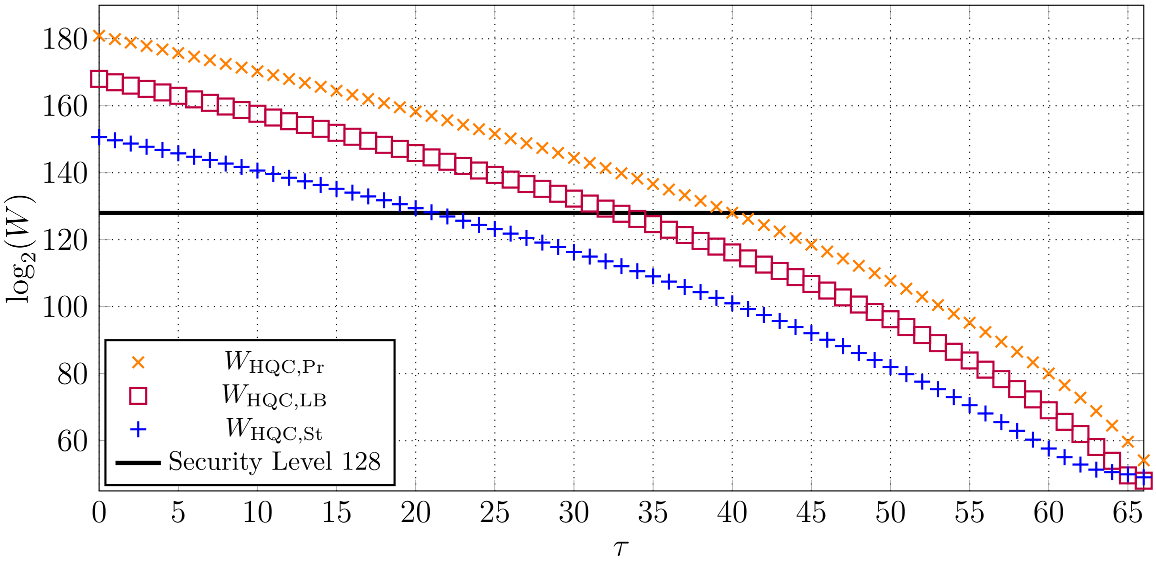


Figure 4.3: Complexities of the modifications of Prange's, Lee and Brickell's, and Stern's algorithms (Algorithm 15, 16, and 17) for the parameter set of HQC-128 as a function of τ, where τ is the number of known non-zero entries in $\boldsymbol{y}$.

gorithm is already below the security level if approximately 20 non-zero entries in $\boldsymbol{y}$ are known, and the complexity of all considered algorithms is far lower than the claimed security level if τ is close to w_{y}. Note that the shown values are conservative estimations of the ISD complexity since we assume that per guess of an information set, we have to solve a large system of equations, which can be replaced by a more sophisticated algorithm as shown in [198].

4.2.5 A Potential Countermeasure Against the Proposed Attacks

The proposed attacks rely on the oracle $\mathcal{O}^{\mathrm{Dec}}_{(\boldsymbol{x},\boldsymbol{y})}$. To realize $\mathcal{O}^{\mathrm{Dec}}_{(\boldsymbol{x},\boldsymbol{y})}$, a power analysis of the syndrome-based decoding of the BCH code in the decryption algorithm is used [187], [189, Sec. 4]. This analysis builds on the fact that the applied BCH decoder first computes the syndrome, and then, it determines the error using only this syndrome and independently of the codeword. This directly implies that codeword masking in combination with the currently applied syndrome-based decoder cannot prevent the proposed attack. However, the author of this thesis believes that codeword masking in combination with an interpolation-based decoding [75] of the BCH code constitutes a potential countermeasure against the proposed attacks. Such a modified decryption al-

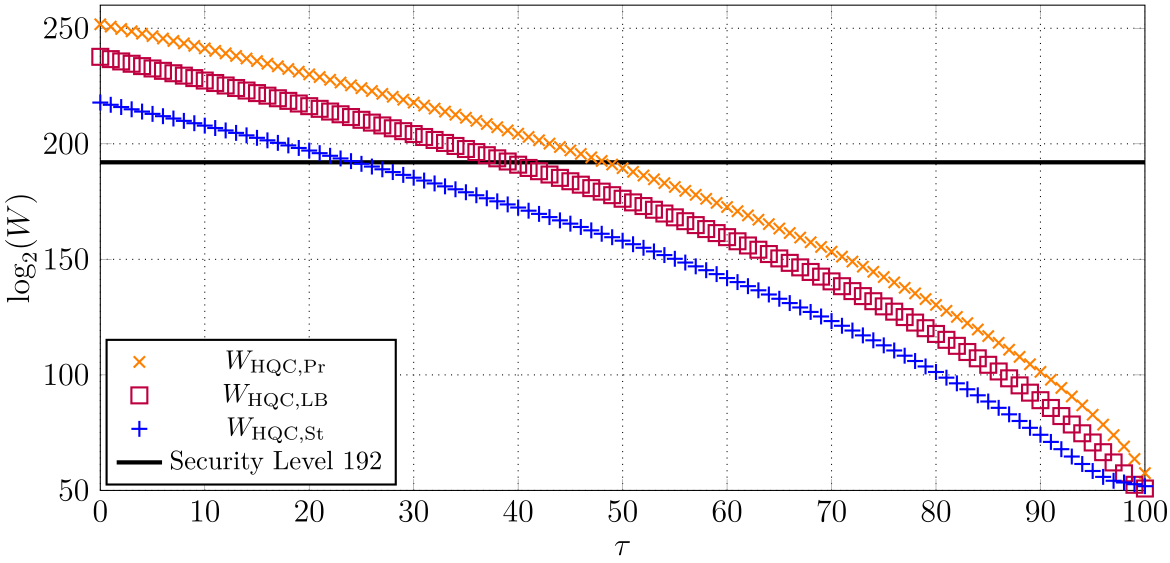


Figure 4.4: Complexities of the modifications of Prange's, Lee and Brickell's, and Stern's algorithms (Algorithm 15, 16, and 17) for the parameter set of HQC-192 as a function of τ, where τ is the number of known non-zero entries in $\boldsymbol{y}$.

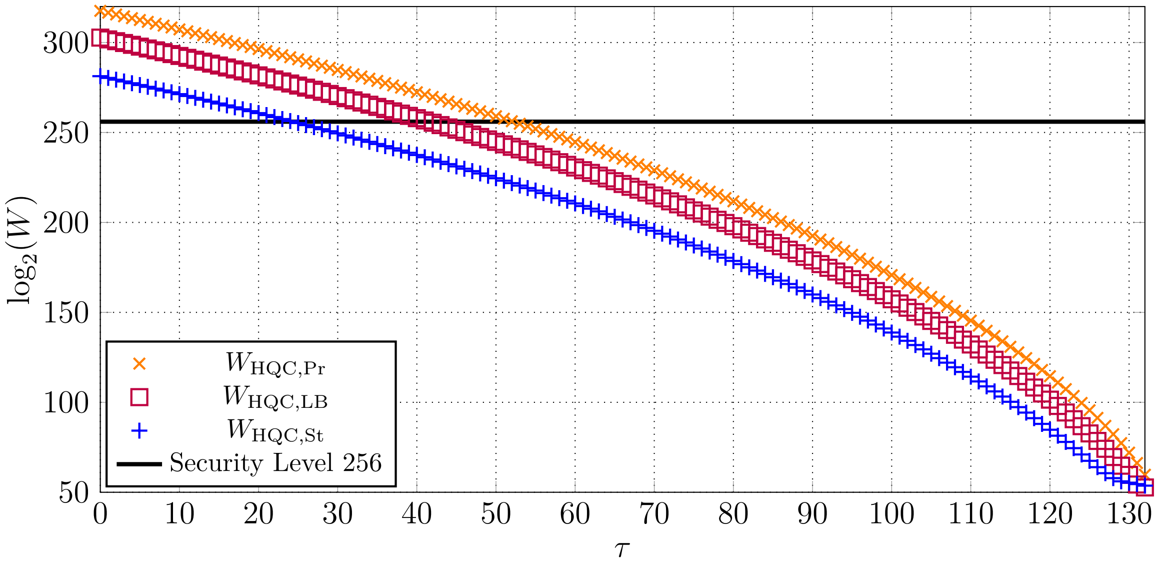


Figure 4.5: Complexities of the modifications of Prange's, Lee and Brickell's, and Stern's algorithms (Algorithm 15, 16, and 17) for the parameter set of HQC-256 as a function of τ, where τ is the number of known non-zero entries in $\boldsymbol{y}$.

gorithm is shown in Algorithm 22, where the interpolation-based decoder InterDecHQC is presented in Algorithm 23, and GetCodeLocators refers to a function that outputs the code locators used to construct the BCH code. It can be observed that the modified decoding of the BCH code strongly dependents on the (random) codeword, and therefore, the measure prevents the construction of the oracle $\mathcal{O}^{\mathrm{Dec}}_{(\boldsymbol{x},\boldsymbol{y})}$ using the power analysis shown in [187], [189, Sec. 4].

Note that the described countermeasure is only a theoretical thought at this stage, and side-channel analyses of implementations of this measure are required to evaluate its efficacy. However, this is beyond the scope of this dissertation and left for future research.

Algorithm 22: ModifiedDecrypt$_{\mathrm{HQC}}$

```
Input  : Ciphertext $(\boldsymbol{u},\boldsymbol{v}) \in \mathbb{F}_2^n \times \mathbb{F}_2^n$
         Private key $(\boldsymbol{x},\boldsymbol{y}) \in \mathbb{F}_2^n \times \mathbb{F}_2^n$
Output: Plaintext vector $\boldsymbol{m} \in \mathbb{F}_2^k$
$\boldsymbol{m}' \leftarrow \mathbb{F}_2^k$
$\boldsymbol{y}' \leftarrow$ EncHQC$(\mathcal{C}_{\mathrm{HQC}}, \boldsymbol{m}') + \boldsymbol{u}\boldsymbol{y} \in \mathbb{F}_2^n$
$\boldsymbol{v}' \leftarrow \boldsymbol{v} - \boldsymbol{y}' \in \mathbb{F}_2^n$
$\hat{\boldsymbol{m}} \leftarrow$ InterDecHQC$(\mathcal{C}_{\mathrm{HQC}}, \boldsymbol{v}') \in \mathbb{F}_2^k$
$\boldsymbol{m} \leftarrow \hat{\boldsymbol{m}} + \boldsymbol{m}' \in \mathbb{F}_2^k$
return Plaintext vector $\boldsymbol{m}$
```

```
Algorithm 23: InterDecHQC
Input  : Code $\mathcal{C}_{\mathrm{HQC}} \subseteq \mathbb{F}_2^n$
         Received word $\boldsymbol{v}' = [\boldsymbol{v}'_1, \dots, \boldsymbol{v}'_{n_1}, \boldsymbol{v}'_{n_1+1}] \in \mathbb{F}_2^n$
Output: Message $\boldsymbol{m} \in \mathbb{F}_2^k$

Step 1: Majority-based decoding of the repetition code
$\tilde{\boldsymbol{v}} = [\tilde{v}_1, \dots, \tilde{v}_{n_1}] \leftarrow \mathbf{0} \in \mathbb{F}_2^{n_1}$
for $i \in [1:n_1]$ do
    if $\mathrm{wt}_{\mathrm{H}}(\boldsymbol{v}'_i) \geq \lceil \frac{n_2}{2} \rceil$ then
        $\tilde{v}_i \leftarrow 1$
    else
        $\tilde{v}_i \leftarrow 0$

Step 2: Interpolation-based decoding of the $[n_1, k]_{\mathbb{F}_2}$ BCH code
$\tau_{\mathrm{RS}} \leftarrow \lfloor \frac{n_1-k}{2} \rfloor$
$\boldsymbol{N} \leftarrow \mathbf{0} \in \mathbb{F}_{2^m}^{n_1 \times (2(n_1-\tau_{\mathrm{RS}})-k+1))}$
$[\alpha_1, \dots, \alpha_{n_1}] \leftarrow \mathsf{GetCodeLocators}(\mathcal{C}_{\mathrm{HQC}}) \in \mathbb{F}_{2^m}^{n_1}$
for $i \in [1:n_1]$ do
    for $j \in [1:n_1 - \tau_{\mathrm{RS}}]$ do
        $N_{i,j} = \alpha_i^{j-1}$
    for $j \in [1:n_1 - \tau_{\mathrm{RS}} - k + 1]$ do
        $N_{i,(n_1-\tau_{\mathrm{RS}}+j)} = \tilde{v}_i \alpha_i^{j-1}$

$\boldsymbol{q} = [q_1, \dots, q_{2(n-\tau_{\mathrm{RS}})-k+1}] \xleftarrow{\$} \mathcal{K}_{2^m}(\boldsymbol{N})$
$Q_1(X) \leftarrow \sum_{i=1}^{n_1-\tau_{\mathrm{RS}}} q_i X^{i-1} \in \mathbb{F}_{2^m}[X]$
$Q_2(X) \leftarrow \sum_{i=1}^{n_1-\tau_{\mathrm{RS}}-k+1} q_{n_1-\tau_{\mathrm{RS}}+i} X^{i-1} \in \mathbb{F}_{2^m}[X]$
$M(X) = \sum_{i=1}^{k} m_i X^{i-1} \leftarrow -Q_1(X)/Q_2(X) \in \mathbb{F}_{2^m}[X]$
return Message $\boldsymbol{m} = [m_1, \dots, m_k]$
```

4.3 Concluding Remarks

In the first part of this chapter, we proposed an efficient key-recovery attack on the McEliece variant that deploys TRS codes [84]. The attack does not disprove the structural properties derived in [84], but constitutes a method to retrieve the structure of a *subfield subcode* of the public TRS code. This structure, in turn, allows us to determine a description of the supercode. We proved that this attack recovers a valid private key from the public key for all practical parameters in $O(n^4)$ field operations. Furthermore, we confirmed the feasibility of the attack by experiments, where we recovered a valid private key for a claimed security level of 128 bits within a few minutes. The presented subfield subcode approach disproves the widespread belief that the restriction of a code to a subfield is an operation that breaks its algebraic structure. Our cryptanalysis shows that subfield subcodes, as well as punctured codes and shortened codes, must also be considered when assessing the security of variants of the McEliece cryptosystem. Although we have shown that the variant of the McEliece cryptosystem based on the subfamily of TRS codes proposed in [84] is not secure, this does not imply that *any* subfamily of TRS codes is not suitable. In fact, TRS codes represent a very large family of codes, and further research is required to determine if other subfamilies exist that could be used in the design of secure McEliece systems. In [199], a rank-based McEliece system that deploys a subfamily of twisted Gabidulin codes was proposed. It was shown in [188, Sec. 5.2] that the presented attack cannot straightforwardly be applied to the aforementioned system, but the authors of [188] stated potential weaknesses that could be analyzed in a future work.

In the second part of this chapter, we presented the first power side-channel attack against the HQC encryption scheme and its KEM version. The presented attack exploits a power side-channel to construct an oracle that returns whether the BCH decoder in the decryption algorithm of HQC corrects an error for a chosen ciphertext. Based on the applied decoding algorithm of the product code, we showed how to design queries to the oracle such that its output allows us to recover a large part of the private key. The remaining part of the key can then be determined by an algorithm based on linear algebra. We observed that the success of the shown attack depends on the Hamming support of the private key, and the attack retrieves 93.20%, 77.98%, and 58.99% of the private keys of HQC-128, HQC-192, and HQC-256, respectively. For the remaining keys or in case of noisy side-channel information, we proposed to apply the presented modifications of Prange's, Lee and Brickell's, and Stern's algorithms in order to determine a valid private key with significantly less operations than the claimed security level. In future work, the presented attack could be adapted to the new variant of HQC, which uses a code concatenation of a Reed–Muller and an RS code instead of the originally proposed product code. Furthermore, the described countermeasure could be implemented and analyzed with respect to its performance and its vulnerability to side-channel attacks.

5

LIGA: A Rank-Metric Code-Based Encryption Scheme

The FL rank-based encryption scheme [62, 200] relies on the problem of reconstructing linearized polynomials and constitutes the linearized equivalent of the broken Augot–Finiasz cryptosystem [61]. While the Augot–Finiasz encryption scheme is closely connected to list decoding of RS codes, the FL system is based on the difficulty of list decoding of Gabidulin codes. In contrast to the original McEliece encryption scheme, where the public key is a *matrix* [13], in the FL system, the public key is only a *vector*, resulting in a much smaller public key size. When the FL scheme was proposed for the first time, it was only conjectured that Gabidulin codes cannot be list decoded efficiently. As this was proven in the last years for many families of Gabidulin codes [175–177], the FL system could be a promising post-quantum secure public-key cryptosystem. However, the recent attack by Gaborit, Otmani, and Talé Kalachi (GOT) [65] retrieves an alternative public key in cubic time complexity.

In this chapter, we present the new rank-based encryption scheme LIGA, which is based on the original FL system. The security of the FL system relies on the proven hardness of list decoding Gabidulin codes, but it is vulnerable to the attack from [65]. To derive the new system, we first propose a new coding-theoretic interpretation of the original FL system, and we develop an alternative decryption algorithm. Then, we show that the public key is equivalent to a corrupted codeword of an interleaved Gabidulin code, and we show that the failure condition of the GOT attack [65] is equal to the failure condition of decoding the public key as a corrupted interleaved Gabidulin codeword. This observation allows us to design the new rank-based public-key encryption scheme as well as the corresponding KEM, which are based on the hardness of list and interleaved decoding of Gabidulin codes. Under the hardness assumptions of problems related to list and interleaved decoding of Gabidulin codes, we show that the encryption version of LIGA is IND-CPA secure in the standard model and the KEM variant is IND-CCA2 secure in the random oracle model. We investigate possible exponential-time attacks on the aforementioned hard problems, provide sets of parameters for security levels 128 bit, 192 bit, and 256 bit and compare them to the NIST proposals RQC [151], ROLLO [150], BIKE [148], and Classic McEliece [135] as well as to Loidreau's McEliece-like system [51, 201].

The results of Chapter 5 are partly published in the proceedings of the *2018 IEEE International Symposium on Information Theory (ISIT)* [152] and in the journal *Designs, Codes and Cryptography* [106]. The author of this dissertation significantly contributed to the design of the encryption variant of LIGA and the analysis of existing attacks on it. He solely developed the KEM variant of LIGA, and he solely proved that the encryption variant is IND-CPA secure and that the KEM variant is IND-CCA2 secure under some hardness assumptions of problems related to list and interleaved decoding of Gabidulin codes. Furthermore, he implemented the encryption scheme in software, and he designed the parameter sets of LIGA and compared them to the parameter sets of other code-based encryption schemes.

Note that Bombar and Couvreur derived a new message recovery attack on the presented encryption scheme LIGA [202]. The authors are not disproving the IND-CPA and IND-CCA2 security claims under the assumption that the underlying problems are hard. They rather devise an efficient decoding routine for certain Gabidulin supercodes, which shows that the underlying hardness assumptions are incorrect. In addition, this new routine enables them to develop an efficient plaintext recovery attack on the system.

5.1 Key Generation in the FL System

In this section, we review the key-generation algorithm of the original FL encryption scheme, and we give a coding-theoretic interpretation of the original public key. Furthermore, we analyze the success condition of the GOT attack [65].

Table 5.1: Summary of the publicly known parameters of LIGA.

Name	Use	Restriction
q	small field size	prime power
m	extension degree	$1 \leq m$
n	code length	$n \leq m$
k	code dimension	$k < n$
u	extension degree	$2 \leq u < k$
w	error weight in public key	$w \geq \max\left\{n - k - \frac{k-u}{u-1}, \left\lfloor \frac{n-k}{2} \right\rfloor + 1\right\}$ $w < \frac{u}{u+2}(n-k)$
t_{pub}	error weight in ciphertext	$t_{\mathrm{pub}} = \left\lfloor \frac{n-k-w}{2} \right\rfloor$
ζ	$\mathbb{F}_{q^m}$-dimension of error vector in the public key	$\zeta < \frac{w}{n-k-w}$ and $\zeta q^{\zeta w - m} \leq \frac{1}{2}$

5.1.1 The Original Algorithm

Let q, m, n, k, u, w, and t_{pub} be positive integers that fulfill the restrictions given in Table 5.1. Then, the original FL key generation routine is shown in Algorithm 24, where $\mathbf{0}_{n-w}$ is the zero vector of length $n - w$.

Algorithm 24: Original FL Key Generation

```
Input  : Integers $q, m, n, k, u, w$ as in Table 5.1
Output: Private key $(\boldsymbol{x}, \boldsymbol{P}_{:,[w+1:n]}) \in \mathbb{F}_{q^{mu}}^{k} \times \mathbb{F}_{q}^{n \times (n-w)}$
        Public key $(\boldsymbol{g}, \boldsymbol{k}_{\mathrm{pub}}) \in \mathbb{F}_{q^m}^{n} \times \mathbb{F}_{q^{mu}}^{n}$
$\boldsymbol{g} \xleftarrow{\$} \{\boldsymbol{a} \in \mathbb{F}_{q^m}^{n} : \mathrm{rk}_q(\boldsymbol{a}) = n\}$
$\boldsymbol{x} \xleftarrow{\$} \{\boldsymbol{a} \in \mathbb{F}_{q^{mu}}^{k} : \dim_{q^m}(\langle a_{k-u+1}, \ldots, a_k \rangle_{q^m}) = u\}$
$\boldsymbol{s} \xleftarrow{\$} \{\boldsymbol{a} \in \mathbb{F}_{q^{mu}}^{w} : \mathrm{rk}_q(\boldsymbol{a}) = w\}$
$\boldsymbol{P} \xleftarrow{\$} \{\boldsymbol{A} \in \mathbb{F}_{q}^{n \times n} : \mathrm{rk}_q(\boldsymbol{A}) = n\}$
$\boldsymbol{G}_{\mathcal{G}} \leftarrow \mathcal{M}_{k,q}(\boldsymbol{g})$
$\boldsymbol{z} \leftarrow [\boldsymbol{s}, \mathbf{0}_{n-w}] \cdot \boldsymbol{P}^{-1}$
$\boldsymbol{k}_{\mathrm{pub}} \leftarrow \boldsymbol{x} \cdot \boldsymbol{G}_{\mathcal{G}} + \boldsymbol{z} \in \mathbb{F}_{q^{mu}}^{n}$
return Private key $(\boldsymbol{x}, \boldsymbol{P}_{:,[w+1:n]})$, Public key $(\boldsymbol{g}, \boldsymbol{k}_{\mathrm{pub}})$
```

5.1.2 Coding-Theoretic Interpretation of the Original Public Key

The public key $\boldsymbol{k}_{\text{pub}}$ of the FL system is a corrupted codeword of a u-interleaved Gabidulin code. To our knowledge, this connection between the public key and interleaved Gabidulin codes has not been known before. This interpretation is central to this chapter and is used in Section 5.2.1 to define the public key of LIGA such that is not vulnerable against the GOT attack [65].

Theorem 5.1. *Fix a basis $\boldsymbol{\gamma}$ of $\mathbb{F}_{q^{mu}}$ over $\mathbb{F}_{q^m}$. Let $\boldsymbol{\gamma}^*$ be a dual basis to $\boldsymbol{\gamma}$ and write $\boldsymbol{k}_{\text{pub}} = \sum_{i=1}^{u} \boldsymbol{k}_{\text{pub}}^{(i)} \gamma_i^*$. Then,*

$$\begin{bmatrix} \boldsymbol{k}_{\text{pub}}^{(1)} \\ \boldsymbol{k}_{\text{pub}}^{(2)} \\ \vdots \\ \boldsymbol{k}_{\text{pub}}^{(u)} \end{bmatrix} = \begin{bmatrix} \boldsymbol{c}_{\mathcal{G}}^{(1)} \\ \boldsymbol{c}_{\mathcal{G}}^{(2)} \\ \vdots \\ \boldsymbol{c}_{\mathcal{G}}^{(u)} \end{bmatrix} + \begin{bmatrix} \boldsymbol{z}_1 \\ \boldsymbol{z}_2 \\ \vdots \\ \boldsymbol{z}_u \end{bmatrix}, \tag{5.1}$$

where $\boldsymbol{c}_{\mathcal{G}}^{(1)}, \dots, \boldsymbol{c}_{\mathcal{G}}^{(u)} \in \mathbb{F}_{q^m}^n$ are codewords of the Gabidulin code $\mathcal{G}_k(\boldsymbol{g})$ with generator matrix $\boldsymbol{G}_{\mathcal{G}}$, and $\boldsymbol{z}_1, \dots, \boldsymbol{z}_u \in \mathbb{F}_{q^m}^n$ are obtained from the vector $\boldsymbol{z} \in \mathbb{F}_{q^{mu}}^n$ by $\boldsymbol{z} = \sum_{i=1}^{u} \boldsymbol{z}_i \gamma_i^*$.

Proof. Recall the definition of the public key

$$\boldsymbol{k}_{\text{pub}} = \boldsymbol{x} \cdot \boldsymbol{G}_{\mathcal{G}} + \boldsymbol{z},$$

where $\boldsymbol{x} \in \mathbb{F}_{q^{mu}}^k$, $\boldsymbol{G}_{\mathcal{G}} \in \mathbb{F}_{q^m}^{k \times n}$ is a generator matrix of the code $\mathcal{G}_k(\boldsymbol{g})$, and $\boldsymbol{z} \in \mathbb{F}_{q^{mu}}^n$ with $\operatorname{rk}_q(\boldsymbol{z}) = w$.

Let $\boldsymbol{x} = \sum_{i=1}^{u} \boldsymbol{x}_i \gamma_i^*$, where $\boldsymbol{x}_1, \dots, \boldsymbol{x}_u$ have coefficients in $\mathbb{F}_{q^m}$. Then, we obtain the following representation of the public key $\boldsymbol{k}_{\text{pub}}$ as a $u \times n$ matrix in $\mathbb{F}_{q^m}$:

$$\begin{bmatrix} \boldsymbol{k}_{\text{pub}}^{(1)} \\ \boldsymbol{k}_{\text{pub}}^{(2)} \\ \vdots \\ \boldsymbol{k}_{\text{pub}}^{(u)} \end{bmatrix} = \begin{bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \\ \vdots \\ \boldsymbol{x}_u \end{bmatrix} \cdot \boldsymbol{G}_{\mathcal{G}} + \begin{bmatrix} \boldsymbol{z}_1 \\ \boldsymbol{z}_2 \\ \vdots \\ \boldsymbol{z}_u \end{bmatrix} = \begin{bmatrix} \boldsymbol{x}_1 \cdot \boldsymbol{G}_{\mathcal{G}} \\ \boldsymbol{x}_2 \cdot \boldsymbol{G}_{\mathcal{G}} \\ \vdots \\ \boldsymbol{x}_u \cdot \boldsymbol{G}_{\mathcal{G}} \end{bmatrix} + \begin{bmatrix} \boldsymbol{z}_1 \\ \boldsymbol{z}_2 \\ \vdots \\ \boldsymbol{z}_u \end{bmatrix}.$$

Since $\boldsymbol{x}_i \cdot \boldsymbol{G}_{\mathcal{G}}$ is a codeword of the code $\mathcal{G}_k(\boldsymbol{g})$, $\forall i \in [1:u]$, the matrix representation of $\boldsymbol{k}_{\text{pub}}$ can be seen as a codeword from the code $\mathcal{G}_k^{(u)}(\boldsymbol{g})$, corrupted by an additive error. ■

Note that the error $[\boldsymbol{z}_1^\top, \dots, \boldsymbol{z}_u^\top]^\top$ in (5.1) has $\mathbb{F}_q$-rank at most w due to the structure of $\boldsymbol{z} = [\boldsymbol{s}, \boldsymbol{0}_{n-w}]\boldsymbol{P}^{-1}$.

5.1.3 Efficient Key Recovery of the Original Public Key

The GOT attack [65] on the original FL system is an efficient structural attack which computes a valid private key of the FL system in cubic time if the public key fulfills certain conditions. We recall this attack in the following and derive an alternative, equally powerful, attack based on the decoding of interleaved Gabidulin codes. We prove that the failure conditions of both attacks are equivalent. The interleaved decoding attack does not have any advantage in terms of cryptanalysis compared to the GOT attack, but enables us to exactly predict for which public keys both attacks work and for which the attacks fail.

GOT Attack

The GOT attack [65] is shown in Algorithm 25 and succeeds under the conditions of the following theorem.

```
Algorithm 25: GOT Attack
  Input  : Public key $(\boldsymbol{g}, \boldsymbol{k}_{\mathrm{pub}}) \in \mathbb{F}_{q^m}^n \times \mathbb{F}_{q^{mu}}^n$
  Output: Private key $(\boldsymbol{x}, \boldsymbol{P}_{:,[w+1:n]}) \in \mathbb{F}_{q^{mu}}^k \times \mathbb{F}_q^{n\times(n-w)}$
  Choose $\gamma_1, \dots, \gamma_u$ to be a basis of $\mathbb{F}_{q^{mu}}$ over $\mathbb{F}_{q^m}$
  for $i \in [1:u]$ do
      $\boldsymbol{k}_{\mathrm{pub}}^{(i)} \leftarrow \mathrm{Tr}(\gamma_i \boldsymbol{k}_{\mathrm{pub}})$
  $\boldsymbol{G}_{\mathcal{G}} \leftarrow \mathcal{M}_{k,q}(\boldsymbol{g})$
  Pick at random a non-zero vector $\widetilde{\boldsymbol{h}} \in \mathbb{F}_{q^m}^n$ such that
      $$\mathcal{M}_{n-w-k,q}\left(\begin{bmatrix} \boldsymbol{G}_{\mathcal{G}} \\ \boldsymbol{k}_{\mathrm{pub}}^{(1)} \\ \vdots \\ \boldsymbol{k}_{\mathrm{pub}}^{(u)} \end{bmatrix}\right) \cdot \widetilde{\boldsymbol{h}}^\top = \boldsymbol{0}.$$
  Choose $\boldsymbol{P} \in \mathbb{F}_q^{n\times n}$ and $\boldsymbol{h}' \in \mathbb{F}_{q^m}^{n-w}$ such that $\widetilde{\boldsymbol{h}}\left(\boldsymbol{P}^{-1}\right)^\top = [\boldsymbol{0}, \boldsymbol{h}']$
  Choose $\boldsymbol{x}$ such that $\boldsymbol{x}\boldsymbol{G}_{\mathcal{G}}\boldsymbol{P}' = \boldsymbol{k}_{\mathrm{pub}}\boldsymbol{P}'$, where $\boldsymbol{P}' = \boldsymbol{P}_{:,[w+1:n]} \in \mathbb{F}_q^{n\times(n-w)}$
  return Private key $(\boldsymbol{x}, \boldsymbol{P}_{:,[w+1:n]})$
```

Theorem 5.2 (Success Condition of the GOT Attack [65, Thm. 1])**.** *Let $\gamma_1, \dots, \gamma_u \in \mathbb{F}_{q^{mu}}$ be a basis of $\mathbb{F}_{q^{mu}}$ over $\mathbb{F}_{q^m}$ and let $\boldsymbol{z}_i = \mathrm{Tr}(\gamma_i \boldsymbol{z})$, for $i \in [1:u]$. If the matrix $\boldsymbol{Z} \in \mathbb{F}_{q^m}^{u \times n}$ with $\boldsymbol{z}_1, \dots, \boldsymbol{z}_u$ as rows, satisfies*

$$\mathrm{rk}_{q^m}(\mathcal{M}_{n-k-w,q}(\boldsymbol{Z})) = w,$$

then $(\boldsymbol{x}, \boldsymbol{P}_{:,[w+1:n]})$ can be recovered from $(\boldsymbol{g}, \boldsymbol{k}_{\mathrm{pub}})$ with $\mathcal{O}(n^3)$ operations in $\mathbb{F}_{q^{mu}}$ by using Algorithm 25.

From Theorem 5.2 follows that if the key is generated by Algorithm 24, the GOT attack breaks the original FL system *with high probability.*

Interleaved Decoding Attack

Recall from Theorem 5.1 that the public key $\boldsymbol{k}_{\mathrm{pub}}$ is a corrupted interleaved codeword. Based on this observation, we derive a structural attack on the original FL system to which we refer as *Interleaved Decoding Attack* in the following. We prove that interleaved decoding and the GOT attack *fail* for the *same public keys.* The idea is to decode $\boldsymbol{k}_{\mathrm{pub}}$ in an interleaved Gabidulin code. Since $w \leq \frac{u}{u+1}(n-k)$, such a decoder returns $\boldsymbol{x}$ with high probability, but fail in certain cases. Since $\mathrm{rk}_{q^m}(\mathcal{M}_{n-w-1,q}(\boldsymbol{g})) = n-w-1$, the interleaved decoder fails if

$$\mathrm{rk}_{q^m}\left(\tilde{\boldsymbol{Z}}\right) := \varphi < w, \tag{5.2}$$

where

$$\tilde{\boldsymbol{Z}} = \begin{bmatrix} \mathcal{M}_{n-k-w,q}(\boldsymbol{z}_1) \\ \mathcal{M}_{n-k-w,q}(\boldsymbol{z}_2) \\ \vdots \\ \mathcal{M}_{n-k-w,q}(\boldsymbol{z}_u) \end{bmatrix}, \tag{5.3}$$

see Lemma 2.5

Equivalence of the GOT Attack and the Interleaved Decoding Attack

In the following, we prove that the failure condition of the GOT attack is equivalent to the condition that decoding $\boldsymbol{k}_{\mathrm{pub}}$ in an interleaved Gabidulin code fails.

Theorem 5.3. *The GOT attack [65] fails if and only if the interleaved decoding attack fails. In particular, both fail if* (5.2) *holds.*

Proof. The matrix $\mathcal{M}_{n-k-w,q}(\boldsymbol{Z})$ from Theorem 5.2 and the matrix $\tilde{\boldsymbol{Z}}$ in (5.3) only differ in row permutations, and therefore, they have the same rank. Furthermore, the rank of the matrix $\tilde{\boldsymbol{Z}}$ in (5.3) cannot be larger than w since any vector in the right kernel of this matrix has rank weight at least $n - w$ [167, Algorithm 3.2.1]. It follows that the failures of Theorem 5.2 and Lemma 2.5 are equivalent. ■

In the next section, we exploit the observation of Theorem 5.3 to propose a new key-generation algorithm that avoids public keys that can be efficiently decoded by an interleaved decoder, thereby rendering the GOT attack useless.

5.2 The New System LIGA

In this section, we propose the encryption scheme

$$\Pi_{\mathrm{LIGA}}^{\mathrm{Enc}} = (\mathsf{KeyGen}_{\mathrm{LIGA}}, \mathsf{Encrypt}_{\mathrm{LIGA}}, \mathsf{Decrypt}_{\mathrm{LIGA}}).$$

The system is based on the original FL system [62], where we keep both the original encryption and decryption algorithm, but we replace the insecure key-generation algorithm. Furthermore, we present a KEM version of LIGA, which we denote by

$$\Pi_{\mathrm{LIGA}}^{\mathrm{KEM}} = (\mathsf{KeyGen}_{\mathrm{LIGA}}, \mathsf{Encaps}_{\mathrm{LIGA}}, \mathsf{Decaps}_{\mathrm{LIGA}}).$$

In Section 5.3, we analyze the security of the system. We single out problems from coding theory, and we prove that the encryption version is IND-CPA secure and the KEM version is IND-CCA2 secure under the assumption that the stated problems are hard. Furthermore, we study new and known attacks on these problems and show that they all run in exponential time, see Section 5.4.

5.2.1 The New Key-Generation Algorithm

We introduce a new key-generation algorithm that is based on choosing $\boldsymbol{z} = \sum_{i=1}^{u} \boldsymbol{z}_i \gamma_i^*$ such that $\varphi < w$, where φ is the rank of the interleaved Moore matrix of the errors $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_u$ in the public key, see (5.3). Based on the dimension of the span of $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_u$, we upper bound φ in the following Theorem 5.4. Recall that when $\varphi < w$, the GOT attack [65] and interleaved decoding of the public key fail, see Theorem 5.3. In this case, retrieving any knowledge about the private key from the public key requires to solve Problem 5.2 (defined later), which corresponds to decoding the interleaved codeword when error patterns occur for which all known decoders fail.

Theorem 5.4. *Let* $\dim_{q^m}(\langle \boldsymbol{z}_1, \ldots, \boldsymbol{z}_u \rangle_{q^m}) = \zeta$*. Then*

$$\varphi = \mathrm{rk}_{q^m}\left(\tilde{\boldsymbol{Z}}\right) \leq \min\{\zeta(n-k-w), w\}.$$

Proof. The dimension of $\langle \boldsymbol{z}_1, \ldots, \boldsymbol{z}_u \rangle_{q^m}$ implies that at most $\zeta(n-k-w)$ rows of $\tilde{\boldsymbol{Z}}$ are linearly independent over $\mathbb{F}_{q^m}$, meaning that $\varphi \leq \zeta(n-k-w)$. The definition of $\boldsymbol{z} = [\boldsymbol{s}, \boldsymbol{0}_{n-w}] \cdot \boldsymbol{P}^{-1}$ leads to

$$\varphi = \mathrm{rk}_{q^m}(\tilde{\boldsymbol{Z}}) = \mathrm{rk}_{q^m}\left(\begin{bmatrix} \mathcal{M}_{n-k-w,q}(\boldsymbol{s}_1), & \boldsymbol{0} \\ \vdots & \vdots \\ \mathcal{M}_{n-k-w,q}(\boldsymbol{s}_u), & \boldsymbol{0} \end{bmatrix} \boldsymbol{P}^{-1} \right) \leq w,$$

where the last inequality holds since $\boldsymbol{s}_1, \ldots, \boldsymbol{s}_u$ are vectors of length w. ■

We propose the following modification to Line 3 of the original key-generation algorithm, depending on the parameter ζ:

```
3   $\mathcal{A} \xleftarrow{\$} \Big\{ \text{subspace } \mathcal{U} \subseteq \mathbb{F}_{q^m}^w : \dim_{q^m} \mathcal{U} = \zeta, \mathcal{U} \text{ has a basis of full-}\mathbb{F}_q\text{-rank elements} \Big\}$
3'  $\begin{bmatrix} \boldsymbol{s}_1 \\ \vdots \\ \boldsymbol{s}_u \end{bmatrix} \xleftarrow{\$} \left\{ \begin{bmatrix} \boldsymbol{s}'_1 \\ \vdots \\ \boldsymbol{s}'_u \end{bmatrix} : \langle \boldsymbol{s}'_1, \ldots, \boldsymbol{s}'_u \rangle_{q^m} = \mathcal{A},\ \mathrm{rk}_q(\boldsymbol{s}'_i) = w, \forall\, i \in [1:u] \right\}$
```

Clearly, $\dim_{q^m}(\langle \boldsymbol{z}_1, \ldots, \boldsymbol{z}_u \rangle_{q^m}) = \zeta$ in this case. To avoid that the GOT attack [65] runs in polynomial time, Theorem 5.4 implies that the parameter ζ must be chosen such that $\zeta < \frac{w}{n-k-w}$. In Section 5.4, we discuss several further exponential-time attacks on LIGA. Some of these attacks have a work factor depending on ζ, which must be considered in the parameter design.

Furthermore, the condition $\mathrm{rk}_q(\boldsymbol{s}'_i) = w$ ensures that $\mathrm{rk}_q(\boldsymbol{z}_i) = w$, for $i \in [1:u]$. This choice maximizes the work factor of generic decoding attacks on the rows of the public key, see Section 5.4.

The restriction of the choice of $\mathcal{A}$ to subspaces that contain a basis of full-$\mathbb{F}_q$-rank codewords is to ensure that the set from which we sample in Line 3' is non-empty. Hence, the key generation always works.

Compared to the choice of $\boldsymbol{z}$ in Line 3 of the original key-generation algorithm (Algorithm 24), we restrict the choice of $\boldsymbol{z}$. However, we show in Section 5.4 that there are still enough possibilities for $\boldsymbol{z}$ to prevent an efficient naïve brute-force attack.

Appendix C.1 contains a more detailed discussion on how to realize Lines 3 and 3′ in practice.

5.2.2 The Public-Key Encryption Version $\Pi_{\mathrm{LIGA}}^{\mathrm{Enc}}$

The new key-generation algorithm $\mathsf{KeyGen}_{\mathrm{LIGA}}$, the encryption algorithm $\mathsf{Encrypt}_{\mathrm{LIGA}}$, and the decryption algorithm $\mathsf{Decrypt}_{\mathrm{LIGA}}$ are shown in Algorithm 26, Algorithm 27, and Algorithm 28, respectively. Compared to the original key-generation algorithm, the algorithm $\mathsf{KeyGen}_{\mathrm{LIGA}}$ has one more input parameter ζ (cf. Section 5.2.1).

```
Algorithm 26: KeyGen_LIGA
  Input  : Integers $q, m, n, k, u, w, t_{\mathrm{pub}}, \zeta$ as in Table 5.1
  Output: Private key $(\boldsymbol{x}, \boldsymbol{P}_{:,[w+1:n]}) \in \mathbb{F}_{q^{mu}}^{k} \times \mathbb{F}_{q}^{n\times(n-w)}$
          Public key $(\boldsymbol{g}, \boldsymbol{k}_{\mathrm{pub}}) \in \mathbb{F}_{q^m}^{n} \times \mathbb{F}_{q^{mu}}^{n}$
1  $\boldsymbol{g} \xleftarrow{\$} \{\boldsymbol{a} \in \mathbb{F}_{q^m}^{n} \,:\, \mathrm{rk}_q(\boldsymbol{a}) = n\}$
2  $\boldsymbol{x} \xleftarrow{\$} \{\boldsymbol{a} \in \mathbb{F}_{q^{mu}}^{k} \,:\, \dim_{q^m}(\langle a_{k-u+1}, \dots, a_k \rangle_{q^m}) = u\}$
3  $\mathcal{A} \xleftarrow{\$}$
     $\left\{\text{subspace } \mathcal{U} \subseteq \mathbb{F}_{q^m}^{w} \,:\, \dim_{q^m} \mathcal{U} = \zeta, \mathcal{U} \text{ has a basis of full-}\mathbb{F}_q\text{-rank elements}\right\}$
3' $\begin{bmatrix} \boldsymbol{s}_1 \\ \vdots \\ \boldsymbol{s}_u \end{bmatrix} \xleftarrow{\$} \left\{ \begin{bmatrix} \boldsymbol{s}_1' \\ \vdots \\ \boldsymbol{s}_u' \end{bmatrix} \,:\, \langle \boldsymbol{s}_1', \dots, \boldsymbol{s}_u' \rangle_{q^m} = \mathcal{A},\, \mathrm{rk}_q(\boldsymbol{s}_i') = w \,, \forall i \in [1:u] \right\}$
4  $\boldsymbol{s} \leftarrow \sum_{i=1}^{u} \boldsymbol{s}_i \gamma_i^*$
5  $\boldsymbol{P} \xleftarrow{\$} \{\boldsymbol{A} \in \mathbb{F}_q^{n\times n} : \mathrm{rk}_q(\boldsymbol{A}) = n\}$
6  $\boldsymbol{G}_{\mathcal{G}} \leftarrow \mathcal{M}_{k,q}(\boldsymbol{g})$
7  $\boldsymbol{z} \leftarrow [\boldsymbol{s}, \boldsymbol{0}_{n-w}] \cdot \boldsymbol{P}^{-1}$
8  $\boldsymbol{k}_{\mathrm{pub}} \leftarrow \boldsymbol{x} \cdot \boldsymbol{G}_{\mathcal{G}} + \boldsymbol{z} \in \mathbb{F}_{q^{mu}}^{n}$
9  return Private key $(\boldsymbol{x}, \boldsymbol{P}_{:,[w+1:n]})$, Public key $(\boldsymbol{g}, \boldsymbol{k}_{\mathrm{pub}})$
```

The proposed system has no decryption failures as proven in the following theorem.

Theorem 5.5 (Correctness of $\mathsf{Decrypt}_{\mathrm{LIGA}}$ [62])**.** *Algorithm 28 returns the plaintext* $\boldsymbol{m}$*.*

Proof. Line 1 computes

$$\boldsymbol{c}\boldsymbol{P} = (\boldsymbol{m} + \mathrm{Tr}(\alpha\boldsymbol{x}))\boldsymbol{G}_{\mathcal{G}}\boldsymbol{P} + [\mathrm{Tr}(\alpha\boldsymbol{s}), \boldsymbol{0}_{n-w}] + \boldsymbol{e}\boldsymbol{P},$$

Algorithm 27: $\mathsf{Encrypt}_{\mathrm{LIGA}}$

```
Input  : Plaintext $\boldsymbol{m} \in \mathbb{F}_{q^m}^{k-u}$
         Public key $(\boldsymbol{g}, \boldsymbol{k}_{\mathrm{pub}}) \in \mathbb{F}_{q^m}^n \times \mathbb{F}_{q^{mu}}^n$
         Randomness $\theta$
Output: Ciphertext $\boldsymbol{c} \in \mathbb{F}_{q^m}^n$
$\alpha \xleftarrow{\$} \mathbb{F}_{q^{mu}}$ using $\theta$
$\boldsymbol{e} \xleftarrow{\$} \{\boldsymbol{a} \in \mathbb{F}_{q^m}^n : \mathrm{rk}_q(\boldsymbol{a}) = t_{\mathrm{pub}}\}$ using $\theta$
$\boldsymbol{G}_{\mathcal{G}} \leftarrow \mathcal{M}_{k,q}(\boldsymbol{g})$
$\boldsymbol{c} \leftarrow [\boldsymbol{m}, \boldsymbol{0}_u] \cdot \boldsymbol{G}_{\mathcal{G}} + \mathrm{Tr}(\alpha \boldsymbol{k}_{\mathrm{pub}}) + \boldsymbol{e} \in \mathbb{F}_{q^m}^n$
return Ciphertext $\boldsymbol{c}$
```

Algorithm 28: $\mathsf{Decrypt}_{\mathrm{LIGA}}$

```
Input  : Ciphertext $\boldsymbol{c} \in \mathbb{F}_{q^m}^n$
         Private key $(\boldsymbol{x}, \boldsymbol{P}_{:,[w+1:n]}) \in \mathbb{F}_{q^{mu}}^k \times \mathbb{F}_q^{n \times (n-w)}$
Output: Plaintext $\boldsymbol{m} \in \mathbb{F}_{q^m}^{k-u}$
$\boldsymbol{c}' \leftarrow \boldsymbol{c}\boldsymbol{P}_{:,[w+1:n]}$
$\mathcal{G}' \leftarrow$ Gabidulin code generated by $\boldsymbol{G}_{\mathcal{G}}\boldsymbol{P}_{:,[w+1:n]}$
$\boldsymbol{m}' \leftarrow$ decode $\boldsymbol{c}'$ in $\mathcal{G}'$
$\alpha \leftarrow \sum_{i=k-u+1}^{k} m_i' x_i^*$
$\boldsymbol{m} \leftarrow (\boldsymbol{m}' - \mathrm{Tr}(\alpha \boldsymbol{x}))_{:,[1:k-u]} \in \mathbb{F}_{q^m}^{k-u}$
return Plaintext $\boldsymbol{m}$
```

whose last $n-w$ columns are given by

$$\boldsymbol{c}' = (\boldsymbol{m} + \mathrm{Tr}(\alpha \boldsymbol{x}))\boldsymbol{G}' + \boldsymbol{e}',$$

where $\boldsymbol{G}' := \boldsymbol{G}_{\mathcal{G}}\boldsymbol{P}_{:,[w+1:n]} \in \mathbb{F}_{q^m}^{k \times (n-w)}$ and $\boldsymbol{e}' := \boldsymbol{e}\boldsymbol{P}_{:,[w+1:n]}$. By decoding the vector $\boldsymbol{c}'$ in the Gabidulin code generated by $\boldsymbol{G}_{\mathcal{G}}\boldsymbol{P}_{:,[w+1:n]}$, we thus obtain the vector

$$\boldsymbol{m}' = \boldsymbol{m} + \mathrm{Tr}(\alpha \boldsymbol{x}).$$

Since the last u positions of the plaintext $\boldsymbol{m}$ are zero, we get $\alpha = \sum_{i=k-u+1}^{k} m_i' x_i^*$, where $\{x_{k-u+1}^*, \dots, x_k^*\}$ is a dual basis to $\{x_{k-u+1}, \dots, x_k\}$. As we know α and $\boldsymbol{x}$, we can compute the plaintext $\boldsymbol{m}$. ■

Remark 5.1. *Steps 1 to 3 of Algorithm 28 can be interpreted as an error-erasure decoder of a Gabidulin code. As this observation may have advantages, especially for* *implementations, we present this connection formally in Appendix C.2.*

A SageMath v8.8 [191] implementation of the public-key encryption version of LIGA can be downloaded from https://bitbucket.org/julianrenner/liga_pke. The purpose of this implementation is to clarify the shown algorithms but *not* to provide a secure and efficient instance. Developing an implementation that offers these two properties and can serve for a performance comparison with other schemes is outside the scope of this work and is left for future research.

5.2.3 The KEM Version $\Pi_{\mathrm{LIGA}}^{\mathrm{KEM}}$

In [192], generic transformations of IND-CPA-secure public-key encryption schemes into IND-CCA2-secure KEMs are proposed. In the following, we apply one of the transformations directly to $\Pi_{\mathrm{LIGA}}^{\mathrm{Enc}}$ to obtain $\Pi_{\mathrm{LIGA}}^{\mathrm{KEM}} = (\mathsf{KeyGen}_{\mathrm{LIGA}}, \mathsf{Encaps}_{\mathrm{LIGA}}, \mathsf{Decaps}_{\mathrm{LIGA}})$. Later, in Section 5.3.2, we prove that $\Pi_{\mathrm{LIGA}}^{\mathrm{Enc}}$ fulfills the requirements such that the applied transformation is secure.

In Algorithm 29 and Algorithm 30, we show the encapsulation and the decapsulation algorithms of $\Pi_{\mathrm{LIGA}}^{\mathrm{KEM}}$, where $\mathcal{L}$, $\mathcal{H}$, and $\mathcal{K}$ denote hash functions and $\mathcal{L} \neq \mathcal{H}$.[1] The algorithm $\mathsf{KeyGen}_{\mathrm{LIGA}}$ remains Algorithm 26.

Algorithm 29: $\mathsf{Encaps}_{\mathrm{LIGA}}$

```
Input  : Public key $(\boldsymbol{g}, \boldsymbol{k}_{\mathrm{pub}}) \in \mathbb{F}_{q^m}^n \times \mathbb{F}_{q^{mu}}^n$
Output: Ciphertext $(\boldsymbol{c}, \boldsymbol{d}) \in \mathbb{F}_{q^m}^n \times \mathbb{F}_2^{512}$
        Shared key $K$
$\boldsymbol{m} \xleftarrow{\$} \mathbb{F}_{q^m}^{k-u}$
$\theta \leftarrow \mathcal{L}(\boldsymbol{m})$
$\boldsymbol{c} \leftarrow \mathsf{Encrypt}_{\mathrm{LIGA}}(\boldsymbol{m}, (\boldsymbol{g}, \boldsymbol{k}_{\mathrm{pub}}), \theta)$
$K \leftarrow \mathcal{K}(\boldsymbol{m}, \boldsymbol{c})$
$\boldsymbol{d} \leftarrow \mathcal{H}(\boldsymbol{m})$
return Ciphertext $(\boldsymbol{c}, \boldsymbol{d})$, Shared key $K$
```

5.2.4 Running Time and Complexity

Timing Attacks

Resistance against timing attacks is essential in many applications, and systems that

[1] E.g., one can use SHA3-512 for $\mathcal{L}$ and SHA512 for $\mathcal{H}$ as proposed in [151].

Algorithm 30: $\mathsf{Decaps}_{\mathrm{LIGA}}$

```
Input  : Ciphertext $(\boldsymbol{c}, \boldsymbol{d}) \in \mathbb{F}_{q^m}^n \times \mathbb{F}_2^{512}$
         Private key $(\boldsymbol{x}, \boldsymbol{P}_{:,[w+1:n]}) \in \mathbb{F}_{q^{mu}}^k \times \mathbb{F}_q^{n\times(n-w)}$
         Public key $(\boldsymbol{g}, \boldsymbol{k}_{\mathrm{pub}}) \in \mathbb{F}_{q^m}^n \times \mathbb{F}_{q^{mu}}^n$
Output: Shared key $K$
$\boldsymbol{m}' \leftarrow \mathsf{Decrypt}_{\mathrm{LIGA}}(\boldsymbol{c}, (\boldsymbol{x}, \boldsymbol{P}_{:,[w+1:n]}))$
$\theta' \leftarrow \mathcal{L}(\boldsymbol{m}')$
$\boldsymbol{c}' \leftarrow \mathsf{Encrypt}_{\mathrm{LIGA}}(\boldsymbol{m}', (\boldsymbol{g}, \boldsymbol{k}_{\mathrm{pub}}), \theta')$
if $\boldsymbol{c} \neq \boldsymbol{c}' \vee \boldsymbol{d} \neq \mathcal{H}(\boldsymbol{m}')$ then
    $K \leftarrow \perp$
else
    $K \leftarrow \mathcal{K}(\boldsymbol{m}', \boldsymbol{c}')$
return Shared key $K$
```

do not enable a constant-time implementation are therefore considered as insecure. Due to the fact that Step 4 of Algorithm 27 can be easily implemented in constant time, the proposed encryption algorithm does not reveal any secret knowledge through timing attacks. The same holds for the presented decryption algorithm since there is an efficient constant-time decoding algorithm for Gabidulin codes [203], and all other steps of Algorithm 28 can be realized in constant time as well.

Asymptotically Fastest Methods

In some scenarios, a constant-time implementation of the system may not be required, but we want that the key generation, the encryption, and the decryption are as fast as possible. The following results were not known when the original FL system was proposed, but they could have an impact on its efficiency.

The complexity of the key generation and the encryption is dominated by the cost of encoding a Gabidulin code (Line 8 of Algorithm 26 and Line 4 of Algorithm 27).[2] The asymptotically fastest-known algorithms [97, 98, 204] for this require $O^{\sim}(n^{\min\{\frac{\omega+1}{2},1.635\}})$ operations in $\mathbb{F}_{q^m}$ or $O^{\sim}(n^{\omega-2}m^2)$ operations in $\mathbb{F}_q$ in general,[3] and $O^{\sim}(n)$ operations in $\mathbb{F}_{q^m}$ if the entries of $\boldsymbol{g}$ are a normal basis of $\mathbb{F}_{q^m}/\mathbb{F}_q$, where ω is

[2] Note that since $\boldsymbol{x}$ and $\boldsymbol{z}$ have coefficients in the large field $\mathbb{F}_{q^{mu}}$, this line can be realized as encoding u messages over $\mathbb{F}_{q^m}$ with the generator matrix $\boldsymbol{G}_{\mathcal{G}} \in \mathbb{F}_{q^m}^{k\times n}$ and corrupting these codewords with an error.

[3] Which of the two algorithms is the fastest depends on the relation between n and m, as well as the used basis of $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$.

the matrix multiplication exponent and $O^{\sim}$ means that log factors are neglected.

The bottleneck of the decryption is error-erasure decoding of a Gabidulin code (Line 3 of Algorithm 28, see also Appendix C.2), where the asymptotically fastest algorithm costs $O^{\sim}\left(n^{\min\{\frac{\omega+1}{2},1.635\}}\right)$ operations in $\mathbb{F}_{q^m}$ [97, 98] or $O^{\sim}(n^{\omega-2}m^2)$ operations in $\mathbb{F}_q$.

For small lengths n, the algorithms from [101, 205, 206], which have quadratic complexity over $\mathbb{F}_{q^m}$ (or cubic complexity over $\mathbb{F}_q$), could be faster than the mentioned algorithms due to smaller hidden constants in the O-notation. This concern is supported by the results shown in [207].

5.3 Difficult Problems & Semantic Security of LIGA

In this section, we introduce problems in the rank metric that are considered to be difficult. Furthermore, we prove that the public-key encryption version of LIGA is IND-CPA secure in the standard model and the KEM version is IND-CCA2 secure in the random oracle model under the assumption that no PPT algorithm can solve them. A detailed complexity analysis of existing and new algorithms solving the stated problems is given in Section 5.4.

5.3.1 Difficult Problems in the Rank Metric

LIGA is based on several difficult problems which are stated in this section. Note that the search variants of the problems correspond exactly to retrieving information about the private key from the public key (not necessarily a valid private key as explained in the following) or the plaintext from the ciphertext. The decisional problems are equivalent to distinguishing the public key or the ciphertext from random vectors.

Definition 5.1 (Restricted Interleaved Gabidulin Decoding (RIGab) Distribution).
Given: *$q, m, n, k, w > \lfloor \frac{n-k}{2} \rfloor, \zeta < \frac{w}{n-k-w}, u < w$.*
Sample uniformly at random

- *$\boldsymbol{G} \xleftarrow{\$} \mathcal{G}$, where $\mathcal{G}$ is the set of all generator matrices of $[n,k]_{\mathbb{F}_{q^m}}$ Gabidulin codes*
- *$\boldsymbol{M} \xleftarrow{\$} \{\boldsymbol{X} \in \mathbb{F}_{q^m}^{u \times k} : \mathrm{rk}_{q^m}(\boldsymbol{X}_{:,[k-u+1:k]}) = u\}$*
- *$\mathcal{A} \xleftarrow{\$} \{\text{subspace } \mathcal{U} \subseteq \mathbb{F}_{q^m}^{w} : \dim_{q^m} \mathcal{U} = \zeta, \mathcal{U} \text{ has a basis of full-}\mathbb{F}_q\text{-rank elements}\}$*
- *$\bar{\boldsymbol{Z}} \xleftarrow{\$} \left\{ \begin{bmatrix} \boldsymbol{s}'_1 \\ \vdots \\ \boldsymbol{s}'_u \end{bmatrix} \in \mathbb{F}_{q^m}^{u \times w} \;:\; \langle \boldsymbol{s}'_1, \ldots, \boldsymbol{s}'_u \rangle_{q^m} = \mathcal{A},\ \mathrm{rk}_q(\boldsymbol{s}'_i) = w\,, \forall\, i \in [1:u] \right\}$*

- $\boldsymbol{Q} \xleftarrow{\$} \{\boldsymbol{A} \in \mathbb{F}_q^{w \times n} : \mathrm{rk}_q(\boldsymbol{A}) = w\}$
- $\boldsymbol{Z} \leftarrow \bar{\boldsymbol{Z}}\boldsymbol{Q}$

Output: $(\boldsymbol{G}, \boldsymbol{M}\boldsymbol{G} + \boldsymbol{Z})$.

Problem 5.1 (Decisional Restricted Interleaved Gabidulin Decoding (DecRIGab)).

Given: $(\boldsymbol{G}, \boldsymbol{Y}) \in \mathbb{F}_{q^m}^{k \times n} \times \mathbb{F}_{q^m}^{u \times n}$, *where* $\boldsymbol{G}$ *is a generator matrix of a Gabidulin code.*

Objective: *Decide with non-negligible advantage whether* $\boldsymbol{Y}$ *came from the* RIGab *distribution with input* q, m, n, k, w, ζ, u *or the uniform distribution over the set of matrices* $\mathbb{F}_{q^m}^{u \times n}$.

To solve DecRIGab, we do not know a better approach than trying to solve the associated *search* problem SeaRIGab, which is usually done for all decoding-based problems.

Problem 5.2 (Search Restricted Interleaved Gabidulin Decoding (SeaRIGab)).

Given: $(\boldsymbol{G}, \boldsymbol{Y})$ *from the* RIGab *distribution with input* q, m, n, k, w, ζ, u.

Objective: *Find* $\boldsymbol{M}' \in \mathbb{F}_{q^m}^{u \times k}$ *and* $\boldsymbol{Z}' \in \mathbb{F}_{q^m}^{u \times n}$ *s.t.* $\mathrm{rk}_q(\boldsymbol{Z}') \leq w$ *and* $\boldsymbol{M}'\boldsymbol{G} + \boldsymbol{Z}' = \boldsymbol{Y}$.

The problem SeaRIGab is equivalent to decoding a codeword of a u-interleaved Gabidulin code that is corrupted by an error and is the underlying problem of the structural attacks from Section 5.1.3.

Note however that not necessarily every solution of this problem can be used directly as a valid private key since some additional structure on $\boldsymbol{M}$ is introduced in LIGA, i.e., Problem 5.2 is easier to solve than retrieving a valid private key of LIGA.

Definition 5.2 (Restricted Error (ResErr) Distribution).

Given: $q, m, n, k, w, t_{\mathrm{pub}}, u, (\boldsymbol{G}, \boldsymbol{K})$ *from the* RIGab *distribution.*

Sample uniformly at random

- $\boldsymbol{e} \xleftarrow{\$} \{\boldsymbol{x} \in \mathbb{F}_{q^m}^n : \mathrm{rk}_q(\boldsymbol{x}) = t_{\mathrm{pub}}\}$
- $\alpha \xleftarrow{\$} \mathbb{F}_{q^{mu}}$
- $\boldsymbol{k} \leftarrow \mathrm{ext}_{q^{mu}/q^m}^{-1}(\boldsymbol{K})$
- $\boldsymbol{y} \leftarrow \mathrm{Tr}(\alpha \boldsymbol{k}) + \boldsymbol{e} = \mathrm{Tr}(\alpha \boldsymbol{m})\boldsymbol{G} + \mathrm{Tr}(\alpha \boldsymbol{z}) + \boldsymbol{e}$

Output: $\boldsymbol{y}$.

Problem 5.3 (Decisional Restricted Gabidulin Decoding (DecRGab)).

Given: $q, m, n, k, w, t_{\mathrm{pub}}, u, (\boldsymbol{G}, \boldsymbol{K})$ *from the* RIGab *distribution,* $\boldsymbol{y} \in \mathbb{F}_{q^m}^n$.

Objective: *Decide with non-negligible advantage whether* $\boldsymbol{y}$ *came from the* ResErr *distribution with input* $q, m, n, k, w, t_{\mathrm{pub}}, u, (\boldsymbol{G}, \boldsymbol{K})$ *or the uniform distribution over the set of vectors* $\mathbb{F}_{q^m}^n$.

As before, we are not aware of a faster approach to solve DecRGab than through the solution of the associated *search* problem.

Problem 5.4 (Search Restricted Gabidulin Decoding (SeaRGab)).
Given: $q, m, n, k, w, t_{\mathrm{pub}}, u, (\boldsymbol{G}, \boldsymbol{K})$ *from the* RIGab *distribution,* $\boldsymbol{y}$ *from the* ResErr *distribution with input* $(\boldsymbol{G}, \boldsymbol{K})$.
Objective: *Find* $\boldsymbol{m}' \in \mathbb{F}_{q^m}^k$ *and* $\boldsymbol{e}' \in \{\boldsymbol{x} \in \mathbb{F}_{q^m}^n : \mathrm{rk}_q(\boldsymbol{x}) \leq t_{\mathrm{pub}}\}$ *such that* $\boldsymbol{m}'\boldsymbol{G} + \boldsymbol{e}' = \boldsymbol{y}$.

Problem 5.4 is equivalent to decoding a codeword of a Gabidulin code that is corrupted by an error that has with high probability a rank weight greater than $(n-k)/2$, see Appendix C.3.

We see in the following that LIGA is IND-CCA2 secure under the assumption that the problem DecRGab is difficult. As mentioned, there is an obvious reduction of DecRGab to SeaRGab, which can again be efficiently reduced to SeaRIGab. In fact, all relevant attacks studied in Section 5.4 make use of this chain of reduction and aim at solving one of the two search problems.

We are not aware of a reduction of DecRIGab to SeaRIGab or to one of the other problems. Hence, it could be that DecRIGab is significantly easier than the other problems. In Section 5.4.3, we show that there is a distinguisher for DecRIGab that is efficiently computable if the system parameter ζ is chosen too small. Due to the missing reduction, it is not clear whether or not this distinguisher influences the security of the system.

5.3.2 Semantic Security

In this section, we prove that the public-key encryption system $\Pi_{\mathrm{LIGA}}^{\mathrm{Enc}}$ is semantically secure against chosen-plaintext attacks in the standard model under the assumption that DecRGab (Problem 5.3) is difficult. In addition, we show that the IND-CCA2 security of $\Pi_{\mathrm{LIGA}}^{\mathrm{KEM}}$ reduces tightly to the IND-CPA security of $\Pi_{\mathrm{LIGA}}^{\mathrm{Enc}}$ in the random oracle model.

IND-CPA Security of $\Pi_{\mathrm{LIGA}}^{\mathrm{Enc}}$

To show that $\Pi_{\mathrm{LIGA}}^{\mathrm{Enc}}$ is secure against chosen-plaintext attacks, we use the definition of admissibility as in [208].

Definition 5.3 (Admissibility [208])**.** *The public-key encryption scheme* $\Pi^{\text{Enc}}_{\text{LIGA}}$*with a message space* $\mathcal{M}$ *and a random space* $\mathcal{R}$ *is called admissible if there is a pair of deterministic polynomial-time algorithms* $\mathsf{Encrypt}_1$ *and* $\mathsf{Encrypt}_2$ *satisfying the following property:*

- *Partible:* $\mathsf{Encrypt}_1$ *takes as input a public key* pk *and* $r \in \mathcal{R}$*, and outputs a* $p(\lambda)$ *bit-string, where* λ *is the security parameter.* $\mathsf{Encrypt}_2$ *takes as input* pk *and* $\boldsymbol{m} \in \mathcal{M}$ *and outputs a* $p(\lambda)$ *bit-string. Here* p *is some polynomial in the security parameter* λ*. Then, for any* pk *given by* $\mathsf{KeyGen}_{\text{LIGA}}$*,* $r \in \mathcal{R}$*, and* $\boldsymbol{m} \in \mathcal{M}$*, it must hold that* $\mathsf{Encrypt}_1(\mathsf{pk}, r) + \mathsf{Encrypt}_2(\mathsf{pk}, \boldsymbol{m}) = \mathsf{Encrypt}_{\text{LIGA}}(\mathsf{pk}, \boldsymbol{m}, r)$*.*

- *Pseudorandomness: Let* D *be a probabilistic algorithm and let*

$$\begin{aligned}\mathsf{Adv}^{\text{IND}}_{D,\mathsf{Encrypt}_1}(\lambda) =& \Pr\Big(D(\mathsf{pk}, \mathsf{Encrypt}_1(\mathsf{pk}, r)) = 1 \,|\, r \xleftarrow{\$} \mathcal{R}, (\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{KeyGen}_{\text{LIGA}}\left(1^\lambda\right)\Big) \\ &- \Pr\Big(D(\mathsf{pk}, s) = 1 \,|\, s \xleftarrow{\$} \mathcal{U}_{p(\lambda)}, (\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{KeyGen}_{\text{LIGA}}\left(1^\lambda\right)\Big),\end{aligned}$$

where $\mathsf{KeyGen}_{\text{LIGA}}\left(1^\lambda\right)$ *indicates that the key-generation algorithm of* LIGA *is run with parameters that are chosen according to the security level* λ*. Furthermore, we define the advantage function of the problem as follows. For any time-complexity* t*,*

$$\mathsf{Adv}^{\text{IND}}_{\mathsf{Encrypt}_1}(\lambda, t) = \max_{D} \left\{\mathsf{Adv}^{\text{IND}}_{D,\mathsf{Encrypt}_1}(\lambda)\right\},$$

where the maximum is taken over all D*. If* $\mathsf{Adv}^{\text{IND}}_{\mathsf{Encrypt}_1}(\lambda, t)$ *is negligible for every polynomial bounded by* t *and every sufficiently large* λ*, then* $\mathsf{Encrypt}_1$ *fulfills the properties of pseudorandomness.*

In the following we prove that $\Pi^{\text{Enc}}_{\text{LIGA}}$ is IND-CPA secure by showing that it fulfills the definition of admissibility.

Theorem 5.6. *The system* $\Pi^{\text{Enc}}_{\text{LIGA}} = (\mathsf{KeyGen}_{\text{LIGA}}, \mathsf{Encrypt}_{\text{LIGA}}, \mathsf{Decrypt}_{\text{LIGA}})$ *is an IND-CPA-secure encryption scheme in the standard model under the assumption that* $\mathsf{DecRGab}$ *is a difficult problem.*

Proof. Let $\mathsf{Encrypt}_1 := \mathrm{Tr}(\alpha \boldsymbol{k}_{\text{pub}}) + \boldsymbol{e}$ and $\mathsf{Encrypt}_2 := \boldsymbol{m}\boldsymbol{G}_{\mathcal{G}}$. Then, one observes that $\mathsf{Encrypt}_{\text{LIGA}} = \mathsf{Encrypt}_1 + \mathsf{Encrypt}_2$, and therefore, $\Pi^{\text{Enc}}_{\text{LIGA}}$ is *partible.* Since $\mathsf{DecRGab}$ (Problem 5.3) is assumed to be difficult, the encryption scheme fulfills *pseudorandomness*, and it follows that the system is *admissibile.* As proven in [208, Lemma 1], if $\Pi^{\text{Enc}}_{\text{LIGA}}$ fulfills Definition 5.3, then it is an IND-CPA-secure encryption scheme. ■

IND-CCA2 Security of $\Pi_{\mathrm{LIGA}}^{\mathrm{KEM}}$

We use a transformation proposed in [192] to transform the public-key encryption scheme $\Pi_{\mathrm{LIGA}}^{\mathrm{Enc}}$ into the KEM $\Pi_{\mathrm{LIGA}}^{\mathrm{KEM}}$. In the following, we prove that $\Pi_{\mathrm{LIGA}}^{\mathrm{KEM}}$ is IND-CCA2 secure in the random oracle model.

The applied transformation requires that the encryption scheme $\Pi_{\mathrm{LIGA}}^{\mathrm{Enc}}$ is γ-spread, which is proven in the following.

Definition 5.4 (γ-spread, [192, 209])**.** *For valid* $(\mathsf{sk}, \mathsf{pk})$ *and* $\boldsymbol{m}$*, the* min-entropy *function of* $\mathsf{Encrypt}_{\mathrm{LIGA}}$ *is defined by*

$$f_{\mathrm{ent}}(\boldsymbol{m}, \mathsf{pk}) := -\log_2 \max_{\boldsymbol{c}\in\hat{\mathcal{C}}_{\mathsf{J}}} \Pr(\boldsymbol{c} = \mathsf{Encrypt}_{\mathrm{LIGA}}(\boldsymbol{m}, \mathsf{pk}, r)),$$

where the randomness is in $r \in \mathcal{R}$*, and* $\hat{\mathcal{C}}_{\mathsf{J}}$ *is the set of possible ciphertexts. A public-key encryption scheme is called* γ-spread *if for every valid key pair* $(\mathsf{pk}, \mathsf{sk})$ *and every message* $\boldsymbol{m} \in \mathcal{M}$*, it holds that* $f_{\mathrm{ent}}(\boldsymbol{m}, \mathsf{pk}) \geq \gamma$*. It follows that for all* $\boldsymbol{c} \in \hat{\mathcal{C}}_{\mathsf{J}}$*, the inequality*

$$\Pr(\boldsymbol{c} = \mathsf{Encrypt}_{\mathrm{LIGA}}(\boldsymbol{m}, \mathsf{pk}, r)) \leq 2^{-\gamma}$$

must be fulfilled.

Lemma 5.7. *The public-key encryption system* $\Pi_{\mathrm{LIGA}}^{\mathrm{Enc}}$ *is* γ*-spread, where* $\gamma = m(t_{\mathrm{pub}} - u) + t_{\mathrm{pub}}(n - t_{\mathrm{pub}} - 1)$*.*

Proof. We observe that

$$\begin{aligned}
\max_{\boldsymbol{c}\in\hat{\mathcal{C}}_{\mathsf{J}}} \Pr(\boldsymbol{c} = \mathsf{Encrypt}_{\mathrm{LIGA}}(\boldsymbol{m}, \mathsf{pk}, r)) &= \max_{\boldsymbol{c}\in\hat{\mathcal{C}}_{\mathsf{J}}} \Pr(\boldsymbol{c} = [\boldsymbol{m}, \boldsymbol{0}_u]\boldsymbol{G}_{\mathcal{G}} + \mathrm{Tr}(\alpha \boldsymbol{k}_{\mathrm{pub}}) + \boldsymbol{e}) \\
&\overset{\text{(i)}}{\leq} \max_{\boldsymbol{c}'\in\hat{\mathcal{C}}'_{\mathsf{J}}} q^{mu} \Pr(\boldsymbol{c}' = [\boldsymbol{m}, \boldsymbol{0}_u]\boldsymbol{G}_{\mathcal{G}} + \boldsymbol{e}) \\
&= q^{mu} \frac{1}{|\{\boldsymbol{e} \in \mathbb{F}_{q^m}^n : \mathrm{rk}_q(\boldsymbol{e}) = t_{\mathrm{pub}}\}|},
\end{aligned}$$

where $\hat{\mathcal{C}}'_{\mathsf{J}}$ is the set of all vectors in rank distance t_{pub} from $[\boldsymbol{m}, \boldsymbol{0}_u]\boldsymbol{G}_{\mathcal{G}}$, and (i) follows from the fact that there are at most q^{mu} choices for α. In [35, Section IV.B], a constructive way of obtaining rank-t_{pub} matrices is given. More precisely, an injective mapping $\varphi : \mathbb{F}_q^{t(n+m-t-1)} \to \{\boldsymbol{A} \in \mathbb{F}_q^{m\times n} : \mathrm{rk}_q(\boldsymbol{A}) = t\}$ is given. Hence, we have

$$|\{\boldsymbol{e} \in \mathbb{F}_{q^m}^n : \mathrm{rk}_q(\boldsymbol{e}) = t_{\mathrm{pub}}\}| \geq q^{t_{\mathrm{pub}}(n+m-t_{\mathrm{pub}}-1)},$$

and it follows that

$$\begin{aligned}\frac{q^{mu}}{|\{\boldsymbol{e} \in \mathbb{F}_{q^m}^n : \mathrm{rk}_q(\boldsymbol{e}) = t_{\mathrm{pub}}\}|} &\leq \frac{q^{mu}}{q^{t_{\mathrm{pub}}(n+m-t_{\mathrm{pub}}-1)}} \\ &= q^{-m(t_{\mathrm{pub}}-u)-t_{\mathrm{pub}}(n-t_{\mathrm{pub}}-1)} \\ &\leq q^{-\gamma}. \end{aligned}$$ ■

Theorem 5.8. *The KEM scheme* $\Pi_{\mathrm{LIGA}}^{\mathrm{KEM}} = (\mathsf{KeyGen}_{\mathrm{LIGA}}, \mathsf{Encaps}_{\mathrm{LIGA}}, \mathsf{Decaps}_{\mathrm{LIGA}})$ *is IND-CCA2 secure in the random oracle model under the assumption that the Problem* DecRGab *problem is difficult.*

Proof. Assuming the DecRGab is difficult, the encryption $\Pi_{\mathrm{LIGA}}^{\mathrm{Enc}}$ is IND-CPA secure, see Theorem 5.6. Furthermore, it is proven in Lemma 5.7 that $\Pi_{\mathrm{LIGA}}^{\mathrm{Enc}}$ has γ-spread encryptions. Thus, the system $\Pi_{\mathrm{LIGA}}^{\mathrm{KEM}}$ can be tightly reduced to $\Pi_{\mathrm{LIGA}}^{\mathrm{KEM}}$ in the random oracle model as shown in [192]. ■

5.4 Security Analysis of LIGA

In this section, we analyze the security of LIGA. As proven in Theorem 5.6 and 5.8, the encryption version is IND-CPA secure and the KEM version is IND-CCA2 secure under the assumption that DecRGab is difficult. Since there are obvious reductions from DecRGab to SeaRGab and from DecRGab to SeaRIGab, we study the hardness of these two search problems in this section (Section 5.4.1 for SeaRIGab and Section 5.4.2 for SeaRGab). In fact, we are not aware of a more efficient method to solve DecRGab than through these two search problems.

Although no formal reduction from any of the other three studied problems to DecRIGab is known, we study also the hardness of DecRIGab (Section 5.4.3). We derive a distinguisher for the public key with exponential complexity in the system parameters, which can be avoided by proper parameter choice.

Due to the nature of the encryption, there are public keys for which the probability that the work factor of some ciphertext attacks is below the designed minimal work factor is greater than $2^{-\lambda}$. We show in Section 5.4.4 that these weak keys occur with negligible probability, i.e., smaller or equal to $2^{-\lambda}$, during the random key generation if the parameters are chosen in a suitable way.

5.4.1 Exponential-Time Attacks on SeaRIGab

We propose new and summarize known methods that solve SeaRIGab (Problem 5.2). All studied algorithms have exponential complexity in the code parameters.

Recall that in the decryption algorithm of LIGA, the last u positions of the private key $\boldsymbol{x}$ have to be a basis of $\mathbb{F}_{q^{mu}}$ over $\mathbb{F}_{q^m}$. Therefore, not every solution of SeaRIGab can be used as valid private key, and thus, it is a strictly easier problem than retrieving a valid private key corresponding to a given public key.

Brute-Force the Vector $\boldsymbol{z}$ Attack

The number of vectors $\boldsymbol{z} \in \mathbb{F}_{q^{mu}}^n$ that fulfill the conditions stated in Section 5.2.1 is equal to the number of possible vectors $\boldsymbol{s} \in \mathbb{F}_{q^{mu}}^w$ times the number of full rank matrices in $\mathbb{F}_{q^m}^{w \times n}$ in reduced row echelon form. Formally, the number of vectors $\boldsymbol{z}$ is

$$\underbrace{|\{\boldsymbol{z} \, : \, \boldsymbol{z} \text{ can occur in Alg. 26}\}|}_{\geq 1} \cdot \underbrace{|\{\boldsymbol{P} \, : \, \boldsymbol{P} \text{ can occur in Alg. 26}\}|}_{\geq \begin{bmatrix} n \\ w \end{bmatrix}_q} \geq \begin{bmatrix} n \\ w \end{bmatrix}_q .$$

Thus, brute-forcing a vector $\boldsymbol{z}$ that is a solution to SeaRIGab has work factor

$$W_{\boldsymbol{z}} \geq \frac{\begin{bmatrix} n \\ w \end{bmatrix}_q}{N'_{\mathrm{R}}} \geq \frac{q^{w(n-w)}}{N'_{\mathrm{R}}},$$

where the latter inequality follows from a lower bound on Gaussian binomial coefficients [68, Lem. 4], and

$$N'_{\mathrm{R}} := \max \left\{ \frac{\sum_{i=0}^{w} \left[\prod_{j=0}^{i-1} (q^{mu} - q^j) \right] \begin{bmatrix} n \\ i \end{bmatrix}_q}{q^{mu(n-k)}}, \, 1 \right\} \tag{5.4}$$

is the average number of interleaved codewords in a ball of radius w around a uniformly at random chosen interleaved received word.

Interleaved Decoding Attack

As described in Section 5.1.3, an attacker can apply an interleaved decoder on $\boldsymbol{k}_{\text{pub}}$ to retrieve an alternative private key. A major ingredient of LIGA is that the public key is chosen in a way that this decoding always fails, i.e., the corresponding linear system of equations does not have a unique solution. However, it is still possible to brute-force search in the solution space of the involved system of equations. This is analyzed in the following. Notice thereby that *any* interleaved codeword in radius at most w is a solution to SeaRIGab.

Problem 5.2 (SeaRIGab) is equivalent to decoding a codeword of a u-interleaved Gabidulin code that is corrupted by an error $\boldsymbol{E}$. This error $\boldsymbol{E}$ fulfills

$$\left\lfloor \frac{n-k}{2} \right\rfloor < \operatorname{rk}_q(\boldsymbol{E}) \leq \frac{u}{u+1}(n-k) \quad \text{and} \quad \operatorname{rk}_{q^m}(\boldsymbol{E}) < \frac{w}{n-k-w} < w,$$

and therefore, no known algorithm is able to correct it efficiently.

The crucial point of the interleaved decoding algorithms from [105, 115] is solving a linear system of equations based on the syndromes with $w+1$ unknowns and φ linearly independent equations which is equivalent to finding the kernel of the matrix in (5.3), cf. [99, Section 4.1]. For $\zeta \geq \frac{w}{n-k-w}$, the dimension of the solution space is one and all solutions are valid for the remaining decoding steps. For $\zeta < \frac{w}{n-k-w}$, the dimension of the solution space is $w+1-\varphi$ but each valid solution forms only a one-dimensional subspace. An attacker can therefore search in the solution space for a valid solution which requires on average

$$\frac{(q^m)^{w+1-\varphi}}{q^m \cdot N'_{\mathrm{R}}} = \frac{q^{m(w-\varphi)}}{N'_{\mathrm{R}}}$$

trials, where N'_{R} is the average number of interleaved codewords, see (5.4). In this case, the search through the solution space has a work factor of

$$W_{\mathrm{ILD}} = \frac{q^{m(w-\zeta(n-k-w))}}{N'_{\mathrm{R}}}.$$

Since the size of the solution space is maximal for $\varphi = n-k-w$, the repair from Section 5.2.1 with the explicit parameter value $\zeta = \dim_{q^m}\left(\langle \boldsymbol{z}_1, \ldots, \boldsymbol{z}_u \rangle_{q^m}\right) = 1$ is the most secure choice *in this sense*. However, we keep the choice of ζ flexible, as the pair-wise linear dependence of $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_u$ could decrease the security.

Besides the syndrome-based interleaved decoding algorithms in [99, 105, 115], there is an interpolation-based decoding algorithm [99, Section 4.3 (page 72)]. The latter algorithm can be interpreted both as a list decoder of interleaved Gabidulin codes with exponential worst-case and average list size or as a probabilistic unique decoder. The probabilistic unique interpolation-based decoder fails if and only if the decoding algorithms in [99, 105, 115] fail, and therefore, the previous analysis applies here as well. For the list decoder, cf. [99, Lemma 4.5], the work factor of the resulting attack is

$$W_{\text{list, public key}} \leq \frac{q^{m(u-1)k}}{N'_{\mathrm{R}}}.$$

Notice that the list of size $q^{m(u-1)k}$ contains many words which are not valid codewords, but we have to go through the whole list to find all valid codewords within radius w.

List Decoding of the Public Key Attack

Recall that $\boldsymbol{k}_{\text{pub}} = \boldsymbol{x} \cdot \boldsymbol{G}_{\mathcal{G}} + \boldsymbol{z}$. Previously, we have explained why this vector is a corrupted version of a codeword of a u-interleaved Gabidulin code. At the same time, $\boldsymbol{x} \cdot \boldsymbol{G}_{\mathcal{G}}$ can be seen as a short Gabidulin code over a large field $\mathbb{F}_{q^{mu}}$, and therefore, one could apply a list decoding algorithm, if one exists, to decode $\boldsymbol{k}_{\text{pub}}$ and obtain $\boldsymbol{x}$. The weight of the error $\boldsymbol{z}$ is larger than the unique decoding radius, and therefore, a unique decoder cannot be applied to reconstruct $\boldsymbol{x}$ and a list decoder for radius w is required.

However, such an algorithm has not been found yet. It was even shown in [175–177] that for most classes of Gabidulin codes such a polynomial-time list decoding algorithm cannot exist. Note that these results were not known when the original FL cryptosystem was proposed. These results also imply that there is no polynomial-time list decoding algorithm for arbitrary Gabidulin codes beyond the unique decoding radius (such as the Guruswami–Sudan algorithm for RS codes).

Randomized Gabidulin Decoding Attack on the Public Key

The public key can be seen as the sum of a Gabidulin codeword over the field $\mathbb{F}_{q^{mu}}$ and an error of weight $w > \frac{n-k}{2}$. Alternatively, as shown in Section 5.1.2, the public key can be seen as an interleaved Gabidulin codeword that is corrupted by an error of weight $w = \xi + \frac{n-k}{2}$, where $\xi > 0$. Each row of (5.1) is a codeword of a Gabidulin code over $\mathbb{F}_{q^m}$ that is corrupted by an error of rank weight w. Both the corrupted Gabidulin codeword over $\mathbb{F}_{q^{mu}}$ as well as over $\mathbb{F}_{q^m}$ can be decoded using the randomized decoding approach proposed in [158]. Since applying the attack on each row of the unfolded public key is more efficient, we conclude that the randomized Gabidulin decoding attack on the public key has an average complexity of

$$W_{\mathrm{RGD}} = \frac{n}{64} \cdot q^{m(n-k)-w(n+m)+w^2+\min\{2\xi(\frac{n+k}{2}-\xi),wk\}}$$

over $\mathbb{F}_{q^m}$.

Moving to Another Close Error Attack

The following attack was suggested by Rosenkilde [210]. It tries to move the vector $\boldsymbol{z}$ to a close vector of the same or smaller rank weight w for which the interleaved decoder for $\boldsymbol{k}_{\mathrm{pub}}$ does not fail.

The idea is to find a vector $\boldsymbol{y} \in \mathbb{F}_{q^m}^{u \times n}$ such that $\boldsymbol{z}' := \boldsymbol{z} + \boldsymbol{y}$ still has rank weight $\mathrm{rk}_q(\boldsymbol{z}') \leq w$ and that the rank of the matrix from (5.3) over $\mathbb{F}_{q^m}$ is at least w. To guarantee the first condition, we want to construct $\boldsymbol{y}$ such that its extended $um \times n$ matrix over $\mathbb{F}_q$ has a row space $\hat{\mathcal{R}}$ that is contained in the one of $\boldsymbol{z}$. Since the matrix (5.3) has rank $\varphi \leq \zeta(n-k-w)$ for the original error $\boldsymbol{z}$, the space $\hat{\mathcal{R}}$ must have at least $\mathbb{F}_q$-dimension $w - \varphi \geq w(\zeta+1) - \zeta(n-k)$. By choosing a random $\hat{\mathcal{R}}$ with this property and taking a random vector $\boldsymbol{y}$ whose extended matrix has $\mathbb{F}_q$-row space equal to $\hat{\mathcal{R}}$, the second condition is fulfilled with high probability.

The complexity of the attack is hence dominated by the complexity of finding a subspace $\hat{\mathcal{R}} \subseteq \mathbb{F}_q^n$ of dimension $w - \varphi$ that is contained in the w-dimensional row rank support of $\boldsymbol{z}$. Since this is unknown, we can find it in a Las-Vegas fashion by repeatedly drawing a subspace uniformly at random. The expected number of iterations until we find a suitable row space is one over the probability that a random $(w-\varphi)$-dimensional subspace of $\mathbb{F}_q^n$ is contained in a given w-dimensional subspace, which is (cf. [171, Proof of Lemma 7])

$$\frac{\begin{bmatrix} w \\ w-\varphi \end{bmatrix}_q}{\begin{bmatrix} n \\ w-\varphi \end{bmatrix}_q} \approx \frac{q^{\varphi(w-\varphi)}}{q^{(n-w+\varphi)(w-\varphi)}} = q^{-(n-w)(w-\varphi)} \leq q^{-(n-w)(w(\zeta+1)-\zeta(n-k))}.$$

Hence, the attack has work factor

$$W_{\mathrm{MCE}} = q^{(n-w)(w-\varphi)} \geq q^{(n-w)(w(\zeta+1)-\zeta(n-k))}.$$

5.4.2 Exponential-Time Attacks on SeaRGab

Retrieving information about the plaintext from the ciphertext and the public key is equal to solving SeaRGab (Problem 5.3). In this section, we summarize methods to solve this problem.

Randomized Gabidulin Decoding Attack on the Ciphertext

Each ciphertext of LIGA can be seen as a Gabidulin codeword over $\mathbb{F}_{q^m}$ plus an error:

$$\begin{aligned} \boldsymbol{c} &= [\boldsymbol{m}, \boldsymbol{0}_u] \cdot \boldsymbol{G}_{\mathcal{G}} + \mathrm{Tr}(\alpha \boldsymbol{k}_{\mathrm{pub}}) + \boldsymbol{e} \\ &= \underbrace{([\boldsymbol{m}, \boldsymbol{0}_u] + \mathrm{Tr}(\alpha \boldsymbol{x})) \cdot \boldsymbol{G}_{\mathcal{G}}}_{\text{codeword}} + \underbrace{\mathrm{Tr}(\alpha \boldsymbol{z}) + \boldsymbol{e}}_{\text{error}}. \end{aligned}$$

Let $\tilde{w} := \mathrm{rk}_q(\mathrm{Tr}(\alpha \boldsymbol{z}) + \boldsymbol{e})$. Then, we can use the decoding algorithm proposed in [158], which requires, on average, at least

$$\frac{n}{64} \cdot q^{m(n-k)-\tilde{w}(n+m)+\tilde{w}^2+\min\{2\xi(\frac{n+k}{2}-\xi),\tilde{w}k\}} \tag{5.5}$$

operations in $\mathbb{F}_{q^m}$.

Clearly, the complexity of the algorithm strongly depends on the value $\tilde{w}$, which in turn depends on the generated keys. In general, $\tilde{w} = w + t_{\mathrm{pub}}$, but for some choices of $\boldsymbol{z}$, α, and $\boldsymbol{e}$, the rank $\tilde{w}$ is smaller. For this issue, we study the probability that $\tilde{w}$ is small, both for randomness in the encryption (random choice of α and $\boldsymbol{e}$) and the key generation (random choice of $\boldsymbol{z}$), in Section 5.4.4 and Appendix C.3. Some extremely rarely occurring keys thereby result in relatively high probabilities that $\tilde{w}$ is small.

However, we can choose the system parameters such that both the probability of a weak key as well as the conditional probability that $\tilde{w} < w$, given a non-weak key, is below $2^{-\lambda}$. Hence, with overwhelming probability, a random key and ciphertext result in a ciphertext error of rank weight $\tilde{w} \geq w$, and the work factor of this attack is at least as large as the “Randomized Gabidulin Decoding Attack on the Public Key” in Section 5.4.1.

List Decoding of the Ciphertext Attack

As described above, the ciphertext of LIGA is a codeword of a Gabidulin code, corrupted by an error of rank weight $\tilde{w}$. Hence, an attacker can try to decode the ciphertext directly. Since $\tilde{w}$ is always greater than the unique decoding radius $\left\lfloor \frac{n-k}{2} \right\rfloor$ of the Gabidulin code, this would require the existence of an efficient (list) decoding algorithm up to radius $\tilde{w}$. As explained previously, there is no such algorithm and bounds on the list size prove that there cannot exist a generic list decoding algorithm for all Gabidulin codes, which indicates that list decoding is a difficult problem.

However, to be secure, we have considered list decoding as follows for the security level of our system. The list size $\mathcal{L}_{\boldsymbol{c},\text{worst}}$ denotes a lower bound on the *worst-case* work factor of list decoding. For example, for a Gabidulin code with parameters $n \mid m$ and $\gcd(n, n-\tilde{w}) \geq 2$, there is a received word such that there are at least

$$\mathcal{L}_{\boldsymbol{c},\text{worst}} \geq \max \left\{ \frac{\begin{bmatrix} n/g \\ (n-\tilde{w})/g \end{bmatrix}_{q^g}}{q^{n(\tilde{w}/g-1)}} \; : \; g \geq 2, \; g \mid \gcd(n, n-\tilde{w}) \right\} \tag{5.6}$$

codewords in rank distance at most $\tilde{w}$ to it.

Although $\mathcal{L}_{\boldsymbol{c},\text{worst}}$ does not imply any statement about the average list size/average work factor, it provides an estimate of the order of magnitude of the work factor of a hypothetical list decoding attack. For our suggested parameters, we have ensured that the value of $\mathcal{L}_{\boldsymbol{c},\text{worst}}$ is sufficiently large in the proposed sets of parameters in Section 5.5.

Combinatorial Rank Syndrome Decoding Attack

The ciphertext $\boldsymbol{c} \in \mathbb{F}_{q^m}^n$ can be interpreted as a corrupted codeword from an $[n,k]_{\mathbb{F}_{q^m}}$ code generated by the matrix

$$\boldsymbol{G}_{\text{RSD}} := \begin{bmatrix} \mathcal{M}_{k-u,q}(\boldsymbol{g}) \\ \text{Tr}(\gamma_1 \boldsymbol{k}_{\text{pub}}) \\ \vdots \\ \text{Tr}(\gamma_u \boldsymbol{k}_{\text{pub}}) \end{bmatrix},$$

see [62]. This can be transformed into an instance of $\mathsf{SeaSD}_{\mathrm{R}}$, where we choose $\boldsymbol{G}_{\mathrm{RSD}}^{\perp}$ as the parity-check matrix, $\boldsymbol{c}(\boldsymbol{G}_{\mathrm{RSD}}^{\perp})^{\top}$ as the syndrome vector, and t_{pub} as the non-negative integer. Thus, the ciphertext can be decoded with the combinatorial syndrome decoding attack from [140] whose complexity is in the order of

$$W_{\mathrm{CRSD}} = n^3 m^3 q^{t_{\mathrm{pub}} \lceil \frac{(k+1)m}{n} \rceil - m}.$$

Algebraic Rank Syndrome Decoding Attack

As described above, the $\mathsf{SeaRGab}$ problem can be solved by decoding an error of rank weight t_{pub} in an $[n,k]_{\mathbb{F}_{q^m}}$ code. Beside the combinatorial approach, there exist algebraic algorithms to solve the Problem. The complexity of this approach is given in Appendix A.1. For completeness, we also state the work factor of this approach in the following.

In [141], the $\mathsf{SeaSD}_{\mathrm{R}}$ problem is expressed as a multivariate polynomial system and is solved by computing a Gröbner basis. In case there is a unique solution to the system, then the work factor of the algorithm is

$$W_{\mathrm{Gr}} = \begin{cases} \left[\frac{((m+n)t_{\mathrm{pub}})^{t_{\mathrm{pub}}}}{t_{\mathrm{pub}}!} \right]^{\mu}, & \text{if } m\binom{n-k-1}{t_{\mathrm{pub}}} \leq \binom{n}{t_{\mathrm{pub}}}, \\ \left[\frac{((m+n)t_{\mathrm{pub}})^{t_{\mathrm{pub}}+1}}{(t_{\mathrm{pub}}+1)!} \right]^{\mu}, & \text{otherwise,} \end{cases}$$

where μ is the exponent in the complexity expression of the used matrix multiplication algorithm. Like the authors of [141], we use $\mu = 2.807$ to compute the work factors since it corresponds to Strassen's algorithm, which is in practice the fastest algorithm for large matrix sizes.

Recently, a new algebraic algorithm was proposed to solve the $\mathsf{SeaSD}_{\mathrm{R}}$ problem [142]. It divides the problem instances into two categories. If

$$m\binom{n-k-1}{t_{\mathrm{pub}}} \geq \binom{n}{t_{\mathrm{pub}}} - 1,$$

we are in the overdetermined case and the proposed algorithm has work factor

$$W_{\mathrm{Wogr}} = m\binom{n-p-k-1}{t_{\mathrm{pub}}}\binom{n-p}{t_{\mathrm{pub}}}^{\mu-1}$$

in $\mathbb{F}_q$, where $p = \min\left\{i \in [1\!:\!n] : m\binom{n-i-k-1}{t_{\text{pub}}} \geq \binom{n-i}{t_{\text{pub}}} - 1\right\}$. Otherwise, we are in the underdetermined case in which the algorithm has work factor

$$W_{\text{Wogr}} = \min\{W_{\text{Under}}, W_{\text{Hybrid}}\}.$$

We have

$$W_{\text{Hybrid}} = q^{at_{\text{pub}}} m \binom{n-k-1}{t_{\text{pub}}} \binom{n-a}{t_{\text{pub}}}^{\mu-1}$$

with $a = \min\left\{i \in [1\!:\!n] : m\binom{n-k-1}{t_{\text{pub}}} \geq \binom{n-i}{t_{\text{pub}}} \; - 1\right\}$. Furthermore, for $0 < b < t_{\text{pub}} + 2$ and $A_b - 1 \leq B_b + C_b$,

$$W_{\text{Under}} = \frac{B_b\binom{k+t_{\text{pub}}+1}{t_{\text{pub}}} + C_b(mk+1)(t_{\text{pub}}+1)}{B_b + C_b} \left(\sum_{j=1}^{b} \binom{n}{t_{\text{pub}}} \binom{mk+1}{j}\right)^2,$$

where $A_b \coloneqq \sum_{j=1}^{b} \binom{n}{t_{\text{pub}}}\binom{mk+1}{j}$, $B_b \coloneqq \sum_{j=1}^{b} m\binom{n-k-1}{t_{\text{pub}}}\binom{mk+1}{j}$ and

$$C_b \coloneqq \sum_{j=1}^{b} \sum_{i=1}^{j} \left((-1)^{i+1} \binom{n}{t_{\text{pub}}+i} \binom{m+i-1}{i} \binom{mk+1}{j-i}\right).$$

We denote the minimum of the work factors of the two algorithms as the work factor of the algebraic rank syndrome decoding attack, i.e.,

$$W_{\text{ARSD}} = \min\{W_{\text{Gr}}, W_{\text{Wogr}}\}.$$

Note that for algebraic decoding, it is neither known how to improve the complexity by using the fact that there are multiple solutions, nor it is known how to speed up the algorithm in the quantum world.

Linearization Attack

In [62], a message attack was proposed, which succeeds for some parameters with high probability in polynomial time.

Lemma 5.9 (Linearization Attack [62]). *Let $\boldsymbol{k}_{\mathrm{pub}}^{(i)} = \mathrm{Tr}(\gamma_i \boldsymbol{k}_{\mathrm{pub}})$, for $i \in [1\!:\!u]$, and*

$$\boldsymbol{M} = \begin{bmatrix} \mathcal{M}_{t_{\mathrm{pub}}+1,q}(\boldsymbol{c}) \\ -\mathcal{M}_{t_{\mathrm{pub}}+1,q}\left(\boldsymbol{k}_{\mathrm{pub}}^{(1)}\right) \\ \vdots \\ -\mathcal{M}_{t_{\mathrm{pub}}+1,q}\left(\boldsymbol{k}_{\mathrm{pub}}^{(u)}\right) \\ -\mathcal{M}_{k+t_{\mathrm{pub}}-u,q}(\boldsymbol{g}) \end{bmatrix}. \tag{5.7}$$

Then, the encrypted message $\boldsymbol{m}$ can be efficiently recovered if the left kernel of $\boldsymbol{M}$ has dimension 1*.*

If $(u+2)t_{\mathrm{pub}} + k > n$, then $\boldsymbol{M}$ has at least two more rows than columns, and we have that the dimension of the left kernel of $\boldsymbol{M}$ is greater than 1. If $\boldsymbol{k}_{\mathrm{pub}}$ is random and $(u+2)t_{\mathrm{pub}} + k \leq n$, the attack is efficient with high probability [62].

Lemma 5.10. *Let $\boldsymbol{M}$ be as in* (5.7)*. Then,*

$$\mathrm{rk}_{q^m}(\boldsymbol{M}) \leq \min\{\varphi + k + 2t_{\mathrm{pub}} - u, n\}.$$

Proof. We can write

$$\boldsymbol{k}_{\mathrm{pub}}^{(i)} = \mathrm{Tr}(\gamma_i \boldsymbol{k}_{\mathrm{pub}}) = \mathrm{Tr}(\gamma_i \boldsymbol{x}) \cdot \mathcal{M}_{k,q}(\boldsymbol{g}) + \boldsymbol{z}_i,$$

for $i \in [1\!:\!u]$. By elementary row operations, we can transform $\boldsymbol{M}$ into

$$\boldsymbol{M}' = \begin{bmatrix} \mathcal{M}_{t_{\mathrm{pub}}+1,q}(\boldsymbol{c}) \\ -\mathcal{M}_{t_{\mathrm{pub}}+1,q}(\boldsymbol{z}_1) \\ \vdots \\ -\mathcal{M}_{t_{\mathrm{pub}}+1,q}(\boldsymbol{z}_u) \\ -\mathcal{M}_{k+t_{\mathrm{pub}}-u,q}(\boldsymbol{g}) \end{bmatrix}.$$

Due to $w + 2t_{\mathrm{pub}} < n - k$, the matrix $\mathcal{M}_{t_{\mathrm{pub}}+1,q}(\boldsymbol{z}_i)$ is a sub-matrix of $\mathcal{M}_{n-k-w,q}(\boldsymbol{z}_i)$, and the rank $\mathrm{rk}_{q^m}(\boldsymbol{M})$ is equal to

$$\mathrm{rk}_{q^m}(\boldsymbol{M}') \leq \varphi + \mathrm{rk}_{q^m}(\mathcal{M}_{t_{\mathrm{pub}}+1,q}(\boldsymbol{c})) + \mathrm{rk}_{q^m}(\mathcal{M}_{k+t_{\mathrm{pub}}-u,q}(\boldsymbol{g})) = \varphi + k + 2t_{\mathrm{pub}} - u.$$

Furthermore, since the number of columns of $\boldsymbol{M}$ is equal to n, $\mathrm{rk}_{q^m}(\boldsymbol{M}) \leq n$. ■

The linearization attack is inefficient if the rank of $\boldsymbol{M}$ is smaller than its number of rows, which implies the following, stronger version of the original statement in [62].

Theorem 5.11. *If $t_{\mathrm{pub}} > \frac{n-k}{u+2}$ or $\varphi < u(t_{\mathrm{pub}}+1)$, the linearization attack in [62] is inefficient and its work factor is*

$$W_{\mathrm{Lin}} = q^{m \cdot \max\{ut_{\mathrm{pub}}+u+1-\varphi, (u+2)t_{\mathrm{pub}}+k+1-n\}}.$$

The first condition in Theorem 5.11 is again fulfilled by the choice of w in Table 5.1. The second one reads as $t_{\mathrm{pub}} > \frac{\varphi}{u} + 1$, and for any valid φ, there are choices of w such that t_{pub} fulfills this inequality for any $u > 1$.

Algebraic Attacks

Faure and Loidreau [62] also described two message attacks of exponential worst-case complexity. The first one is based on computing Greatest Common Divisors (GCDs) of polynomials of degrees

$$W_{\mathrm{GCD}} = q^{m(u-1)} \frac{q^{t_{\mathrm{pub}}+1}-1}{q-1}. \tag{5.8}$$

Since computing the GCD of two polynomials can be implemented in quasi-linear time in the degree of the polynomials, Equation (5.8) gives an estimate on the work factor of this attack. The second algebraic attack is based on finding Gröbner bases of a system of $n_{\mathrm{p}} = \binom{n}{k+2t_{\mathrm{pub}}-u+1}$ many polynomials of degree approximately $d_{\mathrm{p}} = \frac{q^{t_{\mathrm{pub}}+1}-1}{q-1}$. The attack is only efficient for small code parameters, cf. [62, Sec. 5.3]. Since the average-case complexity of Gröbner bases algorithms is hard to estimate, we cannot directly relate n_{p} and d_{p} to the work factor of the attack. Faure and Loidreau choose the code parameters such that $n_{\mathrm{p}} \approx 2^{32}$ and $d_{\mathrm{p}} = 127$ and claim that the attack is inefficient for these values. Our example parameters in Section 5.5 result in values of at least this size.

Overbeck-like Attack

The key attack described in [200, Ch. 7, Sec. 2.1] is based on a similar principle as the one Overbeck uses to attack the McEliece cryptosystem based on Gabidulin codes [44].

The attack from [200, Ch. 7, Sec. 2.1] cannot be applied if

$$w \geq n - k - \frac{k-u}{u-1}.$$

Note that in the proposed parameter sets, this inequality is fulfilled.

Brute-Force Attack on the Element α

An attacker can brute-force $\alpha \in \mathbb{F}_{q^{mu}}$, which has a complexity of

$$W_\alpha = q^{mu}.$$

By knowing α, he just needs to apply an efficient decoding algorithm on $\tilde{\boldsymbol{c}} = \boldsymbol{c} - \mathrm{Tr}(\alpha \boldsymbol{k}_{\mathrm{pub}})$ to retrieve the secret message.

5.4.3 Exponential-Time Attacks on DecRIGab

We have seen in Section 5.3 that LIGA is IND-CCA2 secure under the assumption that DecRGab is a difficult problem. The two previous subsections analyzed all known attacks on the SeaRGab and SeaRIGab problems, which are relevant since there is an obvious reduction of DecRGab to these search problems.

In the following, we study the Problem DecRIGab, which is different in the sense that we do not know an efficient reduction from DecRGab or one of the search problems to DecRIGab. In other words, even if distinguishing the public key is easy, it could still be hard to distinguish the ciphertext. Nevertheless, we study the hardness of DecRIGab in the following and present a distinguisher, which is efficient to compute if ζ is small. The distinguisher is as follows.

Recall the choice of $\boldsymbol{k}_{\mathrm{pub}}$ in Algorithm 26,

$$\boldsymbol{k}_{\mathrm{pub}} = \boldsymbol{x} \cdot \boldsymbol{G}_{\mathcal{G}} + \boldsymbol{z} \in \mathbb{F}_{q^{mu}}^n.$$

Expand $\boldsymbol{k}_{\mathrm{pub}}$ into a $u \times n$ matrix over $\mathbb{F}_{q^m}$ and choose any $\zeta + 1$ rows. As the $\mathbb{F}_{q^m}$-expansion of the error $\boldsymbol{z}$ has $\mathbb{F}_{q^m}$-rank ζ, there are at least $q^m - 1$ many non-trivial $\mathbb{F}_{q^m}$-linear combinations of these $\zeta + 1$ rows that are codewords of $\mathcal{G}_k(\boldsymbol{g})$. This is not true with high probability for a random $u \times n$ matrix over $\mathbb{F}_{q^m}$.

Thus, by repeatedly forming random linear combinations of these $\zeta + 1$ rows and checking whether the result is a codeword of $\mathcal{G}_k(\boldsymbol{g})$, we obtain a Monte-Carlo algorithm with an expected work factor of

$$W_{\boldsymbol{k}_{\mathrm{pub}},\mathrm{distinguisher}} = q^{m\zeta},$$

neglecting the cost of checking whether a vector in $\mathbb{F}_{q^m}^n$ is a codeword. Hence, if $m\zeta$ is smaller than the security parameter of the system, this distinguisher is feasible to compute.

5.4.4 Avoiding Weak Keys

As already discussed in Section 5.4.2, the work factors of the “Randomized Gabidulin Decoding Attack on the Ciphertext” and the “List Decoding of the Ciphertext Attack” depend on the rank of the error part $\mathrm{Tr}(\alpha\boldsymbol{z})+\boldsymbol{e}$ of the ciphertext. Generically, this error has weight $t_{\mathrm{pub}}+w$, but due to the trace operation and the addition, the rank could be smaller.

In Appendix C.3, we analyze the probability that for a given key and a random encryption, the rank is significantly smaller than expected (we use w as a threshold, see Section 5.4.2). It turns out that this probability heavily depends on the minimum distance of the code $\mathcal{A}$ used to generate $\boldsymbol{z}$ in Algorithm 26. The smaller this minimum distance, the larger the probability that the rank is low. More precisely, for a given $\mathcal{A}$ of minimum distance $2 \leq t \leq w-\zeta+2$, the probability

$$\Pr(\mathrm{rk}_q(\mathrm{Tr}(\alpha\boldsymbol{z})+\boldsymbol{e}) < w) \leq q^{-m\zeta} + 256\min\{t, t_{\mathrm{pub}}\}^2 q^{-(t+t_{\mathrm{pub}}-w+1)\left(n+\frac{-t-w-t_{\mathrm{pub}}+1}{2}\right)},$$

cf. Theorem C.9 in Appendix C.3.

Due to the above discussion, we call a key with $\Pr(\mathrm{rk}_q(\mathrm{Tr}(\alpha\boldsymbol{z})+\boldsymbol{e}) < w) > 2^{-\lambda}$ a *weak key*. In Appendix C.3, we derive an upper bound on the probability of choosing a weak key in Algorithm 26. For $\zeta q^{\zeta w-m} \leq \frac{1}{2}$, this bound is roughly

$$\Pr(\text{weak key}) \leq \Theta\!\left(q^{m[t-(w-\zeta+2)]}\right),$$

cf. Remark C.1 in Appendix C.3, where t is the smallest minimum distance for which the key is not weak.

It can be seen that the parameters of LIGA can be chosen such that there is a t with $2 \leq t \leq w-\zeta+2$ and both $\Pr(\mathrm{rk}_q(\mathrm{Tr}(\alpha\boldsymbol{z})+\boldsymbol{e}) < w)$ and $\Pr(\text{weak key})$ are smaller than $2^{-\lambda}$. This is the case for all parameters proposed in Table 5.3.

5.4.5 Summary of the Work Factors

In this section, we recall the conditions on the choice of the parameters such that all known attacks are inefficient and summarize their work factors. Furthermore, we give specific parameters and compare LIGA to other code-based encryption schemes.

In the following, we choose the parameters q, m, n, k, u, w, and t_{pub} as in Table 5.1. Recall that this choice of w prevents the Overbeck-like attack (Section 5.4.2) and results in an exponential work factor of the linearization attack (Section 5.4.2).

Furthermore, we choose ζ to be small such that the work factor of searching the exponentially large output of the interleaved decoding attack (Section 5.4.1) is large. Note that this attack returns an exponentially-large output if and only if the GOT [65] attack fails, cf. Theorem 5.3.

The resulting work factors are summarized in Table 5.2. In addition to these work factors, we have considered the following requirements:

- The work factor of the second algebraic attack in [62] (cf. Section 5.4.2) is unknown. Hence, we choose the code parameters such that the resulting non-linear system of equations occurring in the attack consists of more than $n_{\mathrm{p}} \approx 2^{32}$ many polynomials of degree at least $d_{\mathrm{p}} = 127$. This is the same choice as in [62].

- Since there is no efficient list decoder for Gabidulin codes, the work factor of the list decoding of the public key or the ciphertext in Section 5.4.2 is not known. However, we do have a lower bound on the worst-case work factor for some codes, which is given by the maximal list size $\mathcal{L}_{\boldsymbol{c},\mathrm{worst}}$ in (5.6). In all examples for which the bound holds, we chose the parameters such that $\log_2(\mathcal{L}_{\boldsymbol{c},\mathrm{worst}})$ is much larger than the claimed security level.

- The probability of generating a weak key should be negligible. Thus, we choose the parameters such that $\zeta q^{\zeta w - m} \leq \frac{1}{2}$ and

$$\begin{aligned}\Pr(\text{weak key}) &\leq \frac{q^{m\zeta}-1}{(q^m-1)(q^{mw}-1)}\left(\sum_{i=0}^{t-1}\begin{bmatrix} w \\ i \end{bmatrix}_q \prod_{j=0}^{i-1}\left(q^m-q^j\right)-1\right) \\ &\leq 2^{-\lambda},\end{aligned}$$

Table 5.2: Summary of the work factors of the discussed attacks.

Name of the attack	Work factor
Brute-force $\boldsymbol{z}$ (Sec. 5.4.1)	$W_{\boldsymbol{z}} = \frac{q^{w(n-w)}}{N'_{\mathrm{R}}}$
Interleaved Decoding (Sec. 5.4.1)	$W_{\mathrm{ILD}} = \frac{q^{m(w-\zeta(n-k-w))}}{N'_{\mathrm{R}}}$
Randomized Decoding (Sec. 5.4.1)	$W_{\mathrm{RGD}} = \frac{n}{64} q^{m(n-k)-w(n+m)+w^2+\min\{2\xi(\frac{n+k}{2}-\xi), wk\}}$
Moving to Close Error (Sec. 5.4.1)	$W_{\mathrm{MCE}} = q^{(n-w)(w(\zeta+1)-\zeta(n-k))}$
Combinatorial RSD (Sec. 5.4.2)	$W_{\mathrm{CRSD}} = n^3 m^3 q^{t_{\mathrm{pub}} \lceil \frac{(k+1)m}{n} \rceil - m}$
Algebraic RSD (Sec. 5.4.2)	W_{ARSD}
Linearization (Sec. 5.4.2)	$W_{\mathrm{Lin}} = q^{m \cdot \max\{u t_{\mathrm{pub}}+u+1-\varphi, (u+2)t_{\mathrm{pub}}+k+1-n\}}$
GCD based attack (Sec. 5.4.2)	$W_{\mathrm{GCD}} = q^{m(u-1) \frac{q^{t_{\mathrm{pub}}+1}-1}{q-1}}$
Brute-force α (Sec. 5.4.2)	$W_{\alpha} = q^{mu}$
Distinguisher for $\boldsymbol{k}_{\mathrm{pub}}$ (Sec. 5.4.3)	$W_{\boldsymbol{k}_{\mathrm{pub}},\mathrm{distinguisher}} = q^{m\zeta}$

where λ is the security parameter and

$$t := \min \left\{ t \in \mathbb{Z} : q^{-m\zeta} + 256 \min\{t, t_{\mathrm{pub}}\}^2 q^{-(t+t_{\mathrm{pub}}-w+1)\left(n+\frac{-t-w-t_{\mathrm{pub}}+1}{2}\right)} \leq 2^{-\lambda} \right\}.$$

5.5 Parameters and Key Sizes

We propose parameters for security levels of 128 bit, 192 bit, and 256 bit in Table 5.3, where $R = \frac{k-u}{n}$ denotes the rate. The parameters are chosen such that we can send at least 256 bit of information, and therefore, the system can be used as a KEM. Furthermore, we use a security margin of at least 20 bits. For all parameters, the algebraic attack based on computing GCDs of polynomials is the most efficient attack.

To evaluate the performance of LIGA, we compare it to the IND-CCA2-secure version of Loidreau's system [51, 201] and the NIST proposals RQC [151], ROLLO [150], BIKE [148], and Classic McEliece [135]. To do so, we present the sizes of the private

Table 5.3: Parameter sets for 128 bit, 192 bit, and 256 bit security.

Parameter Set	q	u	k	n	m	ζ	w	t_{pub}	R
LIGA-128	2	5	53	92	97	2	27	6	0.52
LIGA-192	2	5	69	120	127	2	35	8	0.53
LIGA-256	2	5	85	148	149	2	43	10	0.54

Table 5.4: Comparison of the memory costs of the private key sk, the public key pk, and the ciphertext ct in bytes with IND-CCA2-secure Loidreau [201] and the NIST proposals RQC [151], ROLLO [150], BIKE [148], and Classic McEliece [135]. The entry ’yes’ in the column DFR indicates that a scheme has a decryption failure rate larger than 0.

System name	sk	pk	ct	Security	DFR
LIGA-128	40	5618	1116	128 bit	no
RQC-I	40	1834	3652	128 bit	no
ROLLO-I-128	40	696	696	128 bit	yes
Loidreau-128	—	6720	464	128 bit	no
BIKE-2 Level 1	249	1271	1271	128 bit	yes
McEliece348864	6452	261120	128	128 bit	no
LIGA-192	40	9565	1905	192 bit	no
RQC-II	40	2853	5690	192 bit	no
ROLLO-I-192	40	958	958	192 bit	yes
Loidreau-192	—	11520	744	192 bit	no
BIKE-2 Level 2	387	2482	2482	192 bit	yes
McEliece460896	13568	524160	188	192 bit	no
LIGA-256	40	13823	2757	256 bit	no
RQC-III	40	4090	8164	256 bit	no
ROLLO-I-256	40	1371	1371	256 bit	yes
Loidreau-256	—	16128	1024	256 bit	no
BIKE-2 Level 3	513	4094	4094	256 bit	yes
McEliece6688128	13892	1044992	240	256 bit	no

key sk,[4] the public key pk, and the ciphertext ct in Table 5.4. Note that we can use a similar representation of the private key and the public key as in RQC [151, Sec. 2.3.3]. More precisely, we just store a seed of size 40 bytes to generate the private key $\mathsf{sk} = (\boldsymbol{x}, \boldsymbol{P}_{:,[w+1:n]})$ which leads to a private key size of 40 bytes. The vector $\boldsymbol{g}$ in the public key $\mathsf{pk} = (\boldsymbol{g}, \boldsymbol{k}_{\text{pub}})$ can be also stored as a seed of size 40 bytes. Thus, the size of the public key pk is equal to $\left\lceil \frac{mnu \log_2(q)}{8} \right\rceil + 40$ bytes. The size of the ciphertext ct is given by $\left\lceil \frac{nm \log_2(q)}{8} \right\rceil$ bytes.

In [211], a generalization of Grover's algorithm is proposed that finds the roots of a function f in $\sqrt{2^b/r}$ function evaluations on average, where r is the number of roots and 2^b is the number of possible inputs of f. Thus, in a post-quantum world, all shown attacks on LIGA may be accelerated using Grover's algorithm except the GCD based attack and the Algebraic Rank Syndrome Decoding attack. Similar to the quantum ISD algorithm described in [212], the mentioned attacks have in common that they guess an element from a large set, and then, evaluate in polynomial time whether the guess leads to the desired outcome. If the desired outcome is obtained, the system can be broken in polynomial time using exactly this guess. Thus, the work factor of these algorithms is the product of the complexity of checking whether the guess leads to the desired outcome times the inverse of the probability that the guess leads to the desired outcome. We can easily construct a function f that takes as input a guess and checks in polynomial time whether the guess is as desired. If this is the case, f outputs 0 and otherwise anything except 0. Then, we can apply Grover's algorithm to find a root of that function f. Such a root is an element of the set that leads to the desired outcome. In this way, Grover's algorithm reduces the work factor to the product of the polynomial time required for checking the guess times the *square-root* of the inverse of the probability that the guess is as desired. For the GCD based attack, we do not know how to improve the work factor by a quantum computer since the stated complexity already assumes a running time linear in the degree of the polynomials. Furthermore, at the current state of research, there is no quantum speed up known for the Algebraic Rank Syndrome Decoding attack [151, Sec. 6.3]. Using the described work factors, we obtain a post-quantum security level of 97.5 bit, 127.5 bit, and 157.5 bit for LIGA-128, LIGA-192, and LIGA-256, respectively, where Moving to Close Error is the most efficient attack for all three parameter sets.

[4] The size of the private key of Loidreau's system is not shown since the authors of [201] do not state how they represent sk.

5.6 Concluding Remarks

In this chapter, we presented LIGA, a new rank-metric code-based encryption scheme. LIGA uses a new coding-theoretic interpretation of the FL system. We showed that the ciphertext is a corrupted codeword of a Gabidulin code, where, to an unauthorized receiver, the error weight is too large to be correctable. The authorized user knows the row space of a part of the error and is thus able to correct the error. Furthermore, we derived that the public key can be seen as a corrupted codeword of an interleaved Gabidulin code and that in the original FL system, an interleaved Gabidulin decoder can efficiently recover the private key from the public key with high probability. We proved that the condition under which interleaved Gabidulin decoders fail is equal to the condition under which the severe attack by Gaborit, Otmani, and Talé Kalachi fails. Based on this observation, we chose the key generation algorithm of LIGA such that interleaved Gabidulin decoders fail, which in turn implies that the attack by Gaborit *et al.* fails.

We proposed two versions of LIGA and proved that the public-key encryption version is IND-CPA secure in the standard model and the KEM version is IND-CCA2 secure in the random oracle model under the assumption that the DecRGab problem is hard. We extensively analyzed the security of this decisional problem by studying attacks on SeaRGab, SeaRIGab, and DecRIGab. All studied attacks have an exponential work factor in the proposed parameter ranges and can be avoided by a suitable parameter choice.

Finally, we presented parameters for security levels of 128 bit, 192 bit, and 256 bit and compared them to the NIST proposals RQC, ROLLO, BIKE, Classic McEliece, and a rank-metric McEliece-like system proposed by Loidreau. It was observed that LIGA has small ciphertext sizes as well as relatively small key sizes. The encryption and decryption correspond to encoding and decoding of Gabidulin codes, for which efficient and constant-time algorithms exist. Furthermore, the proposed system guarantees decryption and is not based on hiding the structure of a code.

A New Polynomial-Time Attack on LIGA

In [202], Bombar and Couvreur present a new message recovery attack on the proposed cryptosystem LIGA. The authors are not disproving the IND-CCA2 security claim under the assumption that the underlying problems are hard. They rather derive a unique algorithm that can decode certain Gabidulin supercodes efficiently. Using this algorithm, they can solve SeaRGab efficiently, and therefore, they prove that the hardness assumptions of SeaRGab and DecRGab are incorrect. Furthermore, their new decoder allows them to mount a polynomial-time message recovery attack on LIGA.

6
Conclusions

In this thesis, we considered general coding-theoretic problems with applications in cryptography, we cryptanalyzed two existing code-based cryptographic schemes, and we proposed a new rank-metric encryption scheme.

The first part of Chapter 3 covered the problem of syndrome decoding in the sum-rank metric. We presented new findings about erasure decoding in the sum-rank metric and used these observations to devise a non-trivial generic decoding algorithm. We showed that our algorithm has a significantly lower complexity than known naïve approaches for most parameters and that the problem of syndrome decoding in the Hamming metric can be reduced to the considered problem in a randomized fashion. In the second part of this chapter, we considered the problem syndrome decoding of high-order interleaved rank-metric codes. We devised a new generic decoding algorithm and stated conditions under which the algorithm solves the problem in polynomial time. For high-order interleaved Gabidulin codes, we proved that our generic algorithm has the same error correction capability as all known decoding strategies that are tailored to Gabidulin codes. In the third part, we devised a novel strategy for decoding Gabidulin codes beyond the unique decoding radius and discussed possible modifications to it. We observed that this decoding strategy has a considerably lower complexity than all known algorithms for most parameters. A short summary and open problems related to all three problems concluded this chapter.

Chapter 4 was dedicated to the cryptanalysis of two encryption schemes in the Hamming metric. The first system that we considered was a variant of McEliece's encryption scheme based on TRS codes. We devised a feasible key-recovery attack on this system, where our attack is the first of its kind to exploit structural weaknesses of subfield subcodes of the public code. We proved that for all practical parameters proposed by the designers, the attack recovers a valid private key and has a complexity that is quartic in the code length. Furthermore, we observed that the average runtime of the attack implemented on a general purpose processor is only a few minutes. In the second part of this chapter, we investigated the IND-CPA-secure encryption version and the IND-CCA2-secure KEM version of HQC. After recalling the security assumptions of the system, we developed the first power-based side-channel chosen-ciphertext attack on both variants of HQC. We gave a detailed analysis about the success probability of the attack and its runtime, where we observed that the attack recovers more than 93% of the possible keys of HQC-128. This chapter ended with remarks on both systems and related open problems.

In Chapter 5, we proposed a new rank-metric encryption scheme called LIGA. The system constitutes a variant of the broken FL system, and its security does *not* rely on hiding the structure of a code but is based on the difficulty of list decoding and interleaved decoding of Gabidulin codes. We proved that the public-key encryption variant of LIGA is IND-CPA secure in the standard model and the KEM variant is IND-CCA2 secure in the random oracle model, both under hardness assumptions of formally defined problems. We analyzed several exponential-time attacks on the aforementioned problems, summarized their average complexities, and compared the resulting parameters to Loidreau's system and the NIST proposals RQC, ROLLO, BIKE, and Classic McEliece. We observed that LIGA features short ciphertext sizes, small key sizes, and no decryption failures. We concluded this chapter with a summary and a description of a new polynomial-time attack, which was proposed by Bombar and Couvreur.

A

Remarks on Chapter 2

A.1 Complexity of Algebraic Rank Syndrome Decoding

In [141], Bardet *et al.* reformulate the $\mathsf{DecSD}_{\mathrm{R}}$ problem as a multivariate polynomial system and solve it by determining a Gröbner basis. They show that if there exists a unique solution to the system, then their algorithm has a complexity of

$$W_{\mathrm{Gr}} = \begin{cases} \left[\frac{((m+n)t)^t}{t!}\right]^{\mu} & \text{if } m\binom{n-k-1}{t} \geq \binom{n}{t} - 1 \\ \left[\frac{((m+n)t)^{t+1}}{(t+1)!}\right]^{\mu} & \text{otherwise,} \end{cases}$$

where μ refers to the exponent in the complexity expression of the matrix multiplication algorithm and is assumed to be $\mu = 2.807$.

In [142], an algorithm is presented that builds upon [141]. The complexity of the algorithm depends on the parameter set and can be divided into two cases. If the inequality

$$m\binom{n-k-1}{t} \geq \binom{n}{t} - 1 \tag{A.1}$$

holds, then we are in the overdetermined case, and the algorithm solves the $\mathsf{DecSD}_{\mathrm{R}}$ instance with an average complexity of

$$W_{\mathrm{Wogr}} = m\binom{n-p-k-1}{t}\binom{n-p}{t}^{\mu-1}$$

operations in $\mathbb{F}_q$, where $p = \max\left\{i \in [1\!:\!n] : m\binom{n-i-k-1}{t} \geq \binom{n-i}{t} - 1\right\}$. In case (A.1) does not hold, the underdetermined case occurs, and the algorithm requires, on average,

$$W_{\text{Wogr}} = \min\{W_{\text{Hybrid}}, W_{\text{Under}}\} \tag{A.2}$$

operations in $\mathbb{F}_q$. The first term refers to the complexity of using an exponential-time brute-force step to transform the underdetermined instance to an overdetermined instance. The work factor of this approach is equal to

$$W_{\text{Hybrid}} = q^{at} m \binom{n-k-1}{t} \binom{n-a}{t}^{\mu-1},$$

where $a = \min\left\{i \in [1\!:\!n] : m\binom{n-k-1}{t} \geq \binom{n-i}{t} - 1\right\}$. For a parameter set satisfying $q = 2$, $0 < b < t+2$ and $A_b - 1 \leq B_b + C_b$, the second quantity of (A.2) evaluates to

$$W_{\text{Under}} = \min\left\{(B_b + C_b)A_b^{t-1}, \frac{B_b\binom{k+t+1}{t} + C_b(mk+1)(t+1)}{B_b + C_b} A_b^2\right\},$$

where $A_b := \sum_{j=1}^{b}\binom{n}{t}\binom{mk+1}{j}$, $B_b := \sum_{j=1}^{b} m\binom{n-k-1}{t}\binom{mk+1}{j}$ and

$$C_b := \sum_{j=1}^{b}\sum_{i=1}^{j}\left((-1)^{i+1}\binom{n}{t+i}\binom{m+i-1}{i}\binom{mk+1}{j-i}\right).$$

We refer to the minimum of W_{Gr} and W_{Wogr} as the work factor of algebraic rank syndrome decoding, i.e.,

$$W_{\text{ARSD}} = \min\{W_{\text{Gr}}, W_{\text{Wogr}}\}.$$

Note that it is not known how to improve the complexity of the algorithm by using the fact that there are multiple solutions to problem.

B

Remarks on Chapter 3

B.1 Generating Uniformly at Random Errors of a Given Sum-Rank Weight

Algorithm 31: Drawing Uniformly at Random a Vector of Given Sum-Rank Weight

```
Input   : Parameters $q, m, k, n, \ell_{\mathrm{SR}}, t$
Output: Vector $\boldsymbol{e} \overset{\$}{\leftarrow} \{\boldsymbol{e}' \in \mathbb{F}_{q^m}^n : \mathrm{wt}_{\mathrm{SR}}(\boldsymbol{e}') = t\}$
$D^{(1)} \overset{\$}{\leftarrow} [1 : \mathcal{N}_{q,\eta_{\mathrm{SR}},m}(t, \ell_{\mathrm{SR}})]$
$t^{(1)} \leftarrow t$
for $j \in [1 : \ell_{\mathrm{SR}}]$ do
    $t_j \leftarrow \max\left\{t'' \in \left[0 : t^{(j)}\right] : D'(t'', j) < D^{(j)}\right\}$        ($D'(t'', j)$ is defined in (B.1))
    $D^{(j+1)} \leftarrow D^{(j)} - D'(t_j, j)$
    $t^{(j+1)} \leftarrow t^{(j)} - t_j$
for $j \in [1 : \ell_{\mathrm{SR}}]$ do
    $\boldsymbol{a}_j \overset{\$}{\leftarrow} \{\boldsymbol{a} \in \mathbb{F}_{q^m}^{t_j} : \mathrm{rk}_q(\boldsymbol{a}) = t_j\}$
    $\boldsymbol{B}_j \overset{\$}{\leftarrow} \{\boldsymbol{B} \in \mathbb{F}_q^{t_j \times \eta_{\mathrm{SR}}} : \mathrm{rk}_q(\boldsymbol{B}) = t_j\}$
$\boldsymbol{e} \leftarrow [\boldsymbol{a}_1\boldsymbol{B}_1, \boldsymbol{a}_2\boldsymbol{B}_2, \ldots, \boldsymbol{a}_{\ell_{\mathrm{SR}}}\boldsymbol{B}_{\ell_{\mathrm{SR}}}] \in \mathbb{F}_{q^m}^n$
return $\boldsymbol{e}$
```

The recursion in Theorem 3.1 can be turned into a variant of enumerative encoding [213] to efficiently draw uniformly at random from the set of sum-rank vectors of weight

t. Such an algorithm is outlined in Algorithm 31, where

$$D'(t'', j) \coloneqq \sum_{t'=t^{(j)}-\mu_{\mathrm{SR}}(\ell_{\mathrm{SR}}-j)}^{t''-1} \mathrm{NM}_q(m, \eta_{\mathrm{SR}}, t')\, \mathcal{N}_{q,\eta_{\mathrm{SR}},m}(t^{(j)} - t', \ell_{\mathrm{SR}} - j). \tag{B.1}$$

The correctness of Algorithm 31 is proven in the following proposition:

Proposition B.1. *Let $q, m, k, n, \ell_{\mathrm{SR}}$, and t be integers such that $\ell_{\mathrm{SR}} \mid n$ and $t \leq \mu_{\mathrm{SR}}\ell_{\mathrm{SR}}$. Furthermore, let $D'(t'', j)$ be defined as in* (B.1)*. Then, Algorithm 31 outputs a vector $\boldsymbol{e} \in \mathbb{F}_{q^m}^n$ drawn uniformly at random from $\{\boldsymbol{e}' \in \mathbb{F}_{q^m}^n : \mathrm{wt}_{\mathrm{SR}}(\boldsymbol{e}') = t\}$.*

Proof. The set $\{\boldsymbol{e}' \in \mathbb{F}_{q^m}^n : \mathrm{wt}_{\mathrm{SR}}(\boldsymbol{e}') = t\}$ has cardinality $\mathcal{N}_{q,\eta_{\mathrm{SR}},m}(t, \ell_{\mathrm{SR}})$. Let

$$\varphi : [1\colon \mathcal{N}_{q,\eta_{\mathrm{SR}},m}(t, \ell_{\mathrm{SR}})] \to \{\boldsymbol{e}' \in \mathbb{F}_{q^m}^n : \mathrm{wt}_{\mathrm{SR}}(\boldsymbol{e}') = t\}$$

be a bijective mapping. If we know an efficient algorithm to realize the mapping φ, then the drawing can be realized by uniformly sampling $D^{(1)}$ from $[1\colon \mathcal{N}_{q,\eta_{\mathrm{SR}},m}(t, \ell_{\mathrm{SR}})]$ and outputting $\varphi(D^{(1)})$. However, the drawing algorithm can also be realized with a different method.

Let $\phi : \{\boldsymbol{e} \in \mathbb{F}_{q^m}^n : \mathrm{wt}_{\mathrm{SR}}(\boldsymbol{e}) = t\} \to \mathcal{T}_{t,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}}$, $\boldsymbol{e} \mapsto [\mathrm{rk}_q(\boldsymbol{e}_1), \dots, \mathrm{rk}_q(\boldsymbol{e}_{\ell_{\mathrm{SR}}})]$. Then, the drawing can be conducted by computing $\boldsymbol{t} = (\phi \circ \varphi)(D^{(1)})$ and sampling $\boldsymbol{a}_j \xleftarrow{\$} \{\boldsymbol{a} \in \mathbb{F}_{q^m}^{t_j} : \mathrm{rk}_q(\boldsymbol{a}) = t_j\}$ and $\boldsymbol{B}_j \xleftarrow{\$} \{\boldsymbol{B} \in \mathbb{F}_q^{t_j \times \eta_{\mathrm{SR}}} : \mathrm{rk}_q(\boldsymbol{B}) = t_j\}$, for $j \in [1\colon\ell_{\mathrm{SR}}]$. Since $\boldsymbol{e}_j = \boldsymbol{a}_j\boldsymbol{B}_j \in \mathbb{F}_{q^m}^{\eta_{\mathrm{SR}}}$ is a vector drawn uniformly at random from $\{\boldsymbol{e}' \in \mathbb{F}_{q^m}^{\eta_{\mathrm{SR}}} : \mathrm{rk}_q(\boldsymbol{e}') = t_j\}$, it follows that $\boldsymbol{e} = [\boldsymbol{a}_1\boldsymbol{B}_1, \dots, \boldsymbol{a}_{\ell_{\mathrm{SR}}}\boldsymbol{B}_{\ell_{\mathrm{SR}}}]$ is a vector drawn uniformly at random from $\{\boldsymbol{e}' \in \mathbb{F}_{q^m}^n : \mathrm{wt}_{\mathrm{SR}}(\boldsymbol{e}') = t\}$.

To derive the mapping $\phi \circ \varphi : [1\colon \mathcal{N}_{q,\eta_{\mathrm{SR}},m}(t, \ell_{\mathrm{SR}})] \to \mathcal{T}_{t,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}}$ suppose that $t \leq \mu_{\mathrm{SR}}$. Then, the number of vectors that have a weight decomposition $[0, \dots, 0, t]$ is equal to $\mathrm{NM}_q(m, \eta_{\mathrm{SR}}, t)$, and therefore, we map the elements of the set $[1 : \mathrm{NM}_q(m, \eta_{\mathrm{SR}}, t)]$ to the vector $[0, \dots, 0, t]$. Furthermore, the number of vectors that have a weight decomposition $[0, \dots, 0, 1, t-1]$ is equal to $\mathrm{NM}_q(m, \eta_{\mathrm{SR}}, 1)\mathrm{NM}_q(m, \eta_{\mathrm{SR}}, t-1)$, which means that we map elements of the set

$$[\mathrm{NM}_q(m, \eta_{\mathrm{SR}}, t) + 1\colon\mathrm{NM}_q(m, \eta_{\mathrm{SR}}, t) + \mathrm{NM}_q(m, \eta_{\mathrm{SR}}, 1)\mathrm{NM}_q(m, \eta_{\mathrm{SR}}, t-1)]$$

to the vector $[0, \dots, 1, t-1]$.

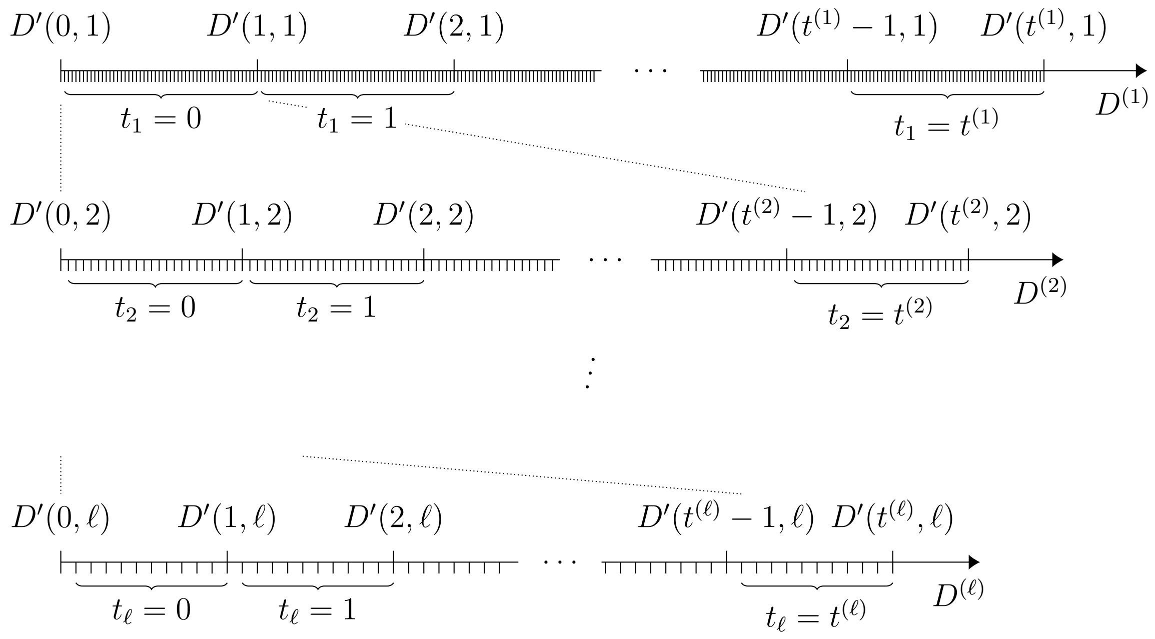


Figure B.1: Illustration of the mapping $\phi\circ\varphi : [1:\mathcal{N}_{q,\eta_{\mathrm{SR}},m}(t,\ell_{\mathrm{SR}})] \to \mathcal{T}_{t,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}}, D^{(1)} \mapsto \boldsymbol{t}$. The variables are defined as in Algorithm 31, and the function $D'(t'',j)$ is defined in (B.1).

It follows by induction that we map

$$\left[\sum_{t'=0}^{t_j-1}\mathrm{NM}_q(m,\eta_{\mathrm{SR}},t')\mathcal{N}_{q,\eta_{\mathrm{SR}},m}(t-t',\ell_{\mathrm{SR}}-j)+1:\sum_{t'=0}^{t_j}\mathrm{NM}_q(m,\eta_{\mathrm{SR}},t')\mathcal{N}_{q,\eta_{\mathrm{SR}},m}(t-t',\ell_{\mathrm{SR}}-j)\right]$$

to $[0,\dots,0,t_j,\dots,t_{\ell_{\mathrm{SR}}}]$, where $\sum_{i=j+1}^{\ell_{\mathrm{SR}}} t_i = t-t_j$.

Algorithm 31 performs this routine. In Line 1, the integer $D^{(1)}$ is drawn uniformly at random from $[1:\mathcal{N}_{q,\eta_{\mathrm{SR}},m}(t,\ell_{\mathrm{SR}})]$, and in Lines 2 to 6, the respective weight distribution vector $(\phi\circ\varphi)(D^{(1)})$ is determined (the case $t>\mu_{\mathrm{SR}}$ is taken into account by starting to sum from $t^{(j)}-\mu_{\mathrm{SR}}(\ell_{\mathrm{SR}}-j)$ instead of 0). The method to compute $(\phi\circ\varphi)(D^{(1)})$ is illustrated in Figure B.1. In Lines 7 to 10, the vectors $\boldsymbol{e}_j\in\mathbb{F}_{q^m}^{\eta_{\mathrm{SR}}}$ are drawn uniformly at random from the set of vectors of rank weight t_j, and the vector $[\boldsymbol{e}_1,\dots,\boldsymbol{e}_{\ell_{\mathrm{SR}}}]$ is returned. ■

B.2 Optimal Support-Drawing Algorithm

In Section 3.1.2, it was shown that the worst-case expected number of iterations of a super-support drawing algorithm that first draws a vector $\boldsymbol{f} \in \mathcal{T}_{s,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}}$ according to a probability distribution $\tilde{p}_{\boldsymbol{f}}$ and then $\mathcal{F} \overset{\$}{\leftarrow} \Xi_{q,\mu_{\mathrm{SR}}}(\boldsymbol{f})$, is equal to

$$\max_{\substack{\boldsymbol{e}\in\mathbb{F}_{q^m}^n:\\ \mathrm{wt}_{\mathrm{SR}}(\boldsymbol{e})=t}} \mathbb{E}[\#\text{iterations}] = \max_{\boldsymbol{t}\in\mathcal{T}_{t,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}}} \left(\sum_{\boldsymbol{f}\in\mathcal{T}_{s,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}}} \tilde{p}_{\boldsymbol{f}}\, \varrho_{q,\mu_{\mathrm{SR}}}(\boldsymbol{f},\boldsymbol{t}) \right)^{-1}.$$

Section 3.1.2 presented an efficient but suboptimal method to choose $\tilde{p}_{\boldsymbol{f}}$. In the following, we reformulate the optimization problem into a linear programming instance. The solution to this problem is optimal but obtaining the solution has a super-polynomial complexity.

Theorem B.2. *Let* $\boldsymbol{f}_1,\dots,\boldsymbol{f}_{|\mathcal{T}_{s,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}}|} \in \mathcal{T}_{s,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}}$, $\boldsymbol{t}_1,\dots,\boldsymbol{t}_{|\mathcal{T}_{t,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}}|} \in \mathcal{T}_{t,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}}$, *and choose*

$$\boldsymbol{c} = [0,\dots,0,1]^\top \in \mathbb{R}^{(|\mathcal{T}_{s,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}}|+1)\times 1}, \quad \boldsymbol{b} = [0,\dots,0,1,-1]^\top \in \mathbb{R}^{|\mathcal{T}_{t,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}}|\times 1},$$

and $\boldsymbol{A} \in \mathbb{R}^{(|\mathcal{T}_{t,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}}|+2)\times(|\mathcal{T}_{s,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}}|+1)}$ *is equal to*

$$\begin{bmatrix}
-\varrho_{q,\mu_{\mathrm{SR}}}(\boldsymbol{f}_1,\boldsymbol{t}_1) & -\varrho_{q,\mu_{\mathrm{SR}}}(\boldsymbol{f}_2,\boldsymbol{t}_1) & \dots & -\varrho_{q,\mu_{\mathrm{SR}}}(\boldsymbol{f}_{|\mathcal{T}_{s,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}}|},\boldsymbol{t}_1) & 1\\
-\varrho_{q,\mu_{\mathrm{SR}}}(\boldsymbol{f}_1,\boldsymbol{t}_2) & -\varrho_{q,\mu_{\mathrm{SR}}}(\boldsymbol{f}_2,\boldsymbol{t}_2) & \dots & -\varrho_{q,\mu_{\mathrm{SR}}}(\boldsymbol{f}_{|\mathcal{T}_{s,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}}|},\boldsymbol{t}_2) & 1\\
\vdots & \vdots & \ddots & \vdots & \vdots\\
-\varrho_{q,\mu_{\mathrm{SR}}}(\boldsymbol{f}_1,\boldsymbol{t}_{|\mathcal{T}_{t,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}}|}) & -\varrho_{q,\mu_{\mathrm{SR}}}(\boldsymbol{f}_2,\boldsymbol{t}_{|\mathcal{T}_{t,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}}|}) & \dots & -\varrho_{q,\mu_{\mathrm{SR}}}(\boldsymbol{f}_{|\mathcal{T}_{s,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}}|},\boldsymbol{t}_{|\mathcal{T}_{t,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}}|}) & 1\\
1 & 1 & \dots & 1 & 0\\
-1 & -1 & \dots & -1 & 0
\end{bmatrix}.$$

Furthermore, let $\boldsymbol{x} \in \mathbb{R}^{(|\mathcal{T}_{s,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}}|+1)\times 1}$ *be a solution to the linear program*

$$\begin{array}{ll}
\textit{Maximize} & \boldsymbol{c}^\top \boldsymbol{x}\\
\textit{subject to} & \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b}\\
\textit{and} & \boldsymbol{x} \geq 0.
\end{array} \tag{B.2}$$

Then $\tilde{p}_{\boldsymbol{f}_i} = x_i$, *for all* $i \in [1:|\mathcal{T}_{s,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}}|]$, *is a distribution that maximizes* (3.2), *and it* *holds that*

$$x^{-1}_{|\mathcal{T}_{s,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}}|+1} = \min\left\{ \max_{\boldsymbol{t}\in\mathcal{T}_{t,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}}} \left(\sum_{\boldsymbol{f}\in\mathcal{T}_{s,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}}} \tilde{p}_{\boldsymbol{f}}\,\varrho_{q,\mu_{\mathrm{SR}}}(\boldsymbol{f},\boldsymbol{t}) \right)^{-1} : \tilde{p}_{\boldsymbol{f}} \in [0,1] \right.$$
$$\left. \forall \boldsymbol{f} \in \mathcal{T}_{s,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}}, \sum_{\boldsymbol{f}\in\mathcal{T}_{s,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}}} \tilde{p}_{\boldsymbol{f}} = 1 \right\}. \tag{B.3}$$

Proof. Let $\tilde{p}_{\boldsymbol{f}_i} = x_i$ and $\xi = x_{|\mathcal{T}_{s,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}}|+1}$ for a solution $\boldsymbol{x}$ of the linear program. The last two rows of the matrix $\boldsymbol{A}$ are equal to

$$\sum_{i=1}^{|\mathcal{T}_{s,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}}|} \tilde{p}_{\boldsymbol{f}_i} = 1.$$

Since $\boldsymbol{x} \geq 0$, it holds that $\tilde{p}_{\boldsymbol{f}_i}$ is a valid discrete probability mass function. The first $|\mathcal{T}_{t,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}}|$ rows of the matrix $\boldsymbol{A}$ represent the constraints

$$\sum_{i=1}^{|\mathcal{T}_{s,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}}|} \tilde{p}_{\boldsymbol{f}_i}\varrho_{q,\mu_{\mathrm{SR}}}(\boldsymbol{f}_i,\boldsymbol{t}_j) \geq \xi \quad \forall j \in [1\colon|\mathcal{T}_{t,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}}|].$$

Since ξ is the largest value for which this constraint is fulfilled for all $j \in [1\colon|\mathcal{T}_{t,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}}|]$ and solutions $\tilde{p}_{\boldsymbol{f}_i}$, it follows that

$$\xi = \max\left\{ \min_{j\in[1:|\mathcal{T}_{t,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}}|]} \left\{ \sum_{i=1}^{|\mathcal{T}_{s,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}}|} \tilde{p}_{\boldsymbol{f}_i}\varrho_{q,\mu_{\mathrm{SR}}}(\boldsymbol{f}_i,\boldsymbol{t}_j) \right\} : \tilde{p}_{\boldsymbol{f}_i} \in [0,1]\,\forall i \in [1\colon|\mathcal{T}_{s,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}}|], \right.$$
$$\left. \sum_{i=1}^{|\mathcal{T}_{s,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}}|} \tilde{p}_{\boldsymbol{f}_i} = 1 \right\}$$

which is equivalent to (B.3). ■

The linear program (B.2) in Theorem B.2 can be solved in polynomial time in the number of variables $|\mathcal{T}_{s,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}}| + 1$. However, depending on s, μ_{SR}, and ℓ_{SR}, this quantity can grow super-polynomially in s, and thus, it is usually not possible to solve the linear program efficiently for large code parameters.

Nevertheless, we consider this optimal solution to the design of $\tilde{p}_{\boldsymbol{f}}$ in the discussion in Section 3.1.3 for all values of $\ell_{\mathrm{SR}}, \mu_{\mathrm{SR}}, s$ for which we can determine a solution. The complexity of this approach is approximately

$$W_{\mathrm{SR}}^{(\mathrm{optimal})} := W_{\mathrm{Iter}}\, x_{|\mathcal{T}_{s,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}}|+1}^{-1},$$

where $x_1, \dots, x_{|\mathcal{T}_{s,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}}|+1}$ is a solution vector to the optimization problem in Theorem B.2, and W_{Iter} is the cost of one iteration. The latter value is at least the cost of erasure decoding, but the real cost could be larger due to the cost of drawing from the distribution $\tilde{p}_{\boldsymbol{f}}$.

C

Remarks on Chapter 5

C.1 Practical Considerations on the Key Generation

We discuss practical aspects related to the following lines of the modified key-generation algorithm (Algorithm 26):

```
3  $\mathcal{A} \xleftarrow{\$} \left\{ \text{subspace } \mathcal{U} \subseteq \mathbb{F}_{q^m}^{w} \,:\, \dim \mathcal{U} = \zeta,\, \mathcal{U} \text{ has a basis of full-}\mathbb{F}_q\text{-rank elements} \right\}$
3' $\begin{bmatrix} \boldsymbol{s}_1 \\ \vdots \\ \boldsymbol{s}_u \end{bmatrix} \xleftarrow{\$} \left\{ \begin{bmatrix} \boldsymbol{s}'_1 \\ \vdots \\ \boldsymbol{s}'_u \end{bmatrix} \,:\, \langle \boldsymbol{s}'_1, \dots, \boldsymbol{s}'_u \rangle_{q^m} = \mathcal{A},\, \mathrm{rk}_q(\boldsymbol{s}'_i) = w\,, \forall\, i \in [1\!:\!u] \right\}$
```

We conjecture that the set from which $\mathcal{A}$ is sampled is almost the set of $[w, \zeta]_{\mathbb{F}_{q^m}}$ codes. Using a combinatorial argument on the known number of full-rank codewords of MRD codes, we prove in Lemma C.6 (Appendix C.3) that MRD codes always have a basis consisting of full-rank codewords. Since the weight enumerator is not known in general for non-MRD codes, we cannot give a proof, but we expect that most codes that are close to MRD also have such a basis. The conjecture is then implied by the fact that close-to MRD codes constitute the majority of linear codes for the parameters considered here [214, 215].

Since it is hard to check if a randomly drawn code admits a basis of full-$\mathbb{F}_q$-rank codewords in the worst case, these arguments also imply a practical method on how to implement Lines 3 and $3'$ in practice: sample uniformly at random from the set of

$[w, \zeta]_{\mathbb{F}_{q^m}}$ codes. With overwhelming probability, the code is close to MRD, and a large proportion of its codewords have full $\mathbb{F}_q$-rank. Randomly choosing u codewords thus gives a generating set consisting of full-rank codewords with high probability. Only if no basis is found after a given number of trials, one needs to formally check if the code does not admit a generating set of full-$\mathbb{F}_q$-rank codewords. This gives a Las-Vegas-type algorithm with small expected running time.

The worst case of this algorithm occurs with extremely small probability.[1] Nevertheless, the worst-case complexity is still quite large. Alternatively, one can draw a new code $\mathcal{A}$ if no generating set is found after a given number of trials. This, however, slightly changes the random experiment from which the code $\mathcal{A}$ is drawn. The only part of Chapter 5 which is influenced by such a modification is Section 5.4.4, which studies weak keys, i.e., keys for which there is a non-negligible probability that the error part of the ciphertext has too low rank and is vulnerable to a feasible ciphertext attack. A key is weak only if the minimum distance of $\mathcal{A}$ is small. By parameter choice, the probability that such a key is generated can be made arbitrarily small, see Appendix C.3. By the same arguments as above, we conjecture that if the probability of obtaining a generating set of full-$\mathbb{F}_q$-rank codewords by drawing u codewords uniformly at random is small, then also the minimum distance of the code must be small. In summary, we expect that this change of drawing procedure results in an even smaller weak-key probability than predicted by Theorem C.9 in Appendix C.3.

C.2 Decryption as Error-Erasure Decoding

In the following, we give a coding-theoretic interpretation of the ciphertext of the original FL system and of LIGA, which—to the best of our knowledge—has not been observed before.

Lemma C.1. *The ciphertext can be written in the form*

$$\boldsymbol{c} = \boldsymbol{c}_{\mathcal{G}} + \boldsymbol{a}_{\mathrm{C}} \boldsymbol{B}_{\mathrm{C}} + \boldsymbol{e},$$

where

- $\boldsymbol{c}_{\mathcal{G}} = (\boldsymbol{m} + \mathrm{Tr}(\alpha \boldsymbol{x}) \cdot \boldsymbol{G}_{\mathcal{G}}) \in \mathbb{F}_{q^m}^n$ *is unknown and a codeword of a Gabidulin code,*

[1] It can be proven that it is close to the probability of drawing a non-MRD code at random, and it could be even smaller in reality since also “near-MRD” could have suitable bases.

- $\boldsymbol{a}_{\mathrm{C}} = \mathrm{Tr}(\alpha\boldsymbol{s}) \in \mathbb{F}_{q^m}^{w}$ *is unknown,*
- $\boldsymbol{B}_{\mathrm{C}} = (\boldsymbol{P}^{-1})_{[1:w],:} \in \mathbb{F}_{q}^{w\times n}$ *is known, and*
- $\boldsymbol{e} \in \mathbb{F}_{q^m}^{n}$ *is unknown.*

Proof. Due to the $\mathbb{F}_{q^m}$-linearity of the trace map Tr and the fact that the entries of the matrices $\boldsymbol{G}_{\mathcal{G}}$ and $\boldsymbol{P}^{-1}$ are in $\mathbb{F}_{q^m}$, we can write

$$\begin{aligned}
\boldsymbol{c} &= \boldsymbol{m}\boldsymbol{G}_{\mathcal{G}} + \mathrm{Tr}(\alpha\boldsymbol{k}_{\mathrm{pub}}) + \boldsymbol{e}\\
&= \boldsymbol{m}\boldsymbol{G}_{\mathcal{G}} + \mathrm{Tr}(\alpha\boldsymbol{x}\boldsymbol{G}_{\mathcal{G}} + \alpha\boldsymbol{z}) + \boldsymbol{e}\\
&= (\boldsymbol{m} + \mathrm{Tr}(\alpha\boldsymbol{x}))\,\boldsymbol{G}_{\mathcal{G}} + \mathrm{Tr}(\alpha\boldsymbol{z}) + \boldsymbol{e}\\
&= (\boldsymbol{m} + \mathrm{Tr}(\alpha\boldsymbol{x}))\,\boldsymbol{G}_{\mathcal{G}} + \mathrm{Tr}(\alpha(\boldsymbol{s},\boldsymbol{0})\boldsymbol{P}^{-1}) + \boldsymbol{e}\\
&= (\boldsymbol{m} + \mathrm{Tr}(\alpha\boldsymbol{x}))\,\boldsymbol{G}_{\mathcal{G}} + \mathrm{Tr}(\alpha\boldsymbol{s}(\boldsymbol{P}^{-1})_{[1:w],:}) + \boldsymbol{e}\\
&= (\boldsymbol{m} + \mathrm{Tr}(\alpha\boldsymbol{x}))\,\boldsymbol{G}_{\mathcal{G}} + \mathrm{Tr}(\alpha\boldsymbol{s})(\boldsymbol{P}^{-1})_{[1:w],:} + \boldsymbol{e}.
\end{aligned}$$

Since the entries of $(\boldsymbol{P}^{-1})_{[1:w],:}$ are in $\mathbb{F}_q$, the ciphertext can be written as above. ■

Theorem C.2. *The message vector* $\boldsymbol{m}$ *can be reconstructed by error-erasure decoders and Steps 4 and 5 of Algorithm 28.*

Proof. As seen in Lemma C.1, we can decompose the ciphertext into a codeword plus an error that is partially known. Therefore, the vector $\boldsymbol{m}+\mathrm{Tr}(\alpha\boldsymbol{x})$ can be reconstructed by error-erasure decoders since the decoding condition reads as

$$w + 2\,\mathrm{rk}_q(\boldsymbol{e}) = w + 2t_{\mathrm{pub}} \leq n - k,$$

see Lemma 2.4, and is fulfilled by Table 5.1. The message $\boldsymbol{m}$ can then be recovered from $\boldsymbol{m} + \mathrm{Tr}(\alpha\boldsymbol{x})$ using the same steps as in Algorithm 28. ■

Theorem C.2 leads to the following observation. With high probability, the ciphertext is a codeword plus an error of rank weight $w + t_{\mathrm{pub}}$, which is beyond the unique decoding radius. The legitimate receiver can only decrypt since she knows the w-dimensional row space of a part of the error. Although the attacker knows the code, she cannot recover the message since she has no further knowledge about the structure of the error.

The procedure implied by Theorem C.2 could have a practical advantage compared to the original decryption algorithm. The code $\mathcal{G}'$ used for decoding in Algorithm 28 depends on the private key. In Theorem C.2, the code is given by $\boldsymbol{g}$, which is public and in fact does not need to be chosen randomly in the key generation.[2] Depending on the used algorithm and type of implementation (e.g., in hardware), it can be advantageous in terms of complexity or implementation size if the code is fixed.

C.3 Probability of Large Enough Ciphertext Error Weight

In this section, we analyze the probability that the error part $\mathrm{Tr}(\alpha \boldsymbol{z}) + \boldsymbol{e}$ of the ciphertext

$$\begin{aligned} \boldsymbol{c} &= [\boldsymbol{m}, \boldsymbol{0}_u] \cdot \boldsymbol{G}_{\mathcal{G}} + \mathrm{Tr}(\alpha \boldsymbol{k}_{\mathrm{pub}}) + \boldsymbol{e} \\ &= \underbrace{([\boldsymbol{m}, \boldsymbol{0}_u] + \mathrm{Tr}(\alpha \boldsymbol{x})) \cdot \boldsymbol{G}_{\mathcal{G}}}_{\text{codeword}} + \underbrace{\mathrm{Tr}(\alpha \boldsymbol{z}) + \boldsymbol{e}}_{\text{error}} \end{aligned}$$

has a large enough rank to avoid the ciphertext attacks discussed in Section 5.4. The results of this appendix are summarized in Section 5.4.4.

For random choices of $\boldsymbol{k}_{\mathrm{pub}}$, α, and $\boldsymbol{e}$, we have $\mathrm{rk}_q(\mathrm{Tr}(\alpha \boldsymbol{z})) = w$, $\mathrm{rk}_q(\boldsymbol{e}) = t_{\mathrm{pub}}$, and $\mathrm{rk}_q(\mathrm{Tr}(\alpha \boldsymbol{z}) + \boldsymbol{e}) = w + t_{\mathrm{pub}}$ with probability close to 1. However, there is a very small probability that the error has a significantly smaller rank than in the generic case. Our aim is to design the system parameters such that this probability is sufficiently small, e.g., $2^{-\lambda}$, to avoid attacks utilizing this behavior.

As we see in this section, the choice of $\boldsymbol{z}$ in the public key significantly influences this probability (fixed $\boldsymbol{z}$, randomness in α and $\boldsymbol{e}$). Since also $\boldsymbol{z}$ is drawn using a random experiment during the key generation, we are interested in the probability that this key is strong, i.e., whether the rank of $\mathrm{Tr}(\alpha \boldsymbol{z}) + \boldsymbol{e}$ is large with sufficiently high probability, where the randomness is only in α and $\boldsymbol{e}$.

We start with a lemma that shows that the probability mass function of the $\mathbb{F}_q$-rank of $\mathrm{Tr}(\alpha \boldsymbol{z})$ for a uniformly drawn α only depends on the weight distribution of the code spanned by $\boldsymbol{a}_1, \dots, \boldsymbol{a}_\zeta$, which are the $\mathbb{F}_{q^m}$-linearly independent vectors over $\mathbb{F}_{q^m}$ from which $\boldsymbol{z}$ is constructed.

Lemma C.3. *Let $\boldsymbol{z}$ be constructed using the randomly chosen $[w, \zeta]_{\mathbb{F}_{q^m}}$ code $\mathcal{A}$ as in Algorithm 26. Denote by $A_0, \dots, A_w$ the rank-weight distribution of $\mathcal{A}$. For α chosen*

[2] Note that we described the key generation as in [62], where $\boldsymbol{g}$ is chosen at random, but this is not necessary for the security of the system.

uniformly at random from $\mathbb{F}_{q^{mu}}$, we have

$$\Pr\left(\mathrm{rk}_q\left(\mathrm{Tr}(\alpha \boldsymbol{z})\right) = i\right) = \frac{A_i}{q^{m\zeta}}.$$

Proof. We use the notation ($\boldsymbol{z}$, $\boldsymbol{s}$, $\mathcal{A}$, $\boldsymbol{P}$, and $\boldsymbol{S}$) from Algorithm 26. First, we observe that $\mathrm{Tr}(\alpha\boldsymbol{z}) = \mathrm{Tr}(\alpha[\boldsymbol{s}, \boldsymbol{0}_{n-w}])\boldsymbol{P}$, and hence, it holds that $\mathrm{rk}_q(\mathrm{Tr}(\alpha\boldsymbol{z})) = \mathrm{rk}_q(\mathrm{Tr}(\alpha\boldsymbol{s}))$. We can expand $\alpha \in \mathbb{F}_{q^{mu}}$ in the dual basis γ_i^* as $\alpha = \sum_{i=1}^{u} \alpha_i \gamma_i^*$. Then,

$$\mathrm{Tr}(\alpha\boldsymbol{s}) = \sum_{i=1}^{u} \alpha_i \boldsymbol{s}_i = [\alpha_1, \dots, \alpha_u] \begin{bmatrix} \boldsymbol{s}_1 \\ \vdots \\ \boldsymbol{s}_u \end{bmatrix} = [\alpha_1, \dots, \alpha_u]\boldsymbol{S} \begin{bmatrix} \boldsymbol{a}_1 \\ \vdots \\ \boldsymbol{a}_\zeta \end{bmatrix},$$

where $\boldsymbol{a}_1, \dots, \boldsymbol{a}_\zeta$ is a basis of $\mathcal{A}$ and $\boldsymbol{S} \in \mathbb{F}_{q^m}^{u \times \zeta}$ is a matrix of full rank ζ. As α is chosen uniformly at random from $\mathbb{F}_{q^{mu}}$, the α_i are chosen independently and uniformly at random from $\mathbb{F}_{q^m}$. As $\mathrm{rk}_{q^m}(\boldsymbol{S}) = \zeta$, this is equivalent to saying that

$$[\beta_1, \dots, \beta_\zeta] \coloneqq [\alpha_1, \dots, \alpha_u]\boldsymbol{S}$$

is chosen uniformly at random from $\mathbb{F}_{q^m}^{\zeta}$. Hence, we have

$$\mathrm{Tr}(\alpha\boldsymbol{s}) = [\beta_1, \dots, \beta_\zeta] \begin{bmatrix} \boldsymbol{a}_1 \\ \vdots \\ \boldsymbol{a}_\zeta \end{bmatrix},$$

i.e., $\mathrm{Tr}(\alpha\boldsymbol{s})$ is a codeword of $\mathcal{A}$, chosen uniformly at random. This immediately implies the claim. ■

A direct consequence of the lemma above is the following statement.

Corollary C.4. *With notation as in Lemma C.3, let d be the minimum rank distance of the $[w, \zeta]_{\mathbb{F}_{q^m}}$ code $\mathcal{A}$. Then,*

$$\Pr\left(\mathrm{rk}_q\left(\mathrm{Tr}(\alpha\boldsymbol{z})\right) < d\right) = q^{-m\zeta}.$$

Corollary C.4 shows that we can bound the probability that $\mathrm{Tr}(\alpha\boldsymbol{z})$ has a small $\mathbb{F}_q$-rank if the code $\mathcal{A}$ (as defined in Lemma C.3) has a large minimum rank distance.

Loosely speaking, if the minimum rank distance of the code is small, we can consider this key to be *weak*, and *strong* otherwise. Since the code is chosen uniformly at random from the set of all $[w, \zeta]_{\mathbb{F}_{q^m}}$ codes, we can use the following result from [215] to bound the probability that the key is weak. In the following, we denote by $\mathrm{d}_{\mathrm{R,min}}(\mathcal{C})$ the minimum rank distance of the code $\mathcal{C}$.

Lemma C.5 ([215, Cor. 5.4])**.** *Let $1 \leq k \leq n$ and $2 \leq d \leq n-k+2$, and let $\mathcal{C} \in \mathbb{F}_{q^m}^n$ be drawn uniformly at random from the set of $[n,k]_{\mathbb{F}_{q^m}}$ codes. Then,*

$$\begin{bmatrix} n \\ k \end{bmatrix}_{q^m}^{-1} \sum_{h=1}^{d-1} \begin{bmatrix} d-1 \\ h \end{bmatrix}_{q^m} \sum_{s=h}^{d-1} \begin{bmatrix} d-1-h \\ s-h \end{bmatrix}_{q^m} \begin{bmatrix} n-s \\ n-k \end{bmatrix}_{q^m} (-1)^{s-h} q^{m\binom{s-h}{2}} \leq$$

$$\Pr\Big(\mathrm{d}_{\mathrm{R,min}}(\mathcal{C}) < d \Big) \leq \frac{q^{mk}-1}{(q^m-1)(q^{mn}-1)} \left(\sum_{i=0}^{d-1} \begin{bmatrix} n \\ i \end{bmatrix}_q \prod_{j=0}^{i-1} \Big(q^m - q^j\Big) - 1 \right).$$

Since the code in Lemma C.5 is chosen uniformly at random, it does not exactly match the distribution of the code $\mathcal{A}$ in Algorithm 26. Hence, we need the following lemma and theorem to estimate the probability of a small minimum rank distance in our case.

Lemma C.6. *If an $[n,k]_{\mathbb{F}_{q^m}}$ code is MRD, then it has a basis consisting of codewords of $\mathbb{F}_q$-rank n.*

Proof. We show that the number of full-rank codewords is at least $q^{m(k-1)}$. Since these codewords are all non-zero, their $\mathbb{F}_{q^m}$-span must have cardinality at least q^{mk} and is hence the entire code.

The weight distribution of an MRD code of length n and minimum distance d is given by

$$A_{d+s} = \begin{bmatrix} n \\ d+s \end{bmatrix}_q \sum_{j=0}^{s} (-1)^{j+s} \begin{bmatrix} d+s \\ d+j \end{bmatrix}_q q^{(s-j)(s-j-1)/2} (q^{m(j+1)} - 1), \tag{C.1}$$

where $s \in [0 : n-d]$, m is the order of the extension field, $n \leq m$, and A_{d+s} denotes the number of rank-$(d+s)$ codewords [86].

We are interested in a lower bound on the number of full-rank codewords, i.e., $s = n-d$. The sum in (C.1) is an alternating sum whose terms increase in j. Therefore, we can be lower bound the sum by

$$A_{d+s} \geq \begin{bmatrix} n \\ d+s \end{bmatrix}_q \left((q^{m(s+1)} - 1) - \begin{bmatrix} d+s \\ d+s-1 \end{bmatrix}_q (q^{ms} - 1) \right).$$

Hence, for $s = n - d$, we obtain

$$\begin{aligned} A_n &\geq \begin{bmatrix} n \\ n \end{bmatrix}_q \left((q^{m(n-d+1)} - 1) - \begin{bmatrix} n \\ n-1 \end{bmatrix}_q (q^{m(n-d)} - 1) \right) \\ &= q^{mk} - 1 - \begin{bmatrix} n \\ n-1 \end{bmatrix}_q (q^{m(k-1)} - 1) \\ &= q^{mk} - 1 - \frac{(q^n - 1)q^{m(k-1)}}{q-1} + \frac{q^n - 1}{q-1} \\ &\geq q^{mk} - \frac{(q^n - 1)q^{m(k-1)}}{q-1} \\ &\geq q^{m(k-1)} \underbrace{\left(q^m - \frac{q^n - 1}{q-1} \right)}_{\geq 1 \text{ since } m \geq n \text{ and } q \geq 2} \\ &\geq q^{m(k-1)}. \end{aligned}$$

■

Theorem C.7. *Let m, ζ, and w be chosen such that*

$$1 - \zeta q^{\zeta w - m} \geq \tfrac{1}{2}. \tag{C.2}$$

Let $\mathcal{A}$ be chosen as in Algorithm 26, i.e., uniformly at random from the set of $[w, \zeta]_{\mathbb{F}_{q^m}}$ codes that have a basis consisting only of codewords with $\mathbb{F}_q$-rank w. Furthermore, let $2 \leq t \leq w - \zeta + 2$. Then,

$$\Pr\left(\mathrm{d}_{\mathrm{R,min}}(\mathcal{A}) < t \right) \leq 2 \frac{q^{m\zeta} - 1}{(q^m - 1)(q^{mw} - 1)} \left(\sum_{i=0}^{t-1} \begin{bmatrix} w \\ i \end{bmatrix}_q \prod_{j=0}^{i-1} \left(q^m - q^j \right) - 1 \right).$$

Proof. We define an alternative random experiment, where a code $\mathcal{A}'$ is chosen uniformly from *all* $[w, \zeta]_{\mathbb{F}_{q^m}}$ codes. The sought probability is then given by the conditional probability

$$\Pr\left(\mathrm{d}_{\mathrm{R,min}}(\mathcal{A}') < t \mid \mathcal{S}\right),$$

where $\mathcal{S}$ is the event that $\mathcal{A}'$ has a basis of maximal-rank codewords. We derive the result using the relation

$$\Pr\left(\mathrm{d}_{\mathrm{R,min}}(\mathcal{A}') < t\right) \geq \Pr\left(\mathrm{d}_{\mathrm{R,min}}(\mathcal{A}') < t \mid \mathcal{S}\right)\Pr\left(\mathcal{S}\right). \tag{C.3}$$

First, note that Lemma C.5 gives us

$$\Pr\left(\mathrm{d}_{\mathrm{R,min}}(\mathcal{A}') < t\right) \leq \frac{q^{m\zeta}-1}{(q^m-1)(q^{mw}-1)}\left(\sum_{i=0}^{t-1}\begin{bmatrix} w \\ i \end{bmatrix}_q \prod_{j=0}^{i-1}\left(q^m - q^j\right) - 1\right).$$

By Lemma C.6, we have

$$\Pr\left(\mathcal{S}\right) \geq \Pr\left(\mathcal{A}' \text{ is MRD}\right).$$

Using [214, Thm. 21], we can lower-bound this probability by

$$\Pr\left(\mathcal{A}' \text{ is MRD}\right) \geq 1 - \zeta q^{\zeta w - m} \geq \tfrac{1}{2},$$

where the last inequality follows from (C.2). The claim follows by combining the two bounds with (C.3). ■

The last building block for a general bound on the probability of $\mathrm{Tr}(\alpha \boldsymbol{z}) + \boldsymbol{e}$ having small rank is the following lemma, which gives a bound on this probability conditioned on the event that $\mathrm{Tr}(\alpha \boldsymbol{z})$ has a given rank.

Lemma C.8. *Let* $\boldsymbol{k}_{\mathrm{pub}} = \boldsymbol{x} \cdot \boldsymbol{G}_{\mathcal{G}} + \boldsymbol{z}$ *be fixed as in Algorithm 27, and let* α *be chosen such that* $\mathrm{rk}_q(\mathrm{Tr}(\alpha \boldsymbol{z})) = t$*. For* $\boldsymbol{e} \xleftarrow{\$} \{\boldsymbol{a} \in \mathbb{F}_{q^m}^n : \mathrm{rk}_q(\boldsymbol{a}) = t_{\mathrm{pub}}\}$*, we have*

$$\Pr\Big(\mathrm{rk}_q\left(\mathrm{Tr}(\alpha\boldsymbol{z}) + \boldsymbol{e}\right) < w \,\Big|\, \mathrm{rk}_q\left(\mathrm{Tr}(\alpha\boldsymbol{z})\right) = t,\, \mathrm{rk}_q(\boldsymbol{e}) = t_{\mathrm{pub}}\Big) \leq 256\min\{t, t_{\mathrm{pub}}\}^2 q^{-(t+t_{\mathrm{pub}}-w+1)\left(n+\frac{-t-w-t_{\mathrm{pub}}+1}{2}\right)}.$$

Proof. For simplicity, we write

$$\begin{aligned}
\boldsymbol{e}_1 &\coloneqq \mathrm{Tr}(\alpha \boldsymbol{z}), & \boldsymbol{E}_1 &\coloneqq \mathrm{ext}_{q^m/q}(\boldsymbol{e}_1), & \mathcal{E}_1^{\mathrm{C}} &\coloneqq \mathcal{R}_q\left(\boldsymbol{E}_1^\top\right) \subseteq \mathbb{F}_q^m \\
&&&& \mathcal{E}_1^{\mathrm{R}} &\coloneqq \mathcal{R}_q(\boldsymbol{E}_1) \subseteq \mathbb{F}_q^n \\
\boldsymbol{e}_2 &\coloneqq \boldsymbol{e}, & \boldsymbol{E}_2 &\coloneqq \mathrm{ext}_{q^m/q}(\boldsymbol{e}_2), & \mathcal{E}_2^{\mathrm{C}} &\coloneqq \mathcal{R}_q\left(\boldsymbol{E}_2^\top\right) \subseteq \mathbb{F}_q^m \\
&&&& \mathcal{E}_2^{\mathrm{R}} &\coloneqq \mathcal{R}_q(\boldsymbol{E}_2) \subseteq \mathbb{F}_q^n.
\end{aligned}$$

It is clear that $\mathrm{rk}_q(\boldsymbol{e}_1 + \boldsymbol{e}_2) = \mathrm{rk}_q(\boldsymbol{E}_1 + \boldsymbol{E}_2)$ and, since $\mathrm{rk}_q(\boldsymbol{e}_1) = \mathrm{rk}_q(\boldsymbol{E}_1) = t$ and $\mathrm{rk}_q(\boldsymbol{e}_2) = \mathrm{rk}_q(\boldsymbol{E}_2) = t_{\mathrm{pub}}$, we have

$$\begin{aligned}
\dim_q\!\left(\mathcal{E}_1^{\mathrm{C}}\right) &= \dim_q\!\left(\mathcal{E}_1^{\mathrm{R}}\right) = t \\
\dim_q\!\left(\mathcal{E}_2^{\mathrm{C}}\right) &= \dim_q\!\left(\mathcal{E}_2^{\mathrm{R}}\right) = t_{\mathrm{pub}}.
\end{aligned}$$

Note that in our probabilistic model, the spaces $\mathcal{E}_1^{\mathrm{C}}$ and $\mathcal{E}_1^{\mathrm{R}}$ are fixed. It follows that $\mathcal{E}_2^{\mathrm{C}}$ and $\mathcal{E}_2^{\mathrm{R}}$ are random variables that are uniformly distributed on the set of t_{pub}-dimensional subspaces of $\mathbb{F}_q^m$ and $\mathbb{F}_q^n$, respectively, and stochastically independent. Due to [216, Thm. 1], for

$$\dim\!\left(\mathcal{E}_1^{\mathrm{C}} \cap \mathcal{E}_2^{\mathrm{C}}\right) = i \quad \text{and} \quad \dim\!\left(\mathcal{E}_1^{\mathrm{R}} \cap \mathcal{E}_2^{\mathrm{R}}\right) = j,$$

we have

$$\mathrm{rk}_q(\boldsymbol{E}_1) + \mathrm{rk}_q(\boldsymbol{E}_2) - i - j \leq \mathrm{rk}_q(\boldsymbol{E}_1 + \boldsymbol{E}_2) \leq \mathrm{rk}_q(\boldsymbol{E}_1) + \mathrm{rk}_q(\boldsymbol{E}_2) - \max\{i, j\}.$$

Since $\mathrm{rk}_q(\boldsymbol{E}_1) + \mathrm{rk}_q(\boldsymbol{E}_2) = t + t_{\mathrm{pub}}$, this implies

$$\begin{aligned}
&\Pr\!\left(\mathrm{rk}_q\left(\boldsymbol{E}_1 + \boldsymbol{E}_2\right) < w \,\middle|\, \mathrm{rk}_q\left(\boldsymbol{E}_1\right) = t,\, \mathrm{rk}_q(\boldsymbol{E}_2) = t_{\mathrm{pub}}\right) \\
&\leq \Pr\left(\dim\!\left(\mathcal{E}_1^{\mathrm{C}} \cap \mathcal{E}_2^{\mathrm{C}}\right) + \dim\!\left(\mathcal{E}_1^{\mathrm{R}} \cap \mathcal{E}_2^{\mathrm{R}}\right) > t + t_{\mathrm{pub}} - w\right) \\
&= \sum_{\substack{i,j=0 \\ i+j > t + t_{\mathrm{pub}} - w}}^{\min\{t, t_{\mathrm{pub}}\}} \Pr\left(\dim\!\left(\mathcal{E}_1^{\mathrm{C}} \cap \mathcal{E}_2^{\mathrm{C}}\right) = i\right) \Pr\left(\dim\!\left(\mathcal{E}_1^{\mathrm{R}} \cap \mathcal{E}_2^{\mathrm{R}}\right) = j\right).
\end{aligned}$$

Due to [171, Proof of Lemma 7], we have

$$\begin{aligned}
\Pr\Big(\dim\Big(\mathcal{E}_1^{\mathrm{C}} \cap \mathcal{E}_2^{\mathrm{C}}\Big) = i\Big) &= \frac{\begin{bmatrix} m-t \\ t_{\mathrm{pub}}-i \end{bmatrix}_q \begin{bmatrix} t \\ i \end{bmatrix}_q q^{(t-i)(t_{\mathrm{pub}}-i)}}{\begin{bmatrix} m \\ t_{\mathrm{pub}} \end{bmatrix}_q} \\
&\leq 16 \frac{q^{(t_{\mathrm{pub}}-i)(m-t-t_{\mathrm{pub}}+i)+i(t-i)+(t-i)(t_{\mathrm{pub}}-i)}}{q^{t_{\mathrm{pub}}(m-t_{\mathrm{pub}})}} \\
&= 16 q^{-i(m-t-t_{\mathrm{pub}}+i)}.
\end{aligned}$$

Likewise, we have

$$\Pr\Big(\dim\Big(\mathcal{E}_1^{\mathrm{R}} \cap \mathcal{E}_2^{\mathrm{R}}\Big) = i\Big) \leq 16 q^{-i(n-t-t_{\mathrm{pub}}+i)}.$$

Due to $n \leq m$, we obtain

$$\begin{aligned}
&\Pr\Big(\mathrm{rk}_q\left(\boldsymbol{E}_1 + \boldsymbol{E}_2\right) < w \,\Big|\, \mathrm{rk}_q\left(\boldsymbol{E}_1\right) = t,\, \mathrm{rk}_q(\boldsymbol{E}_2) = t_{\mathrm{pub}}\Big) \\
&\leq 256 \sum_{\substack{i,j=0 \\ i+j>t+t_{\mathrm{pub}}-w}}^{\min\{t,t_{\mathrm{pub}}\}} q^{-i(n-t-t_{\mathrm{pub}}+i)} q^{-j(m-t-t_{\mathrm{pub}}+j)} \\
&\leq 256 \sum_{\substack{i,j=0 \\ i+j>t+t_{\mathrm{pub}}-w}}^{\min\{t,t_{\mathrm{pub}}\}} q^{-i(n-t-t_{\mathrm{pub}}+i)} q^{-j(n-t-t_{\mathrm{pub}}+j)} \\
&\leq 256 \sum_{\substack{i,j=0 \\ i+j>t+t_{\mathrm{pub}}-w}}^{\min\{t,t_{\mathrm{pub}}\}} q^{-(i+j)(n-t-t_{\mathrm{pub}})-(i^2+j^2)} \\
&\leq 256 \min\{t, t_{\mathrm{pub}}\}^2 q^{-(t+t_{\mathrm{pub}}-w+1)(n-t-t_{\mathrm{pub}})-\frac{(t+t_{\mathrm{pub}}-w+1)^2}{2}} \\
&\leq 256 \min\{t, t_{\mathrm{pub}}\}^2 q^{-(t+t_{\mathrm{pub}}-w+1)\left(n+\frac{-t-w-t_{\mathrm{pub}}+1}{2}\right)}.
\end{aligned}$$

■

Summarized, we have the following:

Theorem C.9. *Let m, ζ, and w be chosen such that $1 - \zeta q^{\zeta w - m} \geq \frac{1}{2}$. Choose $\boldsymbol{z}$ of* *the public key as in Algorithm 26. Let $2 \leq t \leq w - \zeta + 2$. With probability at least*

$$P_{\text{strong,key}}(t) \geq 1 - 2\frac{q^{m\zeta} - 1}{(q^m - 1)(q^{mw} - 1)} \left(\sum_{i=0}^{t-1} \begin{bmatrix} w \\ i \end{bmatrix}_q \prod_{j=0}^{i-1} \left(q^m - q^j\right) - 1 \right)$$

the public key has the following property:

Sample $\alpha \xleftarrow{\$} \mathbb{F}_{q^{mu}}$ and $\boldsymbol{e} \xleftarrow{\$} \{\boldsymbol{a} \in \mathbb{F}_{q^m}^n : \operatorname{rk}_q(\boldsymbol{a}) = t_{\text{pub}}\}$, both uniformly at random. Then, the probability that $\operatorname{Tr}(\alpha \boldsymbol{z}) + \boldsymbol{e}$ has $\mathbb{F}_q$-rank at least w is lower-bounded by

$$\begin{aligned} &\Pr\left(\operatorname{rk}_q\left(\operatorname{Tr}(\alpha \boldsymbol{z}) + \boldsymbol{e}\right) \geq w\right) \\ &\qquad \geq 1 - q^{-m\zeta} - 256 \min\{t, t_{\text{pub}}\}^2 q^{-(t+t_{\text{pub}}-w+1)\left(n + \frac{-t-w-t_{\text{pub}}+1}{2}\right)}. \end{aligned}$$

Proof. This follows directly by combining Corollary C.4, Lemma C.5, Lemma C.8, and a union-bound argument. ■

Remark C.1. *By the asymptotical analysis in [215], we have*

$$P_{\text{strong,key}}(t) \geq 1 - \Theta\left(q^{m[t-(w-\zeta+2)]}\right).$$

Since the hidden constant strongly depends on q, this asymptotic value should only be used for a rough estimation of the strong-key probability, and the exact formula in Theorem C.9 should be used for the parameter design.

Nevertheless, the formula shows that $1 - P_{\text{strong,key}}(t)$ decreases exponentially in m times the difference of t and $w - \zeta + 2$. Hence, usually we can choose t close to the maximal value $w - \zeta + 2$ to achieve a given designed probability for a key to be strong.

For instance, we can choose $t \approx (w - \zeta + 2) - \frac{\lambda}{m} \log_q(2)$ for

$$P_{\text{strong,key}}(t) \geq 1 - 2^{-\lambda},$$

where λ is the security parameter.

D

Notation, Variables, and Abbreviations

In the following, we list the notation, the variables, and the abbreviations that are used throughout this dissertation. Notation and variables that only appear close to their definitions are not listed. Furthermore, the notation and the variables defined in Chapters 3, 4, and 5 are only valid within the scope of the respective chapter. Only the definitions from Chapter 2 are valid throughout the whole thesis.

Mathematical Notation

Sets and Finite Fields

q	Power of a prime.
m, n, u	Integers.
$[n:m]$	Set of integers $\{n, n+1, \dots, m\}$.
$\mathcal{I} + \mathcal{J}$	Sumset of the finite subsets of integers $\mathcal{I}$ and $\mathcal{J}$.
$i \overset{\$}{\leftarrow} \mathcal{I}$	Sampling uniformly from the set $\mathcal{I}$ and assigning it to i.
$\mathbb{Z}$	Set of all integers.
$\mathbb{N}$	Set of all natural numbers.
$\mathbb{R}$	Set of all real numbers.
$\mathbb{F}_q$, $\mathbb{F}_{q^m}$, $\mathbb{F}_{q^{mu}}$	Finite fields of size q, q^m, q^{mu}.
$\mathbb{F}_q^*$, $\mathbb{F}_{q^m}^*$, $\mathbb{F}_{q^{mu}}^*$	Multiplicative groups of $\mathbb{F}_q$, $\mathbb{F}_{q^m}$, $\mathbb{F}_{q^{mu}}$.

$\mathbb{F}_q^{m\times n}$	Set of all $m \times n$ matrices over $\mathbb{F}_q$.
$\mathbb{F}_{q^m}^n = \mathbb{F}_{q^m}^{1\times n}$	Set of all row vectors of length n over $\mathbb{F}_{q^m}$.

Sets, Matrices, and Vectors

$\boldsymbol{A}$	Matrix.
$\boldsymbol{A}^\top$	Transpose of the matrix $\boldsymbol{A}$.
$\boldsymbol{A}^\perp$	Matrix whose rows form a basis of the right kernel of $\boldsymbol{A}$.
$A_{i,j}$	Element in the i-th row and the j-th column of $\boldsymbol{A}$.
$\boldsymbol{A}_{[a:b],[c:d]}$	Matrix $\boldsymbol{A}$ restricted to the rows $a, a+1, \ldots, b$ and to the columns $c, c+1, \ldots, d$.
$\boldsymbol{A}_{[a:b],:}$	Matrix $\boldsymbol{A}$ restricted to the rows $a, a+1, \ldots, b$.
$\boldsymbol{A}_{:,[c:d]}$	Matrix $\boldsymbol{A}$ restricted to the columns $c, c+1, \ldots, d$.
$\mathrm{rk}_q(\boldsymbol{A})$	$\mathbb{F}_q$-rank of the matrix $\boldsymbol{A}$.
$\mathrm{ref}(\boldsymbol{A})$	Reduced row echelon form of the matrix $\boldsymbol{A}$.
$\mathrm{NM}_q(m, n, i)$	Number of $m \times n$ matrices over $\mathbb{F}_q$ of rank i.
$\boldsymbol{a} = [a_1, \ldots, a_n]$	Vector of length n.
$[\boldsymbol{a}, \boldsymbol{d}]$	Concatenation of the vectors $\boldsymbol{a}$ and $\boldsymbol{d}$.
$\mathrm{rot}(\boldsymbol{a})$	Circulant matrix induced by $\boldsymbol{a}$.
$\boldsymbol{a}\boldsymbol{d} := \boldsymbol{a}\,\mathrm{rot}(\boldsymbol{d})^\top$	Product of the vectors $\boldsymbol{a}$ and $\boldsymbol{d}$.
$\boldsymbol{a} \star \boldsymbol{d}$	Componentwise product of the vectors $\boldsymbol{a}$ and $\boldsymbol{d}$.
$\mathcal{M}_{s,q}(\boldsymbol{a})$	Moore matrix with s rows and q-powers for the vector $\boldsymbol{a}$.
$\mathrm{diag}(\boldsymbol{a})$	Diagonal matrix with the elements of $\boldsymbol{a}$ on the main diagonal.
$\mathrm{ext}_{q^m/q}(\boldsymbol{a})$	Extension of the vector $\boldsymbol{a} \in \mathbb{F}_{q^m}^n$ over the field $\mathbb{F}_q$.
$\mathrm{Tr}(\boldsymbol{b})$	Trace of the vector $\boldsymbol{b} \in \mathbb{F}_{q^{mu}}^n$ to $\mathbb{F}_{q^m}^n$.
$\boldsymbol{0}_n$	Zero vector of length n.

Polynomials, Probability Theory, and Vector Spaces

$\mathbb{F}_q[X]$, $\mathbb{F}_{q^m}[X]$	Sets of all univariate polynomials over $\mathbb{F}_q$ and $\mathbb{F}_{q^m}$.
$\deg(f)$	Degree of the polynomial $f \in \mathbb{F}_q[X]$.
$\mathrm{ev}_{\boldsymbol{\alpha}}(f)$	Evaluation map of $f \in \mathbb{F}_q[X]$ for the vector $\boldsymbol{\alpha} \in \mathbb{F}_q^n$.
$\Pr(A)$	Probability of the event A.

$\mathbb{E}[B]$	Expectation of the random variable B.
$\langle \boldsymbol{a}_1, \dots, \boldsymbol{a}_\ell \rangle_q$	$\mathbb{F}_q$-linear vector space spanned by $\boldsymbol{a}_1, \dots, \boldsymbol{a}_\ell$.
$\langle \mathcal{S} \rangle_q$	$\mathbb{F}_q$-linear vector space generated by the elements of the set $\mathcal{S}$.
$\mathcal{V}^\perp$	Dual space of the vector space $\mathcal{V}$.
$\dim_q(\mathcal{V})$	$\mathbb{F}_q$-dimension of the vector space $\mathcal{V}$.
$\mathcal{R}_q(\boldsymbol{A})$	$\mathbb{F}_q$-linear row space of the matrix $\boldsymbol{A}$.
$\mathcal{K}_q(\boldsymbol{A})$	Right $\mathbb{F}_q$-kernel of the matrix $\boldsymbol{A}$.
$\mathrm{rk}_q(\boldsymbol{a})$	Dimension of the $\mathbb{F}_q$-linear space spanned by the entries of $\boldsymbol{a}$.
$\mathrm{Gr}_q(\mathcal{V}, k)$	Set of all k-dimensional $\mathbb{F}_q$-linear subspaces of the space $\mathcal{V}$.
$\begin{bmatrix} j \\ i \end{bmatrix}_q$	Gaussian binomial coefficient.

Linear Codes

$[n,k]_{\mathbb{F}_{q^m}}$	$\mathbb{F}_{q^m}$-linear code of length n and dimension k over $\mathbb{F}_{q^m}$.
$\mathcal{C}^\perp$	Dual code of the linear code $\mathcal{C}$.
$[u; n,k]_{\mathbb{F}_{q^m}}$	(Vertically) u-interleaved $[n,k]_{\mathbb{F}_{q^m}}$ code.
$\mathcal{C}^{(u)}$	(Vertically) u-interleaved code of $\mathcal{C}$.
$\mathcal{C}^{(\star 2)}$	Schur-square (or Hadamard-square) of $\mathcal{C}$.
$\mathrm{supp}_{\mathrm{H}}(\boldsymbol{c})$	Hamming support of the vector $\boldsymbol{c}$.
$\mathrm{wt}_{\mathrm{H}}(\boldsymbol{c})$	Hamming weight of the vector $\boldsymbol{c}$.
$\mathrm{d}_{\mathrm{H}}(\boldsymbol{c}, \boldsymbol{d})$	Hamming distance of the vectors $\boldsymbol{c}$ and $\boldsymbol{d}$.
$[n,k,d_{\min}]^{\mathrm{H}}_{\mathbb{F}_{q^m}}$	$[n,k]_{\mathbb{F}_{q^m}}$ code with minimum Hamming distance $d_{\min}$.
$\mathcal{RS}_k(\boldsymbol{\alpha})$	RS code of dimension k with locators $\boldsymbol{\alpha}$.
$\mathcal{TRS}_k(\boldsymbol{\alpha}, \boldsymbol{\tau}, \boldsymbol{\pi}, \boldsymbol{\eta})$	TRS code of dimension k with locators $\boldsymbol{\alpha}$, twists $\boldsymbol{\tau}$, hooks $\boldsymbol{\pi}$, and field coefficients $\boldsymbol{\eta}$.
$\boldsymbol{G}_{\boldsymbol{\alpha},\boldsymbol{\tau},\boldsymbol{\pi},\boldsymbol{\eta}}$	A generator matrix of $\mathcal{TRS}_k(\boldsymbol{\alpha}, \boldsymbol{\tau}, \boldsymbol{\pi}, \boldsymbol{\eta})$.
$\mathrm{supp}^{(\mathrm{C})}_{\mathrm{R}}(\boldsymbol{c})$	Column rank support of the vector $\boldsymbol{c}$.
$\mathrm{supp}^{(\mathrm{R})}_{\mathrm{R}}(\boldsymbol{c})$	Row rank support of the vector $\boldsymbol{c}$.
$\mathrm{wt}_{\mathrm{R}}(\boldsymbol{c})$	Rank weight of the vector $\boldsymbol{c}$.
$\mathrm{d}_{\mathrm{R}}(\boldsymbol{c}, \boldsymbol{d})$	Rank distance of the vectors $\boldsymbol{c}$ and $\boldsymbol{d}$.
$[n,k,d_{\min}]^{\mathrm{R}}_{\mathbb{F}_{q^m}}$	$[n,k]_{\mathbb{F}_{q^m}}$ code with minimum rank distance $d_{\min}$.

$\mathrm{d}_{\mathrm{R,min}}(\mathcal{C})$	Minimum rank distance of the code $\mathcal{C}$.
$\mathcal{G}_k(\boldsymbol{g})$	Gabidulin code of dimension k with locators $\boldsymbol{g}$.
$\boldsymbol{G}_{\mathcal{G}}$	A generator matrix of $\mathcal{G}_k(\boldsymbol{g})$.
$\mathcal{G}_k^{(u)}(\boldsymbol{g})$	(Vertically) u-interleaved code of $\mathcal{G}_k(\boldsymbol{g})$.
ℓ_{SR}	Blocking parameter.
η_{SR}	Block size.
μ_{SR}	Minimum of η_{SR} and m.
$\mathrm{supp}_{\mathrm{SR}}^{(\mathrm{C})}(\boldsymbol{c})$	Column sum-rank support of the vector $\boldsymbol{c}$.
$\mathrm{supp}_{\mathrm{SR}}^{(\mathrm{R})}(\boldsymbol{c})$	Row sum-rank support of the vector $\boldsymbol{c}$.
$\mathrm{wt}_{\mathrm{SR}}(\boldsymbol{c})$	Sum-rank weight of the vector $\boldsymbol{c}$.
$\mathrm{d}_{\mathrm{SR}}(\boldsymbol{c}, \boldsymbol{d})$	Sum-rank distance of the vectors $\boldsymbol{c}$ and $\boldsymbol{d}$.
$[n, k, d_{\mathrm{min}}]_{\mathbb{F}_{q^m}}^{\mathrm{SR}}$	$[n, k]_{\mathbb{F}_{q^m}}$ code with minimum sum-rank distance d_{min}.
$\mathcal{T}_{t,\ell_{\mathrm{SR}},\mu_{\mathrm{SR}}}$	The set of weight decompositions of sum-rank weight t.
$\Xi_{q,\mu_{\mathrm{SR}}}(\boldsymbol{f})$	Set of products of subspaces of $\mathbb{F}_q^{\mu_{\mathrm{SR}}}$ with weight decomposition $\boldsymbol{f}$.

Coding-Theoretic Problems

DecGab	Decisional Gabidulin Decoding.
$\mathsf{DecSD}_{\mathrm{H}}$	Decisional Hamming Syndrome Decoding.
$\mathsf{DecSD}_{\mathrm{R}}$	Decisional Rank Syndrome Decoding.
$\mathsf{DecISD}_{\mathrm{R}}$	Decisional Interleaved Rank Syndrome Decoding.
$\mathsf{DecRGab}$	Decisional Restricted Gabidulin Decoding.
$\mathsf{DecRIGab}$	Decisional Restricted Interleaved Gabidulin Decoding.
$\mathsf{DecQCSD}_{\mathrm{H}}$	Decisional Quasi-Cyclic Hamming Syndrome Decoding.
$\mathsf{DecSD}_{\mathrm{SR}}$	Decisional Sum-Rank Syndrome Decoding.
SeaGab	Search Gabidulin Decoding.
$\mathsf{SeaISD}_{\mathrm{R}}$	Search Interleaved Rank Syndrome Decoding.
$\mathsf{SeaQCSD}_{\mathrm{H}}$	Search Quasi-Cyclic Hamming Syndrome Decoding.
$\mathsf{SeaRGab}$	Search Restricted Gabidulin Decoding.
$\mathsf{SeaRIGab}$	Search Restricted Interleaved Gabidulin Decoding.
$\mathsf{SeaSD}_{\mathrm{SR}}$	Search Sum-Rank Syndrome Decoding.
$\mathsf{Sea2SD}_{\mathrm{H}}$	Search 2 Hamming Syndrome Decoding.

Complexity Classes

P	Polynomial time.
RP	Randomized polynomial time.
coRP	Co-randomized polynomial time.
ZPP	Zero-error probabilistic polynomial time.
NP	Non-deterministic polynomial time.

Cryptographic Systems

λ	Security level of a cryptographic scheme.
sk	Private key.
pk	Public key.
KeyGen	Key-generation algorithm.
$\mathsf{Encrypt}$	Encryption algorithm.
$\mathsf{Decrypt}$	Decryption algorithm.
Encaps	Key-encapsulation algorithm.
Decaps	Key-decapsulation algorithm.
$\perp$	Decryption or decapsulation failure.
Π^{Enc}	Public-key encryption scheme.
Π^{KEM}	Key-encapsulation mechanism.
$\mathsf{PubEnc}^{\mathsf{CPA}}_{\mathcal{A},\Pi^{\mathrm{Enc}}}(\lambda)$	The indistinguishability under chosen-plaintext attack game for public-key encryption schemes.
$\mathsf{KEM}^{\mathsf{CCA2}}_{\mathcal{A},\Pi^{\mathrm{KEM}}}(\lambda)$	The indistinguishability under adaptive chosen-ciphertext attack game for key-encapsulation mechanisms.

Abbreviations

BCH	Bose–Chaudhuri–Hocquenghem.
BIKE	Bit Flipping Key Encapsulation.
FL	Faure–Loidreau.
GCD	Greatest Common Divisor.
GOT	Gaborit, Otmani, and Talé Kalachi.
GRS	Generalized Reed–Solomon.
HQC	Hamming Quasi-Cyclic.

IND-CCA2	Indistinguishability under Adaptive Chosen-Ciphertext Attack.
IND-CPA	Indistinguishability under Chosen-Plaintext Attack.
ISD	Information-Set Decoding.
KEM	Key-Encapsulation Mechanism.
LAKE	Low Rank Parity Check Codes Key Exchange.
LOCKER	Low Rank Parity Check Codes Encryption.
LRPC	Low-Rank Parity-Check Codes.
MDPC	Moderate-Density Parity-Check Codes.
MDS	Maximum Distance Separable.
MRD	Maximum Rank Distance.
NIST	National Institute of Standards and Technology.
PPT	Probabilistic Polynomial Time.
RAMESSES	Rank Metric Encryption Scheme with Short Keys.
ROLLO	Compilation of Rank-Ouroboros, LAKE, and LOCKER.
RQC	Rank Quasi-Cyclic.
RSA	Rivest–Shamir–Adleman.
RS	Reed–Solomon.
TRS	Twisted Reed–Solomon.

Bibliography

[1] D. Kahn. *The Codebreakers: The Comprehensive History of Secret Communication from Ancient Times to the Internet.* Scribner's and Sons, 1996.

[2] J. Katz and Y. Lindell. *Introduction to Modern Cryptography.* Chapman & Hall/CRC, Taylor & Francis Group, 2014.

[3] W. Diffie and M. Hellman. "New Directions in Cryptography". In: *IEEE Trans. Inf. Theory* 22.6 (1976), pp. 644–654.

[4] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone. *Handbook of Applied Cryptography.* CRC, Taylor & Francis Group, 1996.

[5] *The Transport Layer Security (TLS) Protocol Version 1.3.* URL: https://datatracker.ietf.org/doc/html/rfc8446 (visited on 11/05/2021).

[6] P. W. Shor. "Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer". In: *SIAM J. Comput.* 26.5 (1997), pp. 1484–1509.

[7] C. Gidney and M. Ekera. "How to Factor 2048 Bit RSA Integers in 8 Hours Using 20 Million Noisy Qubits". In: *Quantum* 5 (2021), p. 433.

[8] *List of Quantum Processors.* URL: https://en.wikipedia.org/wiki/List_of_quantum_processors (visited on 11/05/2021).

[9] *IBM's Roadmap For Scaling Quantum Technology.* URL: https://research.ibm.com/blog/ibm-quantum-roadmap (visited on 11/05/2021).

[10] *NIST Post-Quantum Cryptography Standardization Call.* URL: https://csrc.nist.gov/projects/post-quantum-cryptography (visited on 11/05/2021).

[11] D. J. Bernstein. "Introduction to Post-Quantum Cryptography". In: *Int. Conf. Post-Quantum Cryptogr. (PQCrypto).* 2009, pp. 1–14.

[12] R. Overbeck and N. Sendrier. "Code-Based Cryptography". In: *Int. Conf. Post-Quantum Cryptogr. (PQCrypto).* 2009, pp. 95–145.

[13] R. J. McEliece. “A Public-Key Cryptosystem Based on Algebraic Coding Theory”. In: *DSN Prog. Rep.* 44 (1978), pp. 114–116.

[14] H. Niederreiter. “Knapsack Type Cryptosystems and Algebraic Coding Theory”. In: *Probl. Contr. Inf. Theory* 15.2 (1986), pp. 157–166.

[15] M. V. Sidelnikov and O. S. Shestakov. “On Insecurity of Cryptosystems Based on Generalized Reed–Solomon Codes”. In: *Discrete Math. Appl.* 2 (1992), pp. 439–444.

[16] T. P. Berger and P. Loidreau. “How to Mask the Structure of Codes for a Cryptographic Use”. In: *Des. Codes Cryptogr.* 35.1 (2005), pp. 63–79.

[17] C. Wieschebrink. “Cryptanalysis of the Niederreiter Public Key Scheme Based on GRS Subcodes”. In: *Int. Conf. Post-Quantum Cryptogr. (PQCrypto).* 2010, pp. 61–72.

[18] M. V. Sidelnikov. “Public-key Cryptosystem Based on Binary Reed-Muller Codes”. In: *Discrete Math. Appl.* 4 (1994), pp. 191–208.

[19] L. Minder and A. Shokrollahi. “Cryptanalysis of the Sidelnikov Cryptosystem”. In: *Adv. Cryptol. - EUROCRYPT.* 2007, pp. 347–360.

[20] T. P. Berger, P. Cayrel, P. Gaborit, and A. Otmani. “Reducing Key Length of the McEliece Cryptosystem”. In: *Adv. Cryptol. - AFRICACRYPT.* 2009, pp. 77–97.

[21] J.-C. Faugère, A. Otmani, L. Perret, F. de Portzamparc, and J.-P. Tillich. “Structural Cryptanalysis of McEliece Schemes with Compact Keys”. In: *Des. Codes Cryptogr.* 79.1 (2016), pp. 87–112.

[22] H. Janwa and O. Moreno. “McEliece Public Key Cryptosystems Using Algebraic-Geometric Codes”. In: *Des. Codes Cryptogr.* 8.3 (1996), pp. 293–307.

[23] A. Couvreur, I. M. Corbella, and R. Pellikaan. “Cryptanalysis of McEliece Cryptosystem Based on Algebraic Geometry Codes and Their Subcodes”. In: *IEEE Trans. Inf. Theory* 63.8 (2017), pp. 5404–5418.

[24] Y. Wang. “Quantum Resistant Random Linear Code Based Public Key Encryption Scheme RLCE”. In: *IEEE Int. Symp. Inf. Theory (ISIT).* 2016, pp. 2519–2523.

[25] A. Couvreur, M. Lequesne, and J.-P. Tillich. “Recovering Short Secret Keys of RLCE in Polynomial Time”. In: *Int. Conf. Post-Quantum Cryptogr. (PQCrypto)*. 2019, pp. 133–152.

[26] M. Bardet, É. Barelli, O. Blazy, R. Canto Torres, A. Couvreur, P. Gaborit, A. Otmani, N. Sendrier, and J.-P. Tillich. “BIG QUAKE: Binary Goppa Quasi-Cyclic Key Encapsulation”. In: *First round submission to the NIST post-quantum cryptography call* (2017). URL: https://bigquake.inria.fr/.

[27] G. Banegas, P. S. Barreto, B. O. Boidje, P.-L. Cayrel, G. N. Dione, K. Gaj, C. T. Gueye, R. Haeussler, J. B. Klamti, O. N’diaye, D. T. Nguyen, E. Persichetti, and J. E. Ricardini. “DAGS: Key Encapsulation Using Dyadic GS Codes”. In: *J. Math. Cryptol.* 12.4 (2018), pp. 221–239.

[28] É. Barelli and A. Couvreur. “An Efficient Structural Attack on NIST Submission DAGS”. In: *Adv. Cryptol. - ASIACRYPT*. 2018, pp. 93–118.

[29] L. Holzbaur, H. Liu, S. Puchinger, and A. Wachter-Zeh. “On Decoding and Applications of Interleaved Goppa Codes”. In: *IEEE Int. Symp. Inf. Theory (ISIT)*. 2019, pp. 1887–1891.

[30] K. Khathuria, J. Rosenthal, and V. Weger. “Encryption Scheme Based on Expanded Reed-Solomon Codes”. In: *Adv. Math. Commun.* 15.2 (2021), pp. 207–218.

[31] A. Couvreur and M. Lequesne. *On the Security of Subspace Subcodes of Reed–Solomon Codes for Public Key Encryption*. 2020. URL: https://arxiv.org/abs/2009.05826.

[32] J.-C. Faugère, A. Otmani, L. Perret, and J.-P. Tillich. “Algebraic Cryptanalysis of McEliece Variants with Compact Keys”. In: *Adv. Cryptol. - EUROCRYPT*. 2010, pp. 279–298.

[33] E. M. Gabidulin, A. V. Paramonov, and O. V. Tretjakov. “Ideals over a Non-Commutative Ring and Their Application in Cryptology”. In: *Adv. Cryptol. - EUROCRYPT*. 1991, pp. 482–489.

[34] E. M. Gabidulin and A. V. Ourivski. “Modified GPT PKC with Right Scrambler”. In: *Electron. Notes Discrete Math.* 6 (2001), pp. 168–177.

[35] E. M. Gabidulin, A. V. Ourivski, B. Honary, and B. Ammar. “Reducible Rank Codes and Their Applications to Cryptography”. In: *IEEE Trans. Inf. Theory* 49.12 (2003), pp. 3289–3293.

[36] P. Loidreau. “Designing a Rank Metric Based McEliece Cryptosystem”. In: *Int. Conf. Post-Quantum Cryptogr. (PQCrypto)*. 2010, pp. 142–152.

[37] H. Rashwan, E. M. Gabidulin, and B. Honary. “Security of the GPT Cryptosystem and its Applications to Cryptography”. In: *Security Comm. Networks* 4.8 (2011), pp. 937–946.

[38] E. M. Gabidulin. “Attacks and Counter-Attacks on the GPT Public Key Cryptosystem”. In: *Des. Codes Cryptogr.* 48.2 (2008), pp. 171–177.

[39] E. M. Gabidulin, H. Rashwan, and B. Honary. “On Improving Security of GPT Cryptosystems”. In: *IEEE Int. Symp. Inf. Theory (ISIT)*. 2009, pp. 1110–1114.

[40] H. Rashwan, E. M. Gabidulin, and B. Honary. “A Smart Approach for GPT Cryptosystem Based on Rank Codes”. In: *IEEE Int. Symp. Inf. Theory (ISIT)*. 2010, pp. 2463–2467.

[41] J. K. Gibson. “Severely Denting the Gabidulin Version of the McEliece Public Key Cryptosystem”. In: *Des. Codes Cryptogr.* 6.1 (1995), pp. 37–45.

[42] J. K. Gibson. “The Security of the Gabidulin Public Key Cryptosystem”. In: *Adv. Cryptol. - EUROCRYPT*. 1996, pp. 212–223.

[43] R. Overbeck. “Extending Gibson’s Attacks on the GPT Cryptosystem”. In: *Int. Workshop Coding Cryptogr. (WCC)*. 2006, pp. 178–188.

[44] R. Overbeck. “A New Structural Attack for GPT and Variants”. In: *Int. Conf. Cryptol. - MYCRYPT*. 2005, pp. 50–63.

[45] R. Overbeck. “Structural Attacks for Public Key Cryptosystems Based on Gabidulin Codes”. In: *J. Cryptol.* 21.2 (2008), pp. 280–301.

[46] A.-L. Horlemann-Trautmann, K. Marshall, and J. Rosenthal. “Considerations for Rank-Based Cryptosystems”. In: *IEEE Int. Symp. Inf. Theory (ISIT)*. 2016, pp. 2544–2548.

[47] A. Otmani, H. Talé Kalachi, and S. Ndjeya. “Improved Cryptanalysis of Rank Metric Schemes Based on Gabidulin Codes”. In: *Des. Codes Cryptogr.* 86.9 (2016), pp. 1–14.

[48] A.-L. Horlemann-Trautmann, K. Marshall, and J. Rosenthal. “Extension of Overbeck’s Attack for Gabidulin-Based Cryptosystems”. In: *Des. Codes Cryptogr.* 86.2 (2018), pp. 319–340.

[49] T. P. Berger, P. Gaborit, and O. Ruatta. “Gabidulin Matrix Codes and Their Application to Small Ciphertext Size Cryptosystems”. In: *Int. Conf. Cryptol. - INDOCRYPT*. 2017, pp. 247–266.

[50] P. Loidreau. “An Evolution of GPT Cryptosystem”. In: *Int. Workshop Alg. Comb. Coding Theory (ACCT)*. 2016.

[51] P. Loidreau. “A New Rank Metric Codes Based Encryption Scheme”. In: *Int. Conf. Post-Quantum Cryptogr. (PQCrypto)*. 2017, pp. 3–17.

[52] R. Misoczki, J.-P. Tillich, N. Sendrier, and P. S. Barreto. “MDPC-McEliece: New McEliece variants from Moderate Density Parity-Check Codes”. In: *IEEE Int. Symp. Inf. Theory (ISIT)*. 2013, pp. 2069–2073.

[53] J.-P. Tillich. “The Decoding Failure Probability of MDPC Codes”. In: *IEEE Int. Symp. Inf. Theory (ISIT)*. 2018, pp. 941–945.

[54] N. Drucker, S. Gueron, and D. Kostic. “QC-MDPC Decoders with Several Shades of Gray”. In: *Int. Conf. Post-Quantum Cryptogr. (PQCrypto)*. 2020, pp. 35–50.

[55] P. Gaborit, G. Murat, O. Ruatta, and G. Zémor. “Low Rank Parity Check Codes and Their Application to Cryptography”. In: *Int. Workshop Coding Cryptogr. (WCC)*. 2013.

[56] N. Aragon, P. Gaborit, A. Hauteville, O. Ruatta, and G. Zémor. “Low Rank Parity Check Codes: New Decoding Algorithms and Applications to Cryptography”. In: *IEEE Trans. Inf. Theory* 65.12 (2019), pp. 7697–7717.

[57] J. Renner, T. Jerkovits, and H. Bartz. “Efficient Decoding of Interleaved Low-Rank Parity-Check Codes”. In: *Int. Symp. Probl. Redundancy Inf. Control Syst. (REDUNDANCY)*. 2019, pp. 121–126.

[58] J. Renner, S. Puchinger, A. Wachter-Zeh, C. Hollanti, and R. Freij-Hollanti. “Low-Rank Parity-Check Codes over the Ring of Integers Modulo a Prime Power”. In: *IEEE Int. Symp. Inf. Theory (ISIT)*. 2020, pp. 19–24.

[59] J. Renner, A. Neri, and S. Puchinger. “Low-Rank Parity-Check Codes over Galois Rings”. In: *Des. Codes Cryptogr.* 89.2 (2021), pp. 351–386.

[60] M. Alekhnovich. “More on Average Case vs Approximation Complexity”. In: *IEEE Symp. Found. Comput. Sci. (FOCS).* 2003, pp. 298–307.

[61] D. Augot and M. Finiasz. “A Public Key Encryption Scheme Based on the Polynomial Reconstruction Problem”. In: *Adv. Cryptol. - EUROCRYPT.* 2003, pp. 229–240.

[62] C. Faure and P. Loidreau. “A New Public-Key Cryptosystem Based on the Problem of Reconstructing p-Polynomials”. In: *Int. Workshop Coding Cryptogr. (WCC).* 2006, pp. 304–315.

[63] C. Aguilar-Melchor, O. Blazy, J. Deneuville, P. Gaborit, and G. Zémor. “Efficient Encryption from Random Quasi-Cyclic Codes”. In: *IEEE Trans. Inf. Theory* 64.5 (2018), pp. 3927–3943.

[64] J.-S. Coron. “Cryptanalysis of a Public-Key Encryption Scheme Based on the Polynomial Reconstruction Problem”. In: *Int. Conf. Practice Theory Public Key Cryptogr. (PKC).* 2004, pp. 14–27.

[65] P. Gaborit, A. Otmani, and H. Talé Kalachi. “Polynomial-Time Key Recovery Attack on the Faure–Loidreau Scheme Based on Gabidulin Codes”. In: *Des. Codes Cryptogr.* 86.7 (2018), pp. 1391–1403.

[66] T. Migler, K. E. Morrison, and M. Ogle. *Weight and Rank of Matrices over Finite Fields.* 2004. URL: https://arxiv.org/abs/math/0403314.

[67] R. Lidl and H. Niederreiter. *Finite Fields.* Encyclopedia of Mathematics and its Applications. Cambridge University Press, 1996.

[68] R. Kötter and F. R. Kschischang. “Coding for Errors and Erasures in Random Network Coding”. In: *IEEE Trans. Inf. Theory* 54.8 (2008), pp. 3579–3591.

[69] R. W. Hamming. “Error Detecting and Error Correcting Codes”. In: *Bell Labs Tech. J.* 29.2 (1950), pp. 147–160.

[70] R. Singleton. “Maximum Distance Q-Nary Codes”. In: *IEEE Trans. Inf. Theory* 10.2 (1964), pp. 116–118.

[71] I. S. Reed and G. Solomon. “Polynomial Codes over Certain Finite Fields”. In: *J. Soc. Indust. Appl. Math.* 8.2 (1960), pp. 300–304.

[72] D. Gorenstein and N. Zierler. “A Class of Error-Correcting Codes in p^m Symbols”. In: *J. Soc. Ind. Appl. Math.* 9 (1961), pp. 207–214.

[73] E. Berlekamp. “Nonbinary BCH Decoding”. In: *IEEE Trans. Inf. Theory* 14.2 (1968), p. 242.

[74] J. Massey. “Shift-Register Synthesis and BCH Decoding”. In: *IEEE Trans. Inf. Theory* 15.1 (1969), pp. 122–127.

[75] L. Welch and E. Berlekamp. “Error Correction for Algebraic Block Codes”. Pat. US 4,633,470. 1986.

[76] M. Sudan. “Decoding of Reed Solomon Codes Beyond the Error-Correction Bound”. In: *J. Complexity* 13.1 (1997), pp. 180–193.

[77] V. Guruswami and M. Sudan. “Improved Decoding of Reed–Solomon and Algebraic-Geometry Codes”. In: *IEEE Trans. Inf. Theory* 45.6 (1999), pp. 1757–1767.

[78] V. Y. Krachkovsky and Y. X. Lee. “Decoding for Iterative Reed–Solomon Coding Schemes”. In: *IEEE Trans. Magn.* 33.5 (1997), pp. 2740–2742.

[79] D. Bleichenbacher, A. Kiayias, and M. Yung. “Decoding of Interleaved Reed Solomon Codes over Noisy Data”. In: *Int. Colloq. Automata Lang. Program. (ICALP)*. 2003, pp. 97–108.

[80] D. Coppersmith and M. Sudan. “Reconstructing Curves in Three (and Higher) Dimensional Space from Noisy Data”. In: *ACM Symp. Theory Comput. (STOC)*. 2003, pp. 136–142.

[81] F. Parvaresh and A. Vardy. “Multivariate Interpolation Decoding Beyond the Guruswami-Sudan Radius”. In: *Allteron Conf. Commun. Control Comput.* 2004, p. 1362.

[82] F. Parvaresh. “Algebraic List-Decoding of Error-Correcting Codes”. PhD thesis. University of California, San Diego, 2007.

[83] P. Beelen, S. Puchinger, and J. Rosenkilde né Nielsen. “Twisted Reed–Solomon Codes”. In: *IEEE Int. Symp. Inf. Theory (ISIT)*. 2017, pp. 336–340.

[84] P. Beelen, M. Bossert, S. Puchinger, and J. Rosenkilde. “Structural Properties of Twisted Reed–Solomon Codes with Applications to Cryptography”. In: *IEEE Int. Symp. Inf. Theory (ISIT)*. 2018, pp. 946–950.

[85] P. Delsarte. “Bilinear Forms over a Finite Field with Applications to Coding Theory”. In: *J. Comb. Theory Ser. A* 25.3 (1978), pp. 226–241.

[86] E. M. Gabidulin. “Theory of Codes with Maximum Rank Distance”. In: *Probl. Inf. Transm.* 21.1 (1985), pp. 3–16.

[87] R. M. Roth. “Maximum-Rank Array Codes and Their Application to Crisscross Error Correction”. In: *IEEE Trans. Inf. Theory* 37.2 (1991), pp. 328–336.

[88] H. Bartz, L. Holzbaur, H. Liu, S. Puchinger, J. Renner, and A. Wachter-Zeh. “Rank-Metric Codes and Their Applications”. In: *Found. Trends Commun. Inf. Theory* 19.3 (2022), pp. 390–546.

[89] G. Matsaglia and G. P. H. Styan. “Equalities and Inequalities for Ranks of Matrices”. In: *Linear Multilinear Algebra* 2.3 (1974), pp. 269–292.

[90] E. M. Gabidulin. “A Fast Matrix Decoding Algorithm for Rank-Error-Correcting Codes”. In: *Algebraic Coding.* 1992, pp. 126–133.

[91] A. V. Paramonov and O. V. Tretjakov. “An Analogue of Berlekamp-Massey Algorithm for Decoding Codes in Rank Metric”. In: *Moscow Inst. Phys. Technol. (MIPT).* 1991.

[92] G. Richter and S. Plass. “Fast Decoding of Rank-Codes with Rank Errors and Column Erasures”. In: *IEEE Int. Symp. Inf. Theory (ISIT).* 2004, p. 398.

[93] Y. Hassan and V. Sidorenko. “Fast Recursive Linearized Feedback Shift Register Synthesis”. In: *Int. Workshop Alg. Comb. Coding Theory (ACCT).* 2010, pp. 162–167.

[94] V. Sidorenko and M. Bossert. “Fast Skew-Feedback Shift-Register Synthesis”. In: *Des. Codes Cryptogr.* 70.1-2 (2014), pp. 55–67.

[95] P. Loidreau. “A Welch-Berlekamp Like Algorithm for Decoding Gabidulin Codes”. In: *Int. Workshop Coding Cryptogr. (WCC).* 2006, pp. 36–45.

[96] A. Wachter-Zeh, V. Afanassiev, and V. Sidorenko. “Fast Decoding of Gabidulin Codes”. In: *Des. Codes Cryptogr.* 66.1-3 (2013), pp. 57–73.

[97] S. Puchinger and A. Wachter-Zeh. “Sub-Quadratic Decoding of Gabidulin Codes”. In: *IEEE Int. Symp. Inf. Theory (ISIT).* 2016, pp. 2554–2558.

[98] S. Puchinger and A. Wachter-Zeh. “Fast Operations on Linearized Polynomials and Their Applications in Coding Theory”. In: *J. Symb. Comput.* 89 (2018), pp. 194–215.

[99] A. Wachter-Zeh. “Decoding of Block and Convolutional Codes in Rank Metric”. PhD thesis. Ulm University and University of Rennes, 2013.

[100] E. M. Gabidulin, A. V. Paramonov, and O. V. Tretjakov. “Rank Errors and Rank Erasures Correction”. In: *Int. Colloq. Coding Theory.* 1991.

[101] G. Richter and S. Plass. “Error and Erasure Decoding of Rank-Codes with a Modified Berlekamp-Massey Algorithm”. In: *Int. Conf. Source Channel Coding (SCC).* 2004, pp. 203–210.

[102] D. Silva. “Error Control for Network Coding”. PhD thesis. University of Toronto, 2009.

[103] E. M. Gabidulin and N. I. Pilipchuk. “Error and Erasure Correcting Algorithms for Rank Codes”. In: *Des. Codes Cryptogr.* 49.1-3 (2008), pp. 105–122.

[104] D. Silva, F. R. Kschischang, and R. Kötter. “A Rank-Metric Approach to Error Control in Random Network Coding”. In: *IEEE Trans. Inf. Theory* 54.9 (2008), pp. 3951–3967.

[105] P. Loidreau and R. Overbeck. “Decoding Rank Errors Beyond the Error-Correction Capability”. In: *Int. Workshop Alg. Comb. Coding Theory (ACCT)* (2006), pp. 168–190.

[106] J. Renner, S. Puchinger, and A. Wachter-Zeh. “LIGA: A Cryptosystem Based on the Hardness of Rank-Metric List and Interleaved Decoding”. In: *Des. Codes Cryptogr.* 89.6 (2021), pp. 1279–1319.

[107] J. Renner, S. Puchinger, and A. Wachter-Zeh. “Interleaving Loidreau’s Rank-Metric Cryptosystem”. In: *Int. Symp. Probl. Redundancy Inf. Control Syst. (REDUNDANCY).* 2019, pp. 127–132.

[108] V. Sidorenko and M. Bossert. “Decoding Interleaved Gabidulin Codes and Multisequence Linearized Shift-Register Synthesis”. In: *IEEE Int. Symp. Inf. Theory (ISIT).* 2010, pp. 1148–1152.

[109] E. M. Gabidulin, M. Bossert, and P. Lusina. “Space-Time Codes Based on Rank Codes”. In: *IEEE Int. Symp. Inf. Theory (ISIT).* 2000, p. 284.

[110] M. Bossert, E. M. Gabidulin, and P. Lusina. “Space-Time Codes Based on Gaussian Integers”. In: *IEEE Int. Symp. Inf. Theory (ISIT).* 2002, p. 273.

[111] Y. Liu, M. P. Fitz, and O. Y. Takeshita. “A Rank Criterion for QAM Space-Time Codes”. In: *IEEE Trans. Inf. Theory* 48.12 (2002), pp. 3062–3079.

[112] P. Lusina, E. M. Gabidulin, and M. Bossert. "Maximum Rank Distance Codes as Space–Time Codes". In: *IEEE Trans. Inf. Theory* 49.10 (2003), pp. 2757–2760.

[113] G. Robert. "A New Constellation for Space-Time Coding". In: *Int. Workshop Coding Cryptogr. (WCC)*. 2015.

[114] S. Puchinger, S. Stern, M. Bossert, and R. F. Fischer. "Space-Time Codes Based on Rank-Metric Codes and Their Decoding". In: *IEEE Int. Symp. Wireless Commun. Syst. (ISWCS)*. 2016, pp. 125–130.

[115] V. Sidorenko, L. Jiang, and M. Bossert. "Skew-Feedback Shift-Register Synthesis and Decoding Interleaved Gabidulin Codes". In: *IEEE Trans. Inf. Theory* 57.2 (2011), pp. 621–632.

[116] A. Wachter-Zeh and A. Zeh. "List and Unique Error-Erasure Decoding of Interleaved Gabidulin Codes with Interpolation Techniques". In: *Des. Codes Cryptogr.* 73.2 (2014), pp. 547–570.

[117] R. W. Nóbrega and B. F. Uchoa-Filho. "Multishot Codes for Network Coding Using Rank-Metric Codes". In: *IEEE Int. Workshop Wireless Netw. Coding (WiNC)*. 2010, pp. 1–6.

[118] A. Wachter, V. Sidorenko, M. Bossert, and V. V. Zyablov. "On (Partial) Unit Memory Codes Based on Gabidulin Codes". In: *Probl. Inf. Transm.* 47.2 (2011), pp. 117–129.

[119] A. Wachter-Zeh and V. Sidorenko. "Rank Metric Convolutional Codes for Random Linear Network Coding". In: *IEEE Int. Symp. Netw. Coding (NetCod)*. 2012, pp. 1–6.

[120] A. Wachter-Zeh, M. Stinner, and V. Sidorenko. "Convolutional Codes in Rank Metric with Application to Random Network Coding". In: *IEEE Trans. Inf. Theory* 61.6 (2015), pp. 3199–3213.

[121] D. Napp, R. Pinto, J. Rosenthal, and P. Vettori. "MRD Rank Metric Convolutional Codes". In: *IEEE Int. Symp. Inf. Theory (ISIT)*. 2017, pp. 2766–2770.

[122] D. Napp, R. Pinto, J. Rosenthal, and P. Vettori. "Faster Decoding of Rank Metric Convolutional Codes". In: *Int. Symp. Math. Theory Netw. Syst. (MTNS)*. 2018.

[123] U. Martínez-Peñas. “Skew and Linearized Reed–Solomon Codes and Maximum Sum Rank Distance Codes over any Division Ring”. In: *J. Algebra* 504 (2018), pp. 587–612.

[124] D. Boucher. “An Algorithm for Decoding Skew Reed–Solomon Codes with Respect to the Skew Metric”. In: *Des. Codes Cryptogr.* 88.9 (2020), pp. 1991–2005.

[125] U. Martínez-Peñas and F. R. Kschischang. “Reliable and Secure Multishot Network Coding Using Linearized Reed-Solomon Codes”. In: *IEEE Trans. Inf. Theory* 65.8 (2019), pp. 4785–4803.

[126] X. Caruso. *Residues of Skew Rational Functions and Linearized Goppa Codes.* 2019. URL: https://arxiv.org/abs/1908.08430.

[127] H. Bartz, T. Jerkovits, S. Puchinger, and J. Rosenkilde. *Fast Decoding of Codes in the Rank, Subspace, and Sum-Rank Metric.* 2020. URL: https://arxiv.org/abs/2005.09916.

[128] U. Martínez-Peñas. *Sum-Rank BCH Codes and Cyclic-Skew-Cyclic Codes.* 2020. URL: https://arxiv.org/abs/2009.04949.

[129] U. Martínez-Peñas and F. R. Kschischang. “Universal and Dynamic Locally Repairable Codes with Maximal Recoverability via Sum-Rank Codes”. In: *IEEE Trans. Inf. Theory* 65.12 (2019), pp. 7790–7805.

[130] M. Shehadeh and F. R. Kschischang. “Rate-Diversity Optimal Multiblock Space-Time Codes via Sum-Rank Codes”. In: *IEEE Int. Symp. Inf. Theory (ISIT).* 2020, pp. 3055–3060.

[131] E. Byrne, H. Gluesing-Luerssen, and A. Ravagnani. “Fundamental Properties of Sum-Rank-Metric Codes”. In: *IEEE Trans. Inf. Theory* 67.10 (2021), pp. 6456–6475.

[132] C. Ott. In: *Personal Communication.*

[133] E. Berlekamp, R. J. McEliece, and H. Van Tilborg. “On the Inherent Intractability of Certain Coding Problems”. In: *IEEE Trans. Inf. Theory* 24.3 (1978), pp. 384–386.

[134] E. Prange. “The Use of Information Sets in Decoding Cyclic Codes”. In: *IRE Trans. Inf. Theory* 8.5 (1962), pp. 5–9.

[135] M. R. Albrecht, D. J. Bernstein, T. Chou, C. Cid, J. Gilcher, T. Lange, V. Maram, I. von Maurich, R. Misoczki, R. Niederhagen, K. G. Paterson, E. Persichetti, C. Peters, P. Schwabe, N. Sendrier, J. Szefer, C. J. Tjhai, M. Tomlinson, and W. Wang. "Classic McEliece". In: *Third round submission to the NIST post-quantum cryptography call* (2020). URL: https://classic.mceliece.org.

[136] P. Gaborit and G. Zémor. "On the Hardness of the Decoding and the Minimum Distance Problems for Rank Codes". In: *IEEE Trans. Inf. Theory* 62.12 (2016), pp. 7245–7252.

[137] F. Chabaud and J. Stern. "The Cryptographic Security of the Syndrome Decoding Problem for Rank Distance Codes". In: *Int. Conf. Theory and Appl. Cryptology and Inform. Security.* 1996, pp. 368–381.

[138] A. V. Ourivski and T. Johansson. "New Technique for Decoding Codes in the Rank Metric and its Cryptography Applications". In: *Probl. Inf. Transm.* 38.3 (2002), pp. 237–246.

[139] P. Gaborit, O. Ruatta, and J. Schrek. "On the Complexity of the Rank Syndrome Decoding Problem". In: *IEEE Trans. Inf. Theory* 62.2 (2016), pp. 1006–1019.

[140] N. Aragon, P. Gaborit, A. Hauteville, and J.-P. Tillich. "A New Algorithm for Solving the Rank Syndrome Decoding Problem". In: *IEEE Int. Symp. Inf. Theory (ISIT).* 2018, pp. 2421–2425.

[141] M. Bardet, P. Briaud, M. Bros, P. Gaborit, V. Neiger, O. Ruatta, and J.-P. Tillich. "An Algebraic Attack on Rank Metric Code-Based Cryptosystems". In: *Adv. Cryptol. - EUROCRYPT.* 2020, pp. 64–93.

[142] M. Bardet, M. Bros, D. Cabarcas, P. Gaborit, R. Perlner, D. Smith-Tone, J.-P. Tillich, and J. Verbel. "Improvements of Algebraic Attacks for Solving the Rank Decoding and MinRank Problems". In: *Adv. Cryptol. - ASIACRYPT.* 2020, pp. 507–536.

[143] L. Trevisan. *Lecture Notes in Computational Complexity.* 2004.

[144] R. L. Rivest, A. Shamir, and L. Adleman. "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems". In: *Commun. ACM* 21.2 (1978), pp. 120–126.

[145] O. Goldreich, S. Goldwasser, and S. Halevi. “Public-Key Cryptosystems from Lattice Reduction Problems”. In: *Adv. Cryptol. - CRYPTO*. 1997, pp. 112–131.

[146] J. Hoffstein, J. Pipher, and J. H. Silverman. “NTRU: A Ring-Based Public Key Cryptosystem”. In: *Int. Algorithmic Number Theory Symp. (ANTS)*. 1998, pp. 267–288.

[147] J. Bos, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, J. M. Schanck, P. Schwabe, G. Seiler, and D. Stehle. “CRYSTALS - Kyber: A CCA-Secure Module-Lattice-Based KEM”. In: *IEEE Eur. Symp. Secur. Privacy (EuroS&P)*. 2018, pp. 353–367.

[148] N. Aragon, P. S. Barreto, S. Bettaieb, L. Bidoux, O. Blazy, J.-C. Deneuville, P. Gaborit, S. Ghosh, S. Gueron, T. Güneysu, C. Aguilar-Melchor, R. Misoczki, E. Persichetti, N. Sendrier, J.-P. Tillich, V. Vasseur, and G. Zémor. “BIKE: Bit Flipping Key Encapsulation”. In: *Third round submission to the NIST post-quantum cryptography call* (2020). URL: https://bikesuite.org.

[149] C. Aguilar-Melchor, N. Aragon, S. Bettaieb, L. Bidoux, O. Blazy, J. Bos, J. Deneuville, A. Dion, P. Gaborit, J. Lacan, E. Persichetti, J. Robert, P. Véron, and G. Zémor. “Hamming Quasi-Cyclic (HQC)”. In: *Third round submission to the NIST post-quantum cryptography call* (2020). URL: https://pqc-hqc.org.

[150] C. Aguilar-Melchor, N. Aragon, M. Bardet, S. Bettaieb, L. Bidoux, O. Blazy, J. Deneuville, P. Gaborit, A. Hauteville, A. Otmani, R. Ruatta, J.-P. Tillich, and G. Zémor. “ROLLO (Rank-Ouroboros, LAKE and LOCKER)”. In: *Second round submission to the NIST post-quantum cryptography call* (2019). URL: https://pqc-rollo.org.

[151] C. Aguilar-Melchor, N. Aragon, S. Bettaieb, L. Bidoux, O. Blazy, J. Deneuville, P. Gaborit, G. Zemor, A. Couvreur, and A. Hauteville. “Rank Quasi-Cyclic (RQC)”. In: *Second round submission to the NIST post-quantum cryptography call* (2019). URL: https://pqc-rqc.org.

[152] A. Wachter-Zeh, S. Puchinger, and J. Renner. “Repairing the Faure–Loidreau Public-Key Cryptosystem”. In: *IEEE Int. Symp. Inf. Theory (ISIT)*. 2018, pp. 2426–2430.

[153] J. Lavauzelle, P. Loidreau, and B.-D. Pham. *RAMESSES, a Rank Metric Encryption Scheme with Short Keys*. 2019. URL: https://arxiv.org/abs/1911.13119.

[154] S. Puchinger, J. Renner, and J. Rosenkilde. “Generic Decoding in the Sum-Rank Metric”. In: *IEEE Int. Symp. Inf. Theory (ISIT).* 2020, pp. 54–59.

[155] S. Puchinger, J. Renner, and J. Rosenkilde. “Generic Decoding in the Sum-Rank Metric”. In: *IEEE Trans. Inf. Theory* (2022). URL: https://ieeexplore.ieee.org/document/9757164.

[156] J. Renner, S. Puchinger, and A. Wachter-Zeh. “Decoding High-Order Interleaved Rank-Metric Codes”. In: *IEEE Int. Symp. Inf. Theory (ISIT).* 2021, pp. 19–24.

[157] P. Gaborit, A. Hauteville, D. H. Phan, and J.-P. Tillich. “Identity-Based Encryption from Codes with Rank Metric”. In: *Adv. Cryptol. - CRYPTO.* 2017, pp. 194–224.

[158] J. Renner, T. Jerkovits, H. Bartz, S. Puchinger, P. Loidreau, and A. Wachter-Zeh. “Randomized Decoding of Gabidulin Codes Beyond the Unique Decoding Radius”. In: *Int. Conf. Post-Quantum Cryptogr. (PQCrypto).* 2020, pp. 3–19.

[159] U. Martínez-Peñas. “Theory of Supports for Linear Codes Endowed With the Sum-Rank Metric”. In: *Des. Codes Cryptogr.* 87.10 (2019), pp. 2295–2320.

[160] P. Lee and E. Brickell. “An Observation on the Security of McEliece’s Public-Key Cryptosystem”. In: *Adv. Cryptol. - EUROCRYPT.* 1988, pp. 275–280.

[161] J. Stern. “A Method for Finding Codewords of Small Weight”. In: *Int. Colloq. Coding Theory Appl.* 1988, pp. 106–113.

[162] J. J. Metzner and E. J. Kapturowski. “A General Decoding Technique Applicable to Replicated File Disagreement Location and Concatenated Code Decoding”. In: *IEEE Trans. Inf. Theory* 36.4 (1990), pp. 911–917.

[163] C. Haslach and A. Han Vinck. “A Decoding Algorithm with Restrictions for Array Codes”. In: *IEEE Trans. Inf. Theory* 45.7 (1999), pp. 2339–2344.

[164] C. Haslach and A. Han Vinck. “Efficient Decoding of Interleaved Linear Block Codes”. In: *IEEE Int. Symp. Inf. Theory (ISIT).* 2000, p. 149.

[165] R. M. Roth and P. O. Vontobel. “Coding for Combined Block–Symbol Error Correction”. In: *IEEE Trans. Inf. Theory* 60.5 (2014), pp. 2697–2713.

[166] W. P. Wardlaw. “Matrix Representation of Finite Fields”. In: *Math. Mag.* 67.4 (1994), pp. 289–293.

[167] R. Overbeck. “Public Key Cryptography Based on Coding Theory”. PhD thesis. Technical University of Darmstadt, 2007.

[168] L. Holzbaur, S. Puchinger, and A. Wachter-Zeh. “Error Decoding of Locally Repairable and Partial MDS Codes”. In: *IEEE Trans. Inf. Theory* 67.3 (2021), pp. 1571–1595.

[169] V. Sidorenko, G. Schmidt, and M. Bossert. “Decoding Punctured Reed–Solomon Codes up to the Singleton Bound”. In: *Int. Conf. Source Channel Coding (SCC)*. 2008, pp. 1–6.

[170] T. Jerkovits and H. Bartz. “Weak Keys in the Faure–Loidreau Cryptosystem”. In: *Int. Workshop Code-Based Cryptogr. (CBCrypto)*. 2019, pp. 102–114.

[171] T. Etzion and A. Vardy. “Error-Correcting Codes in Projective Space”. In: *IEEE Trans. Inf. Theory* 57.2 (2011), pp. 1165–1173.

[172] A. Wachter, V. Sidorenko, and M. Bossert. “A Basis for all Solutions of the Key Equation for Gabidulin Codes”. In: *IEEE Int. Symp. Inf. Theory (ISIT)*. 2010, pp. 1143–1147.

[173] N. Aragon, O. Blazy, P. Gaborit, A. Hauteville, and G. Zémor. “Durandal: A Rank Metric Based Signature Scheme”. In: *Adv. Cryptol. - EUROCRYPT*. 2019, pp. 728–758.

[174] A.-L. Horlemann-Trautmann and M. Kuijper. “A Module Minimization Approach to Gabidulin Decoding via Interpolation”. In: *J. Algebra Comb. Discrete Struct. Appl.* 5.1 (2017), pp. 29–43.

[175] A. Wachter-Zeh. “Bounds on List Decoding of Rank-Metric Codes”. In: *IEEE Trans. Inf. Theory* 59.11 (2013), pp. 7268–7277.

[176] N. Raviv and A. Wachter-Zeh. “Some Gabidulin Codes Cannot Be List Decoded Efficiently at any Radius”. In: *IEEE Trans. Inf. Theory* 62.4 (2016), pp. 1605–1615.

[177] R. Trombetti and F. Zullo. “On the List Decodability of Rank Metric Codes”. In: *IEEE Trans. Inf. Theory* 66.9 (2020), pp. 5379–5386.

[178] E. Ben-Sasson, S. Kopparty, and J. Radhakrishnan. “Subspace Polynomials and Limits to List Decoding of Reed–Solomon Codes”. In: *IEEE Trans. Inf. Theory* 56.1 (2010), pp. 113–120.

[179] V. Guruswami and A. Vardy. “Maximum-Likelihood Decoding of Reed-Solomon Codes is NP-hard”. In: *IEEE Trans. Inf. Theory* 51.7 (2005), pp. 2249–2256.

[180] S. Puchinger. “Construction and Decoding of Evaluation Codes in Hamming and Rank Metric”. PhD thesis. Ulm University, 2018.

[181] A. Couvreur, P. Gaborit, V. Gauthier-Umaña, A. Otmani, and J.-P. Tillich. “Distinguisher-Based Attacks on Public-Key Cryptosystems Using Reed–Solomon Codes”. In: *Des. Codes Cryptogr.* 73.2 (2014), pp. 641–666.

[182] C. Wieschebrink. “An Attack on a Modified Niederreiter Encryption Scheme”. In: *Int. Conf. Practice Theory Public Key Cryptogr. (PKC)*. 2006, pp. 14–26.

[183] C. Aguilar-Melchor, N. Aragon, S. Bettaieb, L. Bidoux, O. Blazy, J. Bos, J. Deneuville, A. Dion, P. Gaborit, J. Lacan, E. Persichetti, J. Robert, P. Véron, and G. Zémor. “Hamming Quasi-Cyclic (HQC)”. In: *Second round submission to the NIST post-quantum cryptography call* (2019). URL: https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/round-2/submissions/HQC-Round2.zip.

[184] T. B. Paiva and R. Terada. “A Timing Attack on the HQC Encryption Scheme”. In: *Sel. Areas Cryptogr. (SAC)*. 2020, pp. 551–573.

[185] G. Wafo-Tapa, S. Bettaieb, L. Bidoux, P. Gaborit, and E. Marcatel. “A Practicable Timing Attack Against HQC and its Countermeasure”. In: *Adv. Math. Commun.* (2020).

[186] L. Huguenin-Dumittan and S. Vaudenay. “Classical Misuse Attacks on NIST Round 2 PQC: The Power of Rank-Based Schemes”. In: *Int. Conf. Appl. Cryptogr. Netw. Security (ACNS)*. 2020, pp. 208–227.

[187] P. Ravi, S. Sinha Roy, A. Chattopadhyay, and S. Bhasin. “Generic Side-Channel Attacks on CCA-Secure Lattice-Based PKE and KEMs”. In: *IACR Trans. Cryptogr. Hardware Embedded Syst.* 2020.3 (2020), pp. 307–335.

[188] J. Lavauzelle and J. Renner. “Cryptanalysis of a System Based on Twisted Reed–Solomon Codes”. In: *Des. Codes Cryptogr.* 88.7 (2020), pp. 1285–1300.

[189] T. Schamberger, J. Renner, G. Sigl, and A. Wachter-Zeh. “A Power Side-Channel Attack on the CCA2-Secure HQC KEM”. In: *Smart Card Res. Adv. Appl. (CARDIS)*. 2021, pp. 119–134.

[190] A.-L. Horlemann, S. Puchinger, J. Renner, T. Schamberger, and A. Wachter-Zeh. “Information-Set Decoding with Hints”. In: *Int. Workshop Code-Based Cryptogr. (CBCrypto)*. 2021, pp. 60–83.

[191] The Sage Developers. *SageMath, the Sage Mathematics Software System*. 2021. URL: https://www.sagemath.org.

[192] D. Hofheinz, K. Hövelmanns, and E. Kiltz. “A Modular Analysis of the Fujisaki–Okamoto Transformation”. In: *Theory Cryptogr.* 2017, pp. 341–371.

[193] D. J. Bernstein, T. Chou, and P. Schwabe. “McBits: Fast Constant-Time Code-Based Cryptography”. In: *IACR Conf. Cryptogr. Hardware Embedded Syst. (CHES)*. 2013, pp. 250–272.

[194] A. Becker, A. Joux, A. May, and A. Meurer. “Decoding Random Binary Linear Codes in 2n/20: How $1+1=0$ Improves Information Set Decoding”. In: *Adv. Cryptol. - EUROCRYPT*. 2012, pp. 520–536.

[195] I. Dumer. “Two Decoding Algorithms for Linear Codes”. In: *Probl. Inf. Transm.* 25.1 (1989), pp. 24–32.

[196] N. Sendrier. “Decoding One Out of Many”. In: *Int. Conf. Post-Quantum Cryptogr. (PQCrypto)*. 2011, pp. 51–67.

[197] A.-L. Horlemann-Trautmann and V. Weger. “Information Set Decoding in the Lee Metric with Applications to Cryptography”. In: *Adv. Math. Commun.* 15.4 (2021), pp. 677–699.

[198] D. J. Bernstein, T. Lange, and C. Peters. “Attacking and Defending the McEliece Cryptosystem”. In: *Int. Conf. Post-Quantum Cryptogr. (PQCrypto)*. 2008, pp. 31–46.

[199] S. Puchinger, J. Renner, and A. Wachter-Zeh. “Twisted Gabidulin Codes in the GPT Cryptosystem”. In: *Int. Workshop Alg. Comb. Coding Theory (ACCT)*. 2018.

[200] P. Loidreau. “Métrique Rang et Cryptographie (in French)”. Mémoire d’habilitation à diriger des recherches. Université Pierre et Marie Curie, 2007.

[201] H. A. Shehhi, E. Bellini, F. Borba, F. Caullery, M. Manzano, and V. Mateu. “An IND-CCA-Secure Code-Based Encryption Scheme Using Rank Metric”. In: *Adv. Cryptol. - AFRICACRYPT*. 2019, pp. 79–96.

[202] M. Bombar and A. Couvreur. “Decoding Supercodes of Gabidulin Codes and Applications to Cryptanalysis”. In: *Int. Conf. Post-Quantum Cryptogr. (PQCrypto)*. 2021, pp. 3–22.

[203] S. Bettaieb, L. Bidoux, P. Gaborit, and E. Marcatel. “Preventing Timing Attacks Against RQC Using Constant Time Decoding of Gabidulin Codes”. In: *Int. Conf. Post-Quantum Cryptogr. (PQCrypto)*. 2019, pp. 371–386.

[204] X. Caruso and J. Le Borgne. “Fast Multiplication for Skew Polynomials”. In: *Int. Symp. Symb. Algebraic Comput. (ISSAC)*. 2017, pp. 77–84.

[205] M. Gadouleau and Z. Yan. “Complexity of Decoding Gabidulin Codes”. In: *IEEE Annu. Conf. Inf. Sci. Syst. (CISS)*. 2008, pp. 1081–1085.

[206] D. Silva and F. R. Kschischang. “Fast Encoding and Decoding of Gabidulin Codes”. In: *IEEE Int. Symp. Inf. Theory (ISIT)*. 2009, pp. 2858–2862.

[207] J. Kunz, J. Renner, G. Maringer, T. Schamberger, and A. Wachter-Zeh. “On Software Implementation of Gabidulin Decoders”. In: *Int. Workshop Alg. Comb. Coding Theory (ACCT)*. 2020, pp. 95–101.

[208] R. Nojima, H. Imai, K. Kobara, and K. Morozov. “Semantic Security for the McEliece Cryptosystem Without Random Oracles”. In: *Des. Codes Cryptogr.* 49.1-3 (2008), pp. 289–305.

[209] E. Fujisaki and T. Okamoto. “Secure Integration of Asymmetric and Symmetric Encryption Schemes”. In: *J. Cryptol.* 26.1 (2013), pp. 80–101.

[210] J. Rosenkilde. In: *Personal Communication*.

[211] M. Boyer, G. Brassard, P. Høyer, and A. Tapp. “Tight Bounds on Quantum Searching”. In: *Fortschr. Phys.* 46.4-5 (1998), pp. 493–505.

[212] D. J. Bernstein. “Grover vs. McEliece”. In: *Int. Conf. Post-Quantum Cryptogr. (PQCrypto)*. 2010, pp. 73–80.

[213] T. Cover. “Enumerative Source Encoding”. In: *IEEE Trans. Inf. Theory* 19.1 (1973), pp. 73–77.

[214] A. Neri, A.-L. Horlemann-Trautmann, T. Randrianarisoa, and J. Rosenthal. “On the Genericity of Maximum Rank Distance and Gabidulin Codes”. In: *Des. Codes Cryptogr.* 86.2 (2018), pp. 341–363.

[215] E. Byrne and A. Ravagnani. “Partition-Balanced Families of Codes and Asymptotic Enumeration in Coding Theory”. In: *J. Comb. Theory Ser. A* 171 (2020).

[216] G. Marsaglia. *Bounds on the Rank of the Sum of Matrices*. 1964. URL: https://apps.dtic.mil/sti/pdfs/AD0600471.pdf.

Related Publications by the Author

[57] J. Renner, T. Jerkovits, and H. Bartz. “Efficient Decoding of Interleaved Low-Rank Parity-Check Codes”. In: *Int. Symp. Probl. Redundancy Inf. Control Syst. (REDUNDANCY)*. 2019, pp. 121–126.

[58] J. Renner, S. Puchinger, A. Wachter-Zeh, C. Hollanti, and R. Freij-Hollanti. “Low-Rank Parity-Check Codes over the Ring of Integers Modulo a Prime Power”. In: *IEEE Int. Symp. Inf. Theory (ISIT)*. 2020, pp. 19–24.

[59] J. Renner, A. Neri, and S. Puchinger. “Low-Rank Parity-Check Codes over Galois Rings”. In: *Des. Codes Cryptogr.* 89.2 (2021), pp. 351–386.

[88] H. Bartz, L. Holzbaur, H. Liu, S. Puchinger, J. Renner, and A. Wachter-Zeh. “Rank-Metric Codes and Their Applications”. In: *Found. Trends Commun. Inf. Theory* 19.3 (2022), pp. 390–546.

[106] J. Renner, S. Puchinger, and A. Wachter-Zeh. “LIGA: A Cryptosystem Based on the Hardness of Rank-Metric List and Interleaved Decoding”. In: *Des. Codes Cryptogr.* 89.6 (2021), pp. 1279–1319.

[107] J. Renner, S. Puchinger, and A. Wachter-Zeh. “Interleaving Loidreau’s Rank-Metric Cryptosystem”. In: *Int. Symp. Probl. Redundancy Inf. Control Syst. (REDUNDANCY)*. 2019, pp. 127–132.

[152] A. Wachter-Zeh, S. Puchinger, and J. Renner. “Repairing the Faure–Loidreau Public-Key Cryptosystem”. In: *IEEE Int. Symp. Inf. Theory (ISIT)*. 2018, pp. 2426–2430.

[154] S. Puchinger, J. Renner, and J. Rosenkilde. “Generic Decoding in the Sum-Rank Metric”. In: *IEEE Int. Symp. Inf. Theory (ISIT)*. 2020, pp. 54–59.

[155] S. Puchinger, J. Renner, and J. Rosenkilde. “Generic Decoding in the Sum-Rank Metric”. In: *IEEE Trans. Inf. Theory* (2022). URL: https://ieeexplore.ieee.org/document/9757164.

[156] J. Renner, S. Puchinger, and A. Wachter-Zeh. "Decoding High-Order Interleaved Rank-Metric Codes". In: *IEEE Int. Symp. Inf. Theory (ISIT).* 2021, pp. 19–24.

[158] J. Renner, T. Jerkovits, H. Bartz, S. Puchinger, P. Loidreau, and A. Wachter-Zeh. "Randomized Decoding of Gabidulin Codes Beyond the Unique Decoding Radius". In: *Int. Conf. Post-Quantum Cryptogr. (PQCrypto).* 2020, pp. 3–19.

[188] J. Lavauzelle and J. Renner. "Cryptanalysis of a System Based on Twisted Reed–Solomon Codes". In: *Des. Codes Cryptogr.* 88.7 (2020), pp. 1285–1300.

[189] T. Schamberger, J. Renner, G. Sigl, and A. Wachter-Zeh. "A Power Side-Channel Attack on the CCA2-Secure HQC KEM". In: *Smart Card Res. Adv. Appl. (CARDIS).* 2021, pp. 119–134.

[190] A.-L. Horlemann, S. Puchinger, J. Renner, T. Schamberger, and A. Wachter-Zeh. "Information-Set Decoding with Hints". In: *Int. Workshop Code-Based Cryptogr. (CBCrypto).* 2021, pp. 60–83.

[199] S. Puchinger, J. Renner, and A. Wachter-Zeh. "Twisted Gabidulin Codes in the GPT Cryptosystem". In: *Int. Workshop Alg. Comb. Coding Theory (ACCT).* 2018.

[207] J. Kunz, J. Renner, G. Maringer, T. Schamberger, and A. Wachter-Zeh. "On Software Implementation of Gabidulin Decoders". In: *Int. Workshop Alg. Comb. Coding Theory (ACCT).* 2020, pp. 95–101.

[217] S. Puchinger, J. Renner, A. Wachter-Zeh, and J. Zumbrägel. "Efficient Decoding of Gabidulin Codes over Galois Rings". In: *IEEE Int. Symp. Inf. Theory (ISIT).* 2021, pp. 25–30.